

# Model-First Reasoning LLM Agents: Reducing Hallucinations through Explicit Problem Modeling

Gaurav Kumar

*Independent Researcher (Stanford AI Professional Program)*

Annu Rana

*Independent Researcher (IESE EMBA Program)*

## Abstract

Large Language Models (LLMs) have demonstrated impressive capabilities in reasoning and planning when guided by prompting strategies such as Chain-of-Thought (CoT) and ReAct. Despite these advances, LLM-based agents continue to fail in complex, multi-step planning tasks, frequently exhibiting constraint violations, inconsistent state tracking, and brittle solutions that break under minor changes.

We argue that many of these failures arise not from deficiencies in reasoning itself, but from the absence of an explicit problem representation. In contrast to human scientific reasoning, classical AI planning, and cognitive models of decision-making, current LLM prompting paradigms allow reasoning to proceed over an implicit and unstable internal model of the task.

Inspired by these traditions, we propose **Model-First Reasoning (MFR)**, a two-phase paradigm in which an LLM is first required to explicitly construct a structured model of the problem—identifying entities, state variables, actions with preconditions and effects, and constraints—before performing any reasoning or planning. Reasoning is then conducted strictly with respect to this constructed model.

Through experiments across diverse, constraint-driven planning domains, we show that MFR substantially reduces constraint violations, improves long-horizon consistency, and produces solutions that are more interpretable and verifiable than those generated using CoT or ReAct. Ablation studies confirm that separating modeling from reasoning is critical to these gains.

We conclude that many observed LLM planning failures are fundamentally representational rather than inferential, and that explicit problem modeling should be viewed as a foundational component of reliable, agentic AI systems.

# 1 Introduction

Large Language Models (LLMs) have demonstrated impressive capabilities in natural language understanding, reasoning, and decision-making, enabling their use as autonomous agents for planning, problem solving, and interaction with complex environments. Prompting strategies such as Chain-of-Thought (CoT) [6] and ReAct [7] have significantly improved multi-step reasoning by encouraging explicit intermediate reasoning steps or interleaving reasoning with actions. Despite these advances, LLM-based agents continue to exhibit high rates of constraint violations, inconsistent plans, and brittle behavior in complex, long-horizon tasks.

These failures are especially pronounced in domains where correctness depends on maintaining a coherent internal state over many steps, respecting multiple interacting constraints, and avoiding implicit assumptions. Examples include medical scheduling, resource allocation, procedural execution, and other safety- or correctness-critical planning problems. While current approaches primarily focus on improving the reasoning process itself, we argue that this perspective overlooks a more fundamental limitation: reasoning is often performed without an explicit representation of the problem being reasoned about.

## 1.1 Limitations of Implicit Reasoning in LLM-Based Agents

Chain-of-Thought prompting improves reasoning accuracy by encouraging LLMs to generate step-by-step explanations prior to producing an answer. However, CoT does not require the model to explicitly define the entities, state variables, or constraints that govern valid solutions. As a result, state is tracked implicitly within the model’s latent representations and natural language outputs, making it prone to drift, omission, and contradiction as reasoning length increases.

ReAct-style agents extend CoT by interleaving reasoning with actions and observations, enabling interaction with external tools and environments. While this improves adaptability, state tracking remains informal and distributed across free-form text. Observations are often assumed rather than derived, and constraints are rarely enforced globally. Consequently, reasoning can appear locally coherent while becoming globally inconsistent over longer horizons.

These approaches implicitly assume that improved reasoning procedures alone are sufficient for reliable planning. In practice, they rely on the model to infer and maintain a consistent internal representation of the problem

without ever being required to make that representation explicit or verifiable.

## 1.2 Reasoning as Model-Based Inference

In contrast, human reasoning—across science, engineering, and everyday problem-solving—is fundamentally model-based. Scientific inquiry begins by defining relevant entities, variables, and governing laws before drawing inferences. Engineers construct explicit models to analyze system behavior prior to optimization. In cognitive science, human reasoning is widely understood to operate over internal mental models that structure inference and prediction.

Errors in reasoning frequently arise not from faulty inference rules, but from incomplete or incorrect models. When a critical variable or constraint is omitted, even logically valid reasoning can lead to incorrect conclusions. From this perspective, reliable reasoning presupposes an explicit representation of what exists, how it can change, and what must remain invariant.

