From Actions to Understanding: Conformal Interpretability of Temporal Concepts in LLM Agents

Anonymous Author(s)

Abstract

Large Language Models (LLMs) are increasingly deployed as autonomous agents capable of reasoning, planning, and acting within interactive environments. Despite their growing capability to perform multi-step reasoning and decision-making tasks, the internal mechanisms guiding their sequential behavior remain opaque. This paper presents a framework for interpreting the temporal evolution of concepts in LLM agents through a step-wise conformal lens. We introduce the *conformal interpretability framework for temporal tasks*, which combines step-wise reward modeling with conformal prediction to statistically label model's internal representation at each step as successful or failing. Linear probes are then trained on these representations to identify directions of temporal concepts—latent directions in the model's activation space that correspond to consistent notions of success, failure or reasoning drift. Experimental results on two simulated interactive environments, namely ScienceWorld and AlfWorld, demonstrate that these temporal concepts are linearly separable, revealing interpretable structures aligned with task success. We further show preliminary results on improving an LLM agent's performance by leveraging the proposed framework for steering the identified successful directions inside the model. The proposed approach, thus, offers a principled method for early failure detection as well as intervention in LLM-based agents, paving the path towards trustworthy autonomous language models in complex interactive settings.

CCS Concepts

• **Computing methodologies → Artificial intelligence**; **Knowledge representation and reasoning**; **Temporal reasoning**.

Keywords

Large Language Models (LLMs), Sequential Decision-Making, Temporal Interpretability, Conformal Prediction, Representation Analysis, Linear Probes, Steerabilility, Trustworthy Autonomy

ACM Reference Format:
Anonymous Author(s). 2025. From Actions to Understanding: Conformal Interpretability of Temporal Concepts in LLM Agents. In *Proceedings of 29th ACM International Conference on Hybrid Systems: Computation and Control, and 17th ACM/IEEE International Conference on Cyber-Physical Systems (HSCC/ICCPS)*. ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Large Language Models (LLMs) have rapidly evolved from static text generators [1, 35] to autonomous agents capable of navigating, planning, and acting in the physical world [15, 33, 38]. When deployed in embodied simulators such as Alfworld [31], and ScienceWorld [39], these agents exhibit impressive decision-making abilities on sequential goals for accomplishing the assigned task [20]. However, the reasoning process behind such sequential behavior remains largely opaque as these LLM agents continue to operate as *black box*, leaving practitioners without reliable explanations for *why* a model succeeds or *where* it fails within the task trajectory. This has led to increasing interest in the *interpretability* of LLMs, aimed at uncovering how models process inputs to generate outputs in a manner that is transparent and understandable to humans.

Traditional interpretability frameworks such as attribute or feature visualization [23, 41], mechanistic interpretability via sparse autoencoders [9], activation space analysis via representation engineering [48] and universal steering [6] offer valuable insights into mappings between *standalone input* and the model's latent space. These approaches are, however, fundamentally limited in capturing *temporal dynamics* within LLM-based agents as they operate on single standalone inputs such as a image, text or an image-text pair. While performing a sequential task in an interactive environment, the agent's decision at any step depends not only on the current observation but also on the entire trajectory history with prior reasoning traces, environment responses, and accumulated context. Model's internal representation space evolves through time, encoding both correct and incorrect reasoning directions. Understanding these evolving representations requires a temporary-aware interpretability framework interpretability.

A straightforward approach to adapting the existing interpretability frameworks for temporal dynamics would be treating the entire trajectory as a single input and performing analysis on those. This approach, however, does not provide any granularity on *which intermediate step* of the agent contributed to success or failure of the assigned task. An agent that successfully completes the task after ten steps hides the possibility that six of those might be suboptimal or irrelevant. Without a step-level notion of "success", it becomes impossible to pinpoint when the model's internal dynamics shift from aligned reasoning to failure modes such as hallucination or irrelevant exploration.

A motivating example of the Llama-2 [35] agent trained to perform household tasks in the AlfWorld environment is as follows. For the assigned task of *"placing a roll of paper next to the toilet"*, we observed that the agent starts on the right track by moving towards the toilet and searching shelves to locate the roll. Yet midway through the trajectory, its internal reasoning begins to drift. Instead of continuing the search, the agent hallucinates that a non-existent drawer contains the paper roll, issuing an invalid action sequence. This behavioral deviation—despite earlier correct reasoning—leads

