ESP-JETA
ESP Journal of Engineering & Technology Advancements
ISSN: 2583-2646 / Volume 5 Issue 2 May 2025 / Page No: 141-154
Paper Id: JETA-V5I2P115 / Doi: 10.56472/25832646/JETA-V5I2P115

Original Article

Goal Decomposition & Self-Planning in Agentic AI with LLM Backends

Vishnu Lakkamraju
Independent Researcher, USA.

Received Date: 10 February 2025 Revised Date: 14 March 2025 Accepted Date: 09 May 2025

Abstract: *Large Language Models (LLMs), such GPT-4 and its successors, which have let machines understand, generate, and respond to natural language with hitherto unheard-of fluency and contextual awareness, have greatly driven the fast development of Artificial Intelligence (AI). These models are now being included into agentic artificial intelligence systems, which are autonomous agents meant to run with minimum human intervention, therefore changing the way machines understand and act upon complicated directions. Goal decomposition and self-planning capabilities are a major development in this area since they enable agentic artificial intelligence to divide abstract goals into doable tasks and carry out plans with adaptability and flexibility in dynamic surroundings.*

The principles supporting goal decomposition and self-planning in LLM-powered agentic artificial intelligence are thoroughly explored in this work. We discuss several strategic techniques including recursive task modelling, interleaved planning-execution cycles, and decomposition-first approaches. We also look at innovative models as ADaPT (As-needed Decomposition and Planning), AdaPlanner, and GoalAct that let agents decide on their own when and how to break out chores and adjust their plans in response to environmental changes or real-time data. These technologies not only improve operational effectiveness but also enable agents to show cognitive abilities including introspection, reason, and learning from past mistakes.

The study also examines architectural trends such Tree of Thought (ToT), Chain of Thought (CoT), and self-reflective models like ReAct and Reflexion, which taken together form the backbone of contemporary self-planning agents. These patterns give agents organised thinking processes so they may assess several approaches to accomplish tasks and pick the most practical one. We also investigate the translation of goals into machine-readable plans and the integration of LLMs with external symbolic planners, therefore augmenting the value of agentic artificial intelligence systems in challenging fields such legal reasoning, education, and virtual environments.

This work shows how goal decomposition and self-planning provide scalable, interpretable, and strong AI agent architectures by means of a synthesis of present research and empirical performance benchmarks. By doing this, it prepares the way for next developments that will challenge artificial agents' autonomy and intelligence limit.

Keywords: *Agentic Artificial Intelligence, Large Language Models (LLMs), Goal Decomposition, Self-Planning, Autonomous Agents, Adaptive Planning, Task Decomposition, Artificial Reasoning, Chain of Thought (CoT), Tree of Thought (ToT), Reflexive AI, ReAct, Reflexion, Symbolic Planning, Cognitive AI.*

I. INTRODUCTIONS

Driven by the capabilities of big language models (LLMs), such OpenAI's GPT-4, artificial intelligence (AI) has reached a turning point. These models have transformed natural language processing by allowing machines to construct human language with amazing fluency and contextual richness, therefore enabling human language understanding and generation. But a more significant change in artificial intelligence development comes from the evolution from language processing tools into completely autonomous agents able of setting goals, planning actions, and dynamically changing to feedback. Agentic artificial intelligence (agentic AI) is a new class of computers ready to redefine intelligence and autonomy in several fields.

Designed to pursue goals in open-ended, uncertain contexts, agentic artificial intelligence systems run with little human control. Agentic systems, unlike conventional artificial intelligence, which depends on predetermined rules or static programming, constantly assess their environment, create plans, and change them in reaction to fresh data. Two fundamental skills underlie this new paradigm: self-planning and goal deconstruction.

Agent goal decomposition is the ability of an agent to arrange high-level or abstract goals into a series of smaller, doable activities. This reflects human approach to problem-solving, in which main obstacles are addressed methodically. Conversely, self-planning is the autonomous development, adaption, and execution of action sequences aiming at goal accomplishment. These processes are closely related: good goal decomposition is necessary for successful self-planning and vice versa. Together they offer the fundamental mechanics for intelligent, self-directed behaviour.

Including LLMs into agentic artificial intelligence has created fresh avenues for using these features. LLMs can be used to mimic thinking, decision-making, and adaptive planning even if they shine in comprehending and producing text. Reflective models like ReAct and Reflexion as well as architectural ideas like Chain of Thought (CoT) and Tree of Thought (ToT) offer structural frameworks for grouping difficult thinking and planning actions. These techniques let agents clearly express their intermediate ideas, evaluate several courses of action, and grow from past mistakes—qualities that help artificial intelligence approach cognitive intelligence.

Starting with the decomposition-first strategy, which entails upfront job breakdown prior to execution, this work investigates a wide spectrum of planning and decomposition strategies in agentic artificial intelligence systems. While lacking flexibility, this conventional method provides clarity and predictability in steady surroundings. On the other hand, interleaved planning and execution creates a dynamic loop in which agents coordinate and act in unison, real-time strategy modification included. These techniques form the basis for more complex models and reflect the two extremes of planning philosophy—static against adaptive.

Advanced systems as AdaPlanner and ADaPT (As-needed Decomposition and Planning) add subtle means of managing complexity and uncertainty. ADaPT helps agents to determine whether breakdown is required, therefore preventing needless cognitive overhead. By contrast, AdaPlanner uses feedback loops to continually improve plans during implementation. Both technologies help to create an expanding ecosystem of flexible, modular agent designs inspired by human strategic thinking.

GoalAct is a remarkable illustration of scalable and hierarchical planning since it blends worldwide planning with hierarchical execution. GoalAct manages complexity with a combination of abstraction and detail by organising chores into high-level skills and mapping those to sub-actions. Such systems show that agentic artificial intelligence can operate deliberately and aggressively as well as reactively.

Furthermore investigated in this work is how LLM-driven agents might interact with symbolic planning systems employing formal languages as PDDL (Planning Domain Definition Language). These linkages bring the flexible, language-native domain of modern LLMs the rigour and structure of conventional artificial intelligence planning. This hybrid strategy guarantees action validity, improves logical consistency, and increases the spectrum of activities agentic systems can manage.

We assess their performance across benchmark domains—including legal reasoning (Legal Agent Bench), simulated environments (ALFWorld, TextCraft), and general planning systems like AgentGen—to validate these techniques. Especially in multi-step, goal-driven activities, these benchmarks show the benefits of organised planning tools over pure end-to- end language models.

This work tries to give a thorough summary of how self-planning and goal deconstruction are changing the capacities of artificial intelligence agents. We build the foundation for more scalable, interpretable, and intelligent autonomous systems by synthesising architectural strategies, reviewing empirical benchmarks, and analysing future directions.

II. AGENTIC ARTIFICIAL INTELLIGENCE'S CONTRASTING PLANNING PARADIGMS SPAN STRUCTURED DECOMPOSITION TO ADAPTIVE EXECUTION.

Two main planning paradigms have evolved as fundamental tactics in the field of agentic artificial intelligence, where autonomous computers must independently comprehend and execute complex instructions: the Interleaved Planning and Execution approach and the Decomposition-First approach. Each stands for a different perspective on how intelligent agents should evaluate objectives, organise their activities, and change with the times for dynamic conditions during job execution.

The basis of the Decomposition-First strategy is the belief that consistent execution results from organised foresight. Under this approach, an agent starts its work by first carefully considering the main objective and then dissecting it into more doable parts before acting in the real world. This approach is particularly successful in fields where task boundaries are well defined and the environment stays constant throughout operation since it reflects classical project management methods applied in human processes. The process begins with goal analysis, in which the agent determines the main purpose and any implied or explicit needs it calls for. Once the scope is known, the agent moves on task decomposition, separating the goal into distinct, simpler to conceptualise and operationalise sub-tasks.

Taking consideration for dependencies and prerequisites among them, the agent sequences these sub-tasks in a logical and coherent sequence following breakdown. The next important phase is resource allocation; here, the agent distributes time, computational power, and outside tools in line with the requirements of every sub-task. Following this, the agent enters the phase of execution planning, in which it creates a comprehensive road map to implement every element in the predefined sequence. At last, the agent carries out the scheme, closely observing its development and making little changes only when absolutely needed.

There are multiple benefits from this methodical approach. It offers clarity by outlining a thorough road map ahead, therefore lowering uncertainty throughout execution. Predefined plans reduce the need for on-demand decision-making, therefore enabling efficient execution. Furthermore, more predictable outputs let stakeholders better estimate timetables and outcomes. This approach does, however, also show several clear limits. Its stiffness makes it less appropriate for dynamic or volatile surroundings, where unanticipated occurrences could make the initial plan useless. When speed is crucial, much ahead planning increases overhead and could cause delay in action. Furthermore, the limited adaptability of this method implies that significant changes in job conditions could call for a whole re-planning procedure, which can be time-consuming and resource-intensive.

Unlike this fixed, top-down approach, the Interleaved Planning and Execution technique presents a more flexible and dynamic substitute. Agents using this approach start implementation with a partial or temporary plan instead of pledging to follow a whole plan from the start. The agent actively watches the surroundings, gathers input, and adjusts its strategy in response to fresh data as it performs first movements. The agent's constant engagement with the surroundings enables it to remain flexible and nimble, hence the method is especially useful in uncertain or fast changing surroundings.

The cycle starts with first planning grounded on the agent's present knowledge of the goal and situational circumstances. Usually include just the first few sub-tasks, this first blueprint leaves space for the plan to expand and change. The agent then starts working on these little projects, watching results and instantly evaluating performance. This phase of monitoring is crucial since it supplies the information required for good plan development. Whether environmental changes, newly found limits, or new opportunities—the agent adjusts its strategy by adding, deleting, or changing activities as fresh knowledge arises. Until the initial objective is completely met, this iterative cycle of activity, observation, and improvement keeps on.

This method offers significant advantages. Its adaptability lets the agent change with the times and with changing goals. This makes it perfect for settings marked by uncertainty, including those experienced by autonomous cars negotiating erratic traffic, robots working alongside humans, or virtual assistants handling always changing consumer needs. The agent saves effort on contingency plans that might never arise by not overinvesting in a first strategy. Moreover, the iterative character of the method promotes ongoing education, which helps the agent to change its plan with every cycle and enhance over time decision-making.

Still, these benefits are costly. The agent has to manage both planning and execution concurrently, so the interleaved technique brings more complexity. To manage monitoring, data interpretation, and instantaneous plan modifications, this real-time processing calls for significant computer resources. Moreover, the dynamic character of planning can lead to discrepancies, especially if the agent regularly veers off course without keeping coherence between tasks.

These two methods taken together show the trade-offs involved in creating intelligent planning systems for agentic artificial intelligence. Perfect in clarity and predictability, the Decomposition-First approach is appropriate for organised work with little environmental variance. Whereas the Interleaved Planning and Execution approach is best suited for adaptability and flexibility, it thrives in dynamic environments where stationary plans soon become obsolete. The type of the work, the stability of the surroundings, and the degree of tolerance for uncertainty will mostly determine which of them to choose. Practically, hybrid techniques usually integrate aspects of both to maximise the strengths of each, therefore providing the foundation for more sophisticated planning structures covered in later sections of this work.

III. DYNAMIC ARCHITECTURES FOR SCALABLE AGENTIC AI PLANNING WITH ADAPTATION

The limits of conventional static planning approaches become more clear as agentic artificial intelligence systems develop to handle more challenging and dynamic problems. Scholars have responded with fresh ideas including scalability, modularity, and adaptability into the design process. Among the most prominent of these are GoalAct, AdaPlanner, and ADaPT (as-needed decomposition and planning). While using the capability of huge language models to enable autonomous agents to reason, reflect, and modify strategy as situations change, each of these architectures offers a different way for managing the trade-offs between upfront planning and real-time responsiveness.

A. ADaPT: Planning and As-Needed Decomposition

a) Recursive and Reactive Planning Strategies

ADaPT marks a significant break from the conventional wisdom on full-task breakdown prior to execution. ADaPT is a dynamic approach that only breaks out tasks as needed instead than mandating agents to divide a goal into sub-tasks in a single, up-front planning stage. Starting with a broad knowledge of a task, trying first steps, and only breaking it down more when problems develop, this method reflects the way people often work: This adaptability helps agents to avoid needless preparation effort for activities that can be completed immediately while yet allowing them to recursively break down more challenging situations.

b) *Underlying Architecture and Implementation Model*

Two essential components—the planner and the executor—form the basis of the ADaPT architecture. Evaluating the intricacy of every chore the planner comes across and determining whether it calls for decomposition falls to them. Should the planner decide the work is beyond the agent's direct capacity, it breaks it down iteratively into smaller jobs. The executor is then handed these chores to try to complete. This division of duties offers the agent modularity and clarity in their mental process. Recursively repeating this approach helps the agent finally arrive at a set of executable actions bringing the main objective to pass.

c) *Performance Against Comparables*

ADaPT has shown promise in a variety of settings. Over conventional planning baselines, ADaPT attained performance gains of up to 28.3% in the ALFWorld simulation, an interactive domestic setting employing natural language. Other sophisticated benchmarks such as WebShop and TextCraft, where agents must accomplish multi-stage tasks in dynamic textual environments, also showed similar outcomes. These results emphasise how well ADaPT manages different degrees of task complexity without running the computational cost of complete upfront breakdown.

d) *An Advance in Flexible Planning*

ADaPT is unique in that it can strike a mix between dynamic thinking and execution readiness. Particularly in settings where agents must cope with uncertainty and layered decision structures, its recursive decomposition model lets for more scalability and adaptability. Agents applying ADaPT lower planning overhead and increase task completion efficiency by only decomposing when necessary.

B. AdaPlanner: Closed-Loop Planning Driven by Comments

a) *Linking Dynamic Environments with Static Strategies*

While AdaPlanner concentrates on improving plans during implementation based on real-time feedback, ADaPT adjusts planning depth on demand. It is intended as a closed-loop planning system in which every agent's contact with the surroundings shapes the following stage of its strategy. AdaPlanner works particularly well in situations where conditions change often or where unanticipated events routinely throw off set plans.

b) *Organising via Iteration and Improvement*

AdaPlanner starts with the creation of a first plan, maybe broad or partial in nature. The agent gathers environmental feedback as it carries out the planned procedures to ascertain the success of its actions. Two separate uses of this input help to improve the continuing strategy. First, in-plan improvement refers to little, local tweaks meant to accommodate minor modifications or inefficiencies include reordering actions or fixing slight deviations. Second, out-of- plan improvement is more structural and calls for the agent to drastically change or even abandon its present strategy when important chances or challenges present themselves.

A prompt structure intended especially for code-like thinking supports this refining process. AdaPlanner guarantees that the agent's decision-making stays rational, traceable, and resistant to hallucinations—a typical problem in LLM-based planning systems—by using a code-style approach in LLM prompting.

c) *Flexible Use in Complicated Contextues*

Closed-loop design of AdaPlanner has shown to be quite useful in several fields. In navigation-based systems, it lets agents course-correct when goals change or routes are blocked. In dialogue systems, it lets conversational agents change their approaches instantly in reaction to user comments. In multi-stage problem-solving activities, the agent can dynamically re-prioritize sub-goals and steer clear of duplicate or contradicting behaviour. AdaPlanner is a strong and flexible approach for agents to plan across several use cases without turning into a rigid or reactive tool.

d) *Including Comments as a Fundamental Planning Tool*

AdaPlanner emphasises as a first-class component in intelligent agent design the need of environmental feedback. The framework promotes constant reflection and modification instead of seeing plans as fixed obligations. By doing this, it enables agents to keep strong even in non-deterministic, fast-changing environments and to preserve the coherence and direction needed to complete difficult tasks.

C. Goal Act: Combining Hierarchical Execution with Global Strategy

a) A complete approach to long-horizon projects

GoalAct presents an architectural concept fit for multi-step, large-scale projects. It combines the strategic clarity of global planning with the tactical accuracy of hierarchical execution. GoalAct shines in keeping a high-level view of the task structure while allowing flexible adaptation inside each layer of execution when ADaPT breaks based on need and AdaPlanner iteratively refines plans.

b) Dual-Layer Design for Handling Complex Goals

GoalAct operates out of two coordinated components: hierarchical skill execution and global planning. Constant maintenance of a macro-level road map of the high-level skills or competencies needed to reach the ultimate goal marks the global planning module. These abstracted actions—such as "gather evidence," "draft summary," or "search case law"—which dictate the agent's approach help to define her strategy. Every one of these high-level abilities is then broken down into a series of more particular, practical actions. The hierarchical execution system manages them so that the application of any ability is context-aware, effective, and freely editable.

By emphasising the appropriate level of abstraction at any one moment, this two-tiered approach helps GoalAct to control complexity. The executor offers fine-grained control inside each segment; the global planner guarantees coherence and direction across the work.

c) Benchmarked Excellence in Domain Specificity

GoalAct has shown especially good ability in specialist, knowledge-intensive fields like law. GoalAct performed state-of- the-art on the Legal Agent Benchmark, scoring a 12.22% improvement in task success rate over alternative LLM-based agent architectures. Legal work sometimes calls for the smart application of outside instruments, following procedural guidelines, and the interpretation of domain-specific language in line with The layered approach of GoalAct makes it ideally fit for such situations; it helps agents to pursue long-horizon goals while responding appropriately to interim feedback.

d) Giving Agents Strategic Vision and Tactical Agility

GoalAct is particularly effective since it may mix reactivity with foresight. While the hierarchical execution structure guarantees that every step is based on pragmatic reasoning, the worldwide plan gives the agent a clear perspective of its trajectory, therefore helping to avoid aimless or reactive behaviour. This coordination helps agents to manage not only actions but also strategies, contingency planning, and tool interactions in a scalable manner, so operating at a human-like level of planning complexity.

e) Synthesis: Towards More Intelligent Architectural Design for Planning

ADaPT, AdaPlanner, and GoalAct taken together show a move towards increasingly intelligent, context-aware agentic artificial intelligence systems. Every design presents a different approach to control the natural complexity of autonomous behaviour. ADaPT provides flexibility by let one choose when and how to break down. AdaPlanner presents a feedback-oriented loop to maintain environmental responsiveness of plans. GoalAct combines scalable, hierarchical abstract strategy with concrete implementation.

These models draw attention to the variety of strategies in development to handle the same fundamental problem: how can agents be equipped to not only grasp and plan but also to change, evaluate, and grow? Such intelligent planning frameworks will be the backbone of really agentic artificial intelligence as we head towards ever autonomous systems.

IV. DELIBERATIVE REASONING IN LLMS: CHAIN OF THOUGHT AND TREE OF IDEAS AS COGNITIVE SCAFFOLDS

A primary priority in research and development has moved towards improving the reasoning capacity of large language models (LLMs) as they keep advancing. Although LLMs like as GPT-4 can produce fluent, contextually relevant writing, their capacity to participate in profound, methodical problem-solving calls more work. Reacting to this demand, prompting methods including Tree of Thoughts (ToT) and Chain of Thought (CoT) have become transforming tools. These methods reorganise how models approach problems, reflecting human-like cognitive processes by externalising reasoning steps and evaluating alternate lines of thinking, hence improving output accuracy. CoT and ToT taken together have been quite helpful in allowing LLMs to address challenging, multi-step reasoning tasks across many fields including mathematics, logic, strategy, and creative decision-making.

A. Structure of Thought: Organising Sequential Reasoning

a) A Turning Inside Prompt Design

Language models are guided to express their thinking in a step-by-step, rather than straight-forward way via chain of thought prompts, which helps them to avoid directly reaching an answer. This method adds a reflective layer to the processing pipeline of the model, therefore motivating it to "think aloud" prior to producing a result. The approach is basic but effective: it converts opaque inference into a transparent, traceable process by asking the model to divide a task into intermediate phases. This is quite similar to human thought processes, in which people may solve difficulties by jotting down computations, organising arguments, or enumerating advantages and drawbacks before coming to decisions.

b) *Mechanism and Functional Flow-Based Approach*

The Chain of Thought process starts when a model gets a stimulus pushing intermediate thinking. Rather than asking, for instance, "What is the answer to this maths problem?""Explain your steps and then give the answer," a CoT-style prompt may ask. This little change in terminology tells the model to break down the problem, usually resulting in a series of logical stages leading towards the ultimate solution. Every action turns into a cognitive framework supporting the next, producing more precisely explainable results.

Both zero-shot and few-shot situations need for this prompting structure. Under a few-shot environment, models are given exemplars that show how to methodically approach a subject. The wording of the prompt itself is designed in the zero-shot environment to inspire the same sort of collapse. Both times, the effect is that internal decision-making of the model becomes externalised, enabling users and developers to know how a certain outcome was produced.

c) *Improvement in Accuracy and Interpretability*

One of the main advantages of Chain of Thought prompting is better accuracy in activities requiring numerous levels of reasoning. CoT has been demonstrated, for example, to greatly lower hallucinations and boost accuracy in arithmetic and logical deduction-requiring mathematical problems. This is so because, under pressure to specifically think through each stage, the model is less likely to skip over phases or form assumptions.

CoT also improves interpretability, a developing issue in artificial intelligence ethics and safety. Models' mental process helps one to spot mistakes, prejudices, or poor logic. In fields like medical diagnostics, legal analysis, or financial forecasting—where opaque responses could cause mistrust or misguided decisions—this is extremely crucial.

d) *Uses in Several Fields*

The Chain of Thought method has shown success in several spheres. In mathematics and logic puzzles, it helps the model negotiate layered rules and dependencies. By methodically unfolding presumptions, it helps to clarify ambiguity in common sense thinking activities. CoT enables artificial intelligence tutors to teach ideas in a methodical manner, therefore replicating the way in which human teachers assist students across challenges even in educational environments or design.

B. Tree of Thought: Increasing the Reasoning Horizon

a) *Multipath Expansion of Chain of Thought*

Building on the ideas set by Chain of Thought, the Tree of Thoughts (ToT) framework presents a more sophisticated and exploratory model of thinking. CoT moves linearly, one idea after another; ToT allows branching at every turning point. By treating problem-solving as a search over a tree-like structure of possible reasoning paths, this model lets the agent explore several hypotheses or action sequences concurrently before deciding on a solution. This produces a kind of intentional problem-solving that reflects strategic human thought, especially in jobs involving uncertainty, trade-offs, or innovation.

b) *Thought Tree Structure and Execution*

Reasoning starts in the Tree of Thoughts structure not with a single straight path but rather with the development of several first ideas or sub-steps. Every one of these turns into a node on a logical tree. From there, the model investigates possible continuations for every path, creating fresh sub-nodes and assessing them depending on task-specific criteria including logical coherence, probability of success, or novelty. This tree can be built to a limited depth; once enough branches have been investigated, the model assesses the paths as whole and chooses the most likely answer.

Depending on the demands of the situation, ToT can be accomplished using search techniques including depth-limited search or breadth-first search (BFS). Crucially, this framework lets scoring systems or evaluators—learned or human-defined—include priorities for some branches over others. ToT thus becomes not only a motivating technique but also a whole cognitive planning framework stacked on the creative powers of an LLM.

c) *Discussion, Research, and Innovative Solving of Creative Problems*

Particularly in activities when several possible answers exist and the best course of action is not immediately evident, Tree of Thoughts offers several benefits over Chain of Thought. ToT lets models replicate deliberation—choosing not only based on what seems instantly right but also on what emerges as ideal after weighing alternatives—by allowing the investigation of many reasoning routes. This greatly lowers the possibility of early dedication to erroneous or less-than-best choices.

ToT also stimulates imagination. Encouragement of models to spread out and investigate often results in fresh combinations of ideas, tactics, or narrative threads. In fields including game playing, strategic decision-making, interactive storytelling, and design projects where exploration and uniqueness are prized alongside correctness, this makes ToT highly valuable.

d) *Application Versatility and Strategic Depth*

Tree of Thoughts' adaptability has let it flourish in several fields. Strategic games like chess or Go help with forward planning and counterfactual analysis. In challenges involving computational innovation, it allows organised ideas and variance. Key component of advanced decision-making, ToT enables artificial intelligence agents in multi-agent simulations or negotiations to assess possible outcomes based on others' reactions. These programs expose how ToT transforms language models into agents able of structured, deliberate cognition rather than only reactive tools.

e) *Approaching a Unified Cognitive Framework for LLMs*

Tree of Thought and Chain of Thought both mark important developments in enhancing the capacity of big language models to solve problems. They give models the structure they need to reason in an interpretable, logical, more human-aligned manner. For jobs needing unambiguous, deterministic thinking, CoT brings sequential organisation and clarity. In turn, ToT provides width and adaptability so that agents may review several approaches and narrow their decisions depending on comparative analysis.

These approaches are not incompatible. Actually, they can be utilised in concert; CoT forms the linear components in ToT tree branches. The more general conclusion is that motivating is developing from a basic input-output system into a complex interface for cognition—one that enables LLMs to reason, reflect, and improve with hitherto unheard-of capacity.

Agentic artificial intelligence systems will need to be able to not only react but also think, rethink, and explain as they grow increasingly autonomous and accountable for high-stakes chores. Early yet potent first steps towards realising this goal are Chain of Thought and Tree of Thought.

f) *Structured synergy: including external planners into agentic artificial intelligence systems*

The ability of intelligent systems to independently organise, sequence, and perform difficult tasks becomes ever more important as agentic artificial intelligence develops. Although large language models (LLMs) offer great improvement in natural language interpretation and contextual reasoning, their capacity for long-term, multi-step planning is still somewhat young. Integration of outside classical designers into agentic systems has become a transforming architectural option to get over this restriction. AI agents can reach a new degree of dependability, consistency, and strategic depth by combining the rigid, logic-based rigour of symbolic planners with the flexibility and language fluency of LLMs.

g) *Planning's Central Function in Autonomous Agents*

Agentic artificial intelligence systems are built essentially to run without direct human control. They are supposed to follow directions, create strategies, set reasonable goals, and complete chores across a variety of sometimes erratic surroundings. Although LLMs such as GPT-4 are remarkable in understanding goals communicated in common language and even in creating possible action sequences, they lack a natural framework for testing logical consistency or managing long-term relationships between decisions. Here is where symbolic planners come in—introducing into the design of modern autonomous agents decades of study in formal logic, decision theory, and algorithmic planning.

The complimentary strengths of LLMs and classical planners create their synergy. While conventional planners give structure and accuracy in reasoning, LLMs can grasp human intent and manage fuzzy, uncertain data. They taken together create a strong hybrid system combining formalism with generalising ability.

h) *PDDL's Significance in Planning Integration*

Most symbolic planners build their work from the Planning Domain Definition Language (PDDL). Designed to specify the framework of planning issues and the possible options to an agent, PDDL is a standard, machine-readable language. It describes activities with particular preconditions and effects, therefore enabling the planner to methodically find appropriate paths of action to convert a starting state into a desired target state.

Usually, a planning difficulty in PDDL consists in:

- Actions are defined as separate operations an agent is able to execute.
- Preconditions, which have to be satisfied before an action may be carried out.
- Effects are the transformations of the planet following the activity.
- Objectives define the intended result.

Using traditional artificial intelligence techniques as STRIPS, GraphPlan, and heuristic search planners, PDDL's formalism helps to enable effective search-based Planning. Once an issue is written in this way, the planner can create high-fidelity action sequences that agents can run methodically, therefore guaranteeing consistency and lowering mistakes.

i) Agentic AI Comparative Performance With and Without External Planners

Examining a systematic comparison across important performance parameters helps one to fully comprehend the advantages of including external planners into agentic artificial intelligence systems. The following table shows systems supplemented with symbolic planners against those depending just on LLM-based reasoning:

Feature	Agentic AI Without Planner	Agentic AI With External Planner
Planning Logic	Implicit, heuristic-driven	Explicit, formal logic (PDDL-based)
Plan Accuracy	Moderate	High
Scalability to Complex Tasks	Limited	High
Adaptability	High (via LLM generalization)	High (via structured feedback)
Real-Time Responsiveness	Fast but less reliable	Slower, more structured and precise
Error Rate	Higher (due to overlooked constraints)	Lower (validated via logic)
Multi-Step Task Handling	Weak support	Strong support
Natural Language Integration	Native capability	Requires translation
Algorithm Maturity	Emerging	Established and well-studied
Use Case Examples	Chatbots, Q&A, summarization	Robotics, logistics, workflow planning

Particularly in situations when consistency, complexity, and long-term execution integrity are vital, this table shows the great strengths external planners offer.

j) Planner Integration's Strategic advantages

There are three main benefits of including outside planners into agentic artificial intelligence systems. Above importantly, it enables agents to make use of well-known algorithms evolved over many years. These algorithms guarantee that plans are logically consistent, full, and minimally error-prone—qualities that are challenging in simply neural systems.

Second, outside planners increase the effectiveness of design. Symbolic planners can rapidly produce optimal or near-optimal plans without the trial-and-error tendencies of LLMs in fields including robotics, manufacturing, and logistics, where processes must be followed in exact sequences with dependencies.

Improved logical consistency is also another main advantage. In multi-part projects, traditional LLMs sometimes dream or skip important phases. Symbolic planners, on the other hand, confirm that the preconditions of every operation are met before moving forward, therefore preserving a great degree of internal coherence. Applications like autonomous navigation or medical job execution that depend on safety becoming absolutely dependent on this dependability.

Moreover, outside planners help agents to better manage difficult, interconnected activities requiring consideration of resources, time restrictions, and contingencies. These are situations in which LLMs by themselves are not suited for methodically handling.

k) Integration's Technical and Practical Difficulties

Integration of symbolic planners with LLM-driven agents presents unique difficulties even with their benefits. Among them, the most well-known is the translation bottleneck—that is, the method of turning goals in natural language into ordered PDDL representations. This work sometimes calls for semantic parsing and domain-specific mappings that can be error-prone and computationally taxing.

Another issue is scalability. The number of actions and interactions in PDDL files can greatly expand as jobs get more complicated and large, which would increase planning time and memory usage. Research on optimising these files and cutting pointless action pathways is still under progress.

Real-time responsiveness also offers a practical limitation. Although classical designers provide exact and logically solid ideas, they might not always be able to deliver these ideas within the time limits placed by dynamic surroundings. Under such circumstances, a trade-off between speed and accuracy has to be carefully controlled.

l) Future Prospective Research Directions

Dealing with planner integration calls for a multifarious strategy. Semantic parsing models that can automatically convert natural language inputs into organised PDDL goals show great promise. To increase translation fidelity, these models mix rule-based checks with neural approaches.

Rising neuro-symbolic architectures—which combine the deterministic logic of symbolic planners with the statistical generalising capability of LLMs—are another important trend. Such hybrid systems preserve structural rigour in execution while making decisions in uncertain settings.

Moreover, methods aiming at improving scalability are under investigation including hierarchical planning and incremental plan development. These techniques let agents create elements of a plan on-demand or plan at several degrees of abstraction, hence lowering computing overhead.

m) Creating More Reliable, Intelligent AI Agents

A fundamental progress in the search of strong, autonomous intelligence is the inclusion of outside designers into agentic artificial intelligence systems. Classical designers offer structure, dependability, and task execution rigour; LLMs offer flexibility, creativity, and natural language fluency. These technologies together help agents to not only grasp difficult objectives but also be able to reach them precisely and under control.

Though translation, real-time execution, and scalability still present difficulties, continuous study in semantic parsing, neuro-symbolic planning, and hierarchical strategy generation promises to overcome these constraints. Symbolic planning will become increasingly important as the area develops—not as a substitute for language-based thinking, but rather as a partner in creating artificial intelligence capable of thinking, planning, and acting with both freedom and control.

V. APPLICATIONS AND BENCHMARKS OF AGENTIC AI SYSTEMS POSSESSED WITH PLANNING CAPACITY

Agentic artificial intelligence systems, intelligent agents able of autonomous goal-directed behaviour, have become a major advance in machine intelligence as artificial intelligence develops. These systems actively plan, execute, and change depending on real-world goals rather than only passively data processing. Integration of advanced planning features—especially via classical and hybrid planning models—is a fundamental factor driving this progress.

Highly helpful in complicated, multi-step activities, this mix of goal formulation and execution planning enables agentic artificial intelligence systems to operate efficiently in many situations. Their uses cover a wide spectrum, and their performance is currently evaluated using ever more exact benchmarks meant to demonstrate generalisation and real-world utility. Using actual benchmarking data from top research systems, the following overview of important applications evaluates their success.

A. One is Legal Domain: Improving Reason in Domains of Specialised Knowledge

In the legal field, where activities often call for profound reasoning, contextual knowledge, and the use of specialised tools, agentic AI systems with planning capabilities have great promise. Legal AI chores usually involve digesting thick materials, referencing legal precedents, maintaining deadlines, and reacting to structured searches—all of which necessitate more than basic pattern recognition.

One famous system, GoalAct, showed how carefully using tools and structured planning might greatly improve performance in this field. GoalAct demonstrated a 12.22% increase in task accomplishment rates over standard language models in benchmark assessments when applied to legal processes with dynamic tool interaction.

This performance increase captures the advantages of agentic planning. The system selects whether to employ tools like legal databases or search APIs, defines intermediate legal reasoning goals, iteratively over outputs to confirm legal validity, instead of passive reading text. This methodical preparation reflects how human legal assistants work—by following procedural guidelines consistent with legal goals.

There are significant consequences. Agentic artificial intelligence systems can simplify legal procedures, lower human effort, and increase consistency—while still following domain-specific logic and constraints—from contract draughting to case law review.

B. Virtual environments: wise decision-making in simulated worlds

Virtual environments, where agents are assessed on their capacity to see, plan, and act in simulated but controlled surroundings, provide still another rich field for agentic artificial intelligence applications. These settings are extensively applied for both research and benchmarking and provide controlled testing for artificial intelligence behaviour.

Two venues have become well-known in this field: TextCraft and ALFWorld. While TextCraft is a Minecraft-like text-based universe where agents must execute goal-directed behaviours in innovative, spatial settings, ALFWorld models domestic situations and tasks using natural language.

Here, the ADaPT system—adaptive planning transformer—has established new performance benchmarks. ADaPT obtained up to a 33% greater success rate than conventional language model baselines by combining hierarchical planning mechanisms and fine-tuning depending on environment feedback.

The ability of ADaPT to: explains this rise.

- Convert high level objectives into low level activities.
- Apply environmental feedback loops.
- Change plans in real time depending on virtual conditions.