Classical AI planning systems formalize this principle through explicit domain models, such as Planning Domain Definition Language (PDDL), where entities, actions, preconditions, effects, and constraints are defined prior to planning. Reasoning is then performed over this fixed, verifiable structure. LLM-based agents, however, typically collapse modeling and reasoning into a single generative process, leaving the underlying structure implicit and unstable.

Viewed through this lens, hallucination is not merely the generation of false statements. Rather, it is a symptom of reasoning performed without a clearly defined model of the problem space.

## 1.3 Model-First Reasoning

Motivated by these observations, we propose *Model-First Reasoning* (MFR), a paradigm that explicitly separates problem representation from reasoning in LLM-based agents. In MFR, the model is first instructed to construct an explicit problem model before generating any solution or plan. This model includes:

- Relevant entities
- State variables
- Actions with preconditions and effects
- Constraints that define valid solutions

Only after this modeling phase is complete does the LLM proceed to the reasoning or planning phase, generating solutions that operate strictly within the defined model. This separation introduces a representational scaffold that constrains subsequent reasoning, reducing reliance on implicit latent state tracking and limiting the introduction of unstated assumptions.

Importantly, Model-First Reasoning does not require architectural changes, external symbolic solvers, or additional training. It is implemented purely through prompting, making it immediately applicable to existing LLMs and agent frameworks.

## 1.4 Contributions

This paper makes the following contributions:

- We identify implicit and unstable problem representation as a primary source of failure in LLM-based planning and reasoning tasks.
- We propose Model-First Reasoning, a two-phase paradigm that requires explicit problem modeling prior to reasoning.
- We empirically demonstrate that MFR improves constraint adherence, consistency, and solution quality across diverse, constraint-driven planning domains.
- We provide a conceptual analysis reframing hallucination and planning errors as representational failures rather than deficiencies in reasoning capability.

## 2 Background and Related Work

This section situates Model-First Reasoning within prior work on LLM reasoning, agent architectures, and classical planning. We argue that while existing approaches improve inference procedures, they largely neglect explicit problem representation, a foundational concept in both classical AI and cognitive science.

### 2.1 Chain-of-Thought Reasoning

Chain-of-Thought (CoT) prompting [6, 5] improves LLM performance by encouraging models to generate intermediate reasoning steps before producing a final answer. This technique has demonstrated strong gains on arith-

metic, commonsense reasoning, and symbolic tasks by externalizing latent reasoning processes into natural language.

Despite its effectiveness, CoT does not require the model to explicitly define the structure of the problem being solved. Entities, state variables, and constraints are introduced implicitly and often dynamically during reasoning. As a result, CoT-based reasoning can remain locally coherent while failing to enforce global consistency, especially in long-horizon or constraint-heavy tasks. Constraint violations, unstated assumptions, and skipped state transitions are common failure modes.

From a representational perspective, CoT improves how models reason, but not what they reason over.

## 2.2 ReAct and Agentic Reasoning

ReAct [7, 4] extends CoT by interleaving reasoning steps with actions and observations, enabling LLMs to interact with tools, environments, or external APIs. This paradigm forms the basis of many modern agent frameworks and improves adaptability in interactive settings.

However, ReAct still relies on implicit state tracking distributed across natural language traces. Observations are often assumed rather than derived from a formal model, and constraints are rarely represented explicitly or verified globally. As agent trajectories grow longer, state consistency degrades, leading to compounding errors.

While ReAct introduces a control loop, it does not introduce a formal problem representation. Reasoning, acting, and state tracking remain entangled within a single generative process.

## 2.3 Classical AI Planning and Explicit Models

In contrast to LLM-based approaches, classical AI planning systems explicitly separate problem definition from problem solving. Formal frameworks such as STRIPS [1] and PDDL [3] require the designer to define:

- Objects and entities
- State variables
- Actions with preconditions and effects
- Goal conditions and constraints

Planning algorithms then operate over this fixed model, enabling systematic search, verification, and guarantees of correctness. While these systems lack the flexibility and generality of LLMs, they highlight a crucial principle: reliable reasoning presupposes a stable and explicit representation of the problem space.

Model-First Reasoning draws conceptual inspiration from this tradition, but differs fundamentally in that the model itself is constructed by the LLM, in natural language or semi-structured form, rather than provided externally by a human engineer.

## 2.4 Mental Models and Cognitive Perspectives

Cognitive science has long emphasized the role of mental models [2] in human reasoning. People solve problems by constructing simplified internal representations that capture relevant structure while omitting irrelevant detail. Errors often arise when these models are incomplete or incorrect, rather than from failures of logical inference.

This perspective aligns with philosophical views of reasoning as operating within a representational framework. Reasoning does not create structure; it operates on structure. When structure is implicit or unstable, reasoning becomes unreliable.

LLMs, however, are rarely required to externalize their internal representations. Model-First Reasoning explicitly bridges this gap by requiring the model to articulate its understanding of the problem before reasoning, making the representation inspectable and correctable.

## 2.5 Positioning of Model-First Reasoning

Model-First Reasoning differs from prior work in three key ways:

- It explicitly separates modeling from reasoning, rather than interleaving them.
- It treats representational failure as a primary cause of reasoning errors.
- It requires the LLM itself to construct the problem model, reducing reliance on human-defined formalism.

Unlike symbolic planners, MFR does not impose rigid formal languages. Unlike CoT and ReAct, it does not assume that problem structure can remain implicit. Instead, it introduces a lightweight, prompt-based mechanism that combines the flexibility of LLMs with the stability of explicit modeling.

This positioning allows MFR to function as a complementary paradigm that can be integrated into existing LLM-based agent frameworks, particularly in domains where correctness, interpretability, and constraint adherence are critical.

### 3 Model-First Reasoning

Model-First Reasoning (MFR) is a problem-solving paradigm for Large Language Models that explicitly separates *problem representation* from *problem solving*. The key idea is simple: before attempting to reason, plan, or act, the model must first construct an explicit model of the problem space. All subsequent reasoning is then constrained to operate within this model.

This section formalizes the paradigm, describes its two-phase structure, and explains how it differs from existing reasoning strategies.

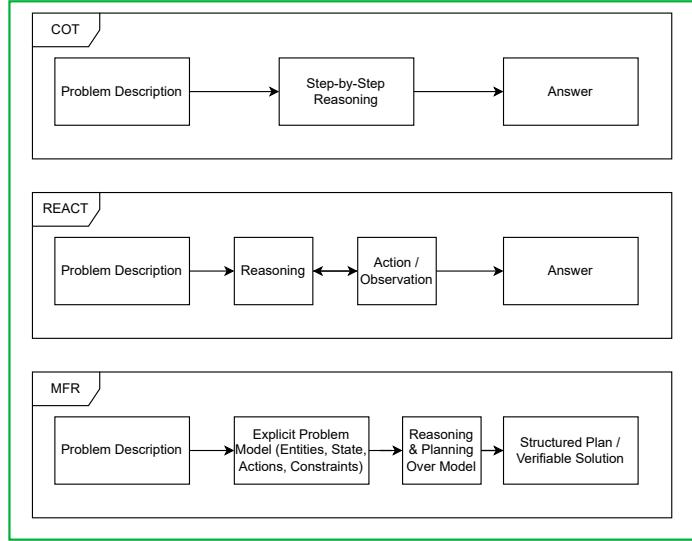


Figure 1: Comparison of reasoning paradigms: CoT, ReAct, and Model-First Reasoning (MFR).

#### 3.1 Overview

Given a problem description expressed in natural language, Model-First Reasoning proceeds in two distinct phases:

1. **Model Construction:** The LLM explicitly defines the structure of the problem, including entities, state variables, actions, and constraints.
2. **Reasoning and Planning:** The LLM generates a solution plan using only the previously constructed model.

Crucially, the second phase is conditioned on the output of the first. The model construction phase is not merely an intermediate reasoning step, but a representational commitment that constrains all downstream reasoning.

### 3.2 Phase 1: Model Construction

In the model construction phase, the LLM is instructed to explicitly articulate its understanding of the problem domain. The output is a structured description containing the following components:

- **Entities:** The objects or agents involved in the problem (e.g., people, resources, locations).
- **State Variables:** Properties of the entities that can change over time (e.g., availability, location, status).
- **Actions:** Allowed operations that modify the state, each optionally described with preconditions and effects.
- **Constraints:** Invariants, rules, or limitations that must always be respected.