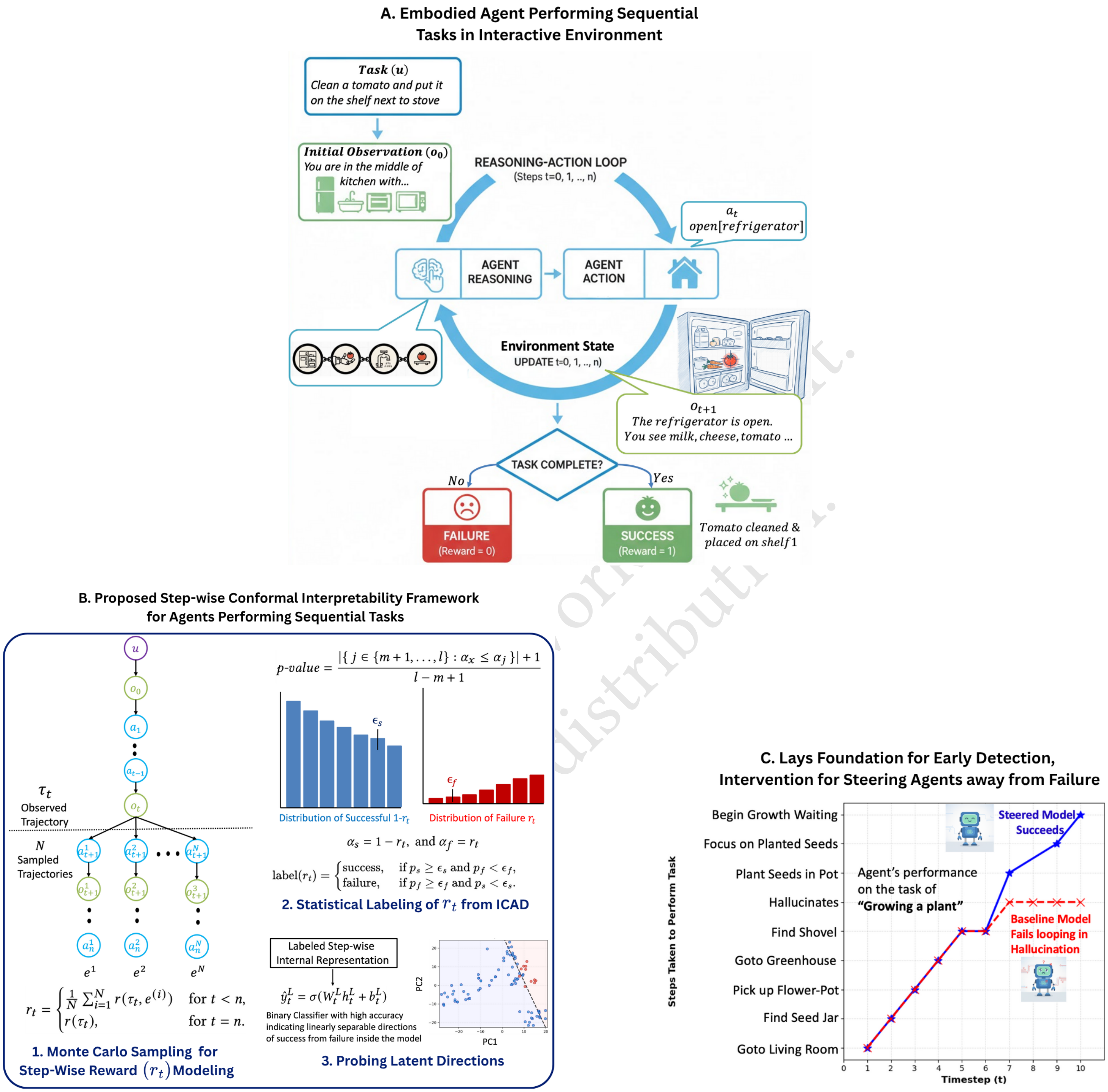

A. Embodied Agent Performing Sequential Tasks in Interactive Environment
Task (u)
Clean a tomato and put it on the shelf next to stove
Initial Observation (o_0)
You are in the middle of kitchen with...
REASONING-ACTION LOOP
(Steps t=0, 1, .., n)
a_t
$open[refrigerator]$
AGENT REASONING
AGENT ACTION
Environment State
UPDATE t=0, 1, .., n)
o_{t+1}
The refrigerator is open. You see milk, cheese, tomato ...
TASK COMPLETE?
No
Yes
FAILURE
(Reward = 0)
SUCCESS
(Reward = 1)
Tomato cleaned & placed on shelf 1
B. Proposed Step-wise Conformal Interpretability Framework for Agents Performing Sequential Tasks
u
o_0
a_1
a_{t-1}
o_t
τ_t
Observed Trajectory
N
Sampled Trajectories
a^1_{t+1}
a^2_{t+1}
a^N_{t+1}
o^1_{t+1}
o^2_{t+1}
o^3_{t+1}
a^1_n
a^2_n
a^N_n
e^1
e^2
e^N
$r_t = \begin{cases} \frac{1}{N}\sum_{i=1}^{N} r(\tau_t, e^{(i)}) & \text{for } t < n, \\ r(\tau_t), & \text{for } t = n. \end{cases}$
1. Monte Carlo Sampling for Step-Wise Reward (r_t) Modeling
$p\text{-}value = \frac{|\{j \in \{m+1, \ldots, l\} : \alpha_x \le \alpha_j\}| + 1}{l - m + 1}$
ϵ_s
ϵ_f
Distribution of Successful 1-r_t
Distribution of Failure r_t
$\alpha_s = 1 - r_t$, and $\alpha_f = r_t$
$\text{label}(r_t) = \begin{cases} \text{success}, & \text{if } p_s \ge \epsilon_s \text{ and } p_f < \epsilon_f, \\ \text{failure}, & \text{if } p_f \ge \epsilon_f \text{ and } p_s < \epsilon_s. \end{cases}$
2. Statistical Labeling of r_t from ICAD
Labeled Step-wise Internal Representation
$\hat{y}^L_t = \sigma(W^L_t h^L_t + b^L_t)$
Binary Classifier with high accuracy indicating linearly separable directions of success from failure inside the model
PC2
PC1
20
10
0
−10
−20
−20
−10
0
10
20
3. Probing Latent Directions
C. Lays Foundation for Early Detection, Intervention for Steering Agents away from Failure
Steps Taken to Perform Task
Begin Growth Waiting
Focus on Planted Seeds
Plant Seeds in Pot
Hallucinates
Find Shovel
Goto Greenhouse
Pick up Flower-Pot
Find Seed Jar
Goto Living Room
Timestep (t)
0
1
2
3
4
5
6
7
8
9
10
Steered Model Succeeds
Agent's performance on the task of "Growing a plant"
Baseline Model Fails looping in Hallucination

Figure 1: A. We consider the problem of temporal interpretability of LLM agents trained to perform sequential tasks in complex interactive environments, B. The proposed framework on step-wise reward modeling combined with conformal labeling to distinguish success and failure at each timestep. Linear probes are then trained on the corresponding model's internal representations to test the hypothesis of linearly separable directions of these step-wise notions of success and failure inside the model. C. This framework establishes the foundation for real-time monitoring, early detection, and targeted intervention prior to task failure. For e.g., it enables steering the model back toward identified success directions when its internal state begins to drift toward failure trajectories.

to task failure, even though the initial steps were coherent and purposeful.

Such examples (more shown in fig. 4) highlight a central challenge in *temporal interpretability*: the final success or failure of a task often obscures *where* in the sequence the agent's internal representations began to diverge. A trajectory that ultimately fails may contain multiple locally successful steps, while an apparently successful run may involve accidental recoveries after erroneous decisions. Current interpretability methods—focused on isolated prompts, static embeddings, or single-step explanations—cannot capture these evolving internal transitions, nor can they identify *when* the model's reasoning begins to deviate.

This work introduces a *step-wise conformal interpretability framework* that transforms the problem of agent's final evaluation on a multi-step sequential task into *temporal representation analysis*. We posit that the *directions of success and failure are geometrically separable* in an LLM fine-tuned to perform sequential tasks, .i.e, these models internally have separable notions of success and failure. To test this hypothesis, we build the proposed interpretability framework on the following three key components (Fig.1(B)):

(1) **Step-Wise Reward Modeling:** We follow Xiong et al. [42]'s approach on generating fine-grained step-wise rewards using Monte Carlo sampling over future trajectories. This transforms sparse final rewards into dense temporal feedback signals on success (or failure) of every step.
(2) **Statistical Labeling via Conformal Prediction:** We propose using Inductive Conformal Prediction (ICP) to label model's internal representation at each step as successful or failing with provable confidence bounds.
(3) **Probing Latent Directions:** We train classifiers (or linear probes) on layer and time-conditioned agent's activations to distinguish the latent space of success from failure.

Key Findings. High accuracy and F1-scores of linear probes on Llama2-7B in two complex simulated interactive environments, namely ScienceWorld [39] and AlfWorld [31], validate our hypothesis that an LLM fine-tuned for sequential tasks develops an internal notion of linearly separable step-wise success across (a) timesteps, (b) layers, and (c) domains—both in-distribution and out-of-distribution.

Steering the model towards Success. As observed in prior work [29, 40] and validated in our experiments, these agents lack the intrinsic ability to self-correct themselves back towards the success directions and often continue drifting along failure trajectories once the deviation occurs. The proposed framework lays the foundation for *steering* LLM agents towards right direction by enabling targeted interventions when early signs of hallucination or misalignment emerge in their internal representations. We conduct preliminary experiments on steering the model towards its (identified) internal 'success' directions and observe that the accuracy of the steered model improves from the baseline model, providing evidence on the practical use of the proposed framework.

2 Related Work

LLM Agents for Interactive Embodied Environment. LLMs have evolved beyond text generation to function as powerful policy models for decision-making in interactive environments. Early systems like WebGPT [26] and SimpleTOD [12] used human feedback or annotated dialogues for interactive learning, while ReAct [44] demonstrated a more scalable approach by integrating natural language reasoning directly into the decision loop, removing human from the loop. Building on this paradigm, works such as IPR [42], SayCan [2] and Inner Monologue [14] extend LLM-based decision-making to embodied and interactive domains, where agents reason over both language and environmental feedback. We perform interpretability analysis on LLMs fine-tuned using ReACT style to autonomously perform sequential tasks in interactive environments.

Interpretability of Large Language Models (LLMs). There has been growing interest in interpreting the decision-making processes of LLMs. A variety of techniques—such as attribution methods that map a model's output to specific input tokens or features using saliency maps [19], or masking-based perturbation approaches [5]—have been proposed to uncover input–output relationships. Other lines of work focus on post-hoc interpretability, where the model itself generates natural language explanations for its outputs [13], or exposes its intermediate reasoning through chain-of-thought traces [24]. While these post-hoc methods apply broadly across both proprietary and open-source models, they often fail to reveal the true internal mechanisms driving the model's behavior. In contrast, mechanistic interpretability [9] seeks to dissect model internals by mapping individual neurons or sub-networks to human-understandable concepts. More scalable variants extend this idea to analyzing distributed activation patterns associated with such concepts [6, 48]. However, these existing approaches remain largely static—linking standalone inputs to interpretable concepts—without accounting for the temporal dynamics of concept evolution that arise when models reason and act over sequential contexts. To our knowledge, time-series interpretability has been applied to LLMs analyzing patterns in temporal data for domains such as weather, finance, health [37, 47], and not in the interactive agentic settings that we consider in this paper.

Use of Probes for Interpretability Analysis. Research on probes in LLMs has developed into a major branch of linguistic and interpretability analysis. Early studies [8, 11], used linear probes to show that intermediate layers of models like BERT [34] encode rich linguistic structures, including parts of speech, syntactic trees, and semantic roles. Foundational work by Alain and Bengio [3] introduced the concept of auxiliary classifiers as probes, inspiring subsequent analysis of internal representations in neural networks. More recently, Marks and Tegmark [25] used linear probes to show that truthful answers to factual statements are represented as a distinct, approximately linear direction in a model's activation space. Building on this line of research, this paper also uses probes as linear classifiers on the hidden representations of an LLM agent to examine whether the model encodes a distinct temporal notion of success and failure while performing sequential tasks. Specifically, similar to existing work [25], we also use the accuracy of these linear probes as the measure of distinct (or separable) directions of step-wise success from failure inside the model.

Conformal Prediction in Explainable AI. Conformal Prediction [4] (CP) provide statistically rigorous uncertainty quantification by producing calibrated prediction intervals or sets with guaranteed coverage, thereby complementing the often opaque confidence estimates of neural models. There is a growing interest in exploring CP for assessing explainability and safety in LLM's decision-making. For instance, Doe et al. [10] highlight CP as a principled way to quantify epistemic uncertainty in generative outputs, while Li et al. [18] extend conformal techniques for out-of-distribution detection in domain-specialized LLMs. Other works, such as Patel et al. [28], leverage CP to optimize LLM decision-making pipelines by pruning low-confidence branches without sacrificing coverage guarantees. In safety-critical domains, Nguyen et al. [27] apply conformal prediction to LLM-based autonomous navigation, providing certified safety bounds for trajectory selection. Moreover, Zhao et al. [46] use interval-based conformal evaluation to assess model reliability in subjective judgment tasks, and Liu et al. [22] introduce formal methods for applying CP to text generation, enhancing interpretability through calibrated likelihood intervals. We propose a novel use of CP framework for labeling temporal representations of the LLM agent as success or failure on the assigned task with bounded error on making labeling errors.

3 Background

3.1 Performing Sequential Tasks in Interactive Environment

Problem Setting: As shown in Fig. 1 (A), we consider a general class of sequential decision-making tasks for LLM agents in an interactive environment. Specifically, an LLM interacts with the environment by receiving textual observations in response to each action performed by the model. The goal of the agent is to execute a coherent sequence of steps that lead to the successful task completion while reasoning and acting at each step through natural language.

Task Formulation: Formally, the task can be represented as a partially observable Markov decision process, described by the tuple (U, S, O, A, T, R) where U represents the space of natural language task instructions, S is the set of environment's states and O is the corresponding observation space providing textual information on the environment's state, and A denotes the discrete action space defined in natural language. The transition dynamics $T : S \times A \longrightarrow S$ govern how the environment's state evolves after an action, and the reward R provides scalar feedback indicating the overall task performance of the agent.

Task Execution: At the start of each episode, the agent is provided with a task instruction $u \in U$ and an initial observation o_0 that describes the environment's initial state s_0. At any discrete time step t, the agent takes an action according to its learned policy π_θ:

$$a_t \sim \pi_\theta(\cdot \mid \tau_{t-1}).$$

Here $\tau_{t-1} = (u, s_0, a_0, \ldots, a_{t-2}, s_{t-1})$ is the episode's history till the previous time-step. Upon executing a_t, the environment transitions to a new state $s_t = T(s_{t-1}, a_t)$ and produces the corresponding observation o_t for the LLM. This iterative process continues until the task is successfully completed or the maximum number of steps is reached. At the end of the episode, the environment provides a scalar reward summarizing the overall task performance of the agent.

Example: Let us consider the following task for a household LLM agent:

$$u = \text{“\textit{clean a tomato and put it on the shelf next to the stove.}”}$$

with the initial observation of the environment as:

$$o_0 = \text{“\textit{You are in the middle of a kitchen. You observe a refrigerator,}}$$
$$\text{\textit{washbasin, microwave, shelf1 next to stove, shelf2 next to basin.}”}$$

To complete this task, the agent must sample a sequence of subtasks from its learned policy such as opening the refrigerator to find the tomato, washing it in the washbasin, and then placing it on the shelf1 as it is next to stove and not shelf2 which is next to basin. After agent's every action, the environment sends its state as textual observation to the agent. For instance, after

$$a_t = \textit{open[refrigerator]},$$

the environment may return this observation on its current state:

$$o_{t+1} = \text{“\textit{The refrigerator is now open. You see milk, tomato, cheese}}$$
$$\text{\textit{and carrots.}”}$$

This iterative action–observation loop continues until the task is completed or the step limit is reached, with the correctness of each action contributing to the overall success. At the end of the episode, the environment provides a scalar reward summarizing the agent's task performance, e.g., a value of 1 if the tomato is washed and correctly placed.

3.2 Supervised Fine-Tuning of LLM Agents

To equip a large language model (LLM) with core agentic capabilities, we perform *supervised fine-tuning* (SFT) on expert demonstrations, aligning the model's policy with trajectories that exemplify correct reasoning, decision-making, and action execution in interactive environments [42].

Expert Demonstrations and the ReAct Paradigm: In agentic settings, the model must not only produce the correct action but also reason about why that action is appropriate given the current context. To capture this reasoning–action interplay, we adopt the *ReAct* (Reasoning and Acting) format [44] of the expert demonstrations for SFT. Here, each step taken by the agent consists of a natural language reasoning trace followed by an executable action. An example on how the ReAct step looks like is as follows:

```
Thought: I need to open the refrigerator to check for
tomato.
Action: open[refrigerator].
```

Such paired data explicitly teach the model how to alternate between reflective reasoning (`Thought`) and executable (`Action`) in the environment, promoting explainability in its downstream behavior.

Expert Trajectory Dataset Construction: Let $ET = \{(u^{(i)}, \tau^{(i)})\}_{i=1}^{|ET|}$ denote a collection of Expert Trajectories, where $u^{(i)}$ is the natural language task instruction and $\tau^{(i)} = (s_o, a_0, s_1, \ldots, a_{final}, s_{final})$ is the sequence of actions and corresponding states (expressed as observations in natural language) from the expert's interaction with the environment. Each trajectory is annotated in ReAct form, providing the reasoning text before each action. These demonstrations can originate from human experts, high-performing teacher models such as GPT [1], or curated datasets collected through scripted interaction with simulation environments.

Training Objective: During SFT, the LLM learns to imitate the expert's behavior by maximizing the likelihood of the expert trajectory given the task instruction:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{(u,\tau)\sim ET}\big[\log \pi_\theta(\tau \mid u)\big],$$

Here, θ is the set of model parameters. In practice, the joint probability of the entire trajectory can be decomposed into a sequence of conditional action probabilities:

$$\pi_\theta(\tau \mid u) = \prod_{t=1}^{final} \pi_\theta(a_t \mid u, s_o, a_0, \ldots, s_{t-1}),$$

which leads to a token-level autoregressive training objective:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{(u,\tau)\sim ET}\Big[\sum_{t=1}^{final} \log \pi_\theta(a_t \mid u, \tau_{t-1})\Big],$$

where τ_{t-1} denotes the trajectory history up to step $t-1$. This objective encourages the agent to reproduce the expert's next action at each step, conditioned on the task and its full prior context on its previous steps and the environment's intermediate observations.

3.3 Conformal Prediction

Conformal prediction [4] is a statistical framework for quantifying how well a new input sample aligns with a reference data distribution. At its core, the framework relies on a *non-conformity measure* (NCM), a real-valued function that quantifies the extent to which an input deviates from the behavior observed in the reference data. Given a dataset $X = \{x_1, x_2, \ldots, x_l\}$ drawn i.i.d. from an underlying data distribution $\mathcal{D}$ of interest, a non-conformity score α_x is assigned to the input x by the NCM defined on $X \cup \{x\}$. Larger value of α_x indicate greater deviation from $\mathcal{D}$ and, consequently, a higher likelihood that the sample is atypical.

Classical Conformal Anomaly Detection: Conformal Anomaly Detection (CAD) utilizes the non-conformity score to assess the likelihood that an unseen input belonging to the same distribution as $\mathcal{D}$. This is done by computing p-value of the input x by comparing its non-conformity score α_x with those of the datapoints in X from the NCM defined on the new set $X \cup \{x\}$:

$$p\text{-}value = \frac{|\{\, i \in \{1, \ldots, l\} : \alpha_x \leq \alpha_i \,\}| + 1}{l+1}. \tag{1}$$

If x is sampled from $\mathcal{D}$, its p-value will tend to be large; conversely, inputs exhibiting substantial deviation from $\mathcal{D}$ yield smaller p-values. An input is deemed anomalous when $p\text{-}value < \epsilon$, where $\epsilon \in (0,1)$ denotes a user-defined significance level controlling the allowable false-alarm probability.

Inductive Conformal Anomaly Detection [36]: While the classical formulation in (1) is statistically sound, recomputing NCM scores across the entire set X for every test input is computationally expensive. The *Inductive Conformal Anomaly Detection* (ICAD) framework mitigates this cost by partitioning the data into a proper training subset $X_{\text{tr}} = \{x_1, \ldots, x_m\}$ and a calibration subset $X_{\text{cal}} = \{x_{m+1}, \ldots, x_l\}$. The NCM is defined on X_{tr} and then evaluated for each calibration datapoint to produce the set of calibration scores $\{\alpha_j\}_{j=m+1}^{l}$. For a new input x, its non-conformity is assessed by comparing its score α_x against the calibration scores:

$$p\text{-}value = \frac{|\{\, j \in \{m+1, \ldots, l\} : \alpha_x \leq \alpha_j \,\}| + 1}{l-m+1}, \tag{2}$$

and, again, detecting x as anomalous if the computed $p\text{-}value < \epsilon$.

This inductive formulation permits all calibration scores to be pre-computed offline, thereby enabling efficient inference while retaining the theoretical validity of the conformal framework.

Statistical Validity and Error Control [36]: Under the standard i.i.d. assumption that both calibration and test samples are drawn from the same distribution $\mathcal{D}$, the p-values obtained via (2) are uniformly distributed in the interval $(0,1)$. Consequently, the probability of a false alarm—incorrectly identifying an in-distribution sample as anomalous—is provably bounded by the significance threshold ϵ:

$$\Pr[p(x \in \mathcal{D}) < \epsilon] \leq \epsilon. \tag{3}$$

The efficacy of the conformal framework depends on the underlying NCM. A variety of NCMs have been proposed in literature, employing methods such as k-nearest neighbors [36], variational autoencoders [7], memory prototypes [43], and transformation equivariance [16, 17]. We propose using the step-wise reward to define the NCM for labeling each step as sampled from the distribution of successful or failing step.

4 Identifying Directions for Temporal Concepts in LLM Agents

For an LLM agent trained to perform a multi-step sequential task, we hypothesize that the directions of success and failure for the task become linearly separable in the agent's internal representation space across layers and time. We aim to validate this hypothesis by training linear probes to classify the agent's activation space as success or failure at each step of the task.

The notion of success (or failure) is quantified by the *step-wise reward* assigned to the agent at each sequential step taken to accomplish the task. Here, we provide details on a) generating these step-wise rewards via Monte-Carlo sampling of the agent's trajectory from its learned policy on performing these tasks, b) leveraging inductive conformal anomaly detection (ICAD) framework for labeling these step-wise rewards as success (or failure) with bounded probability on making labeling errors, and c) training linear probes to classify the agent's internal activation space at each step as success or failure from the labeled step-wise reward.

4.1 Generating Step-Wise Rewards

Traditional learning based approaches rely solely on the *final reward* r assigned to the agent's complete trajectory on its ability to complete the task. This obscures the contribution of individual steps towards the goal. To achieve granular interpretability, we propose to use *step-wise rewards* $\{r_t\}_{t=1}^{T}$, where r_t represents the quantitative measure of success for the agent's partial trajectory τ_t till time t:

$$\tau_t = (s_0, a_0, s_1, a_1, \ldots, s_{t-1}, a_{t-1}, s_t).$$

To estimate r_t, we follow Xiong et al. [42]'s approach on performing Monte Carlo sampling on the agent's action space from its learned policy π_θ conditioned on the observed trajectory. Specifically, given τ_t, we generate N complete, subsequent expected trajectories

$$e^{(i)} = (a^i_{t+1}, s^i_{t+1}, \dots a^i_{final}, s^i_{final}),$$

by performing iterative Monte Carlo sampling on the agents action space starting from a_t till a^i_{final}.

