deepflow

# Orchestrating Human-AI Teams: The Manager Agent as a Unifying Research Challenge

Charlie Masters, Advaith Vellanki, Jiangbo Shangguan, Bart Kultys
Alastair Moore, Stefano V. Albrecht
DeepFlow
London, United Kingdom
{charlie.masters,stefano.albrecht}@deepflow.com

## Abstract

While agentic AI has advanced in automating individual tasks, managing complex multi-agent workflows remains a challenging problem. This paper presents a research vision for autonomous agentic systems that orchestrate collaboration within dynamic human-AI teams. We propose the **Autonomous Manager Agent** as a core challenge: an agent that decomposes complex goals into task graphs, allocates tasks to human and AI workers, monitors progress, adapts to changing conditions, and maintains transparent stakeholder communication. We formalize workflow management as a Partially Observable Stochastic Game and identify four foundational challenges: (1) compositional reasoning for hierarchical decomposition, (2) multi-objective optimization under shifting preferences, (3) coordination and planning in ad hoc teams, and (4) governance and compliance by design. To advance this agenda, we release an open-source simulation and evaluation framework for multi-agent workflow orchestration. Evaluating GPT-5-based Manager Agents across 20 workflows, we find they struggle to jointly optimize for goal completion, constraint adherence, and workflow runtime–underscoring workflow management as a difficult open problem. We conclude with organizational and ethical implications of autonomous management systems.

## CCS Concepts

• **Computing methodologies** → **Multi-agent systems**; *Reinforcement learning*; *Planning and scheduling*; • **Human-centered computing** → Human computer interaction (HCI).

## Keywords

Multi-agent systems, workflow management, human-AI collaboration, task decomposition, ad hoc teamwork

**ACM Reference Format:**
Charlie Masters, Advaith Vellanki, Jiangbo Shangguan, Bart Kultys, Alastair Moore, Stefano V. Albrecht. 2025. Orchestrating Human-AI Teams: The Manager Agent as a Unifying Research Challenge. In *Proceedings of Conference on Distributed Artificial Intelligence (DAI '25).* ACM, New York, NY, USA, 17 pages. https://doi.org/XXXXXXX.XXXXXXX

*DAI '25, London, UK*
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/2025/11
https://doi.org/XXXXXXX.XXXXXXX

## 1 Introduction

AI systems based on large language models (LLMs) have demonstrated remarkable proficiency at discrete, well-defined tasks across diverse domains, such as legal reasoning [34, 37], software engineering [41], drug discovery [39, 81], and finance [21, 91].

However, while current agentic systems are able to dynamically plan and execute well-scoped steps with high efficiency, the overarching strategic layer of workflow management consisting of task decomposition, dynamic resource allocation, progress monitoring, and adaptive re-planning remains fundamentally challenging [53, 82]. This limitation becomes particularly pronounced in complex multi-agent environments where tasks are interdependent, objectives evolve dynamically, and coordination failures can cascade across entire workflows.

The next frontier in distributed AI lies in transcending task-level competence toward autonomous systems that can manage the entire life-cycle of complex, collaborative projects involving multiple stakeholders and evolving objectives. Our research vision is the creation of agentic ecosystems where autonomous AI agents and human workers collaborate as a team [76]. To make progress toward this vision, we propose the **Autonomous Manager Agent** as a specific research challenge.

The Manager Agent is conceived as an autonomous entity responsible for the end-to-end management of complex workflows within a dynamic multi-agent environment. Its purpose is to orchestrate a team of human and AI "workers" to optimally achieve high-level goals specified by a human stakeholder. The responsibilities of the Manager Agent include decomposing complex goals into executable task graphs; allocating human and AI workers to tasks based on their skills, task requirements, and available resources; monitoring progress and proactively identifying potential impediments; adapting to changing environmental constraints and objectives; and maintaining transparent stakeholder communication. This vision aligns with the growing field of LLM-based Multi-Agent Systems (MAS), which aims to emulate the principles of human teamwork and specialization to solve problems collectively at scale [73, 84].