Such adaptability is really vital, particularly as virtual environments get more sophisticated. Robotics, AR/VR interfaces, and interactive learning tools all provide a proving ground for artificial intelligence models that will ultimately run in actual surroundings.

C. Improving Core Reasoning in Diverse Environments: General Planning

Beyond specialized domains like law or virtual worlds, a major goal in AI research is to enhance general planning ability—the capacity of an AI agent to operate across a wide variety of tasks and domains using transferable skills. This type of capability is vital for building generalist systems that can serve as personal assistants, analysts, or collaborators.

In pursuit of this objective, the AgentGen system was developed to improve the planning capabilities of large language models by exposing them to diverse simulated environments and training tasks. Instead of learning in fixed settings, AgentGen continuously generates new problem domains, tasks, and environmental constraints. This diversity forces the system to learn generalized planning strategies rather than memorized patterns.

- The results were impressive: AgentGen outperformed powerful LLMs like GPT-3.5 on several planning-intensive benchmarks. The key to its success was its novel training methodology:
- Environment Diversity: By varying input structures and outcome constraints, AgentGen prevents overfitting and encourages adaptable reasoning.
- Task Generation Framework: New goals are programmatically created, requiring the agent to build unique plans each time.
- Feedback and Iteration: The system refines its strategies through continual interaction and evaluation.

Such advances suggest a future where agentic AI systems can serve as intelligent co-workers, able to take vague instructions and autonomously design strategies to meet objectives across countless scenarios.

D. Agentic AI Progress: Benchmarking

Researchers use a variety of benchmarking techniques, catered to test planning ability, adaptability, tool use, and task performance, to evaluate the effectiveness of these systems. The most often used benchmarks consist in:

Benchmark Environment	Domain	Measured Capabilities	Top Performer	Performance Gain
ALFWorld	Virtual Home Navigation	Goal planning, object manipulation	ADaPT	+33% success rate
TextCraft	Creative Virtual Tasks	Multi-step planning, resource management	ADaPT	+33% success rate
Legal AI Benchmarks	Legal Reasoning	Tool usage, structured response generation	GoalAct	+12.22% success rate
TaskEval (AgentGen)	General Task Planning	Task decomposition, goal generalization	AgentGen	Outperformed GPT-3.5
ToolBench (Various)	Tool Use Across Domains	Decision making, external tool invocation	GoalAct, ReAct	Varies by system

Especially in tasks requiring multi-step reasoning, error management, and adaptability, these benchmarks indicate a continuous pattern: systems including structured planning outperform those depending just on end-to--end language models.

E. Greater Impact and Practical Uses

The immediate consequences of the performance increases found in controlled benchmark environments for practical applications of agentic artificial intelligence: LegalTech Platforms: Automated contract review, case summaries, and more precisely and logically based legal research.

Intelligent non-player characters (NPCs) responding logically to player actions or tutor systems customising learning paths represent virtual agents in games and education. Integration into robots and smart assistants able to carry out duties in homes, factories, or hospitals under little control marks autonomous systems. Task coordination in personal AI assistants that grasp user goals and carry out multi-app processes (e.g., booking travel, generating reports) helps to increase productivity.

F. Conclusion

Advanced planning's inclusion into agentic artificial intelligence systems is enabling hitherto unheard-of capacity in challenging task performance in several spheres. From legal analysis to adaptive virtual agents, systems such as GoalAct, ADaPT, and AgentGen show the pragmatic advantages of organised planning mixed with language comprehension.

By means of exhaustive benchmarking, we now provide unambiguous proof that planning-aware artificial intelligence systems greatly exceed their non-planning counterparts in task success, adaptability, and reasoning. Research keeps extending these capabilities—especially in generalisable planning and real-time adaptation—the future of agentic AI promises smarter, more autonomous, and more helpful AI systems across industries.

VI. AGENTIC AI SYSTEM CHALLENGES AND FUTURE DIRECTIONS

From legal automation to autonomous robotics, as agentic artificial intelligence systems become more adept and integrated across sectors, their development poses basic questions the industry must answer. Currently hampered by problems of scalability, flexibility, evaluation, and transparency, these intelligent agents designed to plan, reason, and act based on abstract goals are Unlocking the full potential of agentic artificial intelligence and deploying it safely and successfully in real-world settings depend on overcoming these obstacles. The main constraints of present systems are investigated in this part together with suggested research paths that might provide more solid, generalisable agents.

A. Planning for Long-Horizon Tasks: The Scalability Challenge

Agentic systems' capacity to scale within the framework of long-horizon planning is one of their most major challenges. These chores span several steps and usually involve complicated dependencies and restrictions. While most artificial intelligence agents lack this degree of sophistication, humans naturally approach such jobs with hierarchical thinking—breaking down big ambitions into smaller, reasonable ones. As job length rises, planning space complexity multiplies exponentially, rapidly exceeding even state-of- the-art technologies.

Think about a real-world situation: planning an international academic conference. Securing venues, handling travel logistics for keynote speakers, planning concurrent sessions, organising media coverage, and guaranteeing adherence to financial limits comprise the work involved. Every sub-task consists in additional layers of conditional dependencies and detail. Agentic artificial intelligence has to not only plan across several levels of abstraction but also monitor the coherence of its approach across these layers if it is to properly handle such complexity.

Often depending on domain-specific rule engineering or static hierarchical templates, current methods restrict generalisability. Aiming to get over this bottleneck, research on dynamic planning architectures including learning-based abstraction, modular sub-goal systems, and incremental and recursive planning is These techniques seem to enable agents to keep coherence in long plans without giving in under the computational weight of thorough search.

Challenge Area	Description	Current Limitation	Emerging Solution
Long-Horizon Planning	Executing tasks across many steps and sub-goals	Exponential planning space, inconsistent decomposition	Hierarchical and modular planning
Task Complexity	Handling tasks with interdependent resources and deadlines	Poor scalability in real-time environments	Plan abstraction, staged execution
Strategic Coherence	Maintaining logical continuity across sub-plans	Replanning triggers conflict with original strategy	Incremental refinement frameworks

B. Navigating Uncertainty: Real-Time Adaptability in Dynamic Systems

Adaptability presents still another basic difficulty. The real world is uncertain; changing user needs, changing surroundings, and unanticipated challenges might make precomputed plans useless. Agents in fields like autonomous driving or emergency response have to be able to dynamically change their plans and react correctly to fresh data.

Still, many present agents find it difficult to adapt to unforeseen circumstances and fail to replan. Once a plan is upset, they lack the reflexive architecture to assess whether to modify, abort, or rebuild a strategy. Actually, this results in brittle behaviour, poor performance, and significant risk in safety-sensitive applications.

Three elements define adaptability: ongoing environmental observation, real-time assessment of the correctness of the plan, and instantaneous action modification capability. Researchers are looking at meta-planning structures that reason not only about chores but also about the planning process itself in order to assist these skills. Other interesting paths include constant learning, reactive planning, and integration of reinforcement learning for dynamic policy modification. Increasing research also investigates lifetime learning in agents—systems that change and improve their planning processes over time depending on experience. Agents that reason across time—not only about what to do now but also about how their planning techniques must change—will probably define the future of agentic adaptation.