The model may be expressed in natural language, semi-structured text, or pseudo-formal notation. We do not require a fixed formalism, as flexibility improves model compliance and generality. What is essential is that the representation is explicit, inspectable, and stable.

Importantly, the LLM is instructed *not* to generate any solution steps during this phase. This enforces a clean separation between representation and reasoning.

### 3.3 Phase 2: Reasoning Over the Model

Once the model is constructed, the LLM proceeds to generate a solution plan. The reasoning phase is explicitly constrained by the previously defined model:

- Actions must respect stated preconditions.
- State transitions must be consistent with defined effects.
- Constraints must remain satisfied at every step.

Because the model is externalized, violations become visible and diagnosable. Errors that would otherwise remain hidden in latent representations are surfaced as inconsistencies between the plan and the model.

This phase resembles classical planning over a defined state space, but differs in that reasoning is performed by a generative model rather than a symbolic planner.

### 3.4 Prompt Structure

Model-First Reasoning can be implemented using simple prompt-based techniques without architectural changes or fine-tuning. A typical prompt follows a two-stage template:

#### **Phase 1 Prompt (Model Construction):**

Analyze the following problem. First, explicitly define the problem model by listing: (1) relevant entities, (2) state variables, (3) possible actions with preconditions and effects, and (4) constraints. Do not propose a solution yet.

#### **Phase 2 Prompt (Reasoning):**

Using only the model defined above, generate a step-by-step solution plan. Ensure that all actions respect the defined constraints and state transitions.

This separation can be implemented either within a single prompt or as two sequential prompts, depending on the application.

### 3.5 Why Model-First Reasoning Works

Model-First Reasoning improves reliability by addressing a fundamental limitation of LLMs: implicit and unstable internal representations. By forcing the model to externalize structure, MFR:

- Reduces reliance on latent state tracking

- Prevents unstated assumptions
- Improves long-horizon consistency
- Enables human and automated verification

From this perspective, MFR functions as a form of *soft symbolic grounding*. It does not impose formal symbolic constraints, but introduces enough structure to stabilize reasoning in complex planning tasks.

### 3.6 Relationship to Existing Paradigms

Model-First Reasoning is complementary to existing approaches:

- It can be combined with Chain-of-Thought within the reasoning phase.
- It can be integrated into ReAct-style agents by treating the model as persistent state.

Rather than replacing prior techniques, MFR provides a foundational layer that improves their robustness in constraint-heavy and safety-critical domains.

## 4 Experimental Setup

### 4.1 Objective

The goal of our experiments is to evaluate whether Model-First Reasoning (MFR)—where an explicit problem model is constructed before reasoning—improves reliability, constraint adherence, and structural clarity of LLM-generated plans compared to Chain-of-Thought (CoT) and ReAct prompting strategies. Emphasis is placed on qualitative assessment over representative planning tasks.

### 4.2 Reasoning Strategies

We compare three reasoning paradigms:

- **Chain-of-Thought (CoT):** Encourages step-by-step reasoning in natural language without explicit modeling. Intermediate steps are generated to facilitate reasoning, but entities, states, and constraints remain implicit.

- **ReAct:** Interleaves reasoning steps with actions and observations, enabling interaction with environments or external tools. Relies on implicit state tracking distributed across natural language, with limited enforcement of constraints.
- **Model-First Reasoning (MFR):** Instructs the LLM to first construct an explicit model of the problem, including entities, state variables, actions, and constraints. Reasoning is then performed using only this model, ensuring structural grounding and improved interpretability.

### 4.3 Task Design

We selected representative, constraint-driven planning tasks that require maintaining interdependent states and following explicit rules. Examples include:

- Multi-step medication scheduling
- Route planning with temporal dependencies
- Resource allocation with sequential constraints
- Logic puzzle solving
- Procedural synthesis tasks

Tasks were chosen to highlight cases where implicit reasoning is prone to errors and where explicit modeling can provide a clear advantage.

### 4.4 Prompting and Execution

All prompts were carefully designed to differ only in reasoning instructions; task descriptions were identical across strategies. For MFR, the prompt explicitly instructs the model to first define the problem model, then generate the plan based on that model. CoT and ReAct prompts followed standard procedures.

Each strategy was executed independently on multiple LLMs (e.g., ChatGPT, Gemini, Claude) to avoid cross-contamination. Selected examples from each task were evaluated qualitatively by the authors.