Realizing human-AI ecosystems demands coordinated research efforts across the Distributed AI community, spanning multiple traditionally separate sub-fields. Historically, scientific fields have often advanced by coalescing around ambitious challenge problems [14, 26, 66]. The Manager Agent problem is designed to serve this role for distributed and agentic AI, requiring the integration of multiple sub-fields in multi-agent systems research, including multi-agent task decomposition and planning [61], multi-objective optimization and learning in dynamic teams with heterogeneous capabilities [4, 56], and governance design [84]. The challenge is

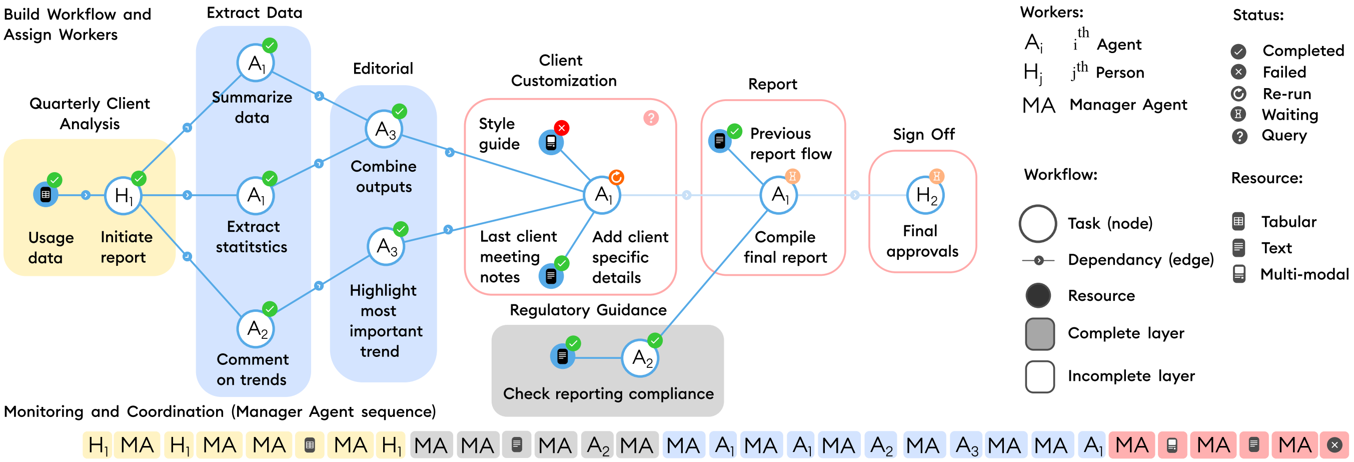

**Figure 1: The Manager Agent (MA) as an orchestrator. Goal: "Write an updated quarterly report for the client". Based on this prompt, the MA is responsible for creating, modifying, and executing actions on a task graph, $G$, with a heterogeneous team of workers, $W$ (details in Section 3). The MA is responsible for coordinating the sequence of activity between workers, illustrated in the bottom row. Here, the `Client Customization` task does not complete as the resource fails verification, and the MA stops the execution, to then determine how to complete the flow successfully.**

grounded in real-world workflow automation with scalable difficulty dimensions (e.g. number of tasks, team size, dependency complexity, constraints). By proposing this challenge, we aim to galvanize research efforts and provide a common ground for evaluating progress in creating collaborative agentic AI systems.

This paper makes the following contributions:

- We propose the **Autonomous Manager Agent** as a unifying research challenge that bridges traditionally separate AI sub-fields including multi-agent coordination, compositional reasoning, and governance design (Section 2).
- We provide a **formal framework** modeling autonomous workflow management as a Partially Observable Stochastic Game (POSG), enabling principled algorithmic development (Section 3).
- We identify **four foundational research problems** for autonomous workflow management – compositional reasoning for task decomposition, multi-objective optimization under non-stationary preferences, coordination in ad hoc teams, and governance by design (Section 4).
- We release *Manager Agent Gym*: an **open-source library and simulator** for multi-agent workflow management (Section 5). We evaluate Manager Agents based on GPT-5 across 20 diverse workflows and find that while they can achieve goal completion, constraint adherence, or workflow runtime, they fail to robustly solve for these qualities in tandem—highlighting workflow management as a difficult problem for agentic AI.
- We discuss the **ethical, regulatory, and privacy implications** of deploying autonomous management systems in real-world organizational contexts (Section 6).