Domain Example	Real-Time Adaptation Requirement	Current Weakness	Research Strategy
Robotics	Navigating unpredictable physical spaces	Fixed plans, poor reactivity	Reactive and adaptive planners
Healthcare AI	Adjusting recommendations in real-time	Static logic, lacks context	Meta-planning, continual learning
Digital Assistants	Changing user intent mid-conversation	Weak dialog memory models	Policy updating, feedback integration

C. Redining benchmarks and metrics helps to clarify the need of better evaluation.

Agentic artificial intelligence has attracted a lot of attention, however the discipline still suffers from few, often false evaluation criteria. Although several benchmarks—including ALFWorld and TextCraft—are great for organised testing—they do not fully represent the range of real-world tasks agents must manage. Moreover, the most often used statistic is still task success rate, which misses qualitative elements as agent planning process logical coherence, clarity, or efficiency.

A better knowledge of agentic intelligence calls for benchmarks evaluating:

- How effectively does an agent break out tasks?
- How adaptably it modifies tactics to fit different settings?
- Whether human partners can understand its choices.

Two agents might both do a task, for example, but one might do it with raw force while another would employ a modest, elegant strategy. Current benchmarks cannot distinguish this. Richer assessment suites including measures for decomposition depth, strategy variability, planning traceability, and interaction quality is thus under demand by the research community.

Metric Type	What It Measures	Example Tool/Benchmark	Gap Addressed
Task Success	Final goal completion	ALFWorld, TextCraft	Ignores process quality
Plan Traceability	Visibility into agent's reasoning steps	TaskEval	Enhances interpretability
Strategic Robustness	Adaptation under constraint or failure	ToolBench	Tests contingency handling
Collaboration Index	Human-AI task co-performance	Human-Agent Dialog Simulators	Evaluates interactivity

D. Future Horizons: Topics of Research Aimed at Generalisable Agents

Development of the next generation of agentic artificial intelligence calls for more than just small model enhancements. It calls for fundamental changes in interaction, reasoning, and planning. Autonomous self-planning—that which lets agents create and edit plans free from outside templates or programming—is one important topic. This entails strongly linking planning logic to language understanding, and more and more merging planning modules straight into LLMs or hybrid neuro-symbolic models.

Agentic artificial intelligence should also be broadening its sensory horizon. Right now systems mostly run on structured or text inputs. But in the actual world, smart decision-making often rely on multimodal information—visual data, music, spatial interactions, even tactile input. Future agents needed for usage in robotics, augmented reality, and healthcare support must learn to comprehend and respond on many sensory data streams.

Another ongoing difficulty is interpretability. Understanding agents' thinking processes becomes crucial for safety, trust, and control as they make more serious judgements. Future agentic systems should not only explain what they did but also why they did it, what alternatives were taken into account, and under what circumstances they would change their approach. Such ability will encourage more efficient human-AI cooperation.

E. Conclusion: Towards Intelligent, Transparent, and Resilient Agentic AI

Although the present generation of agentic artificial intelligence systems shows great potential, releasing their full capability calls for overcoming fundamental problems in planning scalability, real-time flexibility, and meaningful evaluation. Agentic systems will change from narrowly scoped task execution to open-domain strategic collaboration as research improves in fields such hierarchical self-planning, multimodal learning, and explainable reasoning.

The road forward calls for cooperation amongst disciplines—linking artificial intelligence planning, machine learning, cognitive science, and HCI to create agents not merely functionally competent but also contextually aware and communicatively aligned. Built on adaptive intelligence and ethical planning, agentic artificial intelligence is ready to be a necessary friend in every sphere of life in helping to solve practical challenges.

VII. CONCLUSION

What distinguishes very intelligent agents from simple reactive systems is essentially goal decomposition and self-planning. These two skills help agents to not only interpret instructions but also to carry them out with intention, structure, and adaptability as artificial intelligence moves from language understanding to autonomous action.

This work presents research showing how recent developments—especially those using LLMs—have greatly increase what is feasible in agentic artificial intelligence. Frameworks highlighting the variety of planning techniques now at use include ADaPT, AdaPlanner, and GoalAct. These systems let agents make decisions more precisely, learn from feedback, and operate in situations marked by change and uncertainty whether by recursive decomposition, constant improvement, or hierarchical skill execution.

Moreover, encouraging ideas like Tree of Thought and Chain of Thought has been quite helpful in increasing the reasoning capability of LLMs. These methods not only raise output quality but also help agentic decisions to be interpretable by externalising intermediate phases. ReAct and Reflexion are two reflexive models that incorporate learning and self-assessment into daily routines, therefore adding still another level of cognitive complexity.

Beyond architectural design, a major advance is the inclusion of external symbolic planners with LLM-based agents. This mix combines the scalability and accuracy of classical planning systems with the flexibility of language models. Agents can function with more logical consistency by means of languages such as PDDL, particularly in fields demanding strict task validation and multi-stage coordination.

Empirical data corroborate the potential of these developments. With notable performance advancements in legal reasoning and virtual job completion, systems such as GoalAct and ADaPT have demonstrated considerable improvements in benchmark environments. Especially in goal-oriented environments, these measures support the perspective that planning-aware designs outperform merely reactive models.

Still, problems abound. Particularly in long-horizons planning, scalability presents major computational and design challenges. Open research frontiers are adaptability in real-time situations, the desire for richer multimodal inputs, and interpretable and trustworthy AI behaviour. Dealing with these problems calls for a multidisciplinary approach incorporating developments in symbolic reasoning, natural language processing, reinforcement learning, and human-computer interaction.

Looking ahead, our capacity to create systems that are not only autonomous but also cooperative, open, and able of lifetime learning will define agentic artificial intelligence. Agents—from legal assistants to educational tutors to autonomous cars—become more ingrained in real-world processes and must be able to break goals, plan effectively, and change behaviour.

In essence, goal decomposition and self-planning are operational pillars of modern artificial intelligence rather than theoretical capacity. We are launching a new age of intelligent, dependable, and flexible machines by including these faculties into LLM-driven designs. The next frontiers are in honing these skills, spreading them over several modalities, and making sure they line up with human values and goals.