The step-level reward r_t at time t is calculated as:

$$r_t = \begin{cases} \frac{1}{N}\sum_{i=1}^{N} r(\tau_t, e^{(i)}) & \text{for } t < n, \\ r(\tau_t), & \text{for } t = n. \end{cases}$$

This procedure allows the model to evaluate the *expected future success probability* measured in terms of step-wise rewards that are conditioned on its current state. The use of Monte Carlo sampling transforms interpretability into a probabilistic, forward-looking signal, better aligning with how agents internally (intends to) plan the task execution.

4.2 Labeling Step-Wise Rewards

Once step-wise rewards are estimated, the key challenge is determining *when a reward should be interpreted as success or failure.* We introduce the **Conformal Prediction–based labeling mechanism** to statistically calibrate both success and failure labels with bounded guarantees on making the labeling error.

Given the calibration set of step-wise rewards for both successful and failure steps, we can calculate the (non-)conformity of r_t w.r.t success as well as failure. This gives us two p-values for the step: one corresponding to the likelihood of it being a successful step (p_s), and the other corresponding to the likelihood of it being a failure step (p_f). With the intuition of higher rewards for successful steps than failure, we propose

$$\alpha_s = 1 - r_t, \text{ and } \alpha_f = r_t.$$

as the non-conformity score with respect to the successful and failure steps respectively. r_t is then labeled as:

$$\text{label}(r_t) = \begin{cases} \text{success}, & \text{if } p_s \geq \epsilon_s \text{ and } p_f < \epsilon_f, \\ \text{failure}, & \text{if } p_f \geq \epsilon_f \text{ and } p_s < \epsilon_s. \end{cases} \quad (4)$$

Theorem 4.1 (Bounded Guarantees on making Labeling Errors). *The probability of labeling a successful step-wise reward as failure (or False Negative Rate) is strictly bounded by ϵ_s, and the probability of labeling a failure step-wise reward as success (or False Positive Rate) is strictly bounded by ϵ_f.*

Proof. The proof without strict bounds, i.e. without the 'and' (intersection) conditions in (4):

$$\text{label}(r_t) = \begin{cases} \text{success}, & \text{if } p_s \geq \epsilon_s, \\ \text{failure}, & \text{if } p_f \geq \epsilon_f. \end{cases}$$

follows directly from the statistical guarantees of the inductive conformal anomaly detection (ICAD) framework (3). In other words, the probability of p-value for successful r_t less than ϵ_s is bounded by ϵ_s. The second condition $p_f < \epsilon_f$ is an intersection, which reduces the overall probability. Therefore, the false negative rate is strictly bounded by ϵ_s. Similarly, the probability of p-value for failure r_t less than ϵ_f is bounded by ϵ_f. The second condition $p_s < \epsilon_s$ is an intersection, which reduces the overall probability. Therefore, the false positive rate is strictly bounded by ϵ_f.

□

4.3 Representation Probing Across Timesteps

At each time step t, the model's hidden representation captures contextualized knowledge of its decisions till time t. We define the internal state $\mathbf{h}^L_t$ of an LLM agent as its residual stream activations at a specific layer L and at the last token position corresponding to the trajectory τ_t till time t. This state $\mathbf{h}^L_t$ is the object of our interpretability study.

The challenge is to map these latent representations to an interpretable success/failure signal at every timestep. Having obtained calibrated step-wise reward labels, we investigate whether the corresponding model's hidden representations for success and failure are linearly separable. For this, we then train **linear probes** P^L_t to classify h^L_t as success vs. failure:

$$\hat{y}^L_t = \sigma(W^L_t h^L_t + b^L_t),$$

where W^L_t and b^L_t are the (weight and bias) probe parameters.

Classification accuracy of probe provides a quantitative measure of how distinctly the model encodes success and failure trajectories within its internal representation space [3, 25].

5 Experimental Results

5.1 Case Study I: ScienceWorld

Our first case study is on sequential tasks in ScienceWorld [39], a large-scale text-based environment for evaluating an LLM agent's ability to perform scientific reasoning and procedural tasks.

Environment: Each instance of the environment represents a small virtual world inspired by elementary science domains—such as physics, chemistry, and biology—where the agent must explore, manipulate, and reason about objects to accomplish experiment-style goals. The environment comprises interconnected rooms (e.g., greenhouse, laboratory, workshop), each populated with interactive objects that support diverse affordances. Agents operate entirely through natural-language commands and receive textual feedback describing environmental state changes. Fig. 2 illustrates a representative task, "Testing Conductivity", along with the corresponding action-observation trajectory. Tasks in ScienceWorld are typically long-horizon and a normalized reward in $[0, 1]$, which is assigned at the end of each episode to reflect the agent's overall task performance. Specifically, each task is decomposed into multiple sub-goals, and the final reward is computed based on how many of these sub-goals the agent achieves, thereby enabling a fine-grained evaluation of procedural and scientific reasoning capabilities. Examples of these sub-goals are illustrated in Fig. 4, where the preferred sequence of steps is mentioned in the task description.

LLM Agent: We train the `Llama-2-7B` model [35] on 60% (889 out of 1443) of the training trajectories. The agent is trained using Supervised Fine-Tuning (SFT) on a curated dataset of expert trajectories formatted in the ReAct paradigm [44], which explicitly interleaves natural language `Thought` and executable `Action` steps. The remaining 40% is split equally between the calibration

Figure 2: An example of a successful trajectory for the "Test Conductivity" task in `ScienceWorld`, demonstrating the sequential reasoning required by the agent to accomplish the task.

Step	Agent's Action	Environment Observation (Simplified)
1	`go to tool room`	You are in the tool room. You see a light bulb and a battery.
2	`pick up light bulb`	You picked up the light bulb.
3	`pick up battery`	You picked up the battery.
4	`go to workshop`	You are in the workshop. You see a metal fork and a plastic cup.
5	`pick up metal fork`	You picked up the metal fork.
6	`use battery on light bulb`	The battery and the light bulb are now connected in a circuit.
7	`use metal fork on light bulb`	The light bulb illuminates! The metal fork is electrically conductive.

set for conformal labeling of step-wise rewards and the training set of probes on residual stream activations of all 32 layers on the last token of the entire trajectory till time t. We observe that the trained agent is mostly able to successfully complete the assigned task within 10 timesteps.

Testing Scenarios: The environment offers two types of test scenarios: seen and unseen. Seen test set comprises of those tasks (or variations of tasks) that the agent encountered and learned from during its training phase. We refer to this test set as *in-distribution* because it falls within the scope of the data the model was exposed to. Unseen test set comprises of those tasks that the agent never encountered during its training. These involve novel combinations of objects, new environmental layouts, or even entirely new scientific concepts. We refer to this test set as *out-of-distribution* (OOD). We test on the entire set of 360 in-distribution and 165 OOD tasks.