## 2 Conceptual Framework: The Manager Agent

The Manager Agent coordinates activity in a multi-agent system, analogous to a human project manager tasked with optimizing complex, large-scale workflows. Its environment is a dynamic task graph, illustrated in Figure 1, where nodes represent tasks and directed edges represent dependencies. This graph is not static; it is created, modified, and executed by a heterogeneous team of "workers," which can be specialized AI agents (e.g., a code-writing agent, a market-analysis agent) or human collaborators. This aligns with the MAS orchestrator-worker paradigm [45, 73], where a coordinator decomposes problems and delegates sub-tasks to specialized agents. Thus, the Manager Agent focuses on high-level strategy and coordination while worker agents focus on task execution. Worker agents operate autonomously to execute their assigned tasks.

A key aspect of this framework is the shift from a "human-in-the-loop" model, where human intervention is required for most critical decisions, to a "human-on-the-loop" model: the human stakeholder retains control over high-level objectives and oversight, but the Manager Agent autonomously handles the intricate, moment-to-moment operational management. This paradigm shift promises to significantly amplify human productivity by offloading the cognitive burden of complex coordination, while keeping humans in a strategic, supervisory role.

### 2.1 Core Capabilities

To fulfil its role, the Manager Agent must possess a suite of core capabilities that mirror and extend those of a human manager:

(1) **Structuring Workflows:** The Manager Agent must be able to take high-level, often ambiguous, goals from a human stakeholder and decompose them into a structured graph of feasible, concrete tasks and sub-tasks with clearly defined dependencies.
(2) **Assigning Workers:** It must dynamically allocate tasks to the most suitable human or AI workers based on a deep understanding of task requirements, worker skills and availability, and resource constraints.
(3) **Monitoring and Coordination:** The Manager Agent must continually track the progress of all workers on their assigned

tasks, proactively identify and resolve bottlenecks, and ensure synchronized effort across the entire workflow.

(4) **Adaptive Planning and Execution:** The environment is dynamic. The Manager Agent must be able to generate and execute modifications to the workflow in real-time—revising the task graph, adjusting worker roles, or reassigning tasks in response to unexpected events, information, or changing priorities.

(5) **Stakeholder Communication:** It must maintain transparent communication with the stakeholder, providing regular updates on plans, progress, and potential issues to ensure robust oversight and enable informed intervention when necessary.

## 2.2 A Unifying Challenge

The Manager Agent problem is not merely a practical application, but a unifying research challenge that necessitates a synthesis of capabilities from traditionally separate sub-communities within multi-agent systems (MAS) research. Its core capabilities are directly related to long-standing research questions in task decomposition and allocation [44], team formation [49], multi-agent coordination and learning [4], communication [90], agent modeling [6], and ad hoc teamwork [56].

Specifically, the capabilities of **Structuring Workflows** and **Assigning Workers** are a direct instantiation of the central MAS problem of task decomposition and allocation, which seeks optimal methods for breaking down complex goals and mapping sub-tasks to the most appropriate agents [44]. This process is also intrinsically linked to team formation [49], as the Manager Agent must dynamically assemble a group of workers to execute the workflow. The capabilities of **Monitoring and Coordination** and **Adaptive Planning and Execution** are the essence of multi-agent coordination and collaboration, a field focused on managing dependencies and ensuring coherent collective action [61]. To improve over time, the Manager Agent must engage in multi-agent learning to refine its strategies for decomposition, allocation, and coordination [4], as well as agent modeling [6, 59] to infer the capabilities and state of its workers. Effective communication is vital, not only for **Stakeholder Communication** but also for the coordination protocols between the manager and workers, a topic that includes the potential for learning emergent languages [90]. Because the team composition is not fixed, this entire process must occur under the challenging conditions of ad hoc teamwork [56, 65], where the Manager Agent must be able to collaborate with new teammates without pre-coordination between agents (such as prior joint training).

The ambition to build such a unifying agent is not new, but its feasibility has been unlocked by recent breakthroughs in high-capacity foundation models [16, 24]. These models, particularly Large Language Models (LLMs), provide the "cognitive engine" for the Manager Agent, capable of high-level reasoning and planning across a range of complex, real-world tasks that was previously intractable for automated systems [3]. The emergence of Large Reasoning Models (LRMs) in 2024-2025 marks a significant milestone [25, 62]. These models leverage large-scale reinforcement learning to achieve stepwise reasoning required for dynamic planning and adaptation in complex workflows [25]. This creates a unique opportunity to synthesize the reasoning power of foundation models with established MAS frameworks, positioning the Manager Agent as a timely and achievable research goal.

## 3 A Formal Model of Workflow Management

To ground this challenge in a mathematical framework, we model the problem of autonomous workflow management as a **Partially Observable Stochastic Game (POSG)** [36]. A POSG models scenarios where multiple agents interact in a shared environment with incomplete information and different objectives. This is an appropriate model for our domain, as it explicitly accounts for the Manager Agent and the team of worker agents as distinct decision makers with their own action sets, observations, and preferences. A POSG is defined by the tuple $\langle I, S, b^0, \{A_i\}, \{O_i\}, P, \{R_i\}\rangle$, where each component is specified for our domain as follows.

### 3.1 Set of Agents ($I$)

The set of agents $I$ consists of the Manager Agent ($M$) and the set of all worker agents ($W$), which can include both human and AI workers. Thus, $I = \{M\} \cup W$. (We will remark on modeling the stakeholder in Section 3.8.)

### 3.2 State Space ($S$)

The underlying state of the environment at any time $t$, denoted $s^t \in S$, is a comprehensive snapshot of the entire workflow. It is defined as a tuple $s \equiv \langle G, W, C, X, U\rangle$:

- $G$: The complete task-dependency graph, including all task nodes $\{T\}$, their metadata $\mu_T$ (e.g. status, owner, progress to date) and directed edges $E_{T_i,T_j}$ representing dependencies between tasks.
- $W$: The set of all available human and AI workers, including their capabilities, current assignments, availability, and cost rates.
- $C$: A persistent set of communications $\{C_{i,j}\}$ between all agents $i, j \in I$.
- $X$: A registry of all artifacts produced by tasks so far, including documents, code, and other digital assets.
- $U$: A set of preference weights of the stakeholder for how tasks are completed (e.g. cost, speed, quality). These may be fully or partially observable to agents $I$, and can evolve over time.

The initial state $s^0$ is sampled from the initial state distribution $b^0$.

### 3.3 Action Spaces ($A_i$)

Each agent $i \in I$ has its own set of available actions it can perform.

**Manager Agent's Action Space ($A_m$):** The Manager Agent possesses a rich action space designed to orchestrate complex workflows through three primary categories of capabilities:

- *Observability-increasing actions:* These actions allow the agent to reduce its uncertainty about the state by gathering information about the workflow's progress. Representative examples include `Inspect`($T_i$) to view execution logs and outputs $\mu_{T_i}, X_{T_i}$ for specific tasks, and `GetChatHistory`($T_i$) to retrieve the recent communication history related to a given task.
- *Graph-modifying actions:* These actions dynamically alter the structure of the workflow itself. Representative actions include `AddTask`($T_i$) and `RemoveTask`($T_i$) to add or remove tasks $T_i$ from $G$, `AddEdge`($T_i, T_j$) to establish or modify dependencies $E_{T_i,T_j}$

between tasks, and DecomposeTask($T_i$) to decompose a complex task into sub-tasks.

- *Delegation and communication actions:* These actions manage the team of workers. Examples include AssignTask($T_i, W_j$) to assign a task to a human or AI worker, and SendMessage($W_j$,message) to communicate directly with a worker to provide guidance, request updates, or alert them to changing priorities or constraints.

**Worker Agent's Action Space ($A_i$ for $i \in W$):** Worker agents have an action space focused on task completion, including a set of tools which may be fixed or dynamic, or the ability to seek additional information when task requirements are ambiguous.

The evolution of the system depends on the **joint action** $a^t = \langle a_1^t, \dots, a_n^t \rangle \in \times_{i \in I} A_i$ taken by all agents at time $t$.

### 3.4 Observation Spaces ($O_i$)

Each agent has a private, partial view of the state, as well as (potentially) the past actions of other agents.

**Manager Agent's Observation Space ($O_M$):** The Manager Agent has full access to some parts of the state, such as the high-level task graph, resource metadata, and chat history, but may not directly observe other parts such as stakeholder preferences, or the contents of artifacts or detailed worker logs.

**Worker Agent's Observation Space ($O_i$ for $i \in W$):** A worker's observation may be limited to the details of its assigned tasks, communications sent to or by the worker, and artifacts generated by the worker. Additional information about the workflow state may be observable to workers depending on their roles within the team.

### 3.5 Transition and Observation Dynamics ($P$)

The function $P(s', o|s, a)$ defines the probability of transitioning to state $s'$ and all agents receiving a joint observation $o = \langle o_1, \dots, o_n \rangle$, given the current state $s$ and the joint action $a$ taken by all agents. The state dynamics can be deterministic, such as actions that modify the task graph (e.g. removing a task). State dynamics can also be stochastic, particularly when actions involve workers whose performance is non-deterministic (e.g. time to task completion, quality of task output, etc.).

### 3.6 Reward Functions ($R_i$)

Each agent $i \in I$ has its own reward function $r_i = R_i(s, a)$, reflecting its preferences and objectives. We also include workflow-level constraints: hard constraints $\mathcal{H}$ that must always hold, and soft constraints $\mathcal{S}$ that can be violated with penalties.

**Manager Agent's Reward ($R_M$):** The Manager Agent's reward is aligned with the high-level goals of the stakeholder, represented by its preferences $U$. Conceptually, this is a multi-level function that includes a sparse terminal reward for successful workflow completion, and can include additional performance metrics such as time and cost for workflow execution and overall quality of work outputs, weighted according to stakeholder preferences $U$. Additionally, violations of hard and soft constraints ($\mathcal{H}/\mathcal{S}$) result in penalties of varying magnitudes (e.g. large penalty for violating hard constraints).

**Worker Agent's Reward ($R_i$ for $i \in W$):** A worker agent's reward may be simpler, based on metrics such as the timely and successful completion of its assigned tasks. This allows for modeling self-interested behavior, where a worker might prioritize its individual task goals over the global workflow objective.

If all agents share the same reward function, i.e. $R_i = R, \forall i \in I$, then the model simplifies to a Decentralized POMDP (Dec-POMDP), which is a purely cooperative setting [61]. The more general POSG formulation allows for a richer spectrum of mixed cooperative and self-interested behaviors.

### 3.7 Solution Concept

The actions of each agent $i \in I$ in the system are governed by a policy $\pi_i$, which assigns probabilities to the agent's available actions $A_i$ based on the history of the agent's observations $\{o_i^t\}_{t=0,1,2,\dots}$.

The **optimal policy** $\pi_M^*$ of the Manager Agent will depend on the assumptions we make about the behaviors of other agents $W$ in the system. Borrowing concepts from Stochastic Bayesian Games [5] and Interactive POMDPs [28], we may assume that the worker agents $j \in W$ draw their policies from a set of possible policies $\pi_j \in \Pi_W$. In this case, we seek a policy $\pi_M^*$ for the Manager Agent that maximizes the expected return, given by the expected sum of rewards $r_M^t$ it receives over time $t = 0, 1, 2, \dots$, assuming the worker agents can use any of the policies in $\Pi_W$. Such an assumption may be suitable if the possible behaviors of workers are well understood, which is often the case for defined AI workers.

In general, we may also assume that some worker agents may learn and adapt their behaviors over time based on past observations, as is the case with human workers and more advanced AI workers. This brings us into the realm of game theory and multi-agent reinforcement learning (MARL) [4]. In a POSG, where agents may have different preferences, the optimal solution becomes an *equilibrium*—a joint policy from which no agent has a unilateral incentive to deviate. This idea is embodied by the **Nash Equilibrium (NE)** [58], which is a joint policy $\pi^* = (\pi_1^*, \dots, \pi_n^*)$ in which for every agent $i$, its policy $\pi_i^*$ is a best response to the set of other agents' policies $\pi_{-i}^* = \pi^* \setminus \{\pi_i^*\}$. This means that no agