Fundamental Capabilities and Applications of Large Language Models: A Survey

JIAWEI LI, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, China
YANG GAO*, Computer Science and Technology, Beijing Institute of Technology, Beijing, China
YIZHE YANG, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, China
YU BAI, Beijing Institute of Technology, Beijing, China
XIAOFENG ZHOU, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, China
YINGHAO LI, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, China
HUASHAN SUN, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, China
YUHANG LIU, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, China
XINGPENG SI, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, China
YUHAO YE, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, China
YIXIAO WU, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, China
YIGUAN LIN, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, China
BIN XU, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, China
BOWEN REN, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, China
CHONG FENG, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, China
HEYAN HUANG, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, China

*Yang Gao is the corresponding author.

This work has received partial funding from the Major Research Plan of the National Natural Science Foundation of China (Grant No.92370110) and the Joint Funds of National Natural Science Foundation of China (No.U21B2009).
Authors' Contact Information: Jiawei Li, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, Beijing, China; e-mail: jarvi_lee@163.com; Yang Gao, Computer Science and Technology, Beijing Institute of Technology, Beijing, China; e-mail: gyang@bit.edu.cn; Yizhe Yang, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, Beijing, China; e-mail: yizheyang@bit.edu.cn; Yu Bai, Beijing Institute of Technology, Beijing, Beijing, China; e-mail: Yubai@bit.edu.cn; Xiaofeng Zhou, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, Beijing, China; e-mail: zhouxiaofeng@bit.edu.cn; Yinghao Li, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, Beijing, China; e-mail: yhli@bit.edu.cn; Huashan Sun, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, Beijing, China; e-mail: hssun@bit.edu.cn;

ACM 1557-7341/2025/5-ART
https://doi.org/10.1145/3735632

Large Language Models (LLMs) have demonstrated remarkable effectiveness across various domain-specific applications. However, which fundamental capabilities most contribute to their success in different domains remains unclear. This uncertainty complicates LLM evaluation, as existing benchmark-based assessments often fail to capture their real-world performance, where the required capabilities may differ from those measured in the benchmarks. In this survey, we provide a systematic introduction to LLMs' fundamental capabilities, encompassing their definitions, formation mechanisms, and practical applications. We further explore the relationships among these capabilities and discuss how they collectively support complex problem-solving in domain-specific applications. Building on this foundation, we review recent advances in LLM-driven applications across nine specific domains: medicine, law, computational biology, finance, social sciences and psychology, computer programming and software engineering, robots and agents, AI for disciplines, and creative work. We analyze how specific capabilities are leveraged for each domain to address unique requirements. This perspective enables us to establish connections between these capabilities and domain requirements, and to evaluate the varying importance of different capabilities across different domains. Based on these insights, we propose evaluation strategies tailored to the essential capabilities required in each domain, offering practical guidance for selecting suitable backbone LLMs in real-world applications.

CCS Concepts: • **Computing methodologies** → **Network science**; **Artificial intelligence**.

Additional Key Words and Phrases: Large Language Model, Fundamental Capabilities, Applications

1 Introduction

In the current research and application of artificial intelligence, the abundant acquisition of big data, breakthroughs in high-performance computing technology, and innovations in algorithm design have jointly promoted the development and deployment of LLMs [120]. LLMs are also considered to have strong potential value in specific domains, with an increasing number of industries embracing LLMs and already demonstrating outstanding performance [79].

However, applying LLMs in specific domains has encountered a series of challenges. These challenges mainly stem from the inherent characteristics of domain tasks and data, such as the diversity of data sources, the complexity of domain-specific knowledge, and the specificity of application goals and constraints. To enable LLMs to be better applied in specific domains and address the challenges they face in these areas, this paper summarizes two key issues that need to be resolved when applying LLMs in specific domains:

Issue 1: Fundamental capabilities of LLMs and their interactions. LLMs exhibit outstanding performance in comprehending and addressing complex tasks, thus demonstrating great potential in specific domain applications. Numerous studies broadly summarize the core capabilities of LLMs as robust comprehension and generation [88], yet fall short of assisting us in aligning the LLMs' fine-grained capabilities with the intricate requirements of real-world scenarios. Consequently, elucidating the inherent fundamental capabilities manifested by LLMs in domain-specific scenarios and the dynamics among these capabilities becomes essential.

Yuhang Liu, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, Beijing, China; e-mail: yhliu@bit.edu.cn; Xingpeng Si, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, Beijing, China; e-mail: xpsi@bit.edu.cn; Yuhao Ye, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, Beijing, China; e-mail: yhye@bit.edu.cn; Yixiao Wu, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, Beijing, China; e-mail: yxwu@bit.edu.cn; Yiguan Lin, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, Beijing, China; e-mail: yglin@bit.edu.cn; Bin Xu, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, Beijing, China; e-mail: binxu@bit.edu.cn; Bowen Ren, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, Beijing, China; e-mail: bwren-bit@bit.edu.cn; Chong Feng, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, Beijing, China; e-mail: fengchong@bit.edu.cn; Heyan Huang, Beijing Institute of Technology School of Computer Science and Technology, Haidian-qu, Beijing, China; e-mail: hhy63@bit.edu.cn.

Issue 2: The Capabilities Assessment of LLMs in a Specific Domain. Due to the disparity between the capabilities evaluated in benchmarks and those required in real-world domains [105], the excellent performance of LLMs in benchmarks may not necessarily translate to actual applications in specific domains. Therefore, conducting capabilities assessments of LLMs to establish a bridge between benchmarks and real-world domains is crucial.

Based on the above two issues, this survey aims to systematically summarize the fundamental capabilities of LLMs and clarify the capabilities assessment of LLMs. The key contributions of this survey paper are summarized below.

1. This paper summarizes the fundamental capabilities of LLMs in domain applications, including memorization, reasoning, generalization, and diversification. It provides detailed descriptions of each capability and how they collaborate to accomplish specific applications.
2. This paper summarizes the applications of LLMs in nine specific domains from the perspective of real scenarios. In addition, this paper summarizes the importance of fundamental abilities corresponding to each domain, addressing the issue of strong performance on benchmarks not necessarily translating to domain scenarios, and providing users with clear model selection strategies.

The remainder of this survey is organized as follows: Section 2 introduces the four fundamental capabilities of Large Language Models, including their definitions, formation mechanisms, and practical applications. We further explore the relationships among these capabilities and discuss how they collectively support complex problem-solving in domain-specific scenarios. This section builds upon our previous work [123]. Section 3 examines LLM applications across nine specific domains: medicine, law, computational biology, finance, social sciences and psychology, computer programming and software engineering, robots and agents, AI for disciplines, and creative work. We analyze which fundamental capabilities are most critical in each domain and how they address domain-specific challenges. Section 4 aligns the fundamental capabilities of LLMs with their domain-specific applications, establishing connections between capabilities and domain requirements. Based on this analysis, we propose evaluation strategies tailored to the essential capabilities required in each domain, offering practical guidance for selecting suitable backbone LLMs in real-world applications.

2 Fundamental Capabilities and Interactive Capabilities

The human brain's information processing has been extensively studied, revealing five core modules: natural language interaction, knowledge, memorization, reasoning, and generalization [255]. Some research suggests that LLMs exhibit a similar information processing mechanism to the human brain [25]. Consequently, we categorize LLM information processing into four fundamental capabilities, including memorization, reasoning, generalization, and diversification. As illustrated in Figure 1, LLMs utilize short-term memory to understand task instructions and long-term memory to retrieve historical data [41]. This data is processed through the reasoning module, which performs logical, commonsense, and symbolic reasoning to generate outputs [263]. Throughout this process, the generalization enables LLMs to manage information across varying lengths, structures, and tasks [49, 114], while diversification allows for tailored results [217].

The four fundamental capabilities of the LLM work together to complete complex domain applications. In the following section, we will introduce each of the fundamental capabilities in detail.

2.1 Memorization Capabilities

The memorization capabilities of LLMs play a pivotal role in their effectiveness and performance across various domains. Memory, in the context of LLMs, refers to the capacity to retain and access information over time. It can be broadly categorized into two types: long-term memory and short-term memory. Long-term memory refers to the LLM's ability to store and recall knowledge, facts, and concepts acquired during training and previous experiences. It encompasses the model's understanding of world knowledge, implicit encoding of substantial

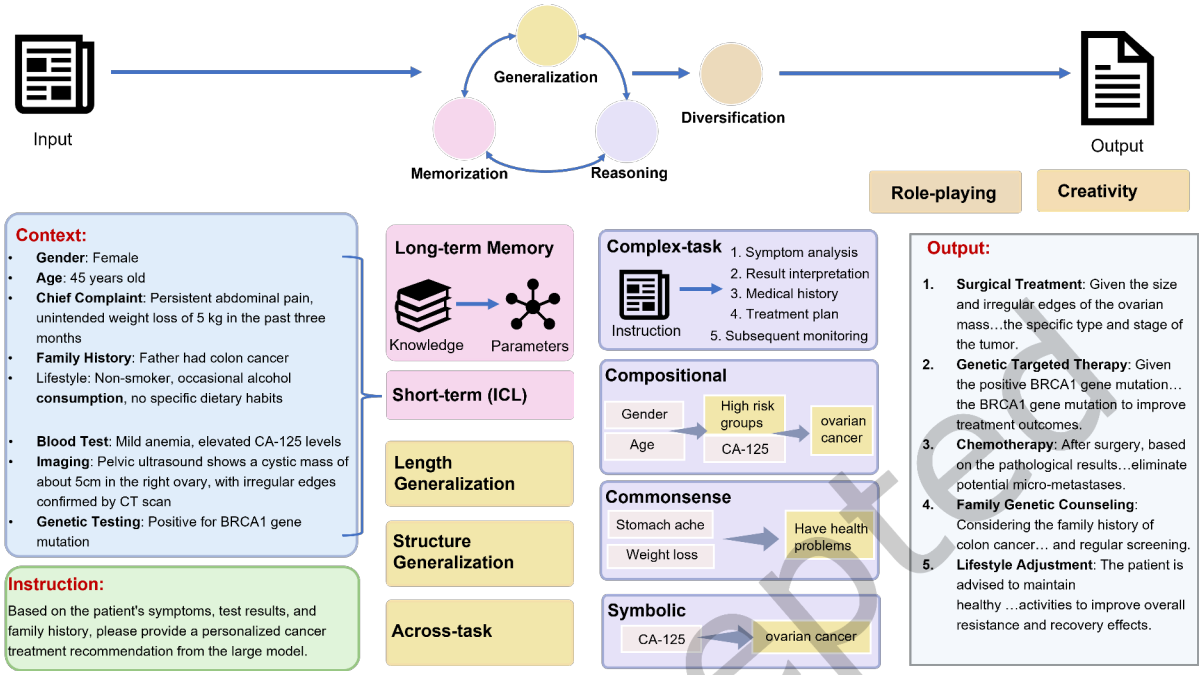

Fig. 1. The relationship between the four fundamental abilities when solving complex domain applications. Take medical treatment as an example.

information, and its capacity to leverage both internal and external knowledge sources. On the other hand, short-term memory focuses on the LLM's in-context learning capabilities and its ability to retain and utilize information within a limited temporal context. Enhancements in short-term memory aim to overcome the contextual limitations of LLMs and enable them to generate coherent content over longer stretches. Recent advancements in the field have addressed challenges related to both long-term and short-term memory aspects, offering valuable methodologies to bolster the retention and application of knowledge within LLMs.

2.1.1 Long-term Memory. The long-term memory of LLMs is intimately connected to their scale, with LLMs showcasing broader knowledge capacity and diversity. Benchmarks like KoLA critically evaluate LLMs' world knowledge across numerous tasks [272], while studies by Tirumala et al. [226] reveal that model size is crucial for efficient memorization. LLMs acting as knowledge bases, as reviewed by AlKhamissi et al. [6], encode substantial information implicitly, and innovations like REALM employ latent retrievers to augment this knowledge [80]. Petroni et al. [184] suggests LLMs can serve as effective knowledge bases even without fine-tuning. Together, these studies underscore the remarkable potential of LLMs to use both internal and external knowledge sources to enhance their long-term memory, applicable across various fields.

Addressing the challenge of preserving the long-term memory of LLMs during continual learning is crucial for their application in specialized fields. Luo et al. [152] propose a novel framework, SCCL, which mitigates catastrophic forgetting—a common obstacle to maintaining long-term knowledge—by employing adaptive classification strategies alongside memory replay and distillation techniques. Complementing this, Luo et al. [153] suggests that initial training on general linguistic tasks and the adoption of a hybrid continual learning strategy

can substantially reduce the loss of long-term syntactic and semantic knowledge. These studies offer valuable methodologies for enhancing the retention of long-term knowledge in LLMs, ensuring their continued effectiveness as they acquire new domain-specific information.

2.1.2 Short-term Memory. Short-term memory in LLMs has been a focus area to enhance their in-context learning (ICL) capabilities. ICL is a capability that allows LLMs to understand and execute tasks based on the immediate context provided within the input text [24]. This skill set eliminates the need for extensive retraining or fine-tuning across different tasks. By analyzing a few examples included in their input, LLMs can infer the task requirements and apply learned patterns to generate accurate responses [292]. This in-context learning ability showcases LLMs' adaptability, making them highly efficient for a broad spectrum of applications with minimal setup [163]. Despite its effectiveness, this approach has limitations in terms of the depth and complexity of understanding it can provide, which is directly influenced by the model's design and the richness of the context provided [282]. Meanwhile, the working mechanism of ICL is also a widely open question and has been investigated a lot by the commuity [16].

2.2 Reasoning Capabilities

The reasoning capabilities of LLMs refer to logically processing information, drawing conclusions, and making decisions based on available data and knowledge [188]. The reasoning capabilities of LLMs have greatly enhanced their application across various industries. For example, they apply commonsense reasoning to user interactions in customer service and healthcare, providing contextually relevant responses. Additionally, advances in symbolic reasoning allow LLMs to support software development and mathematical fields with increased accuracy and clarity. These developments mark significant progress toward more sophisticated AI systems capable of augmenting human tasks. In this section, we summarize the recent advances in the reasoning capabilities of LLMs.

2.2.1 Compositional Reasoning Capabilities. In this section, we synthesize recent findings on compositional reasoning capabilities in LLMs and related systems. Recent advancements demonstrate that augmenting LLMs with specialized modules or training approaches enhances their compositional reasoning capabilities, surpassing traditional methods in diverse and complex tasks. Lu et al. [150] augments LLMs with modules for complex reasoning, achieving significant accuracy improvements on multi-modal tasks. Chen et al. [33] propose a novel prompting strategy, skills-in-context (SKiC), enabling LLMs to exhibit compositional reasoning by solving unseen, complex problems through the innovative composition of pre-existing skills, achieving groundbreaking success on compositional tasks. Ma et al. [157] introduces new benchmarks for evaluating GVLMs' compositional reasoning, with a novel metric to reduce morphological bias. Compositional Task Representations (CTR), a new prompt-free approach, is proposed to learn compositional codes, surpassing prompt-based methods in zero-shot learning [206]. An LLM trained on the PLANE benchmark shows strong capacities in compositional entailment, leveraging subword representations[14].

2.2.2 Complex Task Decomposition. A variety of new prompting techniques such as ADaPT, chain of thought, zero-shot CoT, iterative context-aware, least-to-most, decomposed, and successive prompting have been proposed to enable LLMs to decompose and tackle complex tasks more effectively [109]. Prasad et al. [187] introduces ADaPT, which enhances LLMs' decision-making by planning and decomposing tasks as needed, significantly improving performance on complex tasks. Wei et al. [249] demonstrates that "chain of thought" prompting boosts LLMs' complex reasoning, achieving state-of-the-art results on the GSM8K benchmark. Kojima et al. [110] presents Zero-shot-CoT, using simple prompts to unlock LLMs' underlying reasoning capabilities, achieving substantial gains across diverse reasoning tasks. Wang et al. [232] proposes an iterative prompting framework that progressively extracts PLM's knowledge for multi-step reasoning, overcoming the limitations of traditional prompting methods.

DISC [138] introduces a dynamic solution decomposition framework that adaptively partitions reasoning steps to efficiently allocate computation resources and further improve reasoning performance.

2.2.3 Commonsense Reasoning. Recent work in commonsense reasoning explores innovative approaches like integrating LLMs with search algorithms, conducting comprehensive surveys, and applying code generation models to outperform traditional methods, while also exposing the limitations of LMs in truly understanding commonsense knowledge without specific supervision. Bhargava and Ng [15] surveys recent tasks in commonsense reasoning and generation, evaluating the capabilities and limitations of state-of-the-art pre-trained models. Zhao et al. [291] demonstrates that combining LLMs with MCTS for task planning leverages commonsense knowledge to enhance reasoning and efficiency in complex tasks. Madaan et al. [158] proposes using code generation LMs for structured commonsense reasoning tasks, outperforming traditional LMs in natural language processing. Li et al. [130] conducts a zero-shot and few-shot evaluation of LMs' commonsense knowledge, revealing limitations and the insufficiency of LLMs to reach human-level performance.

2.2.4 Symbolic Reasoning. Recent work on symbolic reasoning highlights the effectiveness of novel prompting techniques and hybrid frameworks combining LLMs with symbolic solvers or distillation methods. Gaur and Saunshi [70] explores symbolic reasoning in math word problems using LLMs, introducing a self-prompting method that aligns symbolic reasoning with numeric answers, enhancing interpretability and accuracy. Wei et al. [249] demonstrates that chain of thought prompting significantly improves LLMs' performance on complex reasoning tasks including symbolic reasoning. Pan et al. [178] introduces Logic-LM, a framework that combines LLMs with symbolic solvers, resulting in substantial improvements in logical reasoning tasks. Gaur and Saunshi [69] shows that GPT-3's performance on symbolic math word problems can be enhanced with specific prompting techniques that encourage the model to describe its reasoning process. Li et al. [125] reveals that even smaller models can benefit from chain-of-thought prompting through Symbolic Chain-of-Thought Distillation from LLMs, leading to improved reasoning performance.