Conformal Thresholds: We set $\epsilon_s = \epsilon_f = 0.1$ for labeling step-wise rewards as success or failure all timesteps. This strictly bounds both the false negative and false positive labeling errors to 10%.

Results and Observations: Tables 1 and 2 show accuracy on iD and OOD test set respectively from timesteps $t = 2$ to 10. At timestep 1, the model is given instructions on its role in the ScienceWorld environment and it always (irrespective of success or failure) responds with an 'OK'. Probes achieve significantly high accuracy with upto 100% in most test cases for iD set and good accuracy in most test cases for OOD set except for one test case (50% at $t = 10$ for layer 8).

Tables 3 and 4 show F1 scores on the in-distribution and OOD test sets, respectively. Similar to results on accuracy, F1 scores are also high on most test cases of the iD set except for one test case (0.67 at $t = 10$ for layer 8). These scores are also high on most test cases of the OOD set except for three cases: 0.67 at $t = 3$ for layers 24 and 32 and for layer 8 at $t = 8$. We observe similar results across all layers of the model, and selected early (layer 8), middle (16 and 24) and later (32) layers to show the results.

These results validate our hypothesis that directions of success and failure become separable in an LLM fine-tuned to perform sequential tasks.

5.2 Case Study II: AlfWorld

The second case study evaluates our conformal interpretability framework on the *ALFWorld* environment [31]. AlfWorld serves as a widely adopted benchmark for autonomous LLM agents in interactive embodied settings, requiring complex sequential reasoning and navigation in simulated household environment [21, 45].

Environment: ALFWorld is a text-based environment that grounds language instructions in a physical, simulated world using the ALFRED dataset's [30] household tasks. The tasks require the agent to complete multi-step goals, such as fetching, cleaning, heating, or putting away objects (e.g., "put a clean mug in the cabinet"). These tasks necessitate robust sequential decision-making, long-term planning, and interaction with various objects and room states. The environment provides textual observations of the agent's surroundings, and the agent responds with text-based actions (e.g., `go to kitchen`, `pick up mug`). Fig. 3 shows an example trajectory of the agent-environment interaction on one of the household's task of "cleaning a tomato and putting it on the shelf next to stove". The agent's performance is judged based on successful completion of the final goal.

LLM Agent and Training: Similar to ScienceWorld, here also we train Llama-2-7B [35] on 60% (1710 out of 2851) of the training trajectories and split the remaining equally between calibration set and training the probes on all layers and timesteps.

Conformal Thresholds: Again, we set $\epsilon_s = \epsilon_f = 0.1$ for all timesteps $t \in \{1, 2, \ldots, 10\}$, strcitly bounding both false negative and false positive labeling errors to 10%.

Results and Discussion: Tables 5 and 6 show the accuracy and F1 scores of the trained probes on the test set of the AlfWorld, respectively. The accuracy varies from 60% (layer 24 at $t = 4$ and layer 8 at $t = 6$) to 95% (at $t = 2$), and the F1 score varies from 0.56 (layer 24 at $t = 4$ to 0.95 (at $t = 2$). Although these results are comparable to the probe accuracies reported for distinguishing truth and falsehood directions in factual question settings [25], they are lower than our results on ScienceWorld. We hypothesize that this difference arises because ALFWorld provides only a single final reward upon task completion, whereas ScienceWorld offers intermediate rewards for sub-goals at multiple steps. This denser reward structure in ScienceWorld yields more precise step-wise feedback, leading to improved predictive performance.

5.3 Steerability of Models Towards Success

We also perform preliminary experiments on steering the LLM agent towards successful directions early on in its task trajectory

Layer	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9	t=10
8	100	100	100	100	100	83	91	91	100
16	100	100	100	100	100	92	100	91	100
24	100	93	100	100	100	92	100	91	100
32	100	93	100	100	100	92	91	91	100

Table 1: Accuracy(%) of Linear Probes on in-distribution Test Set of ScienceWorld across Layers and Timesteps.

Layer	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9	t=10
8	100	92	100	94	92	80	75	75	50
16	100	92	93	94	92	80	75	75	100
24	100	92	100	94	92	73	75	75	100
32	100	92	100	94	92	73	75	75	100

Table 2: Accuracy(%) of Linear Probes on OOD Test Set of ScienceWorld across Layers and Timesteps.

Layer	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9	t=10
8	1.00	0.95	1.00	0.97	0.96	0.88	0.86	0.80	0.67
16	1.00	0.95	0.96	0.97	0.96	0.88	0.86	0.80	1.00
24	1.00	0.95	1.00	0.97	0.96	0.83	0.86	0.80	1.00
32	1.00	0.95	1.00	0.97	0.96	0.83	0.86	0.80	1.00

Table 3: F1 Score of Linear Probes on in-distribution Test Set of ScienceWorld across Layers and Timesteps.

Layer	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9	t=10
8	1.00	1.00	1.00	1.00	1.00	1.00	0.67	0.86	0.80
16	1.00	1.00	1.00	1.00	1.00	0.80	1.00	0.80	1.00
24	1.00	0.67	1.00	1.00	1.00	0.80	1.00	0.80	1.00
32	1.00	0.67	1.00	1.00	1.00	0.80	0.86	0.80	1.00

Table 4: F1 Score of Linear Probes on OOD Test Set across of ScienceWorld Layers and Timesteps.

for ScienceWorld. Using the labeled internal representations from the proposed framework, we leverage an existing steering approach on *representation engineering (RepE)* [48]. The approach computes contrastive activation directions, i.e. differences in internal activations between preferred (*success in our case*) and undesired (*failure in our cases*) model responses, and adds scaled versions of these contrastive activation directions during inference. The scale or the steering coefficient determines the amount of intervention to be performed on the model. This linear intervention steers the model's internal representations toward target attributes (e.g., success in our case), achieving controllable behavior shifts while preserving general performance.

With the idea of performing early interventions, we steer the supervised fine-tuned (SFT) Llama-2-7B agent on timestep three

Figure 3: An example of a successful trajectory for the “Cleaning a tomato and putting it on the shelf next to stove” task in the ALFWorld environment, demonstrating the multi-step nature of a typical sequential task in the environment requiring commonsense, navigation, and object manipulation.

Step	Agent’s Action	Environment Observation (Simplified)
1	`open refrigerator`	The refrigerator is now open. You see milk, tomato, cheese, and carrots.
2	`take tomato from refrigerator`	You are now holding the tomato.
3	`go to washbasin`	You are at the washbasin. You see running water.
4	`wash tomato in washbasin`	The tomato is now clean.
5	`go to shelf1 (next to stove)`	You are near the stove. You see shelf1 next to stove.
6	`put tomato on shelf1`	The tomato has been placed on shelf1 next to the stove.

Layer	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9	t=10
8	95	85	75	75	60	80	75	70	75
16	95	90	65	65	75	65	80	80	75
24	95	90	60	85	80	60	80	75	75
32	95	85	75	85	80	75	90	80	75

Table 5: Accuracy (%) of Linear Probes on ALFWorld Test Set across Layers and Timesteps.

Layer	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9	t=10
8	0.95	0.82	0.71	0.76	0.64	0.82	0.76	0.62	0.71
16	0.95	0.89	0.59	0.63	0.74	0.63	0.78	0.78	0.67
24	0.95	0.89	0.56	0.84	0.80	0.60	0.80	0.71	0.67
32	0.95	0.82	0.74	0.86	0.80	0.74	0.89	0.78	0.67

Table 6: F1 Score of Linear Probes on ALFWorld Test Set across Layers and Timesteps.

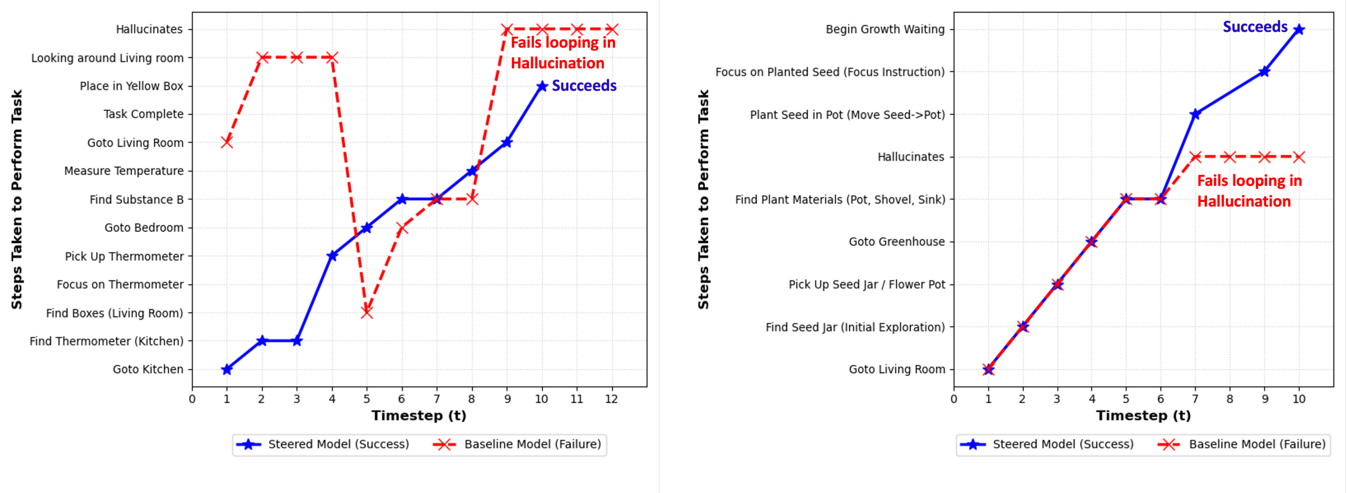
with a low steering coefficient of 0.025. This results in 1.1% boost in the accuracy of the base model. While the gain may appear small, it is notable given that significantly more expensive techniques—such as Best-of-N reward sampling on the SFT agent, Rejection Sampling Fine-Tuning (RFT) on the SFT agent, and Direct Preference Optimization (DPO) Fine-Tuning on the SFT agent —achieve only 2.8%, 4.2%, and 6.8% boosts respectively on the same test settings (Llama-2-7B model and the test set) for ScienceWorld [32]. Fig. 4 shows examples of two test cases where the steered model is able to rectify two commonly observed mistakes by the agent in ScienceWorld: (a) not focusing on the correct sequence of sub-goals, and (b) going off track in the middle of the trajectory.

The reported steering results are preliminary as we applied an off-the-shelf steering approach (RepE) only at timestep 3. RepE has been proposed to steer standalone inputs. Leveraging the proposed framework for developing and applying steering approaches for temporal data across steps is one of the future directions.

6 Conclusion

This work introduces a conformal, time-series–based interpretability framework for analyzing the evolving internal representations of large language model (LLM) agents performing sequential tasks. By integrating step-wise reward modeling with inductive conformal prediction, the framework provides statistically bounded success and failure labels at each timestep, enabling fine-grained temporal analysis of model behavior. Linear probing of hidden activations across layers and timesteps reveals that success and failure directions are linearly separable, validating the hypothesis that LLMs fine-tuned for sequential reasoning implicitly encode step-wise notions of success within their representation space. Empirical evaluations on two complex embodied interactive environments demonstrate the framework’s effectiveness in uncovering interpretable internal directions and steering agents toward successful outcomes.

This framework establishes the foundation for monitoring and early detection of misalignment, enabling step-level diagnosis of reasoning drift and timely intervention before task failure. Future work will extend it to multimodal embodied environments and



(a) Task: "Measure the temperature of unknown substance B in bedroom. First, focus on the thermometer. Next, focus on the unknown substance B. Then, if temperature $> 0.0C \rightarrow$ place it in the yellow box; if $< 0.0C \rightarrow$ place it in the purple box. Boxes are in the living room." The steered model (blue) prioritizes the correct order of instructions and completes the task, while the baseline (red) starts with the wrong order and fails with looping in the hallucination that it has the thermometer to measure the temperature.

(b) Task: "Grow Plant to Reproduction. Seeds can be found in the living room. First, focus on a seed. Then, make changes to the environment that grow the plant until it reaches the reproduction life stage." Here, the baseline model (red) starts with the correct sequence on the task execution but drifts in between hallucinating that it has already planted the seeds. Steering corrects reasoning drift, enabling successful planting and growth.

Figure 4: Comparison of baseline (SFT Llama-2-7B) and the steered LLM agents across two ScienceWorld tasks. Steering along learned *success directions* mitigates reasoning drift, preventing hallucinations and ensuring task completion.

explore proactive steering of autonomous LLM agents through interpretable temporal feedback.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691* (2022).

[3] Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes. In *Proceedings of the International Conference on Learning Representations (ICLR) Workshop Track*. https://arxiv.org/abs/1610.01644

[4] Vineeth Balasubramanian, Shen-Shyang Ho, and Vladimir Vovk. 2014. *Conformal prediction for reliable machine learning: theory, adaptations and applications*. Newnes.

[5] Oren Barkan, Yonatan Toib, Yehonatan Elisha, Jonathan Weill, and Noam Koenigstein. 2024. LLM Explainability via Attributive Masking Learning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. 9522–9537.

[6] Daniel Beaglehole, Adityanarayanan Radhakrishnan, Enric Boix-Adsera, and Mikhail Belkin. 2025. Toward universal steering and monitoring of AI models. *arXiv preprint arXiv:2502.03708* (2025).

[7] Feiyang Cai and Xenofon Koutsoukos. 2020. Real-time out-of-distribution detection in learning-enabled cyber-physical systems. In *2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 174–183.

[8] Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*. 2126–2136. https://aclanthology.org/P18-1198

[9] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600* (2023).