VIII. REFERENCES

[1] Ahn, Y., Choi, E., & Singh, S. (2023). *Language models as zero-shot planners: Extracting actionable plans from LMs* (arXiv:2302.05022). arXiv. https://arxiv.org/abs/2302.05022

[2] Bai, Y., Kadavath, S., & Kundu, S. (2023). *Constitutional AI: Harmlessness from AI feedback* (arXiv:2306.04762). arXiv. https://arxiv.org/abs/2306.04762

[3] Chen, J., Li, H., Yang, J., et al. (2025). *Enhancing LLM-based agents via global planning and hierarchical execution* (arXiv:2504.16563). arXiv. https://arxiv.org/abs/2504.16563

[4] Chen, Y., Li, Z., Zhang, H., & Xu, D. (2022). *Planning with large language models for open-ended tasks* (arXiv:2212.09611). arXiv. https://arxiv.org/abs/2212.09611

[5] Dohan, D., Gehrmann, S., & Gilmer, J. (2022). *Language model cascades* (arXiv:2207.10342). arXiv. https://arxiv.org/abs/2207.10342

[6] Dziri, N., Alabbas, M., & Chang, K. (2023). *Faithfulness in chain-of-thought reasoning* (arXiv:2302.03494). arXiv. https://arxiv.org/abs/2302.03494

[7] Gao, X., Liu, J., & Liu, B. (2023). *PAL: Program-aided language models* (arXiv:2211.10435). arXiv. https://arxiv.org/abs/2211.10435

[8] Ge, T., Li, Z., & Xu, D. (2023). *Toolformer: Language models can teach themselves to use tools* (arXiv:2302.04761). arXiv. https://arxiv.org/abs/2302.04761

[9] Huang, B., & Wang, J. (2023). *Prompting large language models for planning and acting in open worlds* (arXiv:2301.04180). arXiv. https://arxiv.org/abs/2301.04180

[10] Hu, M., Zhao, P., Xu, C., et al. (2024). *AgentGen: Enhancing planning abilities for large language model-based agents via environment and task generation* (arXiv:2408.00764). arXiv. https://arxiv.org/abs/2408.00764

[11] Jiang, Y., Zhang, Q., & Wang, M. (2023). *Decision transformer: Reinforcement learning via sequence modeling* (arXiv:2106.01345). arXiv. https://arxiv.org/abs/2106.01345

[12] Khot, T., Sabharwal, A., & Clark, P. (2022). Explaining via chains of reasoning. *Transactions of the Association for Computational Linguistics*, *10*, 1340–1356. https://doi.org/10.1162/tacl_a_00508

[13] Kim, B., & Rudin, C. (2021). *Interpretable machine learning: A guide for making black box models explainable*. Springer.

[14] Kojima, T., Gu, S. S., Reid, M., et al. (2022). *Large language models are zero-shot reasoners* (arXiv:2205.11916). arXiv. https://arxiv.org/abs/2205.11916

[15] Li, L., & Liang, P. (2023). *Planning with multistep reasoning in language models* (arXiv:2303.04761). arXiv. https://arxiv.org/abs/2303.04761

[16] Liu, Y., Li, Q., & Lin, Z. (2023). *APE: Ask, plan, execute* (arXiv:2305.05087). arXiv. https://arxiv.org/abs/2305.05087

[17] Long, D., & Fox, M. (2003). The language of the PDDL2.1 planning competition. *Journal of Artificial Intelligence Research*, *20*, 279–292. https://doi.org/10.1613/jair.1176

[18] Miller, T., & Binns, R. (2022). Explanation in artificial intelligence: Insights from the social sciences. *AI Magazine*, *43*(1), 7–18. https://doi.org/10.1002/aaai.12041

[19] Nye, M., Hewitt, C., & Lake, B. (2021). Skill acquisition through language and planning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*.

[20] OpenAI. (2023). *GPT-4 Technical Report*. https://openai.com/research/gpt-4

[21] Prasad, A., Koller, A., Hartmann, M., et al. (2023). *ADaPT: As-needed decomposition and planning with language models* (arXiv:2311.05772). arXiv. https://arxiv.org/abs/2311.05772

[22] Reimers, N., & Gurevych, I. (2020). *Making monolingual sentence embeddings multilingual using knowledge distillation* (arXiv:2004.09813). arXiv. https://arxiv.org/abs/2004.09813

[23] Schaeffer, R., Mirhoseini, A., & Bengio, Y. (2023). *Visual reasoning with chain-of-thought prompting* (arXiv:2304.12789). arXiv. https://arxiv.org/abs/2304.12789

[24] Shinn, N., & Labash, A. (2023). *Reflexion: Language agents with verbal reinforcement learning* (arXiv:2303.11366). arXiv. https://arxiv.org/abs/2303.11366

[25] Sun, H., Zhuang, Y., Kong, L., et al. (2023). *AdaPlanner: Adaptive planning from feedback with language models* (arXiv:2305.16653). arXiv. https://arxiv.org/abs/2305.16653

[26] Tamkin, A., Brundage, M., Clark, J., & Ganguli, D. (2021). *Understanding the capabilities, limitations, and societal impact of large language models* (arXiv:2102.02503). arXiv. https://arxiv.org/abs/2102.02503

[27] Touvron, H., Lavril, T., & Simonyan, K. (2023). *LLaMA 2: Open foundation and chat models* (arXiv:2307.09288). arXiv. https://arxiv.org/abs/2307.09288

[28] Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, *30*. https://papers.nips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[29] Wang, B., Lin, Z., & Liu, P. (2023). *ToolLLM: Facilitating tool-use for language models via prompt engineering* (arXiv:2305.12047). arXiv. https://arxiv.org/abs/2305.12047

[30] Wang, J., & Li, D. (2023). Code-as-policies: Language model programs for embodied control. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=BsQ6w2h-4f

[31] Wei, J., Wang, X., Schuurmans, D., et al. (2022). Chain of thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems (NeurIPS)*, *35*, 24824–24837. https://papers.nips.cc/paper_files/paper/2022/file/9d5609613524fd94be3c2142c4cf4020-Paper-Conference.pdf

[32] Wu, S., Agarwal, M., & Xu, Q. (2023). *Zero-shot tool selection with large language models* (arXiv:2303.08128). arXiv. https://arxiv.org/abs/2303.08128

[33] Xie, X., & Pu, Y. (2023). *Multi-agent planning with LLMs* (arXiv:2307.09299). arXiv. https://arxiv.org/abs/2307.09299

[34] Yao, S., Zhao, J., Yu, D., et al. (2023). *Tree of Thoughts: Deliberate problem solving with large language models* (arXiv:2305.10601). arXiv. https://arxiv.org/abs/2305.10601

[35] Ye, Y., Li, X., & Lin, J. (2023). *ToolBench: Evaluating tool use of large language models* (arXiv:2305.17126). arXiv. https://arxiv.org/abs/2305.17126

[36] Yu, W., & Wang, Z. (2023). Knowledge-grounded multi-step reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. https://aclanthology.org/2023.emnlp-main.101

[37] Zhang, J., & Liu, X. (2023). *Prompt-based planning for generalist agents* (arXiv:2302.08295). arXiv. https://arxiv.org/abs/2302.08295

[38] Zhao, P., Yu, H., & Xu, C. (2023). *Integrating PDDL planners into LLMs: A hybrid symbolic-neural framework* (arXiv:2307.01468). arXiv. https://arxiv.org/abs/2307.01468

[39] Zheng, Y., & Chen, B. (2023). *Multi-hop symbolic reasoning with LLMs* (arXiv:2304.08510). arXiv. https://arxiv.org/abs/2304.08510

[40] Zhou, W., Zhang, R., & Wang, J. (2023). *Can large language models solve planning problems?* (arXiv:2303.12705). arXiv. https://arxiv.org/abs/2303.12705