A Survey of Scaling in Large Language Model Reasoning

Zihan Chen[1], Song Wang[1], Zhen Tan[2], Xingbo Fu[1], Zhenyu Lei[1], Peng Wang[1], Huan Liu[2], Cong Shen[1], Jundong Li[1]
[1]University of Virginia, Charlottesville, VA, USA
[2]Arizona State University, Tempe, AZ, USA
{brf3rx, sw3wv, xf3av, vjd5zr, pw7nc, cong, jundong}@virginia.edu
{ztan36, huanliu}@asu.edu

ABSTRACT

The rapid advancements in large Language models (LLMs) have significantly enhanced their reasoning capabilities, driven by various strategies such as multi-agent collaboration. However, unlike the well-established performance improvements achieved through scaling data and model size, the scaling of reasoning in LLMs is more complex and can even negatively impact reasoning performance, introducing new challenges in model alignment and robustness. In this survey, we provide a comprehensive examination of scaling in LLM reasoning, categorizing it into multiple dimensions and analyzing how and to what extent different scaling strategies contribute to improving reasoning capabilities. We begin by exploring scaling in input size, which enables LLMs to process and utilize a more extensive context for improved reasoning. Next, we analyze scaling in reasoning steps that improve multi-step inference and logical consistency. We then examine scaling in reasoning rounds, where iterative interactions refine reasoning outcomes. Furthermore, we discuss scaling in training-enabled reasoning, focusing on optimization through iterative model improvement. Finally, we outline future directions for further advancing LLM reasoning. By synthesizing these diverse perspectives, this survey aims to provide insights into how scaling strategies fundamentally enhance the reasoning capabilities of LLMs and further guide the development of next-generation AI systems.

1. INTRODUCTION

Large Language Models (LLMs) have evolved rapidly, demonstrating remarkable advancements across various natural language processing (NLP) tasks, including text generation, comprehension, and problem-solving [66; 154; 214; 216; 215; 65]. One of the key driving forces behind these improvements is scaling, where increasing the size of training data and model parameters has led to substantial performance gains [86; 69; 193]. Scaling has played a pivotal role in the development of state-of-the-art LLMs such as GPT-4 [134], and Gemini [176], enabling them to generalize across a broad range of tasks with unprecedented accuracy and fluency [184]. The empirical success of scaling laws has reinforced the notion that simply increasing model size and data availability can significantly enhance LLM capabilities [25; 130; 31]. However, while such scaling strategies have led to more powerful models, they do not fully explain improvements in complex reasoning tasks, which require structured thinking and logical consistency [40; 155; 47].

Notably, unlike simpler tasks that rely on memorization or direct retrieval of information, reasoning demands deeper cognitive-like processes, including step-by-step deductions, counterfactual reasoning, and planning [142; 83]. While early LLMs exhibited shallow reasoning abilities [11; 117], recent advancements have introduced techniques aimed at enhancing LLM reasoning performance through various strategies [33; 54; 164]. For instance, s1 [131] explicitly extends the reasoning length, enabling models to engage in deeper, iterative reasoning that can identify and correct errors in previous inference steps. However, scaling reasoning length does not always guarantee improved performance—simply increasing the number of reasoning steps may introduce redundancy, compounding errors, or even diminished accuracy [149; 73; 125; 204; 18; 220]. This highlights the complex and non-trivial nature of scaling in reasoning, necessitating a deeper investigation into how different scaling strategies influence LLM reasoning effectiveness and when they yield diminishing returns. In this survey, we use *reasoning* to refer to tasks in which the model must perform nontrivial transformation over information, such as multi-step deduction or iterative refinement, rather than merely retrieve or fluently generate content. Under this view, not every improvement in general LLM capability should be interpreted as a reasoning improvement. For example, larger context windows, retrieval, or memory may improve performance simply by supplying missing evidence, while their reasoning benefit is most meaningful when that evidence must be integrated in a multi-step decision process. Conversely, these strategies may offer limited gains on tasks dominated by direct factual recall or shallow pattern matching. Throughout this survey, we focus on scaling strategies that strengthen reasoning behavior and highlight the task settings and failure modes under which their gains may diminish.

Several recent surveys have covered adjacent areas, including test-time scaling, general LLM reasoning, and post-training scaling [233; 71; 84]. However, these works typically emphasize a particular stage of scaling, a specific reasoning regime, or a broader architectural view of reasoning systems. More generally, existing surveys often organize reasoning methods by technique families (e.g., RAG, CoT, multi-agent systems, and RL). In contrast, our survey focuses on a different question: how different forms of scaling specifically influence reasoning. We organize the literature by *what form of computation or information is scaled* at inference or training time, which enables a unified comparison of otherwise dis-

Survey	Scope	Organizing lens	Trade-offs	Main emphasis
Zhang et al. [233]	Test-time scaling	*What, how, where, and how well* to scale	Yes	Taxonomy, assessment, and deployment guidance for *test-time* scaling.
Ke et al. [71]	LLM reasoning	*Regimes* and *architectures*, with input/output perspectives	Partial	Broad survey of inference scaling, learning to reason, and agentic systems.
Lai et al. [84]	Post-training scaling	*SFT, RLxF, and TTC* in post-training	Partial	Scaling after pre-training, especially alignment and post-training methodologies.
Ours	Scaling in LLM reasoning	**Input sizes, reasoning steps, reasoning rounds, and training-enabled reasoning**	**Yes**	**Reasoning-centric synthesis across scaling dimensions, with unified comparison of gains, costs, and failure modes.**

Table 1: Comparison with closely related recent surveys. Prior work focuses on test-time scaling, general reasoning regimes/architectures, or post-training scaling, while our survey emphasizes a unified, reasoning-centric view across multiple scaling dimensions.

Dimension	What is scaled	Main benefit	Main cost	Typical failure mode	Best-fit tasks
Input sizes	External information / context / demonstrations	Better grounding and task adaptation	Retrieval and long-context cost	Irrelevant context, distraction, lost-in-the-middle	Knowledge-intensive QA, long-context tasks
Reasoning steps	Intermediate reasoning depth	Better decomposition and verification	Higher token and search cost	Overthinking, compounding errors	Math, logic, planning
Reasoning rounds	Interaction rounds across agents/humans	Critique, diversity, iterative refinement	Coordination and latency overhead	Redundancy, premature consensus, noisy debate	Open-ended reasoning, collaborative decision-making
Model optimization	Internal reasoning via optimization / latent computation	Stronger amortized reasoning	Training compute and data cost	Underthinking, overthinking, and compute-scaling saturation	High-value domains with reusable reasoning improvements

Table 2: Cross-dimensional comparison of scaling strategies for LLM reasoning.

connected lines of work in terms of their reasoning gains, costs, limitations, and characteristic failure modes. Specifically, we categorize reasoning scaling into four dimensions. We first discuss *input scaling*, which expands the external information available to the model through larger contexts, retrieval, demonstrations, or memory. We then examine *reasoning step scaling*, which allocates more intermediate computation to decomposition, verification, and search. Next, we study *reasoning round scaling*, in which LLMs iteratively refine their outputs through interaction, such as multi-agent collaboration, debate, and human-in-the-loop feedback. Finally, we consider *training-enabled reasoning*, which improves reasoning by internalizing stronger reasoning behaviors through optimization. Table 1 compares our survey with prior surveys, and Table 2 summarizes the core trade-offs across these four scaling dimensions, including what is scaled, their main benefits and costs, typical failure modes, and the task settings for which they are best suited.

Table 2 summarizes the core trade-offs across these four scaling dimensions, including what is scaled, their main benefits and costs, typical failure modes, and the task settings for which they are best suited. Although these dimensions are often studied separately, they differ systematically in where computation is allocated, what benefits they provide, and what limitations they introduce. The following sections examine each dimension in detail.

By systematically reviewing the scaling of reasoning in LLMs, this survey aims to bridge the gap between empirical scaling strategies and reasoning improvements. Beyond summarizing representative methods, we seek to clarify when and why scaling improves reasoning, where its returns diminish, and what new challenges it introduces. We hope this survey serves as a useful resource for both researchers and practitioners seeking effective, efficient, and reliable ways to advance LLM reasoning.

2. SCALING IN INPUT SIZES

Scaling input sizes expands the amount of information available to an LLM during reasoning, rather than increasing the depth of reasoning itself. This dimension improves performance by supplying richer contextual evidence, including demonstrations, retrieved documents, and persistent memory, which can enhance grounding, task adaptation, and long-horizon consistency. Its central trade-off is that additional context often improves coverage and robustness, but also increases retrieval overhead, long-context computation,

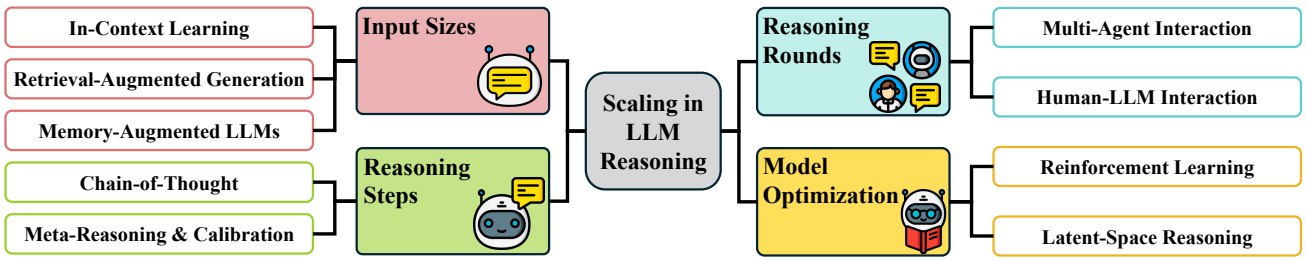

Figure 1: Taxonomy for Scaling in Large Language Model Reasoning.

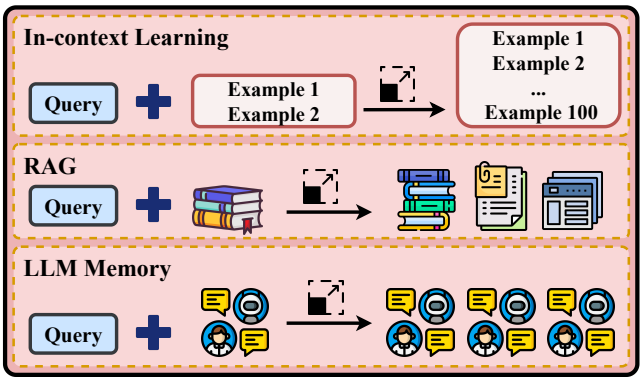

Figure 2: Scaling in LLM input sizes.

and the risk of distraction from irrelevant or poorly organized information. In this section, we examine three major strategies for input scaling—ICL, RAG, and memory-augmented LLMs—and analyze how they improve reasoning performance as well as the bottlenecks they introduce.

2.1 In-Context Learning

In-Context Learning (ICL) enables LLMs to adapt to new tasks without parameter updates by conditioning on demonstrations provided in the input prompt. Various algorithms have been developed to improve ICL performance by optimizing demonstration selection [186; 151; 222; 26], ordering [110; 105], and formatting [179; 109; 77]. More broadly, ICL illustrates a core form of input scaling: increasing the amount of task-relevant context so that the model can better infer the desired behavior from examples alone. Research has shown that model performance often improves as the number of in-context examples increases [1; 126; 11; 117]. However, traditional ICL methods remain constrained by the maximum input context length, which has historically limited them to the few-shot regime [38]. Although some works, such as SAICL [12], modify the attention structure to scale ICL to hundreds of demonstrations [93; 94; 55], they do not fully explore the broader benefits and challenges of operating with substantially larger demonstration sets.

With the expansion of context windows, researchers have increasingly investigated many-shot ICL, in which models leverage hundreds or even thousands of demonstrations [7; 2]. Scaling from few-shot to many-shot ICL has yielded substantial gains across a wide range of generative and discriminative tasks [169; 245; 140]. However, these gains are not unbounded: as the number of in-context demonstrations continues to grow, performance often plateaus and can even decline. This highlights a key limitation of input scaling: more context is only useful when the added information remains relevant, diverse, and well organized. To address these challenges, several methods have been proposed to improve the effectiveness and robustness of many-shot ICL [5; 234; 180]. For example, DrICL [234] adjusts demonstration weights using reinforcement-learning-inspired cumulative advantages to improve generalization, while BRIDGE [180] identifies a subset of influential examples and uses them to generate additional high-quality demonstrations.

2.2 Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) has become a widely adopted strategy to address the limitations of LLMs, such as hallucinations and restricted generalization to concepts beyond their training data [66; 70; 53; 88]. By incorporating retrieved external information, RAG enhances factual grounding and expands the model's accessible knowledge base. However, traditional RAG operates on short retrieval units, requiring the retriever to scan a massive document corpus to find relevant passages [147; 213; 15]. This approach is constrained by input context length limitations, making long-context RAG a challenge. A common strategy is document chunking [154; 214], where LLMs retrieve relevant chunks instead of full documents. However, defining optimal chunk boundaries is difficult, often leading to semantic incoherence and contextual loss, which degrade retrieval effectiveness [97]. Recent advances in long-context LLMs allow models to process millions of tokens [176]. Integrating RAG with long-context LLMs enables the processing of extended contexts while reducing semantic incoherence in chunked retrieval [96; 216; 215].

As input length increases, the burden on retrieval systems grows. LongRAG [65] mitigates this by grouping related documents, reducing the number of retrieval operations while maintaining relevance. ReComp [214] addresses this challenge by compressing retrieved documents into textual summaries before in-context integration, ensuring information remains concise yet informative. Despite these improvements, a key challenge known as "lost-in-the-middle" bias arises [108], where LLMs assign less importance to passages in the middle of a retrieved context. MOI [86] counters this bias by aggregating inference calls from permuted retrieval orders, ensuring a more balanced weighting across the retrieved information.

Another dimension of scaling RAG involves expanding the

amount of data available at inference time [181; 8; 182; 144]. Shao et al. [160] find that increasing datastore size monotonically improves performance across various language modeling and downstream tasks without clear saturation. Their MASSIVEDS datastore, containing trillions of tokens, is designed to support large-scale retrieval efficiently. Further, Yue et al. [229] explore inference-time scaling, showing that allocating more retrieval computation leads to nearly linear performance gains when optimally distributed. Their work introduces a predictive model for optimizing retrieval parameters under computational constraints. Together, these findings suggest that input scaling in RAG is effective not only through longer contexts, but also through larger and more searchable external knowledge stores.

2.3 Memory-Augmented LLMs

Scaling reasoning capabilities of LLMs often necessitates extending their effective context beyond the limited token windows supported by existing architectures [187]. Although increasing context length allows LLMs to process longer sequences, such scaling alone quickly encounters computational bottlenecks and diminishing returns due to quadratic complexity in attention mechanisms [44]. Moreover, even very long-context models struggle to efficiently capture and retrieve critical historical information from past interactions, leading to degraded reasoning performance over extended contexts [45]. To address these limitations, memory augmentation strategies have emerged, enabling LLMs to persistently store, manage, and dynamically retrieve relevant contextual information. Current memory augmentation approaches typically follow two directions: internal architectural modifications to enhance the model's inherent memory capabilities and external memory mechanisms that extend the model context through additional memory components.

Architectural adaptations focus on internalizing long-term dependencies within the model itself. This includes techniques such as augmenting attention mechanisms to better capture extended context [104; 114], refining key-value cache mechanisms to optimize retrieval efficiency over long sequences [95; 111], and modifying positional encodings to enhance length generalization [236; 237]. While effective, these modifications require direct intervention in the model's structure, making them impractical for proprietary or black-box API-based LLMs.

An alternative approach is the integration of external memory modules to supplement the model's limited native context window. Summarization-based methods [115; 185; 122; 188] condense past interactions into structured representations that can be efficiently retrieved during inference. However, fixed-granularity summarization risks fragmenting the discourse, leading to incoherent retrieval. To address this, recent advancements incorporate dynamic memory mechanisms that adaptively refine stored information. RMM [173] exemplifies this strategy by leveraging retrospective reflection to improve retrieval selection, ensuring that the model accesses the most relevant and contextually cohesive knowledge. Scaling memory-augmented LLMs requires balancing efficiency with contextual fidelity. A key challenge is mitigating memory saturation, where excessive storage of past interactions results in retrieval inefficiencies. Techniques such as hierarchical memory organization [160] and retrieval-conditioned compression [214] help alleviate this issue by structuring and filtering stored context dynamically. As research progresses, the convergence of retrieval-augmented memory with scalable long-context architectures offers a promising avenue for enabling LLMs to maintain reasoning consistency over prolonged interactions.

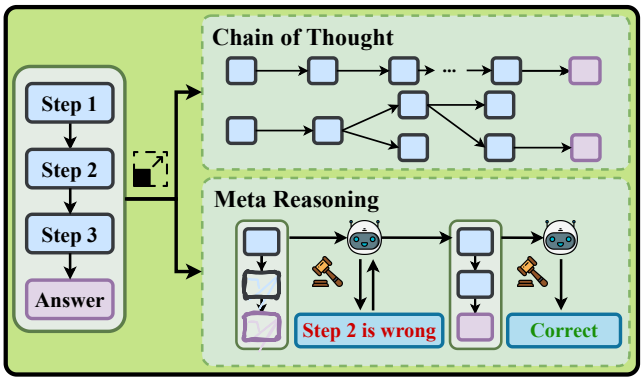

Figure 3: Scaling in LLM reasoning steps.

Overall, input scaling improves reasoning primarily by enriching the information available to the model, rather than by increasing the depth of intermediate reasoning or the amount of interactive refinement. This makes it especially effective for knowledge-intensive QA, long-context understanding, and conversational settings that depend on grounding in demonstrations, retrieved evidence, or prior interactions. At the same time, its gains are constrained by context quality, retrieval precision, and the model's ability to use long inputs effectively. Compared with later dimensions such as reasoning steps and reasoning rounds, the main bottleneck of input scaling is not insufficient inference depth, but whether the added context is relevant, well-structured, and accessible at the right time.

3. SCALING IN REASONING STEPS

Scaling reasoning steps increases the amount of intermediate computation allocated to solving a problem. Unlike input scaling, which improves reasoning by supplying more contextual information, this dimension improves performance by encouraging models to decompose problems, explore candidate solution paths, iteratively refine intermediate outputs, and verify correctness before committing to an answer. Such additional reasoning depth can substantially improve logical consistency and problem-solving ability, especially on tasks requiring structured multi-step inference. However, it also introduces important trade-offs, including higher token and search costs, longer latency, and the risk that additional reasoning may am: Chain-of-Thought prompting and meta-reasoning techniques, and discuss both their benefits and strategies to mitigate the challenges that arise from deeper reasoning processes.

3.1 Chain-of-Thought

Chain-of-Thought (CoT) prompting has emerged as a key technique for improving the reasoning capabilities of LLMs by eliciting explicit step-by-step deliberation, either through zero-shot prompting [79] or few-shot demonstrations [194]. More broadly, CoT represents a direct form of step scaling: rather than supplying the model with more external infor-

mation, it allocates more inference-time computation to constructing intermediate reasoning traces. Since LLMs operate probabilistically [61; 82], greedy decoding does not always yield the best reasoning path or final answer [190]. To mitigate this limitation, repeated sampling approaches such as self-consistency [189] and Best-of-N [132; 10] generate multiple reasoning chains in parallel and then select the final answer based on criteria such as majority agreement, external reward models, or auxiliary verifiers. They improve robustness by exploring multiple candidate trajectories, while introducing substantial computational overhead.

Although simple parallel sampling is computationally straightforward, it remains inefficient and suboptimal by randomly allocating the test-time computation budget to less promising branches [203; 168]. To mitigate this issue, researchers have explored strategies that prioritize promising reasoning paths or intermediate steps over less viable alternatives to effectively prune the search space by applying tree search-enabled reasoning [189; 221; 133; 113; 159; 127; 75] Generally, it structures the reasoning process as a branching tree, where each node represents a discrete thinking step, and branches correspond to different potential solution paths. Like CoT which organizes reasoning in a hierarchical manner, tree search-enabled reasoning enables LLMs to decompose intricate problems into manageable components. However, LLM reasoning with tree search can maintain awareness of multiple hypothesis threads simultaneously and systematically explore the solution space through different search algorithms (e.g., BFS or DFS), making it more powerful for handling complex problems.

The pioneering work CoT-SC [189] extends CoT to the tree structure, where multiple CoTs originate from the same initial (root) prompt, forming a "tree of chains". The chain that provides the best outcome to the initial question, is selected as the final answer. Skeleton-of-Thought (SoT) [133] instead effectively harnesses a tree with a specific level of depth. It performs reasoning through a divide-and-conquer manner, which significantly reduces the generation latency of LLMs. In the first prompt, the LLM is instructed to generate a skeleton of the answer, i.e., a list of points that can be answered independently. For each point, a new prompt is issued in parallel to address only the corresponding part of the question.

Recently, numerous studies have explored Tree of Thoughts (ToT) [221; 113] for tree search-enabled reasoning. Compared to CoT-SC where multiple CoTs originate from the same initial (root) prompt, ToT employs a tree structure to decompose a problem into subproblems and solve them using separate LLM prompts. Unlike ToT using multiple prompts, Algorithm of Thoughts (AoT) [159] uses only a single prompt with in-context examples formulated in an algorithmic fashion. Tree of Uncertain Thought (TouT) [127] enhances ToT with local uncertainty scores by incorporating the variance of multiple LLM responses into the state evaluation function. Tree of Clarifications (ToC) [75] focuses on answering ambiguous questions using ToT. It first retrieves relevant external information and then recursively prompts an LLM to construct a disambiguation tree for the question.

3.2 Meta-Reasoning and Calibration

Numerous works [35; 142; 49; 231; 67] have shown that LLMs have inherited capabilities of self-correction with proper prompt engineering. Typically, an LLM can self-reflect its responses by generating feedback on its answers. It first generates an initial response to an input question. Next, it generates feedback given the original input and its initial response. Finally, it generates a refined response given the input, initial response, and feedback. Generally, self-correction may rely on different sources of feedback, including intrinsic prompts and external information. Intrinsic prompts let LLMs generate feedback on their own responses. For example, CoVe [35] plans verification questions to check an initial response and then systematically answers those questions in order to finally produce an improved revised response. FLARE [67] performs self-correction by iteratively generating a temporary next sentence and check whether it contains low-probability tokens. In contrast, external information enables LLMs to rely on external tools, such as external knowledge from search engines, oracle information, and task-specific metrics, to enhance self-correction. For example, REFINER [142] interacts with a critic model that provides automated feedback on the reasoning. CRITIC [49] interacts with external tools like search engines and code interpreters to verify the desired aspects of an initial output and subsequently amends the output based on the critiques from the verification.

One major concern centers around the efficiency of self-refinement: LLMs need to generate feedback and refined responses iteratively, which can significantly increase the inference time of LLMs. To overcome the scaling issue, Quiet-STaR [231] designs a tokenwise parallel sampling algorithm, using learnable tokens indicating a thought's start and end, and an extended teacher-forcing technique. Another concern is caused by generation-time correction. Prevalent self-correction approaches are based on generation-time correction, heavily depending on the capacity of the critic model to provide accurate quantifiable feedback for intermediate outputs. Nevertheless, this might be quite challenging for many NLP tasks with long token sizes, such as summarization—the summary can be accurately assessed only after the entire summary is generated. This limitation makes generation-time correction infeasible in many NLP tasks. One solution to this issue is post-hoc correction [138]. Unlike general generation-time correction which generates feedback on the intermediate reasoning steps, post-hoc correction involves refining the output after it has been generated.

Overall, step scaling improves reasoning by allocating more computation to decomposition, search, verification, and correction. Compared with input scaling, which primarily addresses deficiencies in available evidence or context, step scaling is most effective when the main bottleneck lies in the reasoning process itself, as in mathematical, logical, and planning tasks. However, its gains are constrained by search efficiency, verification reliability, and the model's ability to avoid overthinking or compounding early mistakes. Relative to later dimensions such as reasoning rounds, which broaden reasoning through interaction, step scaling deepens a single model's inference process and therefore offers a more direct but often less diverse form of reasoning improvement.

4. SCALING IN REASONING ROUNDS

Scaling reasoning rounds expands the reasoning process through iterative interaction rather than through a single forward pass or a single model's internal chain of thought. Un-

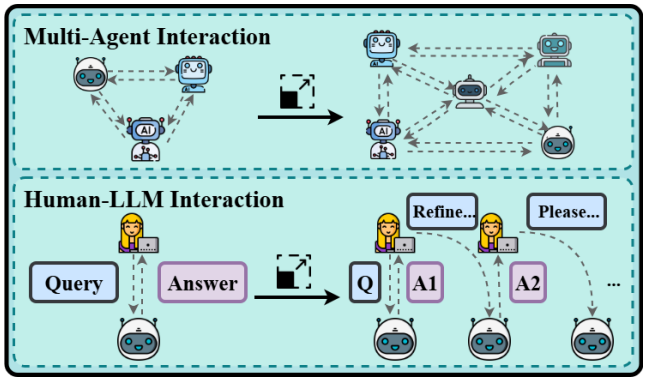

Figure 4: Scaling in reasoning rounds.

like input scaling, which enriches the information available to the model, and step scaling, which deepens intermediate computation within one reasoning trajectory, round scaling improves performance by introducing repeated communication, critique, and refinement across multiple turns. This additional interaction can increase diversity of perspectives, expose hidden errors, and support iterative improvement, but it also comes with significant trade-offs, including coordination overhead, latency, redundancy, and diminishing returns as the number of rounds grows. This section examines two major paradigms of round scaling: multi-agent interaction, where multiple LLMs coordinate or debate across rounds, and human–LLM interaction, where iterative human feedback guides and stabilizes the reasoning process.

4.1 Multi-Agent Interaction

Multi-agent interaction has emerged as a powerful paradigm for scaling LLM reasoning by enabling multiple models to iteratively exchange information, challenge assumptions, and refine outputs. Broadly, existing approaches fall into two major categories: *collaborative* frameworks, which emphasize specialization and division of labor, and *debate-based* frameworks, which introduce adversarial reasoning to expose errors and strengthen the final decision.

In collaborative settings, multiple LLMs work together in a coordinated manner to achieve improved problem-solving capabilities [100; 72; 90]. In particular, in these frameworks, each LLM (agent) is assigned a distinct role—such as planner, executor, verifier, or critic—and iteratively refines its output through structured interactions with other agents [217]. For example, CAMEL [90] introduced a framework where LLM agents assume different personas and interact through structured role-playing, enabling more effective task completion through multi-turn communication. The core idea is to enhance the specialization and division of labor among LLMs, ensuring that different agents contribute unique perspectives to improve overall task performance. Unlike single-agent systems, which rely on an LLM's internal reasoning capability [51; 198], multi-agent frameworks distribute tasks across multiple agents that engage in iterative interactions [90].

Increasing the number of agents can improve task diversity and allow for role specialization, where different agents assume distinct functions such as problem decomposition, tool usage, or evaluation [52]. Research has demonstrated that larger multi-agent systems can achieve greater accuracy and better adaptability in open-ended reasoning tasks, as seen in software development frameworks like MetaGPT [57]. However, these gains are not monotonic. Beyond a certain scale, performance may plateau or even decline due to conflicting reasoning paths, redundancy, and growing coordination overhead [100]. Similarly, [149] shows that structured dialogue among LLM agents improves reasoning depth and solution diversity, but also finds that too many interaction rounds lead to diminishing returns, as agents increasingly reinforce each other's biases rather than contribute genuinely new insights. These findings suggest that naive scaling of agent count or communication depth is insufficient; effective round scaling requires careful coordination protocols and complementary role assignment. Several works therefore introduce explicit communication structures to mitigate these issues. Hierarchical frameworks, in which some LLMs act as supervisors while others serve as task executors, have shown consistent gains in both accuracy and efficiency [13]. Another interesting finding is introduced in LLM Harmony [149], which optimizes inter-agent communication by structuring dialogue between multiple LLM agents. Instead of simple turn-based exchanges, this framework enables agents to dynamically negotiate task objectives, delegate subtasks, and refine outputs iteratively. The results suggest that scaling the number of interacting agents improves performance only when they are given complementary roles, while increasing homogeneous agents leads to redundant reasoning patterns.

In contrast to collaborative frameworks, debate-based methods deliberately assign *adversarial* roles to LLMs and often introduce an explicit judge. In these setups, each agent acts as a debater that challenges others and attempts to persuade a judge, with the goal of surfacing errors and stronger arguments rather than directly solving subtasks. A pioneering example is Multi-Agent Debate (MAD) [101], which proposes a structured debate protocol with a "tit-for-tat" mechanism: multiple debaters exchange arguments over several rounds, and a designated judge aggregates the discussion to reach a final decision. The key idea is to amplify disagreement and critical scrutiny. Compared with self-reflection approaches [120; 165], MAD induces stronger disagreement, helping to avoid premature convergence on incorrect answers. Building on this idea, subsequent debate-based frameworks improve reasoning robustness and factual accuracy by refining debate protocols and judge designs [40].

The scaling effect in debate frameworks manifests in multiple dimensions. In [73], the authors find that when employing a judge LLM to evaluate responses from debater LLMs, increasing the number of debate rounds does not necessarily lead to greater clarity—especially for weaker models, where additional rounds introduce confusion rather than improving accuracy. However, in consultancy-based interactions, where a single LLM attempts to persuade a judge LLM, the judge's accuracy improves over successive rounds. Notably, enhancing the persuasiveness of debater LLMs—making them more effective at convincing the judge—has been shown to yield performance improvements. This scaling effect provides further insights into optimizing debate-based reasoning frameworks. Similarly, [125] suggests that scaling LLM debates with increasingly skilled debaters (e.g., progressing from AI to human debaters) enhances oversight mechanisms, improving overall debate efficacy, whereas consul-

tancy frameworks tend to perform worse under similar conditions. Distinct from these approaches, CIPHER [143] proposes embedding-based communication to facilitate debate, enabling smaller LLMs to retain stronger debate capabilities by mitigating information loss. Their findings indicate that increasing the number of debate rounds improves performance up to a threshold of three rounds, beyond which additional rounds provide diminishing returns. Overall, multi-agent interaction shows that round scaling can improve reasoning not only by increasing deliberation length, but also by introducing complementary roles, disagreement, and iterative critique. At the same time, it reveals a defining challenge of this dimension: additional rounds are helpful only when they generate genuinely new information or perspectives, rather than repeated, biased, or poorly exchanges.

4.2 Human-LLM Interaction

Reasoning rounds can also be scaled through iterative interaction between humans and LLMs. In this setting, improvement does not come from communication among multiple models, but from repeated user guidance that helps the model clarify goals, correct mistakes, and refine its responses. This human-in-the-loop paradigm shifts LLMs from static inference engines to adaptive assistants whose reasoning can be steered and stabilized through feedback [3; 202]. Recent work explores multi-turn reasoning scenarios where users provide incremental clarifications or corrections, allowing models to refine their responses iteratively [120; 80]. This process mirrors how humans engage in collaborative problem-solving, gradually converging on an accurate and well-structured answer. Methods such as self-reflection prompting [165] and feedback-based reinforcement learning [17] demonstrate improvements in factual consistency and reasoning depth by enabling LLMs to assess and revise their outputs.

A key challenge in human-LLM interaction is balancing efficiency with adaptability. Over-reliance on explicit feedback mechanisms can introduce cognitive overhead for users, while insufficient adaptability limits the model's ability to incorporate nuanced human guidance. Recent strategies mitigate this tradeoff through adaptive interaction mechanisms, such as retrieval-enhanced dialogue memory [139] and user-intent modeling [92], allowing LLMs to anticipate user needs and refine responses proactively.

As interaction frameworks scale, ensuring alignment with human cognitive processes remains critical. Fine-tuning strategies that incorporate user feedback loops have shown promise in enhancing model interpretability and trustworthiness [76]. Furthermore, inference-time intervention mechanisms [123; 172] enable LLMs to allocate computational resources efficiently based on user engagement patterns. By refining the synergy between LLMs and human oversight, interactive reasoning systems hold the potential to scale beyond static prompt-response architectures, evolving towards more adaptive and contextually aware AI assistants.

Overall, round scaling improves reasoning by broadening the inference process through interaction, critique, and iterative refinement. Compared with step scaling, which deepens a single reasoning trajectory, round scaling introduces external feedback and perspective diversity, making it particularly useful for open-ended reasoning, oversight, and collaborative decision-making. However, its gains are constrained by communication quality, role complementarity, and coordination efficiency, and its characteristic failure modes include redundancy, premature consensus, noisy debate, and excessive reliance on user effort. Relative to training-enabled reasoning, which seeks to internalize better reasoning policies in advance, round scaling remains more flexible at inference time but also more dependent on well-designed interaction protocols.

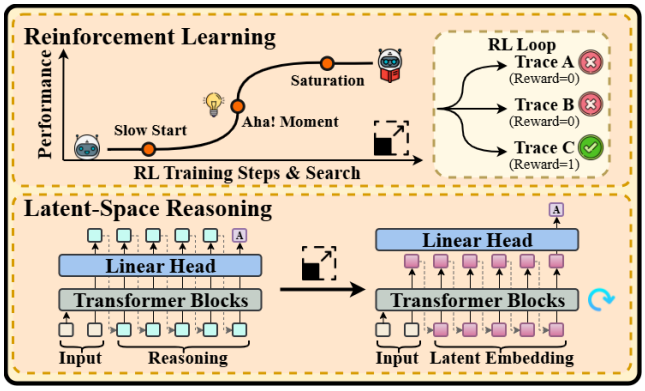

Figure 5: Scaling in model optimization.

5. SCALING IN MODEL OPTIMIZATION

Scaling model optimization improves reasoning not by supplying more external information or extending explicit inference trajectories, but by strengthening the model's internal reasoning capacity through training or latent computation. Unlike the previous three dimensions, which primarily allocate additional resources at inference time, this dimension seeks to internalize better reasoning policies in advance or to expand internal computation without proportionally increasing explicit reasoning length. Reinforcement learning (RL)–based methods scale reasoning by optimizing the model's policies over increasingly complex tasks, aligning behavior with human intentions and enabling deeper multi-step inference. Complementing RL approaches, latent-space reasoning methods scale internal computation without increasing sequence length or model size. By iterating over hidden representations, such as an in-looped transformer, these methods allow models to perform additional internal reasoning steps, effectively scaling "thinking time" while keeping inference cost manageable. Together, Reinforcement Fine-Tuning (RFT) and latent-space reasoning illustrate how optimization-time scaling can deepen LLM reasoning capacity, offering alternatives to traditional scaling via larger models or longer explicit reasoning traces.

5.1 Reinforcement Learning

Although previous studies have shown that distilling knowledge from superior LLMs, regardless of whether supervised fine-tuning (SFT) data are amassed in large quantities or carefully curated [240; 223], can enhance the reasoning abilities of smaller models for solving complex tasks [166; 121; 56], recent studies contend that, merely increasing the volume of SFT data typically yields only a log-linear performance improvement [228]. Moreover, models trained exclusively on SFT data tend to overfit by memorizing the training set, thereby struggling to generalize to out-of-distribution (OOD) tasks [30]. To mitigate these issues, reinforcement

learning (RL) has emerged as a key approach in LLM post-training, aligning models with human preferences [136; 148] and enhancing their reasoning abilities [161; 218; 50].

Recent studies indicate that conducting RL-based fine-tuning following SFT can further improve the reasoning abilities of LLMs. ReFT [119] first performs a warm-up SFT on distilled CoT data followed by fine-tuning the SFT model using Proximal Policy Optimization (PPO) [158] on the same training questions, which eventually leads to significant performance gains on mathematical reasoning tasks. T1 [58] employs a similar training strategy, but emphasizes scaling sampling diversity during RL training through techniques such as high-temperature sampling, on-policy KL normalization, and rule-based reward penalties for undesirable repetition responses. They observed that increasing the number of sampled responses, raising the sampling temperature during RL training, and extending inference-time reasoning steps collectively contribute to improved reasoning performance. DeepSeek-R1 [50] shares a similar strategy as ReFT but employs self-training by directly applying Group Relative Policy Optimization (GRPO) [161] to the base model. This base model is then used to generate long-form CoT data for the warm-up SFT stage, after which GRPO is applied again to the SFT model, ultimately achieving reasoning performance comparable to OpenAI-o1 [60]. They observed an "aha-moment" during the training of DeepSeek-R1-Zero, where the model learned to rethink as the response length increased. Following DeepSeek-R1, recent works observed similar "aha-moment" and think related words on different tasks, including real-world software engineering [197], logical puzzles [210], and automated theorem proving [37] when scaling up the training steps and response length using RL-based fine-tuning.

At the same time, RL-trained reasoning models that produce long CoT traces can exhibit notable failure modes, including "underthinking" [191], where the model frequently switches between shallow reasoning branches without engaging in sustained deliberation, and "overthinking" [22], where excessive reasoning on simple instances can degrade accuracy. Beyond these empirical observations, recent work has begun to systematically characterize the compute–scaling behavior of RL post-training. ScaleRL [74] demonstrates that RL reward improvements follow a predictable sigmoidal compute–performance curve, with early low-gain regions, a sharp transition phase, and a clear asymptotic limit. Their analysis shows that many commonly tuned components, including curriculum design, normalization, and loss aggregation, primarily affect compute efficiency, whereas only a small subset of architectural or algorithmic choices (e.g., loss formulation, precision handling, off-policy setup) materially shifts the ultimate performance ceiling. In parallel, ProRL [107] provides complementary evidence that prolonged RL optimization can elicit qualitatively new reasoning strategies even in relatively small models, though it does not explicitly analyze the predictability of compute scaling. Taken together, these studies suggest that RL post-training is not only empirically effective but also exhibits a structured, saturating scaling pattern, underscoring the importance of understanding compute–performance curves rather than evaluating methods solely at isolated endpoints.

5.2 Latent-Space Reasoning

A second pathway for optimization-based scaling increases reasoning capacity by allocating additional computation in latent space rather than through longer explicit reasoning traces. In explicit reasoning [194], models generate intermediate natural-language steps before producing a final output. While such reasoning improves interpretability and decomposition, it can also be verbose and computationally expensive. Latent-space reasoning aims to address this limitation by performing additional computation over hidden representations without requiring every intermediate thought to be verbalized [33; 164]. This makes it possible to scale "thinking time" without proportionally increasing sequence length or model size.

Several recent methods instantiate this idea in different ways. Deng et al. [33] propose distilling multi-step reasoning into latent representations across layers, allowing the model to solve complex problems in a single forward pass while improving efficiency and scalability. CoCoMix [170] trains LLMs to predict selected semantic concepts directly from hidden states; by interleaving token embeddings with high-level continuous concepts, it enhances abstract reasoning while reducing data and computation costs. More broadly, this line of work reflects a key observation: natural language is not always the most efficient substrate for reasoning. Hao et al. [54] argue that many word tokens mainly serve textual coherence rather than reasoning itself, whereas only certain critical tokens require deeper planning. Based on this insight, Coconut [54] iteratively processes hidden states and enables parallel exploration of multiple reasoning paths in latent space. Additional work explores how iterative latent computation can deepen reasoning without requiring parameter expansion. ITT [23], for example, dynamically allocates computation to critical tokens and iteratively refines hidden representations. The same iterative paradigm has also been explored for test-time scaling [47; 129], where repeated latent transformations improve efficiency relative to explicitly lengthening verbalized reasoning. Similarly, Saunshi et al. [155] show that model depth can effectively be scaled under a limited parameter budget through looping, introducing a new scaling paradigm based on iterative latent-space transformations rather than simply enlarging the model. Collectively, these methods suggest that deeper reasoning need not always correspond to longer visible chains of thought; in some settings, the key resource being scaled is internal computation itself.

Overall, optimization-based scaling improves reasoning by internalizing stronger reasoning policies or by expanding internal computation through latent iterative processing. Compared with input scaling, step scaling, and round scaling, which primarily invest additional resources at inference time, this dimension shifts the trade-off toward up-front optimization cost in exchange for potentially reusable reasoning improvements across many downstream tasks. This makes it particularly attractive in high-value settings where stronger reasoning behavior can be amortized over repeated deployment. However, its gains are constrained by optimization stability, reward fidelity, transferability, and the difficulty of understanding how internalized or latent reasoning generalizes beyond the training conditions.

6. APPLICATION

6.1 AI Research

Scaling in LLMs has fundamentally reshaped AI research, both extending traditional domains and opening entirely new research avenues. This section explores how scaling has influenced three critical areas: LLM-as-a-Judge, fact-checking, and dialogue systems.

LLM-as-a-Judge. Using LLMs to evaluate model outputs or other models has emerged as a pivotal research direction, enabling evaluation at scale beyond traditional approaches and human assessment [89]. Notably, larger models demonstrate a significantly higher correlation with human preferences compared to their smaller counterparts [238]. To further improve evaluation quality, recent work has explored multi-step reasoning processes [152], where scaling the number of reasoning steps enhances evaluation capabilities [29]. Additionally, scaling across multiple judge models has emerged as an effective approach to improve evaluation reliability [99]. Different LLMs functioning as agents collaborate through multi-round discussions before reaching a final judgment, thereby enhancing evaluation consistency [146].

Fact-Checking. The capacity of AI systems to generate misinformation has driven substantial research into automated fact checking [200; 32; 242]. Initial fact verification approaches relied on smaller models with limited contextual understanding, primarily focusing on matching claims to evidence [32]. Large-scale LLMs have shown remarkable fact-checking capabilities by supporting fact-checkers with their extensive knowledge and sophisticated reasoning [175]. Scaling in reasoning steps has been demonstrated to improve claim detection, making the process more methodical [157]. Additionally, RAG has been employed for evidence-backed fact-checking with reduced hallucination and improved performance, with performance scaling with the number of retrieved documents [167]. Multi-agent systems have been widely implemented for fact-checking, where multiple imperfect fact-checkers can collectively provide reliable assessments [178].

Dialogue Systems. Dialogue systems represent the most visible application of LLM scaling [224; 239; 43], where advances in context length, reasoning steps, and training data have dramatically transformed interactive capabilities. Enhanced context handling has significantly impacted dialogue coherence and consistency. Scaling of context provides dialogue agents with more information, enabling more informative long-term conversations [6; 173]. External augmentation has been widely adopted to facilitate long-term dialogue as well. Commonly integrated external knowledge, including commonsense [183], medical [21], and psychological [24] knowledge, serves as supplementary guidance for the reasoning process, ensuring logical coherence across extended contexts. Multi-agent dialogue systems have also demonstrated exceptional capabilities, where multiple LLMs collaborate to comprehensively evaluate and select the most appropriate responses [42].

6.2 Production

The scaling reasoning capabilities of LLMs have significantly enhanced production applications, particularly in software development, data science workflows, and interactive AI systems. This subsection discusses these areas with illustrative examples.

Software Development. The scaling reasoning capabilities of LLMs enhance software development by enabling a better understanding of complex coding tasks and facilitating accurate multi-step reasoning over intricate software dependencies and structures. Advanced reasoning techniques, such as chain-of-thought prompting, allow code-generation assistants to systematically approach and solve coding tasks [20; 64]. Furthermore, structured reasoning strategies can effectively handle larger coding contexts and reduce developer cognitive load during debugging and iterative improvement processes [64].

Data Science Workflows. Scaling reasoning in LLMs substantially improves data science workflows by enabling sophisticated analytical and exploratory tasks. Multi-step reasoning allows LLMs to iteratively explore, interpret, and synthesize insights from diverse datasets [171], effectively supporting hypothesis generation and validation processes [162; 201]. Retrieval-augmented reasoning frameworks extend these capabilities further by dynamically integrating external knowledge during reasoning, thus enriching the comprehensiveness of exploratory analysis [144]. Multi-agent systems are also proposed to collaboratively solve real-world data science challenges [98].

Interactive AI Systems. Scaling reasoning steps and context length transforms interactive AI systems by significantly improving their adaptability and context-awareness. Expanded reasoning capabilities enable dialogue agents to maintain coherent and informative long-term interactions, effectively integrating historical context and external knowledge [6; 43]. Multi-agent systems leverage iterative refinement and structured verification among specialized reasoning agents, further enhancing accuracy and reducing errors such as hallucinations [42]. Interactive AI environments such as LLM-based Cursor [34] leverage LLMs' contextual reasoning to facilitate precise user interactions, enabling targeted queries and refined outputs.

6.3 Science

The scaling of LLMs has significantly benefited scientific domains, with medicine, finance, and disaster management emerging as prominent application areas.

Medical Domain. The medical domain has experienced remarkable advances through scaled LLMs. Research demonstrates that increasing model size leads to enhanced medical reasoning capabilities, with performance on medical questions improving proportionally [9; 102; 116]. This pattern extends to diagnostic reasoning [48; 156], where larger models can identify complex disease progression patterns that smaller models miss [230; 46]. Multi-round reasoning approaches such as CoT have demonstrated exceptional effectiveness in medical diagnosis [199; 106], with additional reasoning steps yielding more accurate diagnoses [59; 16] by enabling consideration of alternative explanations and confounding factors. RAG techniques enhance medical question answering, with performance improving as the number of retrieved snippets increases [212]. Many-shot ICL shows particular efficacy for drug design tasks, with performance scaling with the number of examples provided [128]. Additionally, multi-agent LLM frameworks that simulate medical team consultations have demonstrated superior diagnostic accuracy, with specialized agents collaborating on complex cases to outperform single LLMs when benchmarked against

gold-standard diagnoses [41; 78].

Finance. Financial applications demonstrate improved performance with large-scale LLMs. Studies indicate that fine-tuned large-scale LLMs substantially outperform smaller alternatives [68], with performance scaling with model size [145; 91]. The multi-step reasoning capabilities of scaled LLMs prove particularly valuable for complex financial analysis, significantly outperforming direct approaches [243; 145]. Financial sentiment analysis benefits from increased numbers of examples in many-shot ICL scenarios [2; 196]. RAG-based approaches incorporating banking webpages and policy guides improve question-answering performance, with results scaling with the number of retrieved documents [235]. Multi-agent debate frameworks yield promising results in investment and trading decision scenarios [227; 226; 209], with specialized agents covering distinct functions outperforming single-agent approaches.

Disaster Management. Disaster management has undergone substantial transformation through large-scale LLMs [87]. Social media text classification for disaster types has improved significantly through LLM fine-tuning compared to traditional machine learning methods [225; 39]. The in-context learning capabilities of large-scale LLMs enable context-aware disaster applications including conversational agents for disaster-related queries and situational analysis [135; 150]. Large-scale disaster knowledge graphs enhance in-context learning through retrieval augmentation, enabling LLMs to generate more informative and less hallucinated responses [19; 205]. For high-stakes disaster-related decision-making, multi-agent LLM approaches have been effectively deployed to facilitate adaptive and collaborative decision processes [36; 177], largely outperforming a single LLM.

7. FUTURE DIRECTIONS

Efficiency in Scalable Reasoning. Scaling reasoning capability in LLMs enhances their ability to solve complex problems but also increases response length, making it inefficient for simpler tasks. However, current LLMs apply uniform reasoning effort across all queries, leading to unnecessary computational overhead. A key direction for improvement is adaptive reasoning frameworks, where models dynamically adjust the depth of reasoning based on task difficulty [232; 195]. For example, "Proposer-Verifier" framework [168] offers a promising approach by generating multiple candidate solutions and selecting the most reliable one through verification, reducing redundant reasoning steps while maintaining accuracy. However, achieving dynamic computation allocation requires robust uncertainty estimation, ensuring that models allocate resources efficiently without excessive overhead.

Another challenge is balancing search-based reasoning methods with computational cost. Approaches like ToT and Monte Carlo search refine reasoning iteratively but incur significant compute overhead. Selective pruning strategies that eliminate irrelevant reasoning paths while maintaining solution integrity could help optimize performance [211]. Additionally, RL-based multi-step reasoning faces credit assignment issues, where sparse rewards make optimizing intermediate reasoning steps difficult [82]. Future work should explore hybrid reward models [163] that combine process-based supervision (evaluating stepwise correctness) with some outcome-based rewards (final answer validation) to improve long-horizon reasoning stability and efficiency.

Beyond single-model scaling, collaborative multi-agent systems present a promising avenue for large-scale reasoning [85; 137], but they also introduce significant coordination overhead. As the number of agents increases, computational redundancy and inefficient communication can slow down reasoning instead of improving it [51]. One approach to mitigate this is dynamic agent selection [112], where the system dynamically selects only the most relevant agents for a given reasoning task while discarding redundant ones. Another strategy is hierarchical multi-agent reasoning, where a smaller subset of expert agents handles complex queries, while simpler queries are resolved by lightweight, lower-cost agents. Additionally, inter-agent communication should be optimized through compressed latent representations rather than verbose token-based exchanges, further reducing computational overhead [244]. Future research should explore pruning and optimization techniques that enable multi-agent systems to scale efficiently without unnecessary computational waste, ensuring that reasoning is distributed optimally across agents.

Inverse Scaling and Stability. Inverse scaling refers to the phenomenon where LLMs unexpectedly perform worse on certain tasks, contradicting standard scaling laws that predict consistent improvements with increased model size. Lin et al. [103] first observed this effect when evaluating LLMs such as GPT-2 and GPT-3 on truthfulness tasks, noting that common training objectives incentivize imitative falsehoods, where models produce false but high-likelihood responses due to patterns in their training distribution. McKenzie et al. [124] systematically analyzed different datasets exhibiting inverse scaling and identified key causes like solving distractor tasks instead of intended tasks.

While inverse scaling is widely observed, Wei et al. [192] challenge its universality, showing that some tasks previously exhibiting inverse scaling follow a U-shaped scaling trend—where performance initially declines with increasing model size but later recovers at even larger scales. This suggests that larger models can sometimes unlearn distractor tasks and correct their errors, emphasizing the importance of evaluating scaling trends beyond mid-sized models.

Since scaling laws were originally developed in the context of pretraining, they remain decoupled from downstream task performance, making it an open question of how to systematically predict and mitigate inverse scaling across different reasoning benchmarks. Additionally, challenges like reward hacking [4]—where models exploit superficial signals rather than true reasoning improvements—necessitate adaptive reward models to maintain stability in multi-step reasoning.Future work should focus on developing predictive models for inverse scaling, refining adaptive fine-tuning methods, and leveraging world models for richer environmental feedback, ensuring that multi-step reasoning generalizes effectively across domains such as code generation, planning, question answering, and cross-lingual tasks.

Security Risks in Scaled Reasoning Models. While CoT prompting enhances LLMs' ability to perform structured reasoning, it also introduces new security vulnerabilities, particularly backdoor attacks that manipulate the model's reasoning process. BadChain [207] exploits the model's step-by-step reasoning by injecting backdoor reasoning steps,

causing malicious alterations in the final response when a hidden trigger is present in the query. Similarly, H-CoT [83] manipulates the model's internal reasoning pathways, hijacking its safety mechanisms to weaken its ability to detect harmful content. While defenses such as backdoor detection (CBD) [208] and modified decoding strategies [63] offer some protection, their effectiveness against novel attacks remains largely unexplored. This highlights the urgent need for more robust defenses capable of adapting to emerging threats.

Unlike CoT, RAG integrates external data sources, making them prone to data extraction attacks [28]. Existing defenses primarily focus on retrieval corruption attacks [206; 174; 241], aiming to maintain performance, but data leakage prevention remains an underexplored area. For example, RAG-Thief demonstrates how attackers can extract scalable amounts of private data from proprietary retrieval databases [62]. Beyond attacks on individual LLMs, the scaling of multi-agent reasoning systems introduces new attack surfaces. AgentPoison [27] specifically targets RAG-based and memory-augmented LLM agents, poisoning long-term memory or altering the knowledge base to induce faulty reasoning over time. As multi-agent LLM systems grow in scale, collusive behaviors among malicious agents present an even greater risk [219]. BlockAgents proposes a blockchain-integrated framework for LLM-based cooperative multi-agent systems, mitigating Byzantine behaviors that arise from adversarial agents [14].

As AI adoption increases, the computational and environmental costs of inference also become a growing concern [118; 153; 141]. Large-scale LLMs demand significant energy resources on inference [141]. This opens the door to a new form of attack, OverThink attack [81], where an adversary intentionally inflates the number of reasoning tokens in an LLM's response, drastically increasing financial and computational costs. As LLM reasoning continues to scale, deploying cost-effective safeguards against such attacks will become necessary for sustainable AI deployment.

8. CONCLUSION

In this survey, we presented a comprehensive view of how different scaling strategies shape the reasoning capabilities of large language models. We organized the literature along four major dimensions: input information, reasoning steps, reasoning rounds, and model optimization, and discussed the key methods, benefits, trade-offs, and failure modes associated with each. Our analysis highlights that scaling can substantially improve LLM reasoning across a wide range of domains, but these gains are not uniform: they often come with increased computational cost, instability, diminishing returns, and emerging safety and security risks. We further outlined several promising directions for future research, including adaptive computation allocation, more robust and stable optimization, principled evaluation beyond final-answer accuracy, and safer multi-agent and human-LLM interaction. As LLMs continue to advance, a deeper understanding of how to scale reasoning effectively and responsibly will be essential for building AI systems that are not only more capable but also more efficient, reliable, and trustworthy.

9. ACKNOWLEDGEMENTS

This work is supported in part by the National Science Foundation (NSF) under grants IIS-2006844, IIS-2144209, IIS-2223769, CNS-2154962, BCS-2228534, CMMI-2411248, ECCS-2143559, and CPS-2313110; the Office of Naval Research (ONR) under grant N000142412636; and the Commonwealth Cyber Initiative (CCI) under grant VV1Q24-011.

10. REFERENCES

[1] A. Abedsoltan, A. Radhakrishnan, et al. Context-scaling versus task-scaling in in-context learning. *arXiv*, 2024.

[2] R. Agarwal, A. Singh, et al. Many-shot in-context learning. *NeurIPS*, 2024.

[3] D. Alsagheer, R. Karanjai, et al. Comparing rationality between large language models and humans: Insights and open questions. *arXiv*, 2024.

[4] D. Amodei, C. Olah, et al. Concrete problems in ai safety. *arXiv*, 2016.

[5] J. Baek, S. J. Lee, et al. Revisiting in-context learning with long context language models. *arXiv*, 2024.

[6] J. Bang, H. Noh, et al. Example-based chat-oriented dialogue system with personalized long-term memory. In *BIGCOMP*, 2015.

[7] A. Bertsch, M. Ivgi, et al. In-context learning with long-context models: An in-depth exploration. *arXiv*, 2024.

[8] S. Borgeaud, A. Mensch, et al. Improving language models by retrieving from trillions of tokens. In *ICML*, 2022.

[9] D. Brin, V. Sorin, et al. How large language models perform on the united states medical licensing examination: a systematic review. *MedRxiv*, 2023.

[10] B. Brown, J. Juravsky, et al. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv*, 2024.

[11] T. Brown, B. Mann, et al. Language models are few-shot learners. *NeurIPS*, 2020.

[12] T. Cai, K. Huang, et al. Scaling in-context demonstrations with structured attention. *arXiv*, 2023.

[13] T. Cai, X. Wang, et al. Large language models as tool makers. *arXiv*, 2023.

[14] B. Chen, G. Li, et al. Blockagents: Towards byzantine-robust llm-based multi-agent coordination via blockchain. In *ACM TURC*, 2024.

[15] D. Chen, A. Fisch, et al. Reading wikipedia to answer open-domain questions. *arXiv*, 2017.

[16] J. Chen, Z. Cai, et al. Huatuogpt-o1, towards medical complex reasoning with llms. *arXiv*, 2024.

[17] K. Chen, M. Cusumano-Towner, et al. Reinforcement learning for long-horizon interactive llm agents. *arXiv*, 2025.

[18] L. Chen, J. Q. Davis, et al. Are more llm calls all you need? towards the scaling properties of compound ai systems. *NeurIPS*, 2024.

[19] M. Chen, Z. Tao, et al. Enhancing emergency decision-making with knowledge graphs and large language models. *IJDRR*, 2024.

[20] M. Chen, J. Tworek, et al. Evaluating large language models trained on code. *arXiv*, 2021.

[21] S. Chen, M. Wu, et al. Llm-empowered chatbots for psychiatrist and patient simulation: application and evaluation. *arXiv*, 2023.

[22] X. Chen, J. Xu, et al. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv*, 2024.

[23] Y. Chen, J. Shang, et al. Inner thinking transformer: Leveraging dynamic depth scaling to foster adaptive internal thinking. *arXiv*, 2025.

[24] Y. Chen, X. Xing, et al. Soulchat: Improving llms' empathy, listening, and comfort abilities through fine-tuning with multi-turn empathy conversations. *arXiv*, 2023.

[25] Z. Chen, A. H. Cano, et al. Meditron-70b: Scaling medical pretraining for large language models. *arXiv*, 2023.

[26] Z. Chen, S. Wang, et al. Fastgas: Fast graph-based annotation selection for in-context learning. *ACL*, 2024.

[27] Z. Chen, Z. Xiang, et al. Agentpoison: Red-teaming llm agents via poisoning memory or knowledge bases. *NeurIPS*, 2024.

[28] P. Cheng, Y. Ding, et al. Trojanrag: Retrieval-augmented generation can be backdoor driver in large language models. *arXiv*, 2024.

[29] C.-H. Chiang, H.-y. Lee, et al. Tract: Regression-aware fine-tuning meets chain-of-thought reasoning for llm-as-a-judge. *arXiv*, 2025.

[30] T. Chu, Y. Zhai, et al. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv*, 2025.

[31] H. W. Chung et al. Scaling instruction-finetuned language models. *JMLR*, 2024.

[32] G. Demartini, S. Mizzaro, et al. Human-in-the-loop artificial intelligence for fighting online misinformation: Challenges and opportunities. *IEEE Data Eng. Bull.*, 2020.

[33] Y. Deng, K. Prasad, et al. Implicit chain of thought reasoning via knowledge distillation. *arXiv*, 2023.

[34] D. S. R. Devi, O. U. C. BhagyaSri, R. Sravanthi, S. Chaitrika, M. Priyanka, M. Swarna, and M. Srilekha. Ai-enhanced cursor navigator. *R. and Chaitrika, SL and Priyanka, MN and Swarna, M. and Srilekha, M., AI-Enhanced Cursor Navigator (May 10, 2024)*, 2024.

[35] S. Dhuliawala, M. Komeili, et al. Chain-of-verification reduces hallucination in large language models. In *ACL*, 2024.

[36] A. Dolant and P. Kumar. Agentic llm framework for adaptive decision discourse. *arXiv*, 2025.

[37] K. Dong and T. Ma. Stp: Self-play llm theorem provers with iterative conjecturing and proving. *arXiv*, 2025.

[38] Q. Dong, L. Li, et al. A survey on in-context learning. In *EMNLP*, 2024.

[39] V. G. dos Santos, G. L. Santos, et al. Identifying citizen-related issues from social media using llm-based data augmentation. In *CAiSE*, 2024.

[40] Y. Du, S. Li, et al. Improving factuality and reasoning in language models through multiagent debate. *arXiv*, 2023.

[41] Z. Fan, J. Tang, et al. Ai hospital: Benchmarking large language models in a multi-agent medical interaction simulator. *arXiv*, 2024.

[42] J. Fang, S. Gao, et al. A multi-agent conversational recommender system. *arXiv*, 2024.

[43] L. Friedman, S. Ahuja, et al. Leveraging large language models in conversational recommender systems. *arXiv*, 2023.

[44] Z. Fu, W. Song, et al. Sliding window attention training for efficient large language models. *arXiv*, 2025.

[45] Y. Gao, Y. Xiong, et al. U-niah: Unified rag and llm evaluation for long context needle-in-a-haystack. *arXiv*, 2025.

[46] Á. García-Barragán, A. G. Calatayud, et al. Step-forward structuring disease phenotypic entities with llms for disease understanding. In *CBMS*, 2024.

[47] J. Geiping, S. McLeish, et al. Scaling up test-time compute with latent reasoning: A recurrent depth approach. *arXiv*, 2025.

[48] E. Goh and R. o. Gallo. Large language model influence on diagnostic reasoning: a randomized clinical trial. *JAMA Network Open*, 2024.

[49] Z. Gou, Z. Shao, et al. Critic: Large language models can self-correct with tool-interactive critiquing. *arXiv*, 2023.

[50] D. Guo, D. Yang, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv*, 2025.

[51] T. Guo, X. Chen, et al. Large language model based multi-agents: A survey of progress and challenges. *arXiv*, 2024.

[52] T. Guo, X. Chen, et al. Large language model based multi-agents: A survey of progress and challenges. *arXiv*, 2024.

[53] K. Guu, K. Lee, et al. Retrieval augmented language model pre-training. In *ICML*, 2020.

[54] S. Hao, S. Sukhbaatar, et al. Training large language models to reason in a continuous latent space. *arXiv*, 2024.

[55] Y. Hao, Y. Sun, et al. Structured prompting: Scaling in-context learning to 1,000 examples. *arXiv*, 2022.

[56] N. Ho et al. Large language models are reasoning teachers. *arXiv*, 2022.

[57] S. Hong, M. Zhuge, et al. Metagpt: Meta programming for a multi-agent collaborative framework. In *ICLR*, 2023.

[58] Z. Hou, X. Lv, et al. Advancing language model reasoning through reinforcement learning and inference scaling. *arXiv*, 2025.

[59] Z. Huang, G. Geng, et al. O1 replication journey–part 3: Inference-time scaling for medical reasoning. *arXiv*, 2025.

[60] A. Jaech, A. Kalai, et al. Openai o1 system card. *arXiv*, 2024.

[61] Z. Ji, N. Lee, et al. Survey of hallucination in natural language generation. *ACM Comp.Sur.*, 2023.

[62] C. Jiang, X. Pan, et al. Rag-thief: Scalable extraction of private data from retrieval-augmented generation applications with agent-based attacks. *arXiv*, 2024.

[63] F. Jiang, Z. Xu, et al. Safechain: Safety of language models with long chain-of-thought reasoning capabilities. *arXiv*, 2025.

[64] J. Jiang, F. Wang, et al. A survey on large language models for code generation. *arXiv*, 2024.

[65] Z. Jiang, X. Ma, et al. Longrag: Enhancing retrieval-augmented generation with long-context llms. *arXiv*, 2024.

[66] Z. Jiang, F. F. Xu, et al. Active retrieval augmented generation. In *EMNLP*, 2023.

[67] Z. Jiang, F. F. Xu, et al. Active retrieval augmented generation. In *EMNLP*, 2023.

[68] K. Kalluri. Scalable fine-tunning strategies for llms in finance domain-specific application for credit union, 2024.

[69] J. Kaplan, S. McCandlish, et al. Scaling laws for neural language models. *arXiv*, 2020.

[70] V. Karpukhin, B. Oguz, et al. Dense passage retrieval for open-domain question answering. In *EMNLP*, 2020.

[71] Z. Ke, F. Jiao, Y. Ming, X.-P. Nguyen, A. Xu, D. X. Long, M. Li, C. Qin, P. Wang, S. Savarese, et al. A survey of frontiers in llm reasoning: Inference scaling, learning to reason, and agentic systems. *arXiv preprint arXiv:2504.09037*, 2025.

[72] Z. Kenton, N. Y. Siegel, et al. On scalable oversight with weak llms judging strong llms. *arXiv*, 2024.

[73] A. Khan, J. Hughes, et al. Debating with more persuasive llms leads to more truthful answers. In *ICML*, 2024.

[74] D. Khatri, L. Madaan, et al. The art of scaling reinforcement learning compute for llms. *arXiv*, 2025.

[75] G. Kim, S. Kim, et al. Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models. In *EMNLP*, 2023.

[76] H. Kim, K. Lee, et al. Human implicit preference-based policy fine-tuning for multi-agent reinforcement learning in usv swarm. *arXiv*, 2025.

[77] H. J. Kim, H. Cho, et al. Self-generated in-context learning: Leveraging auto-regressive language models as a demonstration generator. *arXiv*, 2022.

[78] Y. Kim, C. Park, et al. Mdagents: An adaptive collaboration of llms for medical decision-making. *NeurIPS*, 2024.

[79] T. Kojima, S. S. Gu, et al. Large language models are zero-shot reasoners. *NeurIPS*, 2022.

[80] S. Krishna, C. Agarwal, et al. Understanding the effects of iterative prompting on truthfulness. *arXiv*, 2024.

[81] A. Kumar et al. Overthinking: Slowdown attacks on reasoning llms. *arXiv*, 2025.

[82] K. Kumar, T. Ashraf, et al. Llm post-training: A deep dive into reasoning large language models. *arXiv*, 2025.

[83] M. Kuo, J. Zhang, et al. H-cot: Hijacking the chain-of-thought safety reasoning mechanism to jailbreak large reasoning models, including openai o1/o3, deepseek-r1, and gemini 2.0 flash thinking. *arXiv*, 2025.

[84] H. Lai, X. Liu, J. Gao, J. Cheng, Z. Qi, Y. Xu, S. Yao, D. Zhang, J. Du, Z. Hou, et al. A survey of post-training scaling in large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2771–2791, 2025.

[85] H. D. Le et al. Multi-agent causal discovery using large language models. *arXiv*, 2024.

[86] Y. Lee, S.-w. Hwang, et al. Inference scaling for bridging retrieval and augmented generation. *arXiv*, 2024.

[87] Z. Lei, Y. Dong, et al. Harnessing large language models for disaster management: A survey. *arXiv*, 2025.

[88] P. Lewis, E. Perez, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *NeurIPS*, 2020.

[89] D. Li, B. Jiang, et al. From generation to judgment: Opportunities and challenges of llm-as-a-judge. *arXiv*, 2024.

[90] G. Li, H. Hammoud, et al. Camel: Communicative agents for" mind" exploration of large language model society. *NeurIPS*, 2023.

[91] H. Li, Y. Cao, et al. Investorbench: A benchmark for financial decision-making tasks with llm-based agent. *arXiv*, 2024.

[92] J. Li, T. Tang, et al. The web can be your oyster for improving large language models. *arXiv*, 2023.

[93] M. Li, S. Gong, et al. In-context learning with many demonstration examples. *arXiv*, 2023.

[94] X. Li, X.-P. Nguyen, et al. Paraicl: Towards robust parallel in-context learning. *arXiv*, 2024.

[95] Y. Li, H. Jiang, et al. Scbench: A kv cache-centric analysis of long-context methods. *arXiv*, 2024.

[96] Z. Li, C. Li, et al. Retrieval augmented generation or long-context llms? a comprehensive study and hybrid approach. In *EMNLP*, 2024.

[97] Z. Li, J. Xiong, et al. Uncertaintyrag: Span-level uncertainty enhanced long-context modeling for retrieval-augmented generation. *arXiv*, 2024.

[98] Z. Li, Q. Zang, et al. Autokaggle: A multi-agent framework for autonomous data science competitions. *arXiv*, 2024.

[99] J. Liang, R. Ye, et al. Debatrix: Multi-dimensional debate judge with iterative chronological analysis based on llm. *arXiv*, 2024.

[100] T. Liang, Z. He, et al. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv*, 2023.

[101] T. Liang, Z. He, et al. Encouraging divergent thinking in large language models through multi-agent debate. In *EMNLP*, 2024.

[102] V. Liévin, C. E. Hother, et al. Can large language models reason about medical questions? *Patterns*, 2024.

[103] S. Lin, J. Hilton, et al. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv*, 2021.

[104] H. Liu, M. Zaharia, et al. Ring attention with blockwise transformers for near-infinite context. *arXiv*, 2023.

[105] J. Liu, D. Shen, et al. What makes good in-context examples for gpt-3? *arXiv*, 2021.

[106] J. Liu, Y. Wang, et al. Medcot: Medical chain of thought via hierarchical expert. *arXiv*, 2024.

[107] M. Liu, S. Diao, et al. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv*, 2025.

[108] N. F. Liu, K. Lin, et al. Lost in the middle: How language models use long contexts. *TACL*, 2024.

[109] S. Liu, H. Ye, et al. In-context vectors: Making in context learning more effective and controllable through latent space steering. In *ICML*, 2024.

[110] Y. Liu, J. Liu, et al. Let's learn step by step: Enhancing in-context learning ability with curriculum learning. *arXiV*, 2024.

[111] Y. Liu, P. Yang, et al. Chunkkv: Chunk-based key-value cache management for transformer models. *arXiv*, 2025.

[112] Z. Liu, Y. Zhang, et al. Dynamic llm-agent network: An llm-agent collaboration framework with agent team optimization. *arXiv*, 2023.

[113] J. Long. Large language model guided tree-of-thought. *arXiv*, 2023.

[114] C. Lou, Z. Jia, Z. Zheng, et al. Sparser is faster and less is more: Efficient sparse attention for long-range transformers. *arXiv*, 2024.

[115] J. Lu, S. An, et al. Memochat: Tuning llms to use memos for consistent long-range open-domain conversation. *arXiv*, 2023.

[116] K. Lu, Z. Liang, et al. Med-R^2: Crafting Trustworthy LLM Physicians through Retrieval and Reasoning of Evidence-Based Medicine. *arXiv*, 2025.

[117] Y. Lu, M. Bartolo, et al. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv*, 2021.

[118] S. Luccioni, Y. Jernite, et al. Power hungry processing: Watts driving the cost of ai deployment? In *FAccT*, 2024.

[119] T. Q. Luong et al. Reft: Reasoning with reinforced fine-tuning. *arXiv*, 2024.

[120] A. Madaan, N. Tandon, et al. Self-refine: Iterative refinement with self-feedback. *NeurIPS*, 2023.

[121] L. C. Magister et al. Teaching small language models to reason. *arXiv*, 2022.

[122] A. Maharana, D.-H. Lee, et al. Evaluating very long-term conversational memory of llm agents. *arXiv*, 2024.

[123] R. Manvi, A. Singh, et al. Adaptive inference-time compute: Llms can predict if they can do better, even mid-generation. *arXiv*, 2024.

[124] I. McKenzie et al. Inverse scaling: When bigger isn't better. *TMLR*, 2024.

[125] J. Michael et al. Debate helps supervise unreliable experts. *arXiv*, 2023.

[126] S. Min, X. Lyu, et al. Rethinking the role of demonstrations: What makes in-context learning work? In *EMNLP*, 2022.

[127] S. Mo and M. Xin. Tree of uncertain thoughts reasoning for large language models. In *ICASSP*, 2024.

[128] S. Moayedpour, A. Corrochano-Navarro, et al. Many-shot in-context learning for molecular inverse design. *arXiv*, 2024.

[129] A. Mohtashami, M. Pagliardini, et al. Cotformer: A chain-of-thought driven architecture with budget-adaptive computation cost at inference. *arXiv*, 2023.

[130] N. Muennighoff, A. Rush, et al. Scaling data-constrained language models. *NeurIPS*, 2023.

[131] N. Muennighoff, Z. Yang, et al. s1: Simple test-time scaling. *arXiv*, 2025.

[132] R. Nakano, J. Hilton, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv*, 2021.

[133] X. Ning, Z. Lin, et al. Skeleton-of-thought: Prompting llms for efficient parallel generation. *arXiv*, 2023.

[134] OpenAI. Gpt-4 technical report, 2024.

[135] H. T. Otal, E. Stern, et al. Llm-assisted crisis management: Building advanced llm platforms for effective emergency response and public collaboration. In *CAI*, 2024.

[136] L. Ouyang, J. Wu, et al. Training language models to follow instructions with human feedback. *NeurIPS*, 2022.

[137] D. M. Owens, R. A. Rossi, et al. A multi-llm debiasing framework. *arXiv*, 2024.

[138] L. Pan, M. Saxon, et al. Automatically correcting large language models: Surveying the landscape of diverse automated correction strategies. *TACL*, 2024.

[139] Z. Pan, Q. Wu, et al. On memory construction and retrieval for personalized conversational agents. *arXiv*, 2025.

[140] C. F. Park, A. Lee, et al. Iclr: In-context learning of representations. *arXiv*, 2024.

[141] D. Patterson, J. Gonzalez, et al. The carbon footprint of machine learning training will plateau, then shrink. *Computer*, 2022.

[142] D. Paul, M. Ismayilzada, et al. Refiner: Reasoning feedback on intermediate representations. In *EACL*, 2024.

[143] C. Pham, B. Liu, et al. Let models speak ciphers: Multiagent debate through embeddings. In *ICLR*, 2024.

[144] A. Piktus, F. Petroni, et al. The web is your oyster-knowledge-intensive nlp against a very large web corpus. *arXiv*, 2021.

[145] L. Qian, W. Zhou, et al. Fino1: On the transferability of reasoning enhanced llms to finance. *arXiv*, 2025.

[146] Y. Qian, S. Zhang, et al. Enhancing llm-as-a-judge via multi-agent collaboration. 2025.

[147] Y. Qu, Y. Ding, et al. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. *arXiv*, 2020.

[148] R. Rafailov, A. Sharma, et al. Direct preference optimization: Your language model is secretly a reward model. *NeurIPS*, 2023.

[149] S. Rasal. Llm harmony: Multi-agent communication for problem solving. *arXiv*, 2024.

[150] R. Rawat. Disasterqa: A benchmark for assessing the performance of llms in disaster response. *arXiv*, 2024.

[151] O. Rubin, J. Herzig, et al. Learning to retrieve prompts for in-context learning. In *NAACL*, 2022.

[152] S. Saha, X. Li, et al. Learning to plan & reason for evaluation with thinking-llm-as-a-judge. *arXiv*, 2025.

[153] S. Samsi, D. Zhao, et al. From words to watts: Benchmarking the energy costs of large language model inference. In *HPEC*, 2023.

[154] P. Sarthi, S. Abdullah, et al. Raptor: Recursive abstractive processing for tree-organized retrieval. In *ICLR*, 2024.

[155] N. Saunshi, N. Dikkala, et al. Reasoning with latent thoughts: On the power of looped transformers. *arXiv*, 2025.

[156] T. Savage, A. Nayak, et al. Diagnostic reasoning prompts reveal the potential for large language model interpretability in medicine. *NPJ Digital Medicine*, 2024.

[157] M. Sawiński, K. Węcel, et al. Openfact at checkthat! 2023: head-to-head gpt vs. bert-a comparative study of transformers language models for the detection of check-worthy claims. In *CEUR*, 2023.

[158] J. Schulman et al. Proximal policy optimization algorithms. *arXiv*, 2017.

[159] B. Sel, A. Al-Tawaha, et al. Algorithm of thoughts: Enhancing exploration of ideas in large language models. *arXiv*, 2023.

[160] R. Shao, J. He, et al. Scaling retrieval-based language models with a trillion-token datastore. *NeurIPS*, 2024.

[161] Z. Shao, P. Wang, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv*, 2024.

[162] L. Shen, E. Shen, et al. Towards natural language interfaces for data visualization: A survey. *TVCG*, (6), 2022.

[163] W. Shen, X. Zhang, et al. Improving reinforcement learning from human feedback using contrastive rewards. *arXiv*, 2024.

[164] X. Shen, Y. Wang, et al. Efficient reasoning with hidden thinking. *arXiv*, 2025.

[165] N. Shinn, F. Cassano, et al. Reflexion: Language agents with verbal reinforcement learning. *NeurIPS*, 2023.

[166] K. Shridhar, A. Stolfo, et al. Distilling reasoning capabilities into smaller language models. *ACL*, 2023.

[167] R. Singhal, P. Patwa, et al. Evidence-backed fact checking using rag and few-shot in-context learning with llms. *arXiv*, 2024.

[168] C. Snell, J. Lee, et al. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv*, 2024.

[169] M. Song, M. Zheng, et al. Can many-shot in-context learning help llms as evaluators? a preliminary empirical study. *arXiv9*, 2024.

[170] J. Tack et al. Llm pretraining with continuous concepts. *arXiv*, 2025.

[171] Z. Tan, A. Beigi, et al. Large language models for data annotation: A survey. *arXiv*, 2024.

[172] Z. Tan, J. Peng, et al. Tuning-free accountable intervention for llm deployment–a metacognitive approach. *arXiv*, 2024.

[173] Z. Tan, J. Yan, et al. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents, 2025.

[174] Z. Tan and C. a. Zhao. Glue pizza and eat rocks-exploiting vulnerabilities in retrieval-augmented generative models. In *EMNLP*, 2024.

[175] L. Tang, P. Laban, et al. Minicheck: Efficient fact-checking of llms on grounding documents. *arXiv*, 2024.

[176] G. Team, P. Georgiev, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv*, 2024.

[177] K.-T. Tran, D. Dao, et al. Multi-agent collaboration mechanisms: A survey of llms. *arXiv*, 2025.

[178] A. Verma, S. Mohajer, et al. Multi-agent fact checking. *arXiv*, 2025.

[179] X. Wan, R. Sun, et al. Teach better or show smarter? on instructions and exemplars in automatic prompt optimization. *NeurIPS*, 2025.

[180] X. Wan, H. Zhou, et al. From few to many: Self-improving many-shot reasoners through iterative optimization and generation. *arXiv*, 2025.

[181] B. Wang, W. Ping, et al. Shall we pretrain autoregressive language models with retrieval? a comprehensive study. In *EMNLP*, 2023.

[182] B. Wang, W. Ping, et al. Instructretro: instruction tuning post retrieval-augmented pretraining. In *ICML*, 2024.

[183] H. Wang, W. Huang, et al. Unims-rag: A unified multi-source retrieval-augmented generation for personalized dialogue systems. *arXiv*, 2024.

[184] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, W. X. Zhao, Z. Wei, and J. Wen. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6), Mar. 2024.

[185] Q. Wang, L. Ding, et al. Recursively summarizing enables long-term dialogue memory in large language models. *arXiv*, 2023.

[186] S. Wang, Z. Chen, et al. Mixture of demonstrations for in-context learning. *NeurIPS*, 2025.

[187] X. Wang, M. Salmani, et al. Beyond the limits: A survey of techniques to extend the context length in large language models. *arXiv*, 2024.

[188] X. Wang, P. Sen, et al. Adaptive retrieval-augmented generation for conversational systems. *arXiv*, 2024.

[189] X. Wang, J. Wei, et al. Self-consistency improves chain of thought reasoning in language models. *arXiv*, 2022.

[190] X. Wang and D. Zhou. Chain-of-thought reasoning without prompting. *arXiv*, 2024.

[191] Y. Wang, Q. Liu, et al. Thoughts are all over the place: On the underthinking of o1-like llms. *arXiv*, 2025.

[192] J. Wei, N. Kim, et al. Inverse scaling can become u-shaped. In *EMNLP*, 2023.

[193] J. Wei, Y. Tay, et al. Emergent abilities of large language models. *arXiv*, 2022.

[194] J. Wei, X. Wang, et al. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*, 2022.

[195] T.-R. Wei, H. Liu, et al. A survey on feedback-based multi-step reasoning for large language models on mathematics. *arXiv*, 2025.

[196] X. Wei and L. Liu. Are large language models good in-context learners for financial sentiment analysis? *arXiv*, 2025.

[197] Y. Wei, O. Duchenne, et al. Swe-rl: Advancing llm reasoning via reinforcement learning on open software evolution. *arXiv*, 2025.

[198] L. Weng. Llm-powered autonomous agents. *lilianweng.github.io*, Jun 2023.

[199] Y. Weng, B. Li, et al. Large language models with holistically thought could be better doctors. In *NLPCC*, 2024.

[200] R. Wolfe and T. Mitra. The impact and opportunities of generative ai in fact-checking. In *FAccT*, 2024.

[201] C. Wu, S. Yin, et al. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv*, 2023.

[202] X. Wu, L. Xiao, et al. A survey of human-in-the-loop for machine learning. *Futur. Gener. Comput. Syst.*, 2022.

[203] Y. Wu, Z. Sun, et al. Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models. *arXiv*, 2024.

[204] Y. Wu, Y. Wang, et al. When more is less: Understanding chain-of-thought length in llms. *arXiv*, 2025.

[205] Y. Xia, Y. Huang, et al. A question and answering service of typhoon disasters based on the t5 large language model. *IJGI*, 2024.

[206] C. Xiang, T. Wu, et al. Certifiably robust rag against retrieval corruption. *arXiv*, 2024.

[207] Z. Xiang, F. Jiang, et al. Badchain: Backdoor chain-of-thought prompting for large language models. *arXiv*, 2024.

[208] Z. Xiang, Z. Xiong, et al. Cbd: A certified backdoor detector based on local dominant probability. *NeurIPS*, 2023.

[209] Y. Xiao, E. Sun, et al. Tradingagents: Multi-agents llm financial trading framework. *arXiv*, 2024.

[210] T. Xie, Z. Gao, et al. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning. *arXiv*, 2025.

[211] Y. Xie, A. Goyal, et al. Monte carlo tree search boosts reasoning via iterative preference learning. *arXiv*, 2024.

[212] G. Xiong, Q. Jin, et al. Benchmarking retrieval-augmented generation for medicine. In *ACL*, 2024.

[213] L. Xiong, C. Xiong, et al. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv*, 2020.

[214] F. Xu, W. Shi, et al. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *arXiv*, 2023.

[215] P. Xu, W. Ping, et al. Retrieval meets long context large language models. In *ICLR*, 2023.

[216] P. Xu, W. Ping, et al. Chatqa 2: Bridging the gap to proprietary llms in long context and rag capabilities. *arXiv*, 2024.

[217] Z. Xu, C. Yu, F. Fang, Y. Wang, and Y. Wu. Language agents with reinforcement learning for strategic play in the werewolf game, 2024.

[218] A. Yang, B. Yang, et al. Qwen2. 5 technical report. *arXiv*, 2024.

[219] W. Yang, X. Bi, et al. Watch out for your agents! investigating backdoor threats to llm-based agents. *NeurIPS*, 2024.

[220] W. Yang, S. Ma, et al. Towards thinking-optimal scaling of test-time compute for llm reasoning. *arXiv*, 2025.

[221] S. Yao, D. Yu, et al. Tree of thoughts: Deliberate problem solving with large language models. *NeurIPS*, 2023.

[222] J. Ye, Z. Wu, et al. Compositional exemplars for in-context learning. In *ICML*, 2023.

[223] Y. Ye, Z. Huang, et al. Limo: Less is more for reasoning. *arXiv*, 2025.

[224] Z. Yi, J. Ouyang, et al. A survey on recent advances in llm-based multi-turn dialogue systems. *arXiv*, 2024.

[225] K. Yin, C. Liu, et al. Crisissense-llm: Instruction fine-tuned large language model for multi-label social media text classification in disaster informatics. *arXiv*, 2024.

[226] Y. Yu, H. Li, Z. Chen, Y. Jiang, Y. Li, D. Zhang, R. Liu, J. W. Suchow, and K. Khashanah. Finmem: A performance-enhanced llm trading agent with layered memory and character design. In *Proceedings of the AAAI Symposium Series*, number 1, pages 595–597, 2024.

[227] Y. Yu, Z. Yao, et al. Fincon: A synthesized llm multi-agent system with conceptual verbal reinforcement for enhanced financial decision making. *NeurIPS*, 2024.

[228] Z. Yuan, H. Yuan, et al. Scaling relationship on learning mathematical reasoning with large language models. *arXiv*, 2023.

[229] Z. Yue, H. Zhuang, et al. Inference scaling for long-context retrieval augmented generation. In *ICLR*, 2025.

[230] R. Zamora-Resendiz, I. Khurram, et al. Towards maps of disease progression: Biomedical large language model latent spaces for representing disease phenotypes and pseudotime. *medRxiv*, 2024.

[231] E. Zelikman, G. R. Harik, et al. Quiet-star: Language models can teach themselves to think before speaking. In *COLM*, 2024.

[232] L. Zhang, A. Hosseini, et al. Generative verifiers: Reward modeling as next-token prediction. *arXiv*, 2024.

[233] Q. Zhang, F. Lyu, Z. Sun, L. Wang, W. Zhang, W. Hua, H. Wu, Z. Guo, Y. Wang, N. Muennighoff, et al. A survey on test-time scaling in large language models: What, how, where, and how well? *arXiv preprint arXiv:2503.24235*, 2025.

[234] X. Zhang, A. Lv, et al. More is not always better? enhancing many-shot in-context learning with differentiated and reweighting objectives. *arXiv*, 2025.

[235] Y. Zhao and P. a. Singh. Optimizing llm based retrieval augmented generation pipelines in the financial domain. In *NAACL*, 2024.

[236] C. Zheng, Y. Gao, et al. Cape: Context-adaptive positional encoding for length extrapolation. *arXiv*, 2024.

[237] C. Zheng, Y. Gao, et al. Dape: Data-adaptive positional encoding for length extrapolation. *NeurIPS*, 2024.

[238] L. Zheng, W.-L. Chiang, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *NeurIPS*, 2023.

[239] L. Zheng, W.-L. Chiang, et al. Lmsys-chat-1m: A large-scale real-world llm conversation dataset. *arXiv*, 2023.

[240] C. Zhou, P. Liu, et al. Lima: Less is more for alignment. *NeurIPS*, 2023.

[241] H. Zhou, K.-H. Lee, et al. Trustrag: Enhancing robustness and trustworthiness in rag. *arXiv*, 2025.

[242] J. Zhou, Y. Zhang, et al. Synthetic lies: Understanding ai-generated misinformation and evaluating algorithmic and human solutions. In *CHI*, 2023.

[243] F. Zhu, Z. Liu, et al. Tat-llm: A specialized language model for discrete reasoning over financial tabular and textual data. In *ICAIF*, 2024.

[244] H. Zou, Q. Zhao, et al. Genainet: Enabling wireless collective intelligence via knowledge transfer and reasoning. *arXiv*, 2024.

[245] K. Zou, M. Khalifa, et al. Retrieval or global context understanding? on many-shot in-context learning for long-context evaluation. *arXiv*, 2024.