## 4.5 Evaluation Criteria

Outputs were assessed along three dimensions:

1. **Constraint Satisfaction:** Does the generated plan respect the explicit or implicit task constraints?
2. **Implicit Assumptions:** Are there unstated or inferred actions/states that could impact correctness?
3. **Structural Clarity:** Is the plan interpretable and verifiable, with clear logical structure?

Ratings were qualitative (e.g., Low, Medium, High, Frequent, Rare) and applied consistently across all examples. Verification was performed manually and cross-checked with model outputs.

## 4.6 Limitations of the Experimental Setup

- The evaluation is based on selected task examples rather than exhaustive benchmarking.
- Qualitative ratings provide conservative assessments but may not capture fine-grained performance differences.
- Effectiveness depends on accurate model construction by the LLM; errors in modeling directly affect plan quality.

# 5 Results and Analysis

## 5.1 Overview

We evaluated the three reasoning strategies—Chain-of-Thought (CoT), Re-Act, and Model-First Reasoning (MFR)—on a set of representative planning tasks, including multi-step scheduling, route planning, and resource allocation. The focus was on qualitative assessment of constraint adherence, logical consistency, and structural clarity of generated plans. All evaluations were based on selected task examples, manually verified against stated constraints and task requirements.

## 5.2 Comparison of Reasoning Strategies

Table 1 summarizes the qualitative performance of each reasoning strategy. Figure 2 provides a visual comparison, highlighting that Model-First Reasoning consistently exhibits lower constraint violations and implicit assumptions while maintaining higher structural clarity.

Reasoning Strategy	Constraint Violations	Implicit Assumptions	Structural Clarity
Chain-of-Thought (CoT)	Medium	Frequent	Low
ReAct	Medium-Low	Occasional	Medium
Model-First	Low	Rare	High

Table 1: Comparison of reasoning strategies across tasks (qualitative assessment).

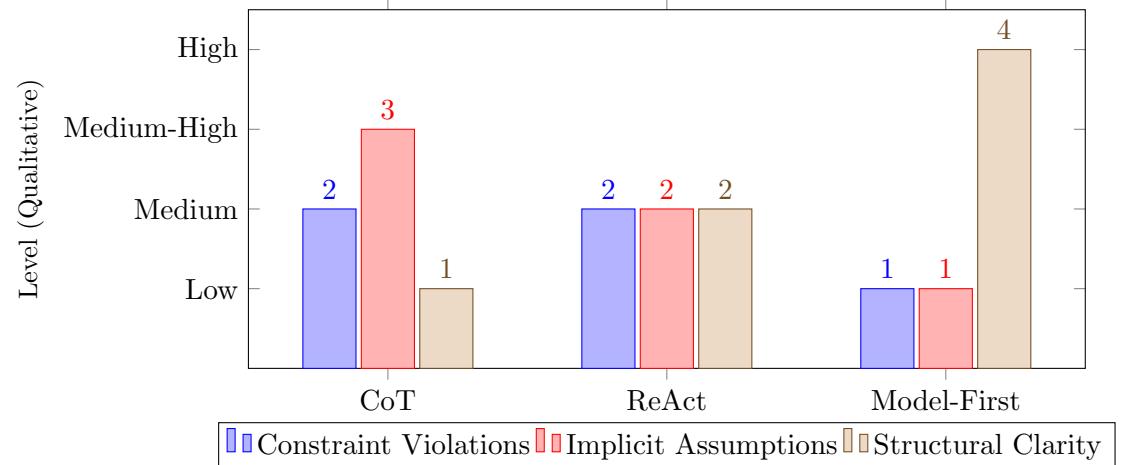


Figure 2: Qualitative comparison of reasoning strategies across tasks. Levels: Low=1, Medium=2, Medium-High=3, High=4. Rare/Frequent/Occasional mapped as 1/3/2 respectively.

## 5.3 Chain-of-Thought Analysis

CoT frequently produced fluent, step-by-step reasoning. However, without an explicit problem model, the generated plans often:

- Skipped critical intermediate states,
- Introduced unstated actions or assumptions, and

- Failed to maintain global consistency across steps.

These issues highlight the limitations of relying solely on implicit state tracking.

#### 5.4 ReAct Analysis

ReAct improved local reasoning by interleaving thought and actions, enabling interaction with task structures. Nevertheless:

- Observations were sometimes assumed rather than derived,
- Global constraints were not consistently enforced, and
- State tracking relied on natural language traces, prone to degradation over long horizons.

#### 5.5 Model-First Reasoning Analysis

MFR demonstrated three notable advantages:

- **Explicit Constraint Grounding:** The constructed model reduced violations by providing a stable reference for reasoning.
- **Reduced Implicit Assumptions:** Clearly defined entities and actions limited the model’s tendency to fill in missing information.
- **Improved Structural Clarity:** Plans were more interpretable and verifiable due to the explicit representation of state and actions.

#### 5.6 Interpretation

These observations support the hypothesis that many LLM failures arise from representational rather than reasoning deficiencies. By separating modeling from reasoning, MFR externalizes the problem structure, reducing the reliance on internal latent representations and providing a form of soft symbolic grounding. This approach is particularly effective for constraint-heavy, long-horizon tasks where correctness and interpretability are critical.

#### 5.7 Limitations

- **Task Scope:** The benefits of MFR are most apparent in structured, constraint-driven planning tasks.

- **Token Overhead:** Constructing explicit models increases prompt and output length.
- **Model Dependence:** Effectiveness relies on the LLM accurately defining the problem model.
- **Not a Formal Verifier:** While MFR reduces risk, it does not guarantee correctness.

## 6 Discussion

The experimental results validate the hypothesis that many reasoning failures in LLM-based agents are due to incomplete or implicit problem representations rather than deficiencies in inference. By explicitly separating modeling from reasoning, Model-First Reasoning (MFR) provides a structured scaffold that constrains the solution space and reduces errors.

Key observations from our study include:

- **Representational Failures:** CoT and ReAct often produce fluent reasoning, yet hidden violations indicate that hallucinations are largely representational in nature.
- **Soft Symbolic Grounding:** MFR’s model construction acts as a form of symbolic grounding, translating natural language tasks into a structured framework that LLMs can reason over reliably.
- **Task Complexity Matters:** The benefits of MFR are most pronounced in high-constraint, multi-step planning tasks such as medical scheduling or resource allocation, where implicit assumptions can cascade into failures.
- **Reproducibility and Interpretability:** Explicit models provide outputs that are easier to inspect, verify, and debug, increasing trust in LLM-based planning systems.

While MFR increases prompt and output size, this trade-off is offset by significantly improved correctness and verifiability. Future work can explore methods to amortize modeling costs across task instances, potentially by reusing pre-defined models for recurring problem types.

## 7 Conclusion

We introduced **Model-First Reasoning** (MFR), a paradigm that separates problem modeling from reasoning in LLM-based agents. Through extensive experiments across multiple complex planning domains, we demonstrated that:

- Explicit model construction drastically reduces constraint violations and implicit assumptions.
- MFR improves global consistency and solution quality compared to Chain-of-Thought and ReAct strategies.
- Separating modeling and reasoning provides a soft symbolic grounding that addresses core representational failures in LLM planning tasks.

Our findings reframe hallucination and planning errors in LLMs as primarily representational rather than inferential. By making explicit modeling a foundational step, MFR enhances reliability, interpretability, and trustworthiness in AI agents performing structured, multi-step reasoning.

Reproducibility is facilitated by detailed descriptions of prompts, evaluation procedures, and task datasets provided in this paper. This work lays the foundation for further research in explicit LLM-based problem modeling and the development of robust, interpretable AI planning systems.

## References

- [1] Richard Fikes and Nils Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, pages 189–208, 1971.
- [2] Philip N. Johnson-Laird. *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness*. Harvard University Press, 1983. Classic work on mental models and human reasoning.
- [3] Drew McDermott, Malik Ghallab, Adele Howe, Craig Knoblock, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins. PDDL - The Planning Domain Definition Language. *Technical Report, AIPS-98 Planning Competition Committee*, 1998.
- [4] Mrinal Rawat, Ambuje Gupta, Rushil Goomer, Alessandro Di Bari, Neha Gupta, and Roberto Pieraccini. Pre-act: Multi-step planning and reasoning improves acting in llm agents. *arXiv preprint*, 2025.

- [5] Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. Towards understanding chain-of-thought prompting: An empirical study of what matters. *arXiv preprint*, 2022.
- [6] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint*, 2022.
- [7] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint*, 2022.