[10] Jane Doe, Alex Smith, and Ravi Kumar. 2024. Addressing Uncertainty in LLMs to Enhance Reliability in Generative AI. *arXiv preprint arXiv:2403.01234* (2024). https://arxiv.org/abs/2403.01234

[11] John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*. 4129–4138. https://aclanthology.org/N19-1419

[12] Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *Advances in Neural Information Processing Systems* 33 (2020), 20179–20191.

[13] Shiyuan Huang, Siddarth Mamidanna, Shreedhar Jangam, Yilun Zhou, and Leilani H Gilpin. 2023. Can large language models explain themselves? a study of llm-generated self-explanations. *arXiv preprint arXiv:2310.11207* (2023).

[14] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. 2022. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608* (2022).

[15] Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. 2024. Understanding the planning of LLM agents: A survey. *CoRR* (2024).

[16] Ramneet Kaur, Susmit Jha, Anirban Roy, Sangdon Park13, Edgar Dobriban, Oleg Sokolsky, and Insup Lee. 2022. iDECODe: In-Distribution Equivariance for Conformal Out-of-Distribution Detection. (2022).

[17] Ramneet Kaur, Yahan Yang, Oleg Sokolsky, and Insup Lee. 2024. Out-of-Distribution Detection in Dependent Data for Cyber-Physical Systems with Conformal Guarantees. *ACM Transactions on Cyber-Physical Systems* 8, 4 (2024), 1–27.

[18] Wei Li, Hao Chen, and Emily Zhang. 2024. Polysemantic Dropout: Conformal OOD Detection for Specialized LLMs. *arXiv preprint arXiv:2405.05678* (2024). https://arxiv.org/abs/2405.05678

[19] Fuxiao Liu. [n. d.]. Towards Understanding In-Context Learning with Contrastive Demonstrations and Saliency Maps. In *NeurIPS 2024 Workshop on Scientific Methods for Understanding Deep Learning*.

[20] Xiaotian Liu, Ali Pesaranghader, Hanze Li, Punyaphat Sukcharoenchaikul, Jaehong Kim, Tanmana Sadhu, Hyejeong Jeon, and Scott Sanner. 2025. Open-world planning via lifted regression with llm-inferred affordances for embodied agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 20881–20897.

[21] Yang Liu, Weixing Chen, Yongjie Bai, Xiaodan Liang, Guanbin Li, Wen Gao, and Liang Lin. 2025. Aligning cyber space with physical world: A comprehensive survey on embodied ai. *IEEE/ASME Transactions on Mechatronics* (2025).

[22] Yifan Liu, Daniel Kim, and Sam Goldstein. 2023. Conformal Language Modeling. *arXiv preprint arXiv:2312.06789* (2023). https://arxiv.org/abs/2312.06789

[23] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems* 30 (2017).

[24] Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful Chain-of-Thought Reasoning. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*. 305–329.

[25] Samuel Marks and Max Tegmark. 2024. The Geometry of Truth: Emergent Linear Structure in LLM Representations of True/False Datasets. In *Proceedings of the Conference on Language Modeling (COLM)*. https://arxiv.org/abs/2310.06824 arXiv preprint arXiv:2310.06824.

[26] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332* (2021).

[27] Minh Nguyen, Luca Rossi, and Pedro Alvarez. 2024. SafePath: Conformal Prediction for Safe LLM-Based Autonomous Navigation. In *Proceedings of the 2024 International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/2406.04567

[28] Rohan Patel, Sarah Williams, and Grace Lin. 2024. Prune 'n Predict: Optimizing LLM Decision-making with Conformal Prediction. *arXiv preprint arXiv:2404.09876* (2024). https://arxiv.org/abs/2404.09876

[29] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. [n. d.]. Reflexion: language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

[30] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10740–10749.

[31] Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Cote, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. [n. d.]. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *International Conference on Learning Representations*.

[32] Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. 2024. Trial and Error: Exploration-Based Trajectory Optimization of LLM Agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 7584–7600.

[33] Kaustubh Sridhar, Souradeep Dutta, Dinesh Jayaraman, and Insup Lee. [n. d.]. RICL: Adding In-Context Adaptability to Pre-Trained Vision-Language-Action Models. In *9th Annual Conference on Robot Learning*.

[34] Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*. 4593–4601. https://aclanthology.org/P19-1452

[35] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[36] Vladimir Vovk, Alex Gammerman, and Glenn Shafer. 2005. *Algorithmic learning in a random world*. Springer Science & Business Media.

[37] Jiahao Wang, Mingyue Cheng, and Qi Liu. 2025. Can Slow-thinking LLMs Reason Over Time? Empirical Studies in Time Series Forecasting. *arXiv e-prints* (2025), arXiv–2505.

[38] Jun Wang, David Smith Sundarsingh, Jyotirmoy V Deshmukh, and Yiannis Kantaros. [n. d.]. ConformalNL2LTL: Translating Natural Language Instructions into Temporal Logic Formulas with Conformal Correctness Guarantees. ([n. d.]).

[39] Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. 2022. ScienceWorld: Is your Agent Smarter than a 5th Grader?. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 11279–11298.

[40] Tianyu Wu, Shizhu He, Jingping Liu, Siqi Sun, Kang Liu, Qing-Long Han, and Yang Tang. 2023. A brief overview of ChatGPT: The history, status quo and potential future development. *IEEE/CAA Journal of Automatica Sinica* 10, 5 (2023), 1122–1136.

[41] Xiaoying Xing, Chia-Wen Kuo, Li Fuxin, Yulei Niu, Fan Chen, Ming Li, Ying Wu, Longyin Wen, and Sijie Zhu. 2025. Where do Large Vision-Language Models Look at when Answering Questions? *arXiv e-prints* (2025), arXiv–2503.

[42] Weimin Xiong, Yifan Song, Xiutian Zhao, Wenhao Wu, Xun Wang, Ke Wang, Cheng Li, Wei Peng, and Sujian Li. 2024. Watch Every Step! LLM Agent Learning via Iterative Step-level Process Refinement. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 1556–1572.

[43] Yahan Yang, Ramneet Kaur, Souradeep Dutta, and Insup Lee. 2024. Memory-based distribution shift detection for learning enabled cyber-physical systems with statistical guarantees. *ACM Transactions on Cyber-Physical Systems* 8, 2 (2024), 1–28.

[44] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

[45] Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, et al. 2025. The Landscape of Agentic Reinforcement Learning for LLMs: A Survey. *arXiv e-prints* (2025),

arXiv–2509.

[46] Ling Zhao, Xiaoyu He, and John Thomas. 2024. Analyzing Uncertainty of LLM-as-a-Judge: Interval Evaluations with Conformal Prediction. *arXiv preprint arXiv:2407.08921* (2024). https://arxiv.org/abs/2407.08921

[47] Jiahui Zhou, Dan Li, Lin Li, Zhuomin Chen, Shunyu Wu, Haozheng Ye, Jian Lou, and Costas J Spanos. 2025. Enhancing LLM Reasoning for Time Series Classification by Tailored Thinking and Fused Decision. *arXiv e-prints* (2025), arXiv–2506.

[48] Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. 2023. Representation Engineering: A Top-Down Approach to AI Transparency. *CoRR* (2023).

Received 13 November 2025