2.3 Generalization Capabilities

Generalization refers to a model's ability to apply learned knowledge from past experiences to new, unseen situations [207]. This capability is essential for real-world applications where models encounter various data. In this section, we will explore the generalization capabilities of LLMs, focusing on three key aspects: length, structural, and across-task generalization.

2.3.1 Length Generalisation. Length generalization in LLMs, which refers to the model's capacity to extend acquired skills to longer problem instances outside the training range, is crucial for addressing complex problems with extensive descriptions [8]. Improving length generalization is key to enhancing the practical use of LLMs in diverse real-world situations.

Theoretical insights, such as Anil et al. [8], have examined the length generalization capabilities of transformer-based models and identified conditions for length generalization in reasoning tasks. For arithmetic tasks, Jelassi et al. [94] introduced train set priming to improve generalization, while the innovative LM-Infinite [83], RASP-Generalization Conjecture [295], and Attention Bias Calibration (ABC) [56] employ different variants of attention mechanisms for longer text generation. On the other hand, Awasthi and Gupta [10] focused on multitask training with task hinting to address length generalization.

These studies collectively present a range of strategies, from practical methodologies like task hinting to theoretical frameworks, enhancing LLMs' ability to manage longer input sequences. They mark significant progress in overcoming the challenges of length generalization, paving the way for more capable and adaptable LLMs.

2.3.2 Structure Generalization. Structure generalization of LLMs refers to the capability to process and interpret complex data structures, such as graphs and tables even though the models are trained on text-only datasets. This ability is crucial for applications extending beyond traditional text-based tasks, spanning various domains including bioinformatics and social network analysis.

Numerous studies have aimed at enhancing the capabilities of LLMs to process and generate diverse data forms beyond traditional text, including graphs [222], tables [287], and visualization charts [239]. This expansion into handling various data types is particularly notable in fields such as healthcare [224], recommendation [239], question answering [96], and biomedical science [236], significantly broadening the practical applications of LLMs.

These studies collectively underscore the expanding versatility of LLMs in handling structured data, revealing a trend toward more sophisticated AI models capable of complex reasoning and diverse applications.

2.3.3 Generalization Across Tasks. Task generalization in LLMs refers to their ability to manage a wide range of tasks, especially those not seen during training. This capacity allows the models to tackle a variety of novel and unexpected challenges, showcasing their flexibility, efficiency, and versatility.

To enhance task generalization in LLMs, two prevalent strategies are employed. Firstly, fine-tuning approaches, such as multi-task [196], instruction tuning [247], and meta tuning [294], fine-tune language models across various NLP tasks to augment their comprehension of instructions, thereby achieving significant zero-shot learning capabilities. These methods highlight LLMs' potential in managing an array of tasks through enhanced instruction understanding. However, fine-tuning parameters of LLMs can be resource-intensive. In response, Ye et al. [268] and Brown et al. [24] investigate few-shot or in-context learning mechanisms. By providing a few task examples, LLMs can infer the task's requirements and format, allowing them to address new tasks effectively. This approach circumvents the need for extensive fine-tuning, instead leveraging examples to foster the models' abilities.

In summary, these studies highlight the generalization of LLMs through diverse methodologies, ranging from fine-tuning to prompting strategies. The overarching objective is to enhance models that not only perform proficiently on familiar tasks but also demonstrate remarkable adaptability to novel challenges, thereby creating more adaptable and intelligent systems.

2.4 Diversification Capabilities

The concept of diversification in LLMs pertains to their capability to produce unique content tailored to various contexts. This diversification arises during the inference process, where a model generates a new token, y_t, based on the previously generated tokens, y_{t-1}, and a specific condition, x, according to the formula $y_t \sim p(y_t|y_{:t-1}, x)$. Notably, the architecture of most LLMs is decoder-only, meaning that conditions such as prompts or in-context examples are incorporated as initial tokens. Thus, we regard these initial inputs as x and the sequence of generated tokens as $y_{:t-1}$. By manipulating these inputs and conditions, LLMs can produce a wide array of content.

We delve into the diversification of the expanding capabilities of LLMs in terms of role-playing and creativity. These two areas highlight the versatility of LLMs, showcasing their ability to adapt to diverse scenarios and tasks. Role-playing enhances the dynamism and context-awareness of LLMs, enabling more nuanced interactions across different scenarios by utilizing role profiles as the condition x.Furthermore, creativity plays a crucial role in unlocking the potential of LLMs for generating innovative and valuable content. This is achieved by modifying the generation process, specifically the sequence of previously generated tokens, y_{t-1}. Together, these advancements in role-playing and creativity signify a substantial leap in the diversification of LLM applications, illustrating their evolving role in artificial intelligence as tools not only for replicating human language but also for exhibiting human-like ingenuity and adaptability.

2.4.1 Role-playing. Role-playing in LLMs represents a significant advancement in the field of natural language processing and artificial intelligence. It involves LLMs assuming specific characters or personas, enabling them to engage in more dynamic, context-rich, and human-like interactions. This capability is particularly important as it allows for more immersive and realistic conversational experiences, catering to diverse applications like customer service, entertainment, education, and therapy. By embodying different roles, i.e., x we defined before, LLMs can offer tailored responses based on character-specific knowledge and behavior patterns, enhancing the relevance and engagement of user interactions.

Wei et al. [248] investigate multi-party conversations, revealing that LLMs can significantly improve group dynamics when trained on datasets like MultiLIGHT. Wang et al. [246]'s RoleLLM framework enhances role-playing in LLMs, leading to advancements in English and Chinese models. Shanahan et al. [205] discusses the importance of role play in understanding dialogue agents' behaviors, focusing on aspects like deception and self-awareness. Li et al. [117] develop ChatHaruhi, demonstrating enhanced role-playing in mimicking anime characters. Personalization in LLMs is the focus of Salemi et al. [195]'s LaMP benchmark, which improves model outputs by incorporating user profiles. Finally, Li et al. [119] explore autonomous cooperation among LLMs through role-playing, showcasing the potential of inception prompting in multi-agent systems.

These studies collectively represent a significant stride in enhancing the role-playing capabilities of LLMs. They demonstrate how role-playing can transform LLMs into more adaptable, engaging, and effective conversational partners, capable of nuanced interactions across various domains by adjusting the condition x. This corpus of research not only broadens the scope of LLM applications but also provides valuable insights into the development of more sophisticated, human-like AI conversational agents.

2.4.2 Creativity. The creativity in LLMs is gaining traction, emphasizing their potential to generate novel and valuable content. Emphasizing creativity in LLMs is key to developing AI systems that not only replicate human language but also exhibit a degree of ingenuity akin to human creativity.

Recent studies in this area offer diverse insights. Chakrabarty et al. [26] develop a framework for evaluating the creativity of LLMs, revealing their current limitations compared to human writers. Franceschelli and Musolesi [64] explore LLMs' creative writing potential, examining their development through various creativity theories and considering their societal impact. Summers-Stay et al. [216] demonstrate that LLMs can enhance their creativity by mimicking human brainstorming techniques. Swanson et al. [219] introduce tools to assist creative writers in leveraging LLMs' capabilities, while Sinha et al. [211] propose a model to balance creativity with factual accuracy in LLM outputs. Bhavya et al. [17] focus on creative analogy mining using PLMs, underscoring the role of LLMs in augmenting human creativity.

Together, these studies contribute significantly to our understanding of LLMs' creativity. They range from developing methods to assess and benchmark LLM creativity to techniques that enhance creative output in writing and problem-solving. This corpus of research underscores LLMs' potential in creative domains and highlights their evolving role in artificial intelligence, pushing the boundaries of what LLMs can achieve in terms of creative thinking and expression.

2.5 Interactive Capabilities

In addition to the four fundamental capabilities, LLMs also possess strong interactive capabilities during domain applications. In specific tasks, LLMs significantly enhance performance by acquiring external information, planning and making decisions regarding the environment, and utilizing external tools [255]. In this section, we will focus on discussing the capabilities of LLMs in terms of using tools and environment interaction, as well as personalized and customized interaction.

2.5.1 Use tools and environment interaction. Integrating specialized tools with LLMs can fully leverage their unique advantages, addressing the limitations of LLMs in specific domain tasks. There are primarily two modes of interaction between LLMs and tools: First, external tools can continuously modify and refine the instructions for LLMs, enabling LLMs to perform more complex tasks. ToolFormer [200] utilizes prompts to guide the model to generate candidate texts that meet the instructions' requirements, followed by an automated process to filter high-quality results. Additionally, ART [179] employs a specific program syntax to build a task repository. When a new task emerges, it retrieves similar tasks from this repository to add to the prompt. Second, LLMs can also play a coordinating role in the system, issuing outlines for solving tasks and automatically matching sub-tasks outlined in the framework with APIs, systems and models that have specific functionalities to complete tasks [182].

LLMs significantly expand their application scope by interacting with external environments through unified natural language interfaces and tool use. For instance, WebGPT interacts with a text-based web browsing environment, enabling end-to-end optimization search and aggregation through imitation and reinforcement learning. WebShop [267] trains LLMs using real-world product information and crowdsourced textual instructions, enabling navigation and various operations on e-commerce websites. HuggingGPT interacts with the Huggingface community, utilizing ChatGPT to process user requests, selecting models based on function descriptions within the community, and executing AI tasks with the chosen models. The interaction of LLMs with database environments adds capabilities such as knowledge base management, unified data vectorization storage and indexing, and automated prompt generation and optimization. This ensures complete control over sensitive data and environments, preventing any data privacy breaches or security risks. Vector databases provide LLMs with expanded memory storage space and enhanced capabilities for advanced query processing [238].

2.5.2 Personalized and customized interaction. The enhancement of LLMs' capabilities has transformed the interaction between humans and personalized systems. Unlike traditional recommendation systems and search engines that passively filter information, LLMs provide a foundation for proactive user participation [32]. Firstly, LLMs extend the capability of fact retrieval into explicit knowledge bases, offering a more comprehensive knowledge source for recommendation systems [233]. This allows for a broader and more accurate understanding of user queries and preferences. Secondly, the instructions tailored for recommendation scenarios can make LLMs significantly outperform traditional recommenders [106]. The characteristics of users and their interaction history can be efficiently transformed into natural language instructions for input to LLMs [39]. Furthermore, the robust interpretability of LLMs enables the creation of precise, natural, and user-preference-aligned custom explanations, alleviating the limitations of traditional, formulaic explanations [124]. Lastly, LLMs with strong reasoning and decision-making capabilities, such as GPT-NAS [269], GENIUS [293], and LLMatic [165], provide enhanced support for personalized customization services. These models leverage their advanced cognitive capabilities to deliver more accurate and user-centric recommendations, enhancing the overall personalization experience.

3 The Application of LLMs in Specific Domains

LLMs have different applications in different domain scenarios. For instance, in the medical and ledge domains, they may function as domain experts engaged in dialogues or summarizing documents [185, 208]. Systematically summarizing the application methods of LLMs in various domains facilitates combining these models with specific scenarios more efficiently. However, some research is often classified and summarized from the perspective of NLP tasks [105]. Kaddour et al. [105] classify applications in medical scenarios into medical question answering and comprehension, and medical information retrieval. However, LLMs may participate in medical diagnosis, diagnostic assistance, and other scenarios. The differences make it difficult for research results to be directly applied to real scenarios. Therefore, as depicted in Figure 2, we explore existing work to divide it from the perspective of real domain scenarios and to summarize the applications of LLMs in Medicine (Sec. 3.1), Law

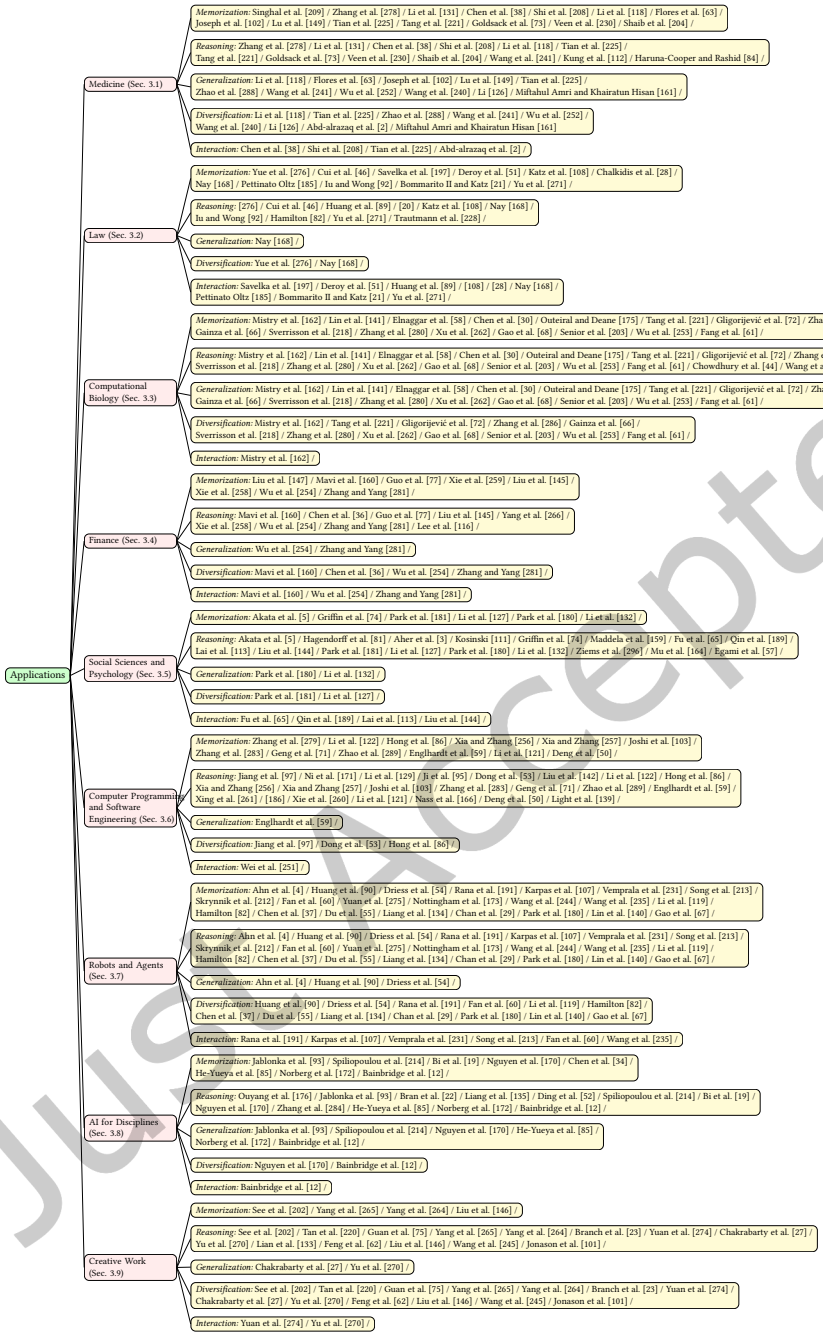

Fig. 2. Correspondence between domains and fundamental capabilities in this paper.

(Sec. 3.2), Computational Biology (Sec. 3.3), Finance (Sec. 3.4), Social Sciences and Psychology (Sec. 3.5), Computer Programming and Software Engineering (Sec. 3.6), Robots and Agents (Sec. 3.7), AI for Disciplines (Sec. 3.8), and Creative Work (Sec. 3.9). The aim is to provide more practical advice for the application of LLMs in various domains.

3.1 Medicine

There has been a longstanding interdisciplinary research line between medicine and artificial intelligence about learning representations and downstream applications [7, 43, 192]. The recent emergence of powerful LLMs [209] has further advanced related research, bringing new research paradigms and impressive abilities. LLMs can collect patient symptom information, process medical materials, and so on. In this section, we introduce different types of applications and analyze the capabilities shown in the applications, involving diagnosis for patients, assistance for doctors, and so on.

3.1.1 For Patients.

Medical Diagnosis. Considering the unequal distribution of medical resources and the lack of healthcare facilities during disease outbreaks, it is imperative to establish an accessible way through which everyone can seek medical assistance. With the Internet, LLMs can be accessed widely and in large numbers. And more importantly, LLMs have the capabilities to diagnose. As a chatbot, LLMs with reasoning and memorization capabilities can serve the role of doctors in diagnoses for every user in the world. A lot of studies build medical chatbots with medical data and technologies like continual pre-training [225], instruction tuning [131], and reinforcement learning from human feedback [278]. For further enhancement, Chen et al. [38] emphasize the ability of LLMs to inquire. They define the multi-turn questioning process as Chain of Questioning (CoQ). They collect the BianQueCorpus with multi-turn health conversations and train their model, BianQue, which has the better ability to conduct Chain of Questioning. Shi et al. [208] focuses on multiple task forms more than dialogue, such as suggestion and question answering (QA). When facing real-world patients with insufficient information and clear goals, it is important to perform various tasks to help real-world patients clarify their goals. More than text, LLaVA-Med [118] focus on training a large language-and-vision medical assistant. Based on LLaVA [143], LLaVA-Med is trained with multimodal biomedical data. The data is split into two stages of curriculum learning, namely concept alignment and instruction-tuning. Therefore, the model can acquire much biomedical knowledge and abilities as a medical assistant.

Knowledge Acquisition. Medical information used by medical professionals often involves complex and difficult knowledge, which is hard for patients to understand. That causes the gap between these two kinds of people. Fortunately, LLMs possess much medical knowledge and cover various language styles for both the average person and medical professionals. So, they can bridge the gap to facilitate the medical conversation. LLMs can answer questions from patients about complex and difficult medical knowledge. To make the knowledge easy to understand, LLMs [63] can also output simple and concise content of complex knowledge with their powerful generation ability. Joseph et al. [102] evaluates LLMs with multilingual medical text simplification. LLMs cannot only simplify text but also use different languages to bridge the communication gap. And Lu et al. [149] shows that after being prompted with summaries, models can generate simplified content with better simplification quality and consistency.

3.1.2 For Doctors.

Diagnostic Assistance. In diagnosis, doctors need key information for the sake of efficiency. However, there is much noisy information in the raw clinical text. This challenge calls for the application of NLP tasks, such as named entity recognition and relation extraction. Many medical LLMs are fine-tuned with diverse medical

tasks and show strong results in medical information extraction tasks [225]. More than understanding text and extracting keywords, generative LLMs [221] can summarize clinical content to provide more human-readable information. External knowledge sources, such as medical knowledge graphs Goldsack et al. [73], can be used by models to improve summarization. The medical summaries from LLMs even outperform human summaries in terms of completeness and correctness [230]. Meanwhile, more studies are conducted with different kinds of challenges, such as multiple documents [204] and ambiguous information [230], showing new problems to be explored.

Medical Report Generation. Reports for medical images are important but time-consuming. Writing reports also requires much medical knowledge and experience. That raises the need for automatic medical report generation. Multimodal models can be adapted to this application [288]. They use the vision ability to understand medical images, the reasoning ability to analyze features, and the language ability to generate reports. ChatCAD [241] combines previous medical models and the powerful LLMs. The features and information contained in the images are extracted into text. Then LLMs use the extracted text to generate reports. They show that LLMs can generate reports of higher quality. RadFM [252] uses a vision encoder to encode 2D/3D images into latent embeddings. With a large scale of multimodal data, the embeddings for vision features and language tokens are modeled together to generate outputs like medical reports. Other studies explore fine-grained representations [240] and pseudo data [126] for medical report generation.

3.1.3 For Medical Teachers and Students.

Courses. Practice is important for medical education. Meanwhile, the LLMs' capability to role-play makes them good actors. They can simulate virtual patients for medical students and provide practice for students. LLMs can generate clinical case studies or act as virtual patients to ask questions and provide subjective feedback. That allows students to practice in various scenarios and avoids risks in the real world [2]. Moreover, LLMs can provide practices related to medical images. Students can use ChatGPT [174] to generate medical images for interpretation practice [161].

Exams. During exam preparation, LLMs are knowledgeable virtual teachers. They not only answer questions from students but also give explanations by organizing knowledge and reasoning. Kung et al. [112] demonstrates that the AI explanation outputs are concordant and insightful. Additionally, LLMs are used as a new way for metrics in many NLP tasks. They also have the potential to participate in medical school assessments. They can generate questions and then identify key features and common mistakes in student responses [84].

3.2 Law

LLMs in the legal domain have started to play a key role, being utilized in legal research, contract analysis, drafting legal documents, and document review, and leverage their core capabilities to transform how legal information is processed, understood, and applied [42, 271]. Memorization allows LLMs to retain vast amounts of legal texts and precedents, ensuring comprehensive legal knowledge [28]. Reasoning enables the model to analyze complex legal scenarios, interpret laws, and suggest reasoned arguments [271]. Generalization helps in applying learned principles to new, unseen cases, enhancing adaptability [42]. Diversification ensures varied perspectives in legal analysis, promoting innovative solutions [169]. Lastly, interactive capabilities facilitate real-time assistance to legal professionals, making legal advice more accessible and efficient [46]. Together, these capabilities make LLMs invaluable for enhancing accuracy, efficiency, and innovation in the legal field.

3.2.1 Legal Education. Legal education often utilizes LLMs in examinations to test their capacity for statute recall, legal reasoning, and document drafting [46, 89, 276].

Many works have explored ChatGPT's performance in legal examinations. Bommarito II and Katz [21] shows that ChatGPT achieved passing scores in two types of bar examinations, validating ChatGPT's general legal understanding capability. Choi et al. [42] task ChatGPT with independently completing law school exams, revealing both challenges and insights for LLMs in legal assessments. Pettinato Oltz [185]'s research indicates that LLMs can assist law professors with administrative tasks, simplifying academic activities. Katz et al. [108] demonstrates that GPT-4 exhibits state-of-the-art performance across the entire UBE test, including multiple-choice questions, essays, and performance tests.

There are also studies based on open-source models, which enhance LLMs' capabilities in the legal domain by fine-tuning on legal datasets. Huang et al. [89] inject legal knowledge during the continued training of LLaMA [227], designing appropriate supervised fine-tuning tasks and integrating a retrieval module to improve authenticity in the text generation process, addressing the challenge of LLMs completing domain-specific tasks. Cui et al. [46] collects real legal consultation data and constructs datasets based on legal regulations and judicial interpretations. They developed ChatLaw based on Ziya-LLaMA [237], employing vector database retrieval methods, keyword search methods, and self-attention mechanisms to enhance the model's performance in the legal field. Yue et al. [276] construct a supervised fine-tuning dataset for the judicial domain using legal logic prompting strategies and fine-tuned an LLM with legal reasoning capabilities based on Baichuan [11].

3.2.2 Legal Aid. In the Legal Aid scenario, LLMs require diversification and interactive capabilities to better assist people in their support work. LLMs can aid law educators in managing their administrative responsibilities and in enhancing the efficiency of their academic research activities [185]. Cui et al. [46] demonstrates the role of LLMs in legal aid by presenting ChatLaw, a model that combines legal domain knowledge with vector databases to address legal problems effectively and reduce hallucinations typical in LLMs. Yue et al. [276] significantly aids legal professionals in case analysis, offers publicly accessible legal consultation, and serves as an educational assistant for law students, showcasing its versatility in various judicial scenarios.

3.2.3 Legal Drafting and Reasoning. The drafting of some legal documents and the reasoning of legal cases require the utilization of LLMs' generalization and reasoning capabilities. LLMs play a pivotal role in the legal sector in the creation of various legal documents, such as contracts, briefs, and pleadings. They ensure that these documents adhere to current legal standards and regulations [51, 92]. Additionally, they provide access to a range of templates and standardized language options, thereby streamlining the document drafting process [197]. Iu and Wong [92] focuses on ChatGPT's capabilities in legal drafting, particularly in creating various legal documents and developing logical legal strategies based on minimal input. Deroy et al. [51] concludes that current abstractive summarization models and LLMs are not yet adept for autonomous legal summarization, suggesting a human-in-the-loop approach and the need for better error detection methods. Savelka et al. [197] evaluates GPT-4's ability to generate accurate and clear explanations of legal terms, comparing a baseline approach with an augmented method using case law context.

3.2.4 Legal Analysis and Research. For legal case analysis and legal research, LLMs need to utilize their memorization and reasoning abilities to analyze relevant information based on legal statutes. LLMs enhance legal analysis and research by enabling the extraction and analysis of crucial contract clauses, identifying potential compliance risks and issues, and suggesting automated modifications for contract improvement Blair-Stanek et al. [20], Deroy et al. [51]. Additionally, LLMs offer advanced legal research tools that provide predictive insights on case outcomes, facilitate the efficient identification of precedents and relevant case laws, and streamline the overall legal research process for more accuracy and efficacy. Deroy et al. [51] evaluates ChatGPT's capabilities in legal research, focusing on case analysis, evidence identification, anticipating defenses, and advising on dispute resolutions, highlighting its potential despite data limitations and a need for more complex testing scenarios.

Blair-Stanek et al. [20] reveals that GPT-3's ability to apply legal statutes to specific scenarios is limited, achieving modest accuracy in both simple and more complex synthetic legal tasks, suggesting a significant need for improvement in LLMs' statutory reasoning capabilities. Hamilton [82] develop a Transformer-based system to simulate U.S. Supreme Court decisions from 2010-2016, finding it accurately predicted real-world rulings and reflected justices' legal ideologies.

3.3 Computational Biology

Computational biology involves leveraging computer science to model and simulate diverse biological systems, enabling a deeper understanding of structures and processes of life [201]. Genes and proteins are responsible for nearly every task of cellular life [167]. However, the structural complexity of these biomolecules leads to a plethora of unknown proteins and genes, rendering traditional analysis methods ineffective due to their high complexity, low robustness, and limited generalization. LLMs, owing to their robust capabilities in representing, understanding, predicting, and generalizing on sequence data, have seen gradual application in protein representation learning and structure prediction tasks, yielding remarkable outcomes.

3.3.1 Embedding Modeling. Protein embeddings are a technique for encoding the functional and structural properties of proteins. These embeddings can be used for a variety of downstream tasks, such as sequence clustering and sequence classification [45].

Top prediction methods [1, 199] typically involve collecting evaluation information (EI) about the family which can be computationally expensive [13] and may not be applicable to all proteins [183, 190].

Leveraging their powerful representation capabilities, LLMs can extract general representations from extensive protein datasets [104, 215] and maintain strong generalization by directly capturing evolutionary patterns and sequence features inherent to protein structures. Based on different levels of training data, we introduce the following three types of LLM application methods.

Sequence-based. Models like ESM-2 [141] and ProtTrans [58] utilize autoregressive, autoencoding, or combined objectives during pre-training on amino acid sequences. Subsequently, these embeddings serve as inputs for downstream tasks, achieving SOTA performance. CaLM [175], pre-trained on codon sequences instead of amino acids, exhibits comparable or superior representation information quality when compared to SOTA models, even those with more than 50 times the parameters.

Structure-based. Relying solely on protein sequences for training may limit the predictive capabilities of LLMs, particularly those heavily reliant on the 3D structures of proteins [234]. 3D protein structure [234, 286] or protein surface [66, 218] information are incorporated during the pre-training stage using graph neural networks or multi-view contrastive learning. The integration of such structural information empowers the learned protein embeddings to make predictions solely based on protein sequences, which is crucial when acquiring a dependable protein structure is challenging or time-intensive [234].

Multimodal-based. To leverage the powerful cross-modal representation and alignment capabilities of language models effectively, OntoProtein [280] harnesses contrastive learning to integrate a knowledge graph consisting of Gene Ontology[1] and its related proteins into the pre-training of protein language models. Likewise, ProtST [262] integrates textual data describing essential attributes (e.g., protein's function) at various granularities through three types of tasks (unimodal mask prediction, multimodal representation alignment, and multimodal mask prediction) during the pre-training phase. The aforementioned methods integrate informative biological knowledge from KGs and textual sources into protein embeddings, thus encapsulating a wider range of essential information.

[1]https://geneontology.org/

3.3.2 Structural Prediction. Predicting protein structures [87] from sequences is a crucial task with implications for function prediction, drug design, and a comprehensive understanding of related biological processes [198].

Traditional approaches utilize different information (structure-related properties, templates, multiple sequence alignment, etc.) to aid in structure prediction (DESTINI [68], AlphaFold [203], AlphaFold2 [104] and RoseTTAFold [156]). However, searching for such information can be resource and time-intensive, and its generalization ability in real-world scenarios may be limited.

Leveraging the robust memorization and understanding capabilities of LLMs, OmegaFold [253] utilized OmegaPLM to extract the residue and pairwise embeddings from a single sequence, which are fed into Geoformer and the structure module enabling the direct prediction of the 3D structure of proteins. Similarly, as LLM excels in capturing abstract features with enhanced generalization, ESMFold [141], HelixFold-Single [61] and RGN2 [44] trained LLMs to learn additional structural and functional properties to replace the role of MSAs (multiple sequence alignment), thereby augmenting the accuracy and generalization of the model for structure prediction. Moreover, utilizing the learning capability of LLM, trRosettaX-Single [242] incorporates two supervised learning tasks—amino acid type prediction and inter-residue geometry prediction—to bolster the model's capacity for extracting embedding information, which combined with an MSA knowledge-distilled multi-scale neural network to further enhances the performance of the structural prediction system.

3.4 Finance

In the financial domain, substantial volumes of textual data accumulate, encompassing intricate professional knowledge and real-world scenarios. Concurrently, users' inquiries exhibit diversity and pronounced personalized requirements. Leveraging the robust language understanding, short-term memory, reasoning capabilities, and other inherent features of LLMs like ChatGPT, this data can be efficiently processed. Such capability not only unveils novel prospects for artificial intelligence applications in finance but also effectively aids users in addressing a myriad of financial challenges, thereby fostering the advancement of the financial domain [254, 281], such as Finance Question Answering [160], Stock movement prediction [266], Robo-Advisor [145], and financial decision-making [273].

3.4.1 Stock Movement Prediction. Stock movement prediction is a crucial task in the financial domain, with the potential to greatly influence investment strategies and decision-making processes. Accurate predictions empower investors and traders to devise more informed strategies, ultimately improving their market decision-making prowess. With their robust language understanding and short-term memory capabilities, LLMs have prompted some researchers to explore their application in predicting stock movement. Specifically, both MindLLM [266] and COT-ChatGPT [258] employ the COT method, utilizing the text analysis and numerical reasoning capabilities of LLM to thoroughly analyze the latent semantics in the context, aiming to achieve precise predictions of stock movement. More recently, TIMECAP[116] leverages LLMs as context-aware tools for time series event prediction, further improving performance through contextualization and enhancement mechanisms.

3.4.2 Financial Question Answering. The question answering system is crucial in NLP. Similarly, it's a vital application in the financial field. Researchers are increasingly applying LLMs to financial question answering systems, inspired by their success in broader question answering tasks [77]. Liu et al. [147] and Xie et al. [259] leverage the long-term memory capacities inherent in LLMs to undergo training on financial datasets, thereby accomplishing financial question-answering tasks. Mavi et al. [160] develop an algorithm in the financial domain's question-answering scenarios, incorporating a retrieval-augmented generation model, which relies on regular expressions to match semi-structured documents while fully leveraging the reasoning and generative capabilities of LLM. What's more, a range of complex issues arises, including situations that require the application of numerical reasoning capabilities. To address these challenges, methods such as DocMath [36], POT [290], and

COT-GPT4 [128] employ LLMs based on program generation to tackle partial problems involving large-scale numerical computations or complex mathematical expressions. These models generate intermediate results, effectively performing numerical reasoning, and leverage the reasoning and generative capabilities of LLMs to ultimately produce answers.

3.4.3 Robo-Advisor and Decision-Making. The paramount objective of a robo-advisor is to furnish human users with easily comprehensible financial advice. Considering the exemplary language understanding, generation capabilities, and role-playing abilities of LLMs, LLMs can serve effectively as robo-advisors, delivering personalized financial guidance tailored to user preferences. Against this background, FinGPT[115] built a robo-advisor framework with data as the core element. After the user provides their own characteristics, the model provides personalized financial advice to the user by accessing information from the data center and analyzing the potential impact of the data, as well as the user's characteristics. On the other hand, FinMem [273] uses an LLM to build an Agent framework specifically designed for financial decision-making, which includes three key modules: Profiling (feature description) is used to outline the characteristics of the agent; the Memory module has hierarchical processing functions. Helps agents absorb hierarchical financial data from reality; finally, the Decision-making module is responsible for transforming insights gained from Memory into specific investment decisions. This design aims to improve the decision-making performance of robot advisors in the financial domain.

3.5 Social Sciences and Psychology

The emergence of LLMs provides new research avenues for social sciences and psychology. In psychology, LLMs are utilized across experimental to applied psychology, enhancing the exploration of theories and applications. In the field of social sciences, the use of LLMs to simulate social relationships and behavioral patterns provides new ways to study social science theories. Additionally, LLMs have also played a significant role in the development of computational social science. The powerful reasoning and generalization capabilities of LLMs enable researchers to efficiently process and analyze vast amounts of data. Furthermore, they facilitate the extraction of more precise analytical results.

3.5.1 Experimental Psychology. Experimental psychology is the use of experimental methods to study psychology and contribute to the development of theory. In practical applications, LLMs can be used to simulate human behavior, leveraging their generalization capabilities to conduct various psychological experiments[5]. These experiments aim to determine if LLMs can replicate results consistent with human studies. Research indicates that LLMs often demonstrate intuitive behavior akin to humans[81]. Furthermore, it's observed that LLMs tend to simulate human behavior more accurately and faithfully than smaller ones[3]. LLMs can also be used to conduct tests of Theory-of-Mind(ToM)[111]. Experiments indicate a significant advancement in the capabilities of these models to mimic human-like understanding and reasoning. Additionally, LLMs are effective in modeling psychological changes. Comparisons between human judgments and those derived from LLMs in behavioral tests often show consistency, suggesting LLMs have the potential to act as models of the effect of influence[74].

3.5.2 Applied Psychology. Applied psychology uses psychological principles to solve specific problems, such as treating mental health issues. In applied psychology, LLMs can be used to provide psychological interventions for users [159]. LLM-Counselors Support System [65] is a psychological intervention model for users with negative emotions built based on the LLMs. It analyses the conversations between volunteers and online users, identifies potential psychological problems, and provides suggestions for volunteers to respond. LLMs have long text comprehension capabilities, and thus can be used for depression detection based on social media content. Chat-Diagnose[189] analyzes and detects social media content for diagnosis and provides personalized advice through interactive dialogue. In addition, the interactive capability of LLMs can be used for psychological counseling, Psy-LLM[113] is an online psychological counseling model that provides professional answers to users' requests

for psychological support and mental health advice, and ChatCounselor[144] is based on real conversations between counseling clients and professional psychologists and can be used as a personalized counseling assistant.

3.5.3 Social Simulation. The experiments on human societies are usually expensive and may even be ethically problematic. Then with the development of LLMs, there is a gradual shift towards the use of LLM-based agents to build virtual environments and simulate social phenomena[127, 181]. Generative Agents[180] builds multiple agents to simulate human interactions in a sandbox environment and demonstrates that they produce believable simulacra of both individual and emergent group behavior. In addition, LLMs can be used for task-oriented social simulation. MetaAgents[132] introduce collaborative generative agents that endow LLM-based Agents with consistent behavior patterns and task-solving capabilities. MetaAgents also proposes a multi-module framework to equip collaborative generative agents with human-like reasoning capabilities and expertise. Moreover, using the reasoning and role-playing capabilities of LLMs, a social network simulation system can be constructed to mimic the behavior of real humans in social networks[285].

3.5.4 Computational social science. Computational social science(CSS) analyses large-scale data from the web, social media platforms, and other sources to answer important social science questions or advance theories of human behavior. However, enabling the analysis of these massive datasets can be very time-consuming and complex. LLM has excellent comprehension and reasoning capabilities to process and analyse large amounts of textual data. Therefore, LLMs are currently used in CSS mainly to reduce costs and increase the efficiency of social science analyses. Ziems et al. [296] provides researchers with a road map for using LLMs as CSS tools by evaluating LLMs on an extensive suite of CSS tasks. Mu et al. [164] evaluates the zero-shot performance of two publicly accessible LLMs, ChatGPT and OpenAssistant, in the context of six computational social science classification tasks, and also investigates the effects of various prompting strategies. For statistical properties such as asymptotic unbiasedness and proper uncertainty quantification in CSS research, Egami et al. [57] proposed a new algorithm for downstream statistical analyses using the output of the LLMs while guaranteeing statistical properties.

3.6 Computer Programming and Software Engineering

LLMs are also gradually playing a key role in the fields of computer programming and software engineering. LLMs can not only fully understand the natural language needs of developers and rely on their powerful code generation capabilities to generate code that meets the needs; they can also rely on their excellent semantic understanding capabilities to understand the code, identify errors in the code, and finally generate correct code. In addition, LLMs can use role-playing capabilities to simulate different roles to better complete tasks and assist developers in their work.

3.6.1 Computer Programming. In the field of computer programming, LLMs mainly assist developers in code-related aspects, reflected in the three aspects of code generation, automatic program repair, and code comment generation.

Code Generation. Code generation is a technique that aims to automatically generate code based on developers' requirements. It can reduce repetitive coding efforts and improve software development productivity. Within this sphere, LLMs excel in generating code through interactive dialogues, skillfully interpreting natural language requirements, and crafting code that precisely meets these specifications. Additionally, LLMs adeptly employ brainstorming strategies, generating a variety of ideas from natural language inputs and identifying the most effective solutions for code generation[129]. Moreover, LLMs play versatile roles in a multi-stage collaborative process and fully embody their role-playing ability. As knowledge providers, they supply pertinent solution-related insights, and as self-reflective expert programmers, they refine and correct the generated code[97]. Moreover,

LLMs can also simulate different roles such as analysts, coders, and testers to complete code generation in the form of multi-stage mutual cooperation[53]. LLMs, functioning as collaborative agents, can assume diverse roles like Product Manager, Architect, Project Manager, Engineer, and QA Engineer. This multi-agent collaboration, following a pipeline paradigm, ensures a more comprehensive and efficient code generation process, leveraging the unique strengths of each role within the LLM framework[86]. LLMs not only enhance code quality through meticulously crafted prompts, such as utilizing the causal analysis to analyze the causal relations between prompts and generated code and adjusting prompts accordingly[95], Chain-of-Thought (CoT)[142], and Structured Chain-of-Thought (SCoT)[122], but also improve the quality of generated code by verifying the correctness of the generated code[171] and modifying the generated code based on the generated code and its running results[279]. Recently, optimization-based techniques such as SCATTERED FOREST SEARCH (SFS) have also been proposed to further enhance the reasoning scalability and solution diversity of LLMs in code generation tasks [139].

Automatic Program Repair. Automated Program Repair (APR) seeks to reduce the manual bug-fixing effort of developers by automatically synthesizing patches given the original buggy code. LLM can fully understand the semantic meaning of the program through the iteration of the incorrect patches and corresponding test result information in the dialogue, conduct compositional reasoning, and finally generate the correct patch [256, 257]. In addition, LLM can also simulate different roles such as testers, developers, and reviewers to implement interactive program repair in the form of multi-stage collaboration[283], fully embodying role-playing ability. LLM can not only handle multilingual program repair[103] but also generate more effective patches by combining with Completion Engine[251].

Code Comment Generation. Code comment generation (a.k.a. code summarization) aims at generating natural language descriptions for a code snippet to facilitate developers' program comprehension activities. In this context, LLMs excel at generating code annotations through a question-and-answer format. LLMs are capable of learning from context and deducing appropriate comments for a given piece of code by analyzing a set of similar code comment generation examples[289]. Moreover, LLMs can generate annotations based on prompts with various intentions. This ability to produce multiple annotations reflecting different intentions[71] offers developers a multifaceted understanding of the code, allowing them to grasp its functionality and purpose from diverse angles.

3.6.2 Software Engineering. In the field of software engineering, LLMs can assist developers in developing and testing different types of software.

Assist Software Development. LLMs play a pivotal role in aiding developers with software development tasks, including Embedded Systems Development and the development of AI-Native Services. In the embedded system development workflow [59], LLMs engage collaboratively with developers through an interactive, multi-round dialogue. LLMs can understand the requirements provided by the developers and generate corresponding code, efficiently identify and report errors, and deduce executable code. During the entire development process of AI-native services[261], LLMs adeptly craft and test AI chains tailored to the developers' requirements, paving the way for the creation of sophisticated AI-native services. LLMs can also complete the development of AI-native services based on the logic and functions of the AI chain.

Assist Software Testing. Software testing plays a vital role in ensuring the quality of software applications, and LLMs can assist testers in software testing. LLMs can fully understand user-written bug reports and output corresponding test cases to facilitate software fault localization[186]. Beyond this, LLMs can not only be used to build automated unit test generation tools, generate unit test code and automatically verify repairs[260], but can also learn from historical reported bugs to generate similar bug-triggering code snippets to find new bugs[50]. Additionally, LLMs can augment static analysis tools, providing deeper insights into code to unearth hidden

bugs[121]. LLMs can also be integrated with web element localization algorithms to improve the accuracy of web element positioning in GUI testing[166].

3.7 Robots and Agents

Agents built with LLM as core controllers have expanded from basic automation tasks to complex interactions and decision-making. The ability of LLMs to understand complex instructions and context, and utilize tools effectively, as well as a wide range of knowledge enables agents to accomplish more challenging tasks in both the real and virtual worlds. Moreover, the interaction between multiple agents can better perform tasks in various scenarios.

3.7.1 Agents and real-world interaction. LLM-based agents can encode a wealth of semantic knowledge about the world and have excellent text understanding and generation capabilities. To interact with the real world, LLMs require real-world experience. SayCan [4] combines robots with low-level skills and LLMs. The robot acts as the LLM's "hands and eyes," while the LLM supplies high-level semantic knowledge about the task. Specifically, the LLM provides a task-grounding to determine useful actions and the affordance function of each skill provides a world-grounding to determine what is possible to execute upon the plan.

The limitation of SayCan is that the system can only receive text input, which is not sufficient for many real-world tasks that require visual graphics. Inner Monologue [90] is built on SayCan, combines multiple perception models that perform various tasks, and utilizes multimodal environmental feedback through inner monologue to enable closed-loop planning. PaLM-E [54] trains visual and language data jointly to enable the LLM to make more valid plans and decisions for tasks in real-world scenarios. Exploiting the growing body of 3D scene graph research, SayPlan [191] allows LLMs to conduct a semantic search for task-relevant subgraphs from a smaller, collapsed representation of the full graph, scaling LLMs' task planning capabilities to large-scale environments.

Giving LLMs access to external tools can help LLMs solve challenges in areas that require professional knowledge such as engineering and reduce hallucinations. Embodied intelligence is a comprehensive and diverse field, so there are a variety of libraries and APIs. MRKL system [107] consists of the LLM, smaller specialized language models, and API calls to databases, significantly alleviating the hallucinations of the LLM on arithmetic operations tasks. To apply ChatGPT to the robot system, Vemprala et al. [231] create a high-level function library for ChatGPT to deal with so that ChatGPT can parse user intent and convert it to a logical chaining of high-level function calls. RestGPT [213] connects LLMs with RESTful APIs, which adhere to the widely adopted software architectural style for web service development. Aiming at fulfilling practical user instructions.

3.7.2 Agents and virtual world interaction. To investigate the capabilities of agents in tackling diverse tasks in complex domains, open-world virtual games like Minecraft serve as a suitable platform. Early research [212] mainly uses reinforcement learning to empower agents in completing low-level tasks, such as crafting simple items in Minecraft. Current researches rely on the powerful reasoning and planning capabilities of LLMs based on RL to enable agents to accomplish high-level tasks in a virtual environment, such as constantly exploring and developing new skills.

Minedojo [60] introduced a knowledge base with Minecraft videos, tutorials, and related pages. It leverages a large video-language model pre-trained on this base as a reward function to train the agent. This approach leverages the long-term memory capabilities and cross-modal learning capabilities of LLMs. Plan4MC [275] focuses on learning fine-grained basic skills and planning based on them, thus solving multi-task learning problems. DECKARD [173] employs a dual-phase approach utilizing LLMs for item crafting in Minecraft. In the first phase, the task is decomposed into a sequence of subgoals. In the second stage, the agent executes subgoals and explores the environment. This approach makes use of the short-term memory capabilities of LLMs.

DEPS [244] prompts the LLM to explain the failures in the execution phase, to better correct errors in the initial plan.

However, the method of decomposing high-level tasks into a set of sub-goals based on Minecraft recipes lacks sufficient exploration flexibility. Voyager [235] employs an automated curriculum by prompt GPT-4 [174] that enables open-ended exploration driven by curiosity. To store and retrieve complex behaviors, it introduces a growing library of executable code skills, using code as the action space rather than low-level motor commands. Additionally, to generate executable code for actions, Voyager incorporates an iterative prompting mechanism that encompasses environment feedback, execution errors, and self-verification.

3.7.3 Multi-agent interaction. As an isolated entity, a single agent lacks communication and interaction with other entities, which limits their ability in complex scenarios. Multi-agent systems (MAS) [229] focus on how to organize and coordinate multiple agents to complete tasks collaboratively. In MAS, agents communicate using natural language, making it easier for humans to understand and engage with them.

Researchers usually adopt a role-playing framework, where different agents play distinct roles. In CAMEL [119], one agent plays the role of assistant and another plays the role of user. This configuration adds specificity to the tasks assigned to the agents, allowing for a more concrete and targeted interaction between them. They chat with each other to communicate and solve their assigned tasks. Blind Judgment [82] trains nine independent models with the opinions of nine Supreme Court justices and uses this MAS to simulate judicial rulings. AgentVerse [37] is a framework that dynamically adjusts the composition of a group based on current progress, effectively simulating the problem-solving process of human groups. Additionally, it has revealed some emergent behaviors exhibited by agents in multi-agent collaborations, such as volunteer and conformity behaviors.

Du et al. [55], Liang et al. [134], and Chan et al. [29] employ a multi-agent adversarial debate framework to complete challenging tasks such as commonsense machine translation, counterintuitive arithmetic reasoning, open-ended QA and dialogue response generation while finding that this framework aligns more closely with human preferences.

Multi-agent systems can also be employed to simulate human societies, necessitating the involvement of more agents, with each agent emulating human behaviors observed in real-world societies. Generative Agents [180] and AgentSims [140] build a virtual town with multiple agents, each of which has the capabilities of observation, memory, planning, and reflection, allowing human users to interact in natural language. Gao et al. [67] focuses on the propagation of attitudes and emotions in a virtual society, making progress at the cognitive level.

3.8 AI for Disciplines

The progress of many research-oriented disciplines is currently hindered by limitations in data utilization capabilities and reliance on human expertise. There is an urgent need for a role with exceptional data comprehension capabilities and exploratory thinking to address these limitations. The recent emergence of LLMs has demonstrated significant potential in memory, reasoning, and generation, particularly in disciplines such as physics, chemistry, climate, and mathematics. This section primarily introduces the roles played by LLMs in these subject areas and the specific capabilities they exhibit.

3.8.1 Physics and Chemistry. The realm of physical chemistry encompasses the complexities of atomic structures, chemical bonds, states of matter, and their properties. In addressing these challenges, accumulating extensive experience and precise reasoning are crucial. LLMs have emerged as significant computational tools in physical chemistry due to their advanced reasoning capabilities and systematic analytical approach to complex molecular phenomena.

In chemical research, traditional chemical formula deduction usually relies on manually summarizing experience, which may be time-consuming and unstable. LLMs can directly help researchers efficiently deduce chemical

formulas [176]. In addition, for structured chemical experimental data, manual analysis by human resources is often time-consuming and limited in effectiveness. LLMs can help researchers extract important information from the data, which is beneficial for predicting the properties of molecules and materials [93].

In the field of physics, predicting crystal properties requires consideration of the complex interactions between atoms and molecules within the crystal [194]. Similarly, when designing quantum computer architectures, it is necessary to consider complex physical processes. LLMs can simulate these interactions and physical processes, helping researchers intuitively understand these processes and assist in related research [135]. Moreover, LLMs can also bring benefits to physics research and education. They can be directly used to solve physics application problems and provide detailed solution steps for teaching purposes [52]. Furthermore, it can also reason about the states and events of physical entities, helping to understand the relevant properties of various physical entities [214].

3.8.2 Math. Traditional methods of generating mathematical application problems typically require manual construction of both the problem and its solution, which is costly in terms of labor, time, and resources. In contrast, large language models can quickly construct mathematical application problems and generate complete and detailed solution steps [85]. Moreover, their abilities extend beyond this - they can also make analogies and rewrite different types of application problems with multiple solution methods [172]. Additionally, understanding complex mathematical problems may be challenging for students, but LLMs can annotate metadata for mathematical problems, helping students better understand and learn these problems [12].

3.8.3 Weather. The field of climate prediction demands LLMs capable of handling various types of data (such as images and text), historical data from multiple periods, and a grasp of relevant domain-specific physical knowledge [34]. This requirement necessitates LLMs to possess memory, reasoning, and generalization capabilities.

Traditional weather forecasting methods incur high numerical computation costs. The Pangu Weather Model [18], leveraging its outstanding long-term memory and memory retrieval capabilities, effectively harnesses meteorologists' knowledge, achieving, for the first time, accuracy surpassing that of traditional numerical computation models under low computational loads. LLMs not only reduce costs in terms of computational load but also exhibit superior performance in handling highly complex data scenarios. ClimaX climate models [170] demonstrate exceptional data perception and multimodal prediction capabilities despite training with limited resolution and computational resources. The learning and utilization of physical laws are also pivotal in weather prediction, a task LLMs are adept at accomplishing. NowcastNet [284] generates credible near-term precipitation forecasts in Physics, displaying clear multiscale patterns across a 2048 km $\times$ 2048 km area with lead times extending up to 3 hours. Leveraging historical climate data over short periods accelerates weather prediction and enhances accuracy. Fengwu [34] relies on the inherent short-term memory capacity provided by the Replay Buffer mechanism, extending forecast lead times to 10.75 days.

3.9 Creative Work

With advancing LLM capabilities, research has expanded beyond traditional machine learning into human-machine collaborative creation and heuristic domains. These models enhance creative efficiency and provide diverse inspirational stimuli, facilitating novel ideation processes.

3.9.1 Story and Script Generation. In the field of story and script generation, it is necessary for models to generate fluent and grammatically correct sentences, but more importantly, to maintain narrative coherence and thematic relevance in the content [202]. Enhancing the long-text generation capability and context-learning ability of LLMs can improve the quality and usability of the model's generated output.

Tan et al. [220] proposed the ProGen method. This approach utilizes TF-IDF to calculate the informativeness of words, and based on a set of words with the highest information content, the model iteratively expands to generate

a complete passage. Guan et al. [75] proposed the HINT model, which generates a high-level representation of each decoded sentence to model long-range coherence. Two pre-training objectives, i.e., similarity prediction and order discrimination, are used to learn the representations at the sentence level and at the chapter level.

Guiding LLM appropriately in prompts can stimulate the potential of LLMs in storytelling. The Re3 framework [265] generates stories exceeding 2000 words through a recursive prompting and revision framework. This framework utilizes GPT-3 to generate a structured story framework, which serves as a guide for continuing the story. With each iteration of generation, context-relevant outline information is injected into the model. Through repeated injections and filtering of content, paragraph coherence and quality are ensured.

Similarly, DOC framework [264] ensures the integrity of the narrative by separating the guidance of the outline from the generation of paragraphs. The DOC framework consists of a detailed outliner and a detailed controller. The detailed outline creates a more detailed and hierarchical outline, shifting the creative burden from the main drafting process to the planning stage.

Furthermore, Several applications leverage LLMs for collaborative authoring. Branch et al. [23] utilized GPT-3 for collaborative storytelling, where the AI tracked story progress and character behavior in real-time during live performances, generating spontaneous narrative to advance the plot. CoPoet [27] features a language model finetuned on poetry-writing instructions, offering a natural language interface for users to request suggestions through various instruction types, enabling poems that meet both content and stylistic requirements. Yuan et al. [274] introduced WordCraft, a GPLM-based AI assistant with a rich graphical interface supporting vocabulary replacement, style transformation, continuation, and rewriting tasks.

3.9.2 Multimodel Generation. According to the statistics from Davies et al. [48], the most widely applied areas of AI in the creative industry are music, performing, and visual art, which account for 20% of the projects in the Gateway to Research statistical results. The comprehension ability of LLMs for complex instructions and their enhanced capability to interact with the environment further strengthen the practicality of AI in these domains.

In music composition, researchers have developed AI systems to enhance creativity. Yu et al. [270] integrates multiple tools with LLMs as task planners to select optimal processing methods. Hussain et al. [91]introduced the M2UGEN framework, which processes images, audio, and video for generation or recognition tasks. This framework comprises three components: an Encoder utilizing ViT and ViViT for image/video processing and MERT for music; a Controller employing LLAMA2 to interpret input signals and intentions; and a Generator leveraging AudioLDM 2 and MusicGen for music creation. The Composer LLM [101] facilitates music composition through retrieval-augmented generation, particularly for abc-symbolized folk tunes. Connected to a music database, this LLM analyzes and retrieves information based on user requests, first generating a brief commentary to structure its thinking before producing musical symbols responsive to provided prompts.

In the domain of visual design, LLMs have been employed for image generation and style design. The potent image generation capability of diffusion models is well-known, and building upon this foundation, Lian et al. [133] utilize LLM to generate a benchmark structure in natural language form, subsequently controlling the diffusion model's generation through this structure. Feng et al. [62] introduced the LayoutGPT model, which exhibits stronger control over the structural relationships of objects in images and can be used to enhance the generation effects of diffusion models.

3.9.3 Heuristic task. Heuristic tasks are those where the primary goal is not generation but rather to inspire users' creativity and ideas. These tasks test the model's ability to recall knowledge and explore ideas.

Liu et al. [146] designed the CoQuest system for the extension of research topics. This system, supported by extensive knowledge bases and facilitated by frequent interactions and feedback between humans and AI, progressively refines and expands research topics proposed by users, making research tasks more efficient. Wang et al. [245] introduced the SSP framework for creative tasks, which involves collaborative efforts between LLM and multiple roles. By leveraging their respective strengths and knowledge, this framework unleashes

Domains	Memorization	Reasoning	Generalization	Diversification	Interaction	Total
Medicine	16	15	12	10	5	25
Law	11	10	1	2	9	19
Computational Biology	17	20	17	13	2	20
Finance	21	21	3	4	5	24
Social and Psychology	6	20	2	3	4	22
Programming	12	21	1	3	1	23
Robots and Agents	22	22	3	13	6	22
AI for Disciplines	9	13	7	3	2	15
Creative Work	4	16	2	15	2	16

Table 1. We analyzed the fundamental capabilities across various domains in this table. For example, we analyzed 19 papers in the law domain. Among these 19 papers, 11 focused on memorization capabilities, 10 on reasoning capabilities, 1 on generalization capabilities, 2 on diversification capabilities, and 9 on interaction capabilities. Based on this table, we construct the radar chart in Figure 3.

the collaborative cognitive capabilities of LLM, enhancing the model's capabilities and overall performance in creative tasks.

4 The Capabilities Assessment of LLMs in Specific Domains

LLMs' performance in specific domains depends on their fundamental capabilities. However, we often evaluate LLMs based on their performance on benchmarks, but their strong performance on benchmarks may not necessarily translate to domain scenarios [78]. For example, while InstructBLIP exhibits outstanding performance in image caption tests, its performance significantly diminishes in online interactive evaluations closer to real-world scenarios [47].

Guo et al. [78] highlights that the range of model capabilities assessed by different benchmarks varies, leading to discrepancies between benchmarks and the model's performance in domain scenarios. Therefore, the quantitative assessment of fundamental capabilities within specific domains is crucial for users in choosing the most appropriate benchmarks. We employ a case study approach to conduct a case-by-case statistical analysis of the articles in Section 3, deriving quantitative values for the important capabilities in each domain through expert evaluation (see Figure 2). Taking the medical field as an example, among the 25 papers categorized for this study, methods involving memorization capabilities are present in 16 papers, reasoning capabilities in 15, generalization capabilities in 12, diversification capabilities in 10, and interactive capabilities in 5. Therefore, memorization capabilities emerge as the most critical in the medical domain, accounting for 64% of the focus. In table 1, we list the number of papers covered in each domain and analyze the fundamental capabilities demonstrated by these papers across various domains. Based on this data, we create radar charts for each domain.

As shown in Figure 3, we constructed radar charts to illustrate the relative importance of different fundamental capabilities in various domains. Based on these radar charts, researchers can quantify the differences between benchmarks and real scenarios. In the following chapters, we will introduce our selection strategy in the medical and computer programming domains as examples.

4.1 Medical

The radar chart shows that memorization capabilities (64%) and reasoning capabilities (60%) are important fundamental capabilities in the medical domain. For Medical Diagnosis and Knowledge Acquisition, LLMs need to engage in dialogues with patients using their medical knowledge. In this context, long-term memory related to the domain knowledge and reasoning capabilities to assist in answering questions is crucial for model performance.

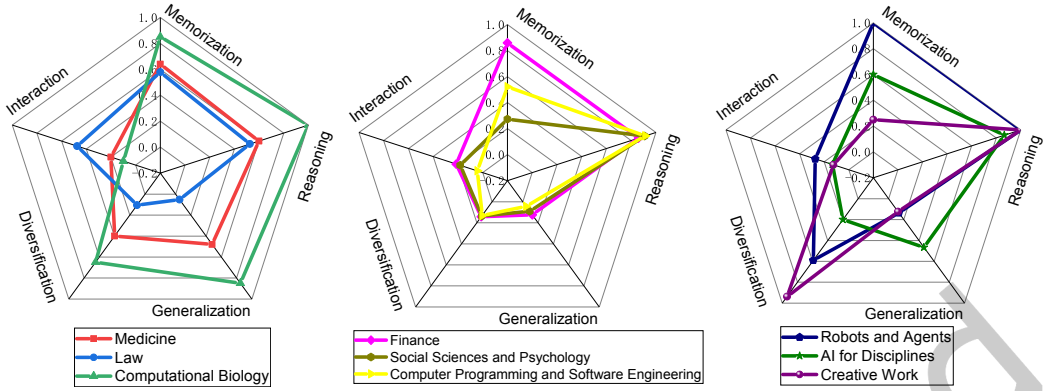

Fig. 3. The radar charts of LLMs' fundamental capabilities in various domains.

MedQA		MedMCQA		PubMedQA	
Model	Acc(%)	Model	Acc(%)	Model	Acc(%)
LLAMA2-70B [40]	61.5	PubmedBERT [177]	41.0	BioGPT [151]	78.2
Flan-PaLM [209]	67.6	BioMedGPT-10B [154]	51.4	Flan-PaLM [209]	79.0
Meditron-70B [40]	70.2	Codex [136]	62.7	Med-PaLM 2 [210]	79.2
Med-PaLM 2 [210]	85.4	VOD [137]	62.9	BioGPT-Large [151]	81.0
GPT4 [174]	90.2	Med-PaLM 2 [210]	72.3	Meditron-70B [40]	81.6

Table 2. The performance of different LLMs on MedQA, MedMCQA, and PubMedQA.

For Diagnostic Assistance and Medical Report Generation, LLMs typically assist doctors in reading patient case information and generating treatment plans. In this context, short-term memory capabilities and reasoning capabilities play a crucial role.

Therefore, we recommend applying LLMs that excel in memorization capabilities and reasoning abilities benchmarks to the medical domain. Here, we provide three recommended medical benchmarks, with Table 2 summarizing the performance of different LLMs on these three benchmarks.

MedQA [98] is a medical text question and answer dataset in a multiple-choice format. It aims to test the professional knowledge and clinical decision-making abilities of LLMs. The examination of professional knowledge mainly targets the **memorization capabilities** of LLMs, while the assessment of clinical decision-making abilities primarily focuses on the **reasoning and generalization capabilities** of LLMs.

MedMCQA [177] is a large-scale multiple-choice question and answer dataset, with data sourced from All India Institute of Medical Sciences (AIIMS) and National Eligibility cum Entrance Test (NEET PG). Different from the MedQA dataset, besides directly examining the **memorization capabilities** of LLMs, the MedMCQA dataset includes detailed explanations for each answer, requiring LLMs to possess deep language **reasoning capabilities**.

PubMedQA [99] is a biomedical question answering dataset collected from PubMed abstracts. The task involves generating an answer in a multiple-choice format of yes/no given a question. This dataset demands **reasoning** over biomedical research texts, especially the capabilities to understand and analyze quantitative content, to answer questions.

HumanEval		MBPP	
Model	pass@1	Model	pass@1
XwinCoder-34B [223]	75.6	StarCoder2-15B [148]	66.2
Magicoder-6.7B [250]	76.8	XwinCoder-34B [223]	67.7
Coderllama-34B [193]	79.9	Coderllama-70B [193]	75.4
WizardCoder-33B [155]	77.4	WizardCoder-33B [155]	78.9
DeepSeek-Coder-33B [76]	79.9	DeepSeek-Coder-33B [76]	78.7
GPT-4-Turbo [174]	88.4	GPT-4-Turbo [174]	83.5

Table 3. The performance of different LLMs on HumanEval and MBPP.

We chose the medical domain as a case study to further illustrate the practicality and effectiveness of our approach. We identify memorization capabilities and reasoning capabilities as the most crucial fundamental capabilities in the medical domain. Based on this, we recommend models that perform well on benchmarks (MedQA [99], MedMCQA [177], and PubMedQA [99]) focused on these capabilities. Through this process, we discover that Med-PaLM 2 excels in these benchmarks. This finding is consistent with the industry recognition that the model has received in the medical domain. Specifically, renowned organizations such as HCA Healthcare, BenchSci, Accenture, and Deloitte have deployed the Med-PaLM 2 [210] model across various medical scenarios, validating its value in real-world applications. In contrast, although RobotGPT-30B [100] and jianpeiGPT [243] performed well on the CMB benchmark [243], their performance in real-world applications does not match that of the former, further proving the effectiveness of our selection methods.

4.2 Computer Programming

According to the radar chart, reasoning capabilities (91%) are considered the most crucial skill in computer programming and software development, followed by memorization capabilities (52%). Since code is a symbolic, hierarchical, and logic-driven language commonly used for handling complex tasks, the reasoning capabilities of LLMs are applied in various code scenarios. Short-term memory helps LLMs understand requirements and gather contextual information in code generation and automatic program repair.

Evaluation criteria for programming-related tasks evolve from single-type code language and static metrics to multi-type code languages and metrics [277]. Among these, evaluation standards involving multi-language, multi-type metrics require models to possess stronger reasoning capabilities. In this paper, we recommend HumanEval [35] and MBPP [9] benchmarks as the basis for selecting LLMs for programming-related scenarios. Table 3 presents the performance of some LLMs on these benchmarks.

5 Conclusion

In this paper, we summarize the fundamental capabilities of LLMs in domain applications and illustrate how they collaborate. Simultaneously, we summarize the applications of LLMs in various domains from real-world perspectives. Furthermore, we outline the key capabilities emphasized in different domains, aiding users in more accurately applying LLMs in domain applications.

References

[1] Almagro A. 2017. DeepLoc: prediction of protein subcellular localization using deep learning. *Bioinformatics* (2017).

[2] Alaa Abd-alrazaq, Rawan AlSaad, Dari Alhuwail, Arfan Ahmed, Padraig Mark Healy, Syed Latifi, Sarah Aziz, Rafat Damseh, Sadam Alabed Alrazak, and Javaid Sheikh. 2023. Large Language Models in Medical Education: Opportunities, Challenges, and Future Directions. *JMIR Med Educ* (2023).

[3] Gati V. Aher, Rosa I. Arriaga, and Adam Tauman Kalai. 2023. Using Large Language Models to Simulate Multiple Humans and Replicate Human Subject Studies. In *Proc. of ICML*.

[4] Michael Ahn, Anthony Brohan, Noah Brown, and et al. 2022. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. *CoRR* (2022).

[5] Elif Akata, Lion Schulz, Julian Coda-Forno, Seong Joon Oh, Matthias Bethge, and Eric Schulz. 2023. Playing repeated games with Large Language Models. *CoRR* (2023).

[6] Badr AlKhamissi, Millicent Li, Asli Celikyilmaz, Mona Diab, and Marjan Ghazvininejad. 2022. A Review on Language Models as Knowledge Bases. *arXiv preprint arXiv: 2204.06031* (2022).

[7] Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, and et al. 2019. Publicly Available Clinical BERT Embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*.

[8] Cem Anil, Yuhuai Wu, Anders Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Ramasesh, Ambrose Slone, Guy Gur-Ari, and et al. 2022. Exploring Length Generalization in Large Language Models. In *Proc. of NeurIPS*.

[9] Jacob Austin, Augustus Odena, Maxwell I. Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, and et al. 2021. Program Synthesis with Large Language Models. *CoRR* (2021).

[10] Pranjal Awasthi and Anupam Gupta. 2023. Improving Length-Generalization in Transformers via Task Hinting. arXiv:2310.00726 [cs.LG]

[11] Baichuan-inc. 2023. *Baichuan-13b.* https://github.com/baichuan-inc/Baichuan-13B

[12] Katie Bainbridge, Candace A. Walkington, Armon Ibrahim, Iris Zhong, Debshila Basu Mallick, Julianna Washington, and Richard G. Baraniuk. 2023. A Case Study using Large Language Models to Generate Metadata for Math Questions. In *AIED 2023, Tokyo, Japan, July 7, 2023*.

[13] Michael Bernhofer, Christian Dallago, Tim Karl, and et al. 2021. PredictProtein - Predicting Protein Structure and Function for 29 Years. *Nucleic Acids Research* (2021).

[14] Lorenzo Bertolini, Julie Weeds, and David Weir. 2022. Testing Large Language Models on Compositionality and Inference with Phrase-Level Adjective-Noun Entailment. In *Proc. of COLING*.

[15] Prajjwal Bhargava and Vincent Ng. 2022. Commonsense Knowledge Reasoning and Generation with Pre-trained Language Models: A Survey. *Proc. of AAAI* (2022).

[16] Satwik Bhattamishra, Arkil Patel, Phil Blunsom, and Varun Kanade. 2023. Understanding In-Context Learning in Transformers and LLMs by Learning to Learn Discrete Functions. *arXiv preprint arXiv:2310.03016* (2023).

[17] Bhavya Bhavya, Jinjun Xiong, and Chengxiang Zhai. 2023. Cam: A large language model-based creative analogy mining framework. In *Proceedings of the ACM Web Conference 2023*.

[18] Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. 2022. Pangu-Weather: A 3D High-Resolution Model for Fast and Accurate Global Weather Forecast. *CoRR* (2022).

[19] Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. 2023. Accurate medium-range global weather forecasting with 3D neural networks. *Nature* (2023).

[20] Andrew Blair-Stanek, Nils Holzenberger, and Benjamin Van Durme. 2023. Can GPT-3 perform statutory reasoning? *arXiv preprint arXiv:2302.06100* (2023).

[21] Michael Bommarito II and Daniel Martin Katz. 2022. GPT takes the bar exam. *arXiv preprint arXiv:2212.14402* (2022).

[22] Andres M Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. 2023. ChemCrow: Augmenting large-language models with chemistry tools. arXiv:2304.05376 [physics.chem-ph]

[23] Boyd Branch, Piotr Mirowski, and Kory W. Mathewson. 2021. Collaborative Storytelling with Human Actors and AI Narrators. In *Proceedings of the Twelfth International Conference on Computational Creativity, México City, México (Virtual), September 14-18, 2021*.

[24] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, et al. 2020. Language models are few-shot learners. *Proc. of NeurIPS* (2020).

[25] Charlotte Caucheteux and Jean-Rémi King. 2022. Brains and algorithms partially converge in natural language processing. *Communications biology* (2022).

[26] Tuhin Chakrabarty, Philippe Laban, Divyansh Agarwal, Smaranda Muresan, and Chien-Sheng Wu. 2023. Art or artifice? large language models and the false promise of creativity. *arXiv preprint arXiv:2309.14556* (2023).

[27] Tuhin Chakrabarty, Vishakh Padmakumar, and He He. 2022. Help me write a Poem - Instruction Tuning as a Vehicle for Collaborative Poetry Writing. In *Proc. of EMNLP*.

[28] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559* (2020).

[29] Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. ChatEval: Towards Better LLM-based Evaluators through Multi-Agent Debate. *CoRR* (2023).

[30] Bo Chen, Xingyi Cheng, Yangli ao Geng, Shengyin Li, Xin Zeng, Bo Wang, Jing Gong, Chiming Liu, Aohan Zeng, Yuxiao Dong, Jie Tang, and Leo T. Song. 2023. xTrimoPGLM: Unified 100B-Scale Pre-trained Transformer for Deciphering the Language of Protein. *bioRxiv* (2023).

[31] Bo Chen, Ziwei Xie, Jiezhong Qiu, Zhaofeng Ye, Jinbo Xu, and Jie Tang. 2023. Improved the heterodimer protein complex prediction with protein language models. *Briefings Bioinform.* (2023).
[32] Jin Chen, Zheng Liu, Xu Huang, Chenwang Wu, Qi Liu, Gangwei Jiang, Yuanhao Pu, Yuxuan Lei, Xiaolong Chen, Xingmei Wang, Defu Lian, and Enhong Chen. 2023. When Large Language Models Meet Personalization: Perspectives of Challenges and Opportunities. arXiv:2307.16376 [cs.IR]
[33] Jiaao Chen, Xiaoman Pan, Dian Yu, Kaiqiang Song, Xiaoyang Wang, et al. 2023. Skills-in-Context Prompting: Unlocking Compositionality in Large Language Models. *arXiv preprint arXiv: 2308.00304* (2023).
[34] Kang Chen, Tao Han, Junchao Gong, Lei Bai, Fenghua Ling, Jing-Jia Luo, Xi Chen, Leiming Ma, Tianning Zhang, Rui Su, Yuanzheng Ci, Bin Li, Xiaokang Yang, and Wanli Ouyang. 2023. FengWu: Pushing the Skillful Global Medium-range Weather Forecast beyond 10 Days Lead. *CoRR* (2023).
[35] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harrison Edwards, et al. 2021. Evaluating Large Language Models Trained on Code. *CoRR* (2021).
[36] Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2022. Program of Thoughts Prompting: Disentangling Computation from Reasoning for Numerical Reasoning Tasks. *CoRR* (2022).
[37] Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chen Qian, Chi-Min Chan, Yujia Qin, Yaxi Lu, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2023. AgentVerse: Facilitating Multi-Agent Collaboration and Exploring Emergent Behaviors in Agents. *CoRR* (2023).
[38] Yirong Chen, Zhenyu Wang, Xiaofen Xing, Huimin Zheng, Zhipei Xu, Kai Fang, Junhong Wang, Sihang Li, Jieling Wu, Qi Liu, and Xiangmin Xu. 2023. BianQue: Balancing the Questioning and Suggestion Ability of Health LLMs with Multi-turn Health Conversations Polished by ChatGPT. *CoRR* (2023).
[39] Zheng Chen. 2023. PALR: Personalization Aware LLMs for Recommendation. *CoRR* (2023).
[40] Zeming Chen, Alejandro Hernández-Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, Alexandre Sallinen, Alireza Sakhaeirad, Vinitra Swamy, Igor Krawczuk, Deniz Bayazit, Axel Marmet, Syrielle Montariol, Mary-Anne Hartley, Martin Jaggi, and Antoine Bosselut. 2023. MEDITRON-70B: Scaling Medical Pretraining for Large Language Models. *CoRR* (2023).
[41] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & Deep Learning for Recommender Systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, DLRS@RecSys 2016, Boston, MA, USA, September 15, 2016.*
[42] Jonathan H Choi, Kristin E Hickman, Amy Monahan, and Daniel Schwarcz. 2023. Chatgpt goes to law school. *Available at SSRN* (2023).
[43] Youngduck Choi, Chill Yi-I Chiu, and David A. Sontag. 2016. Learning Low-Dimensional Representations of Medical Concepts. In *Summit on Clinical Research Informatics, CRI 2016, San Francisco, CA, USA, March 21-24, 2016.*
[44] Ratul Chowdhury, Nazim Bouatta, Surojit Biswas, Charlotte Rochereau, George M. Church, et al. 2021. Single-sequence protein structure prediction using language models from deep learning. *bioRxiv* (2021).
[45] The UniProt Consortium. 2022. UniProt: the Universal Protein Knowledgebase in 2023. *Nucleic Acids Research* (2022).
[46] Jiaxi Cui, Zongjian Li, Yang Yan, Bohua Chen, and Li Yuan. 2023. Chatlaw: Open-source legal large language model with integrated external knowledge bases. *arXiv preprint arXiv:2306.16092* (2023).
[47] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C. H. Hoi. 2023. InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning. *CoRR* (2023).
[48] J Davies, J Klinger, J Mateos-Garcia, and K Stathoulopoulos. 2020. The art in the artificial AI and the creative industries. *Creat Ind Policy Evid Centre* (2020).
[49] Ernest Davis and Gary Marcus. 2015. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Commun. ACM* (2015).
[50] Yinlin Deng, Chunqiu Steven Xia, Chenyuan Yang, Shizhuo Dylan Zhang, Shujing Yang, and Lingming Zhang. 2023. Large Language Models are Edge-Case Fuzzers: Testing Deep Learning Libraries via FuzzGPT. *CoRR* (2023).
[51] Aniket Deroy, Kripabandhu Ghosh, and Saptarshi Ghosh. 2023. How Ready are Pre-trained Abstractive Models and LLMs for Legal Case Judgement Summarization? *arXiv preprint arXiv:2306.01248* (2023).
[52] Jingzhe Ding, Yan Cen, and Xinyuan Wei. 2023. Using Large Language Model to Solve and Explain Physics Word Problems Approaching Human Level. *CoRR* (2023).
[53] Yihong Dong, Xue Jiang, Zhi Jin, and Ge Li. 2023. Self-collaboration Code Generation via ChatGPT. *CoRR* (2023).
[54] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, et al. 2023. PaLM-E: An Embodied Multimodal Language Model. *CoRR* (2023).
[55] Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2023. Improving Factuality and Reasoning in Language Models through Multiagent Debate. *CoRR* (2023).

[56] Shaoxiong Duan and Yining Shi. 2023. From Interpolation to Extrapolation: Complete Length Generalization for Arithmetic Transformers. arXiv:2310.11984 [cs.LG]
[57] Naoki Egami, Musashi Jacobs-Harukawa, Brandon M Stewart, and Hanying Wei. 2023. Using Large Language Model Annotations for Valid Downstream Statistical Inference in Social Science: Design-Based Semi-Supervised Learning. *arXiv preprint arXiv:2306.04746* (2023).
[58] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas B. Fehér, et al. 2020. ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Deep Learning and High Performance Computing. *bioRxiv* (2020).
[59] Zachary Englhardt, Richard Li, Dilini Nissanka, Zhihan Zhang, Girish Narayanswamy, Joseph Breda, Xin Liu, Shwetak N. Patel, and Vikram Iyer. 2023. Exploring and Characterizing Large Language Models For Embedded System Development and Debugging. *CoRR* (2023).
[60] Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. 2022. MineDojo: Building Open-Ended Embodied Agents with Internet-Scale Knowledge. In *NeurIPS*.
[61] Xiaomin Fang, Fan Wang, Lihang Liu, Jingzhou He, Dayong Lin, Yingfei Xiang, Xiaonan Zhang, Hua Wu, Hui Li, and Le Song. 2022. HelixFold-Single: MSA-free Protein Structure Prediction by Using Protein Language Model as an Alternative. *CoRR* (2022).
[62] Weixi Feng, Wanrong Zhu, Tsu-Jui Fu, Varun Jampani, et al. 2023. LayoutGPT: Compositional Visual Planning and Generation with Large Language Models. *CoRR* (2023).
[63] Lorenzo Jaime Yu Flores, Heyuan Huang, Kejian Shi, et al. 2023. Medical Text Simplification: Optimizing for Readability with Unlikelihood Training and Reranked Beam Search Decoding. *CoRR* (2023).
[64] Giorgio Franceschelli and Mirco Musolesi. 2023. On the creativity of large language models. *arXiv preprint arXiv:2304.00008* (2023).
[65] Guanghui Fu, Qing Zhao, Jianqiang Li, Dan Luo, Changwei Song, Wei Zhai, Shuo Liu, Fan Wang, Yan Wang, Lijuan Cheng, Juan Zhang, and Bing Xiang Yang. 2023. Enhancing Psychological Counseling with Large Language Model: A Multifaceted Decision-Support System for Non-Professionals. *CoRR* (2023).
[66] Pablo Gainza, Freyr Sverrisson, F. Monti, Emanuele Rodolà, D. Boscaini, Michael M. Bronstein, and Bruno E. Correia. 2019. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods* (2019).
[67] Chen Gao, Xiaochong Lan, Zhihong Lu, Jinzhu Mao, Jinghua Piao, Huandong Wang, Depeng Jin, and Yong Li. 2023. S^3: Social-network Simulation System with Large Language Model-Empowered Agents. *CoRR* (2023).
[68] Mu Gao, Hongyi Zhou, and Jeffrey Skolnick. 2019. DESTINI: A deep-learning approach to contact-driven protein structure prediction. *Scientific Reports* (2019).
[69] Vedant Gaur and Nikunj Saunshi. 2022. Symbolic Math Reasoning with Language Models. *2022 IEEE MIT Undergraduate Research Technology Conference (URTC)* (2022).
[70] Vedant Gaur and Nikunj Saunshi. 2023. Reasoning in Large Language Models Through Symbolic Math Word Problems. *arXiv preprint arXiv: 2308.01906* (2023).
[71] Mingyang Geng, Shangwen Wang, Dezun Dong, Haotian Wang, Ge Li, Zhi Jin, et al. 2024. Large Language Models are Few-Shot Summarizers: Multi-Intent Comment Generation via In-Context Learning. (2024).
[72] Vladimir Gligorijević, P. Douglas Renfrew, Tomasz Kosciólek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C. Taylor, Ian Fisk, Hera Vlamakis, Ramnik J. Xavier, et al. 2021. Structure-based protein function prediction using graph convolutional networks. *Nature Communications* (2021).
[73] Tomas Goldsack, Zhihao Zhang, Chen Tang, Carolina Scarton, and Chenghua Lin. 2023. Enhancing Biomedical Lay Summarisation with External Knowledge Graphs. *CoRR* (2023).
[74] Lewis D. Griffin, Bennett Kleinberg, Maximilian Mozes, Kimberly T. Mai, Maria Vau, Matthew Caldwell, and Augustine Marvor-Parker. 2023. Susceptibility to Influence of Large Language Models. *CoRR* (2023).
[75] Jian Guan, Xiaoxi Mao, Changjie Fan, Zitao Liu, Wenbiao Ding, and Minlie Huang. 2021. Long Text Generation by Modeling Sentence-Level and Discourse-Level Coherence. In *Proc. of ACL*.
[76] Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Y. Wu, Y. K. Li, Fuli Luo, Yingfei Xiong, and Wenfeng Liang. 2024. DeepSeek-Coder: When the Large Language Model Meets Programming - The Rise of Code Intelligence. *CoRR* (2024).
[77] Yue Guo, Zian Xu, and Yi Yang. 2023. Is ChatGPT a Financial Expert? Evaluating Language Models on Financial Natural Language Processing. In *Proc. of EMNLP Findings*.
[78] Zishan Guo, Renren Jin, Chuang Liu, Yufei Huang, Dan Shi, Supryadi, Linhao Yu, Yan Liu, Jiaxuan Li, Bojian Xiong, and Deyi Xiong. 2023. Evaluating Large Language Models: A Comprehensive Survey. *CoRR* (2023).
[79] Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proc. of ACL*.

[80] Kelvin Guu, Kenton Lee, Z. Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: Retrieval-Augmented Language Model Pre-Training. *International Conference on Machine Learning* (2020).
[81] Thilo Hagendorff, Sarah Fabi, and Michal Kosinski. 2022. Machine intuition: Uncovering human-like intuitive decision-making in GPT-3.5. *CoRR* (2022).
[82] Sil Hamilton. 2023. Blind judgement: Agent-based supreme court modelling with gpt. *arXiv:2301.05327* (2023).
[83] Chi Han, Qifan Wang, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. 2023. LM-Infinite: Simple On-the-Fly Length Generalization for Large Language Models. arXiv:2308.16137 [cs.CL]
[84] Lois Haruna-Cooper and Mohammed Ahmed Rashid. 2023. GPT-4: the future of artificial intelligence in medical school assessments. *Journal of the Royal Society of Medicine* (2023).
[85] Joy He-Yueya, Gabriel Poesia, Rose E. Wang, and Noah D. Goodman. 2023. Solving Math Word Problems by Combining Language Models With Symbolic Solvers. *CoRR* (2023).
[86] Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, and Chenglin Wu. 2023. MetaGPT: Meta Programming for Multi-Agent Collaborative Framework. *CoRR* (2023).
[87] Bozhen Hu, Jun-Xiong Xia, Jiangbin Zheng, Cheng Tan, Yufei Huang, Yongjie Xu, and Stan Z. Li. 2022. Protein Language Models and Structure Prediction: Connection and Progression. *ArXiv* (2022).
[88] Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards Reasoning in Large Language Models: A Survey. In *Proc. of ACL Findings*.
[89] Quzhe Huang, Mingxu Tao, Zhenwei An, Chen Zhang, Cong Jiang, Zhibin Chen, Zirui Wu, and Yansong Feng. 2023. Lawyer LLaMA Technical Report. *arXiv preprint arXiv:2305.15062* (2023).
[90] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, et al. 2022. Inner Monologue: Embodied Reasoning through Planning with Language Models. *CoRR* (2022).
[91] Atin Sakkeer Hussain, Shansong Liu, Chenshuo Sun, and Ying Shan. 2023. M^2UGen: Multi-modal Music Understanding and Generation with the Power of Large Language Models. *CoRR* (2023).
[92] Kwan Yuen Iu and Vanessa Man-Yi Wong. 2023. ChatGPT by OpenAI: The End of Litigation Lawyers? *Available at SSRN* (2023).
[93] Kevin Maik Jablonka, Qianxiang Ai, Alexander Al-Feghali, Shruti Badhwar, Joshua D. Bocarsly, Andres M. Bran, Stefan Bringuier, et al. 2023. 14 Examples of How LLMs Can Transform Materials Science and Chemistry: A Reflection on a Large Language Model Hackathon. *CoRR* (2023).
[94] Samy Jelassi, Stéphane d'Ascoli, Carles Domingo-Enrich, Yuhuai Wu, Yuanzhi Li, and François Charton. 2023. Length Generalization in Arithmetic Transformers. arXiv:2306.15400 [cs.LG]
[95] Zhenlan Ji, Pingchuan Ma, Zongjie Li, and Shuai Wang. 2023. Benchmarking and Explaining Large Language Model-based Code Generation: A Causality-Centric Approach. *CoRR* (2023).
[96] Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Xin Zhao, and Ji-Rong Wen. 2023. StructGPT: A General Framework for Large Language Model to Reason over Structured Data. In *Proc. of EMNLP*.
[97] Shuyang Jiang, Yuhao Wang, and Yu Wang. 2023. SelfEvolve: A Code Evolution Framework via Large Language Models. *CoRR* (2023).
[98] Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, et al. 2021. What Disease Does This Patient Have? A Large-Scale Open Domain Question Answering Dataset from Medical Exams. *Applied Sciences* (2021).
[99] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. 2019. PubMedQA: A Dataset for Biomedical Research Question Answering. In *Proc. of EMNLP*.
[100] Yixiang Jin, Dingzhe Li, Yong A, Jun Shi, Peng Hao, Fuchun Sun, Jianwei Zhang, and Bin Fang. 2024. RobotGPT: Robot Manipulation Learning From ChatGPT. *IEEE Robotics Autom. Lett.* 9, 3 (2024), 2543–2550. doi:10.1109/LRA.2024.3357432
[101] Nicolas Jonason, Luca Casini, Carl Thomé, et al. 2023. Retrieval Augmented Generation of Symbolic Music with LLMs. *CoRR* (2023).
[102] Sebastian Joseph, Kathryn Kazanas, Keziah Reina, Vishnesh J. Ramanathan, Wei Xu, Byron C. Wallace, and Junyi Jessy Li. 2023. Multilingual Simplification of Medical Texts. *CoRR* (2023).
[103] Harshit Joshi, José Pablo Cambronero Sánchez, Sumit Gulwani, Vu Le, Gust Verbruggen, and Ivan Radicek. 2023. Repair Is Nearly Generation: Multilingual Program Repair with LLMs. In *Proc. of AAAI*.
[104] John M. Jumper, Richard Evans, Alexander Pritzel, et al. 2021. Highly accurate protein structure prediction with AlphaFold. *Nature* (2021).
[105] Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and Applications of Large Language Models. *CoRR* (2023).
[106] Wang-Cheng Kang, Jianmo Ni, Nikhil Mehta, Maheswaran Sathiamoorthy, Lichan Hong, Ed H. Chi, and Derek Zhiyuan Cheng. 2023. Do LLMs Understand User Preferences? Evaluating LLMs On User Rating Prediction. *CoRR* (2023).
[107] Ehud Karpas, Omri Abend, Yonatan Belinkov, Barak Lenz, Opher Lieber, Nir Ratner, et al. 2022. MRKL Systems: A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning. *CoRR* (2022).
[108] Daniel Martin Katz, Michael James Bommarito, Shang Gao, et al. 2023. Gpt-4 passes the bar exam. *Available at SSRN 4389233* (2023).

[109] Tushar Khot, H. Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2022. Decomposed Prompting: A Modular Approach for Solving Complex Tasks. *International Conference on Learning Representations* (2022).
[110] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large Language Models are Zero-Shot Reasoners. In *Proc. of NeurIPS*.
[111] Michal Kosinski. 2023. Theory of Mind May Have Spontaneously Emerged in Large Language Models. *CoRR* (2023).
[112] Tiffany H. Kung, Morgan Cheatham, Arielle Medenilla, Czarina Sillos, Lorie De Leon, Camille ElepaĆ±o, Maria Madriaga, Rimel Aggabao, Giezel Diaz-Candido, James Maningo, and Victor Tseng. 2023. Performance of ChatGPT on USMLE: Potential for AI-assisted medical education using large language models. *PLOS Digital Health* (2023).
[113] Tin Lai, Yukun Shi, Zicong Du, Jiajie Wu, Ken Fu, Yichao Dou, and Ziqi Wang. 2023. Psy-LLM: Scaling up Global Mental Health Psychological Services with AI-based Large Language Models. *CoRR* (2023).
[114] Brenden M. Lake and Marco Baroni. 2023. Human-like systematic generalization through a meta-learning neural network. *Nat.* (2023).
[115] Kausik Lakkaraju, Sai Krishna Revanth Vuruma, Vishal Pallagani, Bharath Muppasani, et al. 2023. Can LLMs be Good Financial Advisors?: An Initial Study in Personal Decision Making for Optimized Outcomes. *CoRR* (2023).
[116] Geon Lee, Wenchao Yu, Kijung Shin, Wei Cheng, and Haifeng Chen. 2025. TimeCAP: Learning to Contextualize, Augment, and Predict Time Series Events with Large Language Model Agents. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, Toby Walsh, Julie Shah, and Zico Kolter (Eds.). AAAI Press, 18082–18090. doi:10.1609/AAAI.V39I17.33989
[117] Cheng Li, Ziang Leng, Chenxi Yan, Junyi Shen, Hao Wang, Weishi MI, Yaying Fei, et al. 2023. ChatHaruhi: Reviving Anime Character in Reality via Large Language Model. *arXiv preprint arXiv:2308.09597* (2023).
[118] Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, et al. 2023. LLaVA-Med: Training a Large Language-and-Vision Assistant for Biomedicine in One Day. *CoRR* (2023).
[119] Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, et al. 2023. Camel: Communicative agents for" mind" exploration of large scale language model society. *arXiv preprint arXiv:2303.17760* (2023).
[120] Hang Li. 2022. Language models: past, present, and future. *Commun. ACM* (2022).
[121] Haonan Li, Yu Hao, and et al. 2023. The Hitchhiker's Guide to Program Analysis: A Journey with Large Language Models. *CoRR* (2023).
[122] Jia Li, Ge Li, Yongmin Li, and Zhi Jin. 2023. Structured Chain-of-Thought Prompting for Code Generation. *arXiv preprint arXiv:2305.06599* (2023).
[123] Jiawei Li, Yizhe Yang, Yu Bai, and et al. 2024. Fundamental Capabilities of Large Language Models and their Applications in Domain Scenarios: A Survey. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Bangkok, Thailand, 11116–11141. doi:10.18653/v1/2024.acl-long.599
[124] Lei Li, Yongfeng Zhang, and Li Chen. 2023. Personalized Prompt Learning for Explainable Recommendation. *ACM Trans. Inf. Syst.* (2023).
[125] Liunian Harold Li, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. 2023. Symbolic Chain-of-Thought Distillation: Small Models Can Also "Think" Step-by-Step. *Annual Meeting of the Association for Computational Linguistics* (2023).
[126] Qi Li. 2023. Harnessing the Power of Pre-trained Vision-Language Models for Efficient Medical Report Generation. In *Proc. of CIKM*.
[127] Siyu Li, Jin Yang, and Kui Zhao. 2023. Are you in a Masquerade? Exploring the Behavior and Impact of Large Language Model Driven Social Bots in Online Social Networks. *CoRR* (2023).
[128] Xianzhi Li, Samuel Chan, Xiaodan Zhu, Yulong Pei, Zhiqiang Ma, et al. 2023. Are ChatGPT and GPT-4 General-Purpose Solvers for Financial Text Analytics? A Study on Several Typical Tasks. In *Proc. of EMNLP*.
[129] Xin-Ye Li, Jiang-Tian Xue, Zheng Xie, and Ming Li. 2023. Think Outside the Code: Brainstorming Boosts Large Language Models in Code Generation. *CoRR* (2023).
[130] Xiang Lorraine Li, Adhiguna Kuncoro, Jordan Hoffmann, Cyprien de Masson d'Autume, et al. 2022. A Systematic Investigation of Commonsense Knowledge in Large Language Models. In *Proc. of EMNLP*.
[131] Yunxiang Li, Zihan Li, Kai Zhang, Ruilong Dan, and You Zhang. 2023. ChatDoctor: A Medical Chat Model Fine-tuned on LLaMA Model using Medical Domain Knowledge. *CoRR* (2023).
[132] Yuan Li, Yixuan Zhang, and Lichao Sun. 2023. MetaAgents: Simulating Interactions of Human Behaviors for LLM-based Task-oriented Coordination via Collaborative Generative Agents. *CoRR* (2023).
[133] Long Lian, Boyi Li, Adam Yala, and Trevor Darrell. 2023. LLM-grounded Diffusion: Enhancing Prompt Understanding of Text-to-Image Diffusion Models with Large Language Models. *CoRR* (2023).
[134] Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. 2023. Encouraging Divergent Thinking in Large Language Models through Multi-Agent Debate. *CoRR* (2023).
[135] Zhiding Liang, Jinglei Cheng, Rui Yang, Hang Ren, et al. 2023. Unleashing the Potential of LLMs for Quantum Computing: A Study in Quantum Architecture Design. arXiv:2307.08191 [quant-ph]
[136] Valentin Liévin, Christoffer Egeberg Hother, and Ole Winther. 2022. Can large language models reason about medical questions? *CoRR* (2022).

[137] Valentin Liévin, Andreas Geert Motzfeldt, Ida Riis Jensen, and Ole Winther. 2023. Variational Open-Domain Question Answering. In *Proc. of ICML*.
[138] Jonathan Light, Wei Cheng, Yue Wu, Masafumi Oyamada, Mengdi Wang, Santiago Paternain, and Haifeng Chen. 2025. DISC: Dynamic Decomposition Improves LLM Inference Scaling. *CoRR* abs/2502.16706 (2025). doi:10.48550/ARXIV.2502.16706 arXiv:2502.16706
[139] Jonathan Light, Yue Wu, Yiyou Sun, Wenchao Yu, Yanchi Liu, Xujiang Zhao, Ziniu Hu, Haifeng Chen, and Wei Cheng. [n. d.]. SFS: Smarter Code Space Search improves LLM Inference Scaling. In *The Thirteenth International Conference on Learning Representations*.
[140] Jiaju Lin, Haoran Zhao, Aochi Zhang, Yiting Wu, Huqiuyue Ping, and Qin Chen. 2023. AgentSims: An Open-Source Sandbox for Large Language Model Evaluation. *CoRR* (2023).
[141] Zeming Lin, Halil Akin, Roshan Rao, Brian L. Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. 2022. Evolutionary-scale prediction of atomic level protein structure with a language model. *bioRxiv* (2022).
[142] Chao Liu, Xuanlin Bao, Hongyu Zhang, Neng Zhang, Haibo Hu, Xiaohong Zhang, and Meng Yan. 2023. Improving ChatGPT Prompt for Code Generation. *CoRR* (2023).
[143] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual Instruction Tuning. *CoRR* (2023).
[144] June M. Liu, Donghao Li, He Cao, Tianhe Ren, Zeyi Liao, and Jiamin Wu. 2023. ChatCounselor: A Large Language Models for Mental Health Support. *CoRR* (2023).
[145] Xiao-Yang Liu, Guoxuan Wang, and Daochen Zha. 2023. FinGPT: Democratizing Internet-scale Data for Financial Large Language Models. *CoRR* (2023).
[146] Yiren Liu, Si Chen, Haocong Cheng, Mengxia Yu, Xiao Ran, Andrew Mo, et al. 2023. How AI Processing Delays Foster Creativity: Exploring Research Question Co-Creation with an LLM-based Agent. *CoRR* (2023).
[147] Zhuang Liu, Degen Huang, Kaiyu Huang, Zhuang Li, and Jun Zhao. 2020. FinBERT: A Pre-trained Financial Language Representation Model for Financial Text Mining. In *Proc. of IJCAI*.
[148] Anton Lozhkov, Raymond Li, Loubna Ben Allal, Federico Cassano, Joel Lamy-Poirier, Nouamane Tazi, Ao Tang, et al. 2024. StarCoder 2 and The Stack v2: The Next Generation. *CoRR* (2024).
[149] Junru Lu, Jiazheng Li, Byron C. Wallace, Yulan He, and Gabriele Pergola. 2023. NapSS: Paragraph-level Medical Text Simplification via Narrative Prompting and Sentence-matching Summarization. In *Proc. of ACL Findings*.
[150] Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. 2023. Chameleon: Plug-and-Play Compositional Reasoning with Large Language Models. *arXiv preprint arXiv: 2304.09842* (2023).
[151] Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. BioGPT: generative pre-trained transformer for biomedical text generation and mining. *Briefings Bioinform.* (2022).
[152] Yun Luo, Xiaotian Lin, Zhen Yang, Fandong Meng, et al. 2023. Mitigating Catastrophic Forgetting in Task-Incremental Continual Learning with Adaptive Classification Criterion. *arXiv preprint arXiv: 2305.12270* (2023).
[153] Yun Luo, Zhen Yang, Xuefeng Bai, Fandong Meng, Jie Zhou, and Yue Zhang. 2023. Investigating Forgetting in Pre-Trained Representations Through Continual Learning. *arXiv preprint arXiv: 2305.05968* (2023).
[154] Yizhen Luo, Jiahuan Zhang, Siqi Fan, Kai Yang, Yushuai Wu, Mu Qiao, and Zaiqing Nie. 2023. BioMedGPT: Open Multimodal Generative Pre-trained Transformer for BioMedicine. *CoRR* (2023).
[155] Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. 2023. WizardCoder: Empowering Code Large Language Models with Evol-Instruct. *CoRR* (2023).
[156] Baek M, DiMaio F, Anishchenko I, Dauparas J, Ovchinnikov S, Lee Gr, Wang J, Cong Q, Kinch Ln, Schaeffer Rd, et al. 2021. Accurate prediction of protein structures and interactions using a 3-track neural network. *Science (New York, N.Y.)* (2021).
[157] Teli Ma, Rong Li, and Junwei Liang. 2023. An Examination of the Compositionality of Large Generative Vision-Language Models. *arXiv preprint arXiv: 2308.10509* (2023).
[158] Aman Madaan, Shuyan Zhou, Uri Alon, Yiming Yang, and Graham Neubig. 2022. Language Models of Code are Few-Shot Commonsense Learners. *Conference on Empirical Methods in Natural Language Processing* (2022).
[159] Mounica Maddela, Megan Ung, Jing Xu, Andrea Madotto, Heather Foran, and Y-Lan Boureau. 2023. Training Models to Generate, Recognize, and Reframe Unhelpful Thoughts. In *Proc. of ACL*.
[160] Vaibhav Mavi, Abulhair Saparov, and Chen Zhao. 2023. Retrieval-Augmented Chain-of-Thought in Semi-structured Domains. *CoRR* (2023).
[161] Muhammad Miftahul Amri and Urfa Khairatun Hisan. 2023. Incorporating AI Tools into Medical Education: Harnessing the Benefits of ChatGPT and Dall-E. *Journal of Novel Engineering Science and Technology* (2023).
[162] Jaina Mistry, Sara Chuguransky, Lowri Williams, Matloob Qureshi, Gustavo A. Salazar, Erik L. L. Sonnhammer, Silvio C. E. Tosatto, Lisanna Paladin, Shriya Raj, Lorna J. Richardson, Robert D. Finn, and Alex Bateman. 2021. Pfam: The protein families database in 2021. *Nucleic Acids Res.* (2021).
[163] Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, et al. 2023. Few-shot Fine-tuning vs. In-context Learning: A Fair Comparison and Evaluation. *Annual Meeting of the Association for Computational Linguistics* (2023).

[164] Yida Mu, Ben P. Wu, William Thorne, Ambrose Robinson, Nikolaos Aletras, Carolina Scarton, Kalina Bontcheva, and Xingyi Song. 2023. Navigating Prompt Complexity for Zero-Shot Classification: A Study of Large Language Models in Computational Social Science. *CoRR* (2023).
[165] Muhammad U. Nasir, Sam Earle, Julian Togelius, Steven James, and Christopher Cleghorn. 2023. LLMatic: Neural Architecture Search via Large Language Models and Quality Diversity Optimization. arXiv:2306.01102 [cs.NE]
[166] Michel Nass, Emil Alégroth, and Robert Feldt. 2023. Improving web element localization by using a large language model. *CoRR* (2023).
[167] Nature Education. 2010. Protein Function.
[168] John J Nay. 2022. Law informs code: A legal informatics approach to aligning artificial intelligence with humans. *Nw. J. Tech. & Intell. Prop.* (2022).
[169] Ha-Thanh Nguyen. 2023. A Brief Report on LawGPT 1.0: A Virtual Legal Assistant Based on GPT-3. *arXiv preprint arXiv:2302.05729* (2023).
[170] Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K. Gupta, and Aditya Grover. 2023. ClimaX: A foundation model for weather and climate. In *Proc. of ICML*.
[171] Ansong Ni, Srini Iyer, Dragomir Radev, Veselin Stoyanov, Wen-Tau Yih, Sida I. Wang, and Xi Victoria Lin. 2023. LEVER: Learning to Verify Language-to-Code Generation with Execution. In *Proc. of ICML*.
[172] Kole Norberg, Husni Almoubayyed, Stephen E. Fancsali, Logan De Ley, Kyle Weldon, April Murphy, and Steven Ritter. 2023. Rewriting Math Word Problems with Large Language Models. In *Proceedings of the Workshop on Empowering Education with LLMs - the Next-Gen Interface and Content Generation 2023 co-located with 24th International Conference on Artificial Intelligence in Education (AIED 2023), Tokyo, Japan, July 7, 2023*.
[173] Kolby Nottingham, Prithviraj Ammanabrolu, Alane Suhr, et al. 2023. Do Embodied Agents Dream of Pixelated Sheep: Embodied Decision Making using Language Guided World Modelling. In *Proc. of ICML*.
[174] OpenAI. 2023. GPT-4 Technical Report. *CoRR* (2023).
[175] Carlos Outeiral and Charlotte M. Deane. 2022. Codon language embeddings provide strong signals for protein engineering. *bioRxiv* (2022).
[176] Siru Ouyang, Zhuosheng Zhang, Bing Yan, Xuan Liu, Jiawei Han, and Lianhui Qin. 2023. Structured Chemistry Reasoning with Large Language Models. *CoRR* (2023).
[177] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. MedMCQA: A Large-scale Multi-Subject Multi-Choice Dataset for Medical domain Question Answering. In *Conference on Health, Inference, and Learning, CHIL 2022, 7-8 April 2022, Virtual Event*.
[178] Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. 2023. Logic-LM: Empowering Large Language Models with Symbolic Solvers for Faithful Logical Reasoning. *arXiv preprint arXiv: 2305.12295* (2023).
[179] Bhargavi Paranjape, Scott M. Lundberg, Sameer Singh, Hannaneh Hajishirzi, Luke Zettlemoyer, and Marco Túlio Ribeiro. 2023. ART: Automatic multi-step reasoning and tool-use for large language models. *CoRR* (2023).
[180] Joon Sung Park, Joseph C. O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative Agents: Interactive Simulacra of Human Behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, UIST 2023, San Francisco, CA, USA, 29 October 2023- 1 November 2023*.
[181] Joon Sung Park, Lindsay Popowski, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2022. Social Simulacra: Creating Populated Prototypes for Social Computing Systems. In *The 35th Annual ACM Symposium on User Interface Software and Technology, UIST 2022, Bend, OR, USA, 29 October 2022 - 2 November 2022*.
[182] Shishir G. Patil, Tianjun Zhang, Xin Wang, et al. 2023. Gorilla: Large Language Model Connected with Massive APIs. *CoRR* (2023).
[183] Nelson Perdigão, Julian Heinrich, Christian Stolte, Kenneth Sabir, Michael Buckley, Bruce Tabor, Bethany Signal, Brian S. Gloss, Christopher J. Hammang, Burkhard Rost, Andrea Schafferhans, and Seán I. O'Donoghue. 2015. Unexpected features of the dark proteome. *Proceedings of the National Academy of Sciences* (2015).
[184] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, A. Bakhtin, Yuxiang Wu, Alexander H. Miller, and S. Riedel. 2019. Language Models as Knowledge Bases? *Conference on Empirical Methods in Natural Language Processing* (2019).
[185] Tammy Pettinato Oltz. 2023. ChatGPT, Professor of Law. *Professor of Law (February 4, 2023)* (2023).
[186] Laura Plein, Wendkûuni C. Ouédraogo, Jacques Klein, and Tegawendé F. Bissyandé. 2023. Automatic Generation of Test Cases based on Bug Reports: a Feasibility Study with Large Language Models. *CoRR* (2023).
[187] Archiki Prasad, Alexander Koller, Mareike Hartmann, Peter Clark, Ashish Sabharwal, et al. 2023. ADaPT: As-Needed Decomposition and Planning with Language Models. *arXiv preprint arXiv: 2311.05772* (2023).
[188] Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. 2023. Reasoning with Language Model Prompting: A Survey. In *Proc. of ACL*.
[189] Wei Qin, Zetong Chen, Lei Wang, Yunshi Lan, Weijieying Ren, and Richang Hong. 2023. Read, Diagnose and Chat: Towards Explainable and Interactive LLMs-Augmented Depression Detection in Social Media. *CoRR* (2023).

[190] Predrag Radivojac, Zoran Obradovic, David Keith Smith, Guang Zhu, Slobodan Vucetic, Celeste J. Brown, J. David Lawson, and A. Keith Dunker. 2004. Protein flexibility and intrinsic disorder. *Protein Science* (2004).
[191] Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian D. Reid, and Niko Sünderhauf. 2023. SayPlan: Grounding Large Language Models using 3D Scene Graphs for Scalable Task Planning. *CoRR* (2023).
[192] Mucheng Ren, Heyan Huang, Yuxiang Zhou, Qianwen Cao, Yuan Bu, and Yang Gao. 2022. TCM-SD: A Benchmark for Probing Syndrome Differentiation via Natural Language Processing. In *Proc. of CCL*.
[193] Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, , et al. 2023. Code Llama: Open Foundation Models for Code. *CoRR* (2023).
[194] Andre Niyongabo Rubungo, Craig Arnold, Barry P. Rand, and Adji Bousso Dieng. 2023. LLM-Prop: Predicting Physical And Electronic Properties Of Crystalline Solids From Their Text Descriptions. *CoRR* (2023).
[195] Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2023. LaMP: When Large Language Models Meet Personalization. *arXiv preprint arXiv:2304.11406* (2023).
[196] Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2022. Multitask Prompted Training Enables Zero-Shot Task Generalization. In *Proc. of ICLR*.
[197] Jaromir Savelka, Kevin D Ashley, Morgan A Gray, Hannes Westermann, and Huihui Xu. 2023. Explaining Legal Concepts with Augmented Large Language Models (GPT-4). *arXiv preprint arXiv:2306.09525* (2023).
[198] Michael Schauperl and Rajiah Aldrin Denny. 2022. AI-Based Protein Structure Prediction in Drug Discovery: Impacts and Challenges. *Journal of chemical information and modeling* (2022).
[199] Maria Schelling, Thomas A. Hopf, and Burkhard Rost. 2018. Evolutionary couplings and sequence variation effect predict protein binding sites. *Proteins: Structure* (2018).
[200] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language Models Can Teach Themselves to Use Tools. *CoRR* (2023).
[201] David B. Searls. 2018. computational biology.
[202] Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D. Manning. 2019. Do Massively Pretrained Language Models Make Better Storytellers?. In *Proc. of CoNLL*.
[203] Andrew W. Senior, Richard Evans, John M. Jumper, James Kirkpatrick, L. Sifre, Tim Green, Chongli Qin, Augustin Zídek, et al. 2020. Improved protein structure prediction using potentials from deep learning. *Nature* (2020).
[204] Chantal Shaib, Millicent L. Li, Sebastian Joseph, Iain James Marshall, Junyi Jessy Li, and Byron C. Wallace. 2023. Summarizing, Simplifying, and Synthesizing Medical Evidence using GPT-3 (with Varying Success). In *Proc. of ACL*.
[205] Murray Shanahan, Kyle McDonell, and Laria Reynolds. 2023. Role play with large language models. *Nature* (2023).
[206] NAN SHAO, Zefan Cai, Hanwei xu, Chonghua Liao, Yanan Zheng, and Zhilin Yang. 2023. Compositional Task Representations for Large Language Models. In *Proc. of ICLR*.
[207] Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. 2021. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624* (2021).
[208] Xiaoming Shi, Zeming Liu, Chuan Wang, Haitao Leng, Kui Xue, Xiaofan Zhang, and Shaoting Zhang. 2023. MidMed: Towards Mixed-Type Dialogues for Medical Consultation. In *Proc. of ACL*.
[209] Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, et al. 2022. Large Language Models Encode Clinical Knowledge. *CoRR* (2022).
[210] Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, et al. 2023. Towards Expert-Level Medical Question Answering with Large Language Models. *CoRR* (2023).
[211] Ritwik Sinha, Zhao Song, and Tianyi Zhou. 2023. A mathematical abstraction for balancing the trade-off between creativity and reality in large language models. *arXiv preprint arXiv:2306.02295* (2023).
[212] Alexey Skrynnik, Zoya Volovikova, Marc-Alexandre Côté, Anton Voronov, Artem Zholus, Negar Arabzadeh, Shrestha Mohanty, Milagro Teruel, Ahmed Awadallah, Aleksandr Panov, Mikhail Burtsev, and Julia Kiseleva. 2022. Learning to Solve Voxel Building Embodied Tasks from Pixels and Natural Language Instructions. *CoRR* (2022).
[213] Yifan Song, Weimin Xiong, Dawei Zhu, Cheng Li, Ke Wang, Ye Tian, and Sujian Li. 2023. RestGPT: Connecting Large Language Models with Real-World Applications via RESTful APIs. *CoRR* (2023).
[214] Evangelia Spiliopoulou, Artidoro Pagnoni, Yonatan Bisk, and Eduard H. Hovy. 2022. EvEntS ReaLM: Event Reasoning of Entity States via Language Models. In *Proc. of EMNLP*.
[215] Martin Steinegger, Milot Mirdita, and Johannes Söding. 2018. Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold. *Nature Methods* (2018).
[216] Douglas Summers-Stay, Clare R Voss, and Stephanie M Lukin. 2023. Brainstorm, then Select: a Generative Language Model Improves Its Creativity Score. In *Proc. of AAAI*.
[217] Lichao Sun, Yue Huang, Haoran Wang, Siyuan Wu, et al. 2024. TrustLLM: Trustworthiness in Large Language Models. *CoRR* (2024).

[218] Freyr Sverrisson, Jean Feydy, Bruno E. Correia, and Michael M. Bronstein. 2020. Fast end-to-end learning on protein surfaces. *bioRxiv* (2020).
[219] Ben Swanson, Kory Mathewson, Ben Pietrzak, Sherol Chen, and Monica Dinalescu. 2021. Story centaur: Large language model few shot learning as a creative writing tool. In *Proc. of EACL*.
[220] Bowen Tan, Zichao Yang, Maruan Al-Shedivat, Eric P. Xing, and Zhiting Hu. 2021. Progressive Generation of Long Text with Pretrained Language Models. In *Proc. of NAACL*.
[221] Chen Tang, Shun Wang, Tomas Goldsack, and Chenghua Lin. 2023. Improving Biomedical Abstractive Summarisation with Knowledge Aggregation from Citation Papers. *CoRR* (2023).
[222] Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. 2023. GraphGPT: Graph Instruction Tuning for Large Language Models. arXiv:2310.13023 [cs.CL]
[223] Xwin-LM Team. 2023. *Xwin-LM.* https://github.com/Xwin-LM/Xwin-LM
[224] Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. 2023. Large language models in medicine. *Nature medicine* (2023).
[225] Yuanhe Tian, Ruyi Gan, Yan Song, Jiaxing Zhang, and Yongdong Zhang. 2023. ChiMed-GPT: A Chinese Medical Large Language Model with Full Training Regime and Better Alignment to Human Preferences. *CoRR* (2023).
[226] Kushal Tirumala, Aram H. Markosyan, et al. 2022. Memorization Without Overfitting: Analyzing the Training Dynamics of Large Language Models. *Neural Information Processing Systems* (2022).
[227] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971 [cs.CL]
[228] Dietrich Trautmann, Alina Petrova, and Frank Schilder. 2022. Legal prompt engineering for multilingual legal judgement prediction. *arXiv preprint arXiv:2212.02199* (2022).
[229] Wiebe Van der Hoek and Michael Wooldridge. 2008. Multi-agent systems. *Foundations of Artificial Intelligence* (2008).
[230] Dave Van Veen, Cara Van Uden, Louis Blankemeier, Jean-Benoit Delbrouck, Asad Aali, Christian Bluethgen, Anuj Pareek, Malgorzata Polacin, William Collins, et al. 2023. Clinical Text Summarization: Adapting Large Language Models Can Outperform Human Experts. *CoRR* (2023).
[231] Sai Vemprala, Rogerio Bonatti, Arthur Bucker, et al. 2023. ChatGPT for Robotics: Design Principles and Model Abilities. *CoRR* (2023).
[232] Boshi Wang, Xiang Deng, and Huan Sun. 2022. Iteratively Prompt Pre-trained Language Models for Chain of Thought. *Conference on Empirical Methods in Natural Language Processing* (2022).
[233] Cunxiang Wang, Pai Liu, and Yue Zhang. 2021. Can Generative Pre-trained Language Models Serve As Knowledge Bases for Closed-book QA?. In *Proc. of ACL*.
[234] Duolin Wang, Usman L. Abbas, Qing Shao, Jin Chen, and Dong Xu. 2023. S-PLM: Structure-aware Protein Language Model via Contrastive Learning between Sequence and Structure. *bioRxiv* (2023).
[235] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An Open-Ended Embodied Agent with Large Language Models. *CoRR* (2023).
[236] Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. 2023. Can Language Models Solve Graph Problems in Natural Language? arXiv:2305.10037 [cs.CL]
[237] Junjie Wang, Ping Yang, Ruyi Gan, Yuxiang Zhang, Jiaxing Zhang, and Tetsuya Sakai. 2023. Zero-Shot Learners for Natural Language Understanding via a Unified Multiple-Choice Perspective. *IEEE Access* (2023).
[238] Jianguo Wang, Xiaomeng Yi, Rentong Guo, Hai Jin, Peng Xu, Shengjun Li, Xiangyu Wang, and et al. 2021. Milvus: A Purpose-Built Vector Data Management System. In *International Conference on Management of Data, Virtual Event, China, June 20-25, 2021.*
[239] Lei Wang, Songheng Zhang, Yun Wang, Ee-Peng Lim, and Yong Wang. 2023. LLM4Vis: Explainable Visualization Recommendation using ChatGPT. arXiv:2310.07652 [cs.HC]
[240] Siyuan Wang, Bo Peng, Yichao Liu, and Qi Peng. 2023. Fine-grained Medical Vision-Language Representation Learning for Radiology Report Generation. In *Proc. of EMNLP*.
[241] Sheng Wang, Zihao Zhao, Xi Ouyang, Qian Wang, and Dinggang Shen. 2023. ChatCAD: Interactive Computer-Aided Diagnosis on Medical Image using Large Language Models. *CoRR* (2023).
[242] Wenkai Wang, Zhenling Peng, and Jianyi Yang. 2022. Single-sequence protein structure prediction using supervised transformer protein language models. *bioRxiv* (2022).
[243] Xidong Wang, Guiming Hardy Chen, Dingjie Song, Zhiyi Zhang, Zhihong Chen, Qingying Xiao, Feng Jiang, Jianquan Li, Xiang Wan, Benyou Wang, and Haizhou Li. 2023. CMB: A Comprehensive Medical Benchmark in Chinese. *CoRR* abs/2308.08833 (2023). arXiv:2308.08833
[244] Zihao Wang, Shaofei Cai, Anji Liu, Xiaojian Ma, and Yitao Liang. 2023. Describe, Explain, Plan and Select: Interactive Planning with Large Language Models Enables Open-World Multi-Task Agents. *ArXiv* (2023).

[245] Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. 2023. Unleashing Cognitive Synergy in Large Language Models: A Task-Solving Agent through Multi-Persona Self-Collaboration. *CoRR* (2023).
[246] Zekun Moore Wang, Zhongyuan Peng, Haoran Que, Jiaheng Liu, Wangchunshu Zhou, Yuhan Wu, Hongcheng Guo, Ruitong Gan, Zehao Ni, Man Zhang, et al. 2023. Rolellm: Benchmarking, eliciting, and enhancing role-playing abilities of large language models. *arXiv:2310.00746* (2023).
[247] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned Language Models are Zero-Shot Learners. In *Proc. of ICLR*.
[248] Jimmy Wei, Kurt Shuster, Arthur Szlam, Jason Weston, Jack Urbanek, and Mojtaba Komeili. 2023. Multi-party chat: Conversational agents in group settings with humans and models. *arXiv preprint arXiv:2304.13835* (2023).
[249] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of Thought Prompting Elicits Reasoning in Large Language Models. In *Proc. of NeurIPS*.
[250] Yuxiang Wei, Zhe Wang, Jiawei Liu, Yifeng Ding, and Lingming Zhang. 2023. Magicoder: Source Code Is All You Need. *CoRR* (2023).
[251] Yuxiang Wei, Chunqiu Steven Xia, and Lingming Zhang. 2023. Copiloting the Copilots: Fusing Large Language Models with Completion Engines for Automated Program Repair. *CoRR* (2023).
[252] Chaoyi Wu, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2023. Towards Generalist Foundation Model for Radiology. *CoRR* (2023).
[253] Rui Min Wu, Fan Ding, Rui Wang, Rui Shen, Xiwen Zhang, Shitong Luo, Chenpeng Su, Zuofan Wu, Qi Xie, et al. 2022. High-resolution de novo structure prediction from primary sequence. *bioRxiv* (2022).
[254] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David S. Rosenberg, and Gideon Mann. 2023. BloombergGPT: A Large Language Model for Finance. *CoRR* (2023).
[255] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, et al. 2023. The Rise and Potential of Large Language Model Based Agents: A Survey. *CoRR* (2023).
[256] Chunqiu Steven Xia and Lingming Zhang. 2023. Conversational Automated Program Repair. *CoRR* (2023).
[257] Chunqiu Steven Xia and Lingming Zhang. 2023. Keep the Conversation Going: Fixing 162 out of 337 bugs for $0.42 each using ChatGPT. *CoRR* (2023).
[258] Qianqian Xie, Weiguang Han, Yanzhao Lai, Min Peng, and Jimin Huang. 2023. The Wall Street Neophyte: A Zero-Shot Analysis of ChatGPT Over MultiModal Stock Movement Prediction Challenges. *CoRR* (2023).
[259] Qianqian Xie, Weiguang Han, Xiao Zhang, Yanzhao Lai, Min Peng, Alejandro Lopez-Lira, and Jimin Huang. 2023. PIXIU: A Large Language Model, Instruction Data and Evaluation Benchmark for Finance. *CoRR* (2023).
[260] Zhuokui Xie, Yinghao Chen, Chen Zhi, et al. 2023. ChatUniTest: a ChatGPT-based automated unit test generation tool. *CoRR* (2023).
[261] Zhenchang Xing, Qing Huang, Yu Cheng, Liming Zhu, Qinghua Lu, and Xiwei Xu. 2023. Prompt Sapper: LLM-Empowered Software Engineering Infrastructure for AI-Native Services. *CoRR* (2023).
[262] Minghao Xu, Xinyu Yuan, Santiago Miret, and Jian Tang. 2023. ProtST: Multi-Modality Learning of Protein Sequences and Biomedical Texts. In *Proc. of ICML*.
[263] Junbing Yan, Chengyu Wang, Taolin Zhang, Xiaofeng He, Jun Huang, and Wei Zhang. 2023. From Complex to Simple: Unraveling the Cognitive Tree for Reasoning with Small Language Models. In *Proc. of EMNLP Findings*.
[264] Kevin Yang, Dan Klein, Nanyun Peng, and Yuandong Tian. 2023. DOC: Improving Long Story Coherence With Detailed Outline Control. In *ACL*.
[265] Kevin Yang, Yuandong Tian, Nanyun Peng, and Dan Klein. 2022. Re3: Generating Longer Stories With Recursive Reprompting and Revision. In *Proc. of EMNLP*.
[266] Yizhe Yang, Huashan Sun, Jiawei Li, Runheng Liu, Yinghao Li, Yuhang Liu, Heyan Huang, and Yang Gao. 2023. MindLLM: Pre-training Lightweight Large Language Model from Scratch, Evaluations and Domain Applications. *CoRR* (2023).
[267] Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. WebShop: Towards Scalable Real-World Web Interaction with Grounded Language Agents. In *NeurIPS*.
[268] Seonghyeon Ye, Hyeonbin Hwang, Sohee Yang, Hyeongu Yun, Yireun Kim, and Minjoon Seo. 2023. In-context instruction learning. *arXiv preprint arXiv:2302.14691* (2023).
[269] Caiyang Yu, Xianggen Liu, Wentao Feng, Chenwei Tang, and Jiancheng Lv. 2023. GPT-NAS: Evolutionary Neural Architecture Search with the Generative Pre-Trained Model. arXiv:2305.05351 [cs.CV]
[270] Dingyao Yu, Kaitao Song, Peiling Lu, Tianyu He, Xu Tan, Wei Ye, Shikun Zhang, and Jiang Bian. 2023. MusicAgent: An AI Agent for Music Understanding and Generation with Large Language Models. *CoRR* (2023).
[271] Fangyi Yu, Lee Quartey, and Frank Schilder. 2022. Legal Prompting: Teaching a Language Model to Think Like a Lawyer. arXiv:2212.01326 [cs.CL]
[272] Jifan Yu, Xiaozhi Wang, Shangqing Tu, Shulin Cao, Daniel Zhang-Li, et al. 2023. KoLA: Carefully Benchmarking World Knowledge of Large Language Models. arXiv:2306.09296 [cs.CL]

[273] Yangyang Yu, Haohang Li, Zhi Chen, Yuechen Jiang, Yang Li, Denghui Zhang, Rong Liu, Jordan W. Suchow, and Khaldoun Khashanah. 2023. FinMe: A Performance-Enhanced Large Language Model Trading Agent with Layered Memory and Character Design. *CoRR* (2023).
[274] Ann Yuan, Andy Coenen, Emily Reif, and Daphne Ippolito. 2022. Wordcraft: Story Writing With Large Language Models. In *IUI 2022: 27th International Conference on Intelligent User Interfaces, Helsinki, Finland, March 22 - 25, 2022*.
[275] Haoqi Yuan, Chi Zhang, Hongcheng Wang, Feiyang Xie, Penglin Cai, Hao Dong, and Zongqing Lu. 2023. Plan4MC: Skill Reinforcement Learning and Planning for Open-World Minecraft Tasks. *CoRR* (2023).
[276] Shengbin Yue, Wei Chen, Siyuan Wang, Bingxuan Li, Chenchen Shen, Shujun Liu, Yuxuan Zhou, Yao Xiao, Song Yun, Wei Lin, et al. 2023. Disc-lawllm: Fine-tuning large language models for intelligent legal services. *arXiv preprint arXiv:2309.11325* (2023).
[277] Daoguang Zan, Bei Chen, Fengji Zhang, Dianjie Lu, Bingchao Wu, Bei Guan, Yongji Wang, and Jian-Guang Lou. 2023. Large Language Models Meet NL2Code: A Survey. In *Proc. of ACL*.
[278] Hongbo Zhang, Junying Chen, Feng Jiang, Fei Yu, et al. 2023. HuatuoGPT, towards Taming Language Model to Be a Doctor. *CoRR* (2023).
[279] Kechi Zhang, Zhuo Li, Jia Li, Ge Li, and Zhi Jin. 2023. Self-Edit: Fault-Aware Code Editor for Code Generation. In *Proc. of ACL*.
[280] Ningyu Zhang, Zhen Bi, Xiaozhuan Liang, Siyuan Cheng, Haosen Hong, Shumin Deng, Jiazhang Lian, Qiang Zhang, and Huajun Chen. 2022. OntoProtein: Protein Pretraining With Gene Ontology Embedding. *ArXiv* (2022).
[281] Xuanyu Zhang and Qing Yang. 2023. XuanYuan 2.0: A Large Chinese Financial Chat Model with Hundreds of Billions Parameters. In *CIKM*.
[282] Yiming Zhang, Shi Feng, and Chenhao Tan. 2022. Active Example Selection for In-Context Learning. *Conference on Empirical Methods in Natural Language Processing* (2022).
[283] Yuwei Zhang, Zhi Jin, et al. 2023. STEAM: Simulating the InTeractive BEhavior of ProgrAMmers for Automatic Bug Fixing. *CoRR* (2023).
[284] Yuchen Zhang, Mingsheng Long, Kaiyuan Chen, Lanxiang Xing, Ronghua Jin, Michael I. Jordan, and Jianmin Wang. 2023. Skilful nowcasting of extreme precipitation with NowcastNet. *Nat.* (2023).
[285] Ziyin Zhang, Chaoyu Chen, Bingchang Liu, Cong Liao, Zi Gong, Hang Yu, Jianguo Li, and Rui Wang. 2023. A Survey on Language Models for Code. *CoRR* (2023).
[286] Zuobai Zhang, Minghao Xu, Arian R. Jamasb, Vijil Chenthamarakshan, Aurélie C. Lozano, Payel Das, and Jian Tang. 2022. Protein Representation Learning by Geometric Structure Pretraining. *ArXiv* (2022).
[287] Bowen Zhao, Changkai Ji, Yuejie Zhang, Wen He, Yingwen Wang, Qing Wang, Rui Feng, and Xiaobo Zhang. 2023. Large Language Models are Complex Table Parsers. In *Proc. of EMNLP*.
[288] Guosheng Zhao, Yan Yan, and Zijian Zhao. 2023. Normal-Abnormal Decoupling Memory for Medical Report Generation. In *EMNLP Findings*.
[289] Junjie Zhao, Xiang Chen, Guang Yang, and Yiheng Shen. 2023. Automatic Smart Contract Comment Generation via Large Language Models and In-Context Learning. *CoRR* (2023).
[290] Yilun Zhao, Yitao Long, Hongjun Liu, Linyong Nan, Lyuhao Chen, Ryo Kamoi, Yixin Liu, Xiangru Tang, Rui Zhang, and Arman Cohan. 2023. DocMath-Eval: Evaluating Numerical Reasoning Capabilities of LLMs in Understanding Long Documents with Tabular Data. *CoRR* (2023).
[291] Zirui Zhao, Wee Sun Lee, and David Hsu. 2023. Large Language Models as Commonsense Knowledge for Large-Scale Task Planning. *arXiv preprint arXiv: 2305.14078* (2023).
[292] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate Before Use: Improving Few-shot Performance of Language Models. In *Proc. of ICML*.
[293] Mingkai Zheng, Xiu Su, Shan You, Fei Wang, Chen Qian, Chang Xu, and Samuel Albanie. 2023. Can GPT-4 Perform Neural Architecture Search? arXiv:2304.10970 [cs.LG]
[294] Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. 2021. Adapting Language Models for Zero-shot Learning by Meta-tuning on Dataset and Prompt Collections. In *Proc. of EMNLP Findings*.
[295] Hattie Zhou, Arwen Bradley, Etai Littwin, Noam Razin, Omid Saremi, Josh Susskind, Samy Bengio, and Preetum Nakkiran. 2023. What Algorithms can Transformers Learn? A Study in Length Generalization. arXiv:2310.16028 [cs.LG]
[296] Caleb Ziems, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. 2023. Can Large Language Models Transform Computational Social Science? *CoRR* (2023).

Received 10 June 2024; revised 10 June 2024; accepted 8 May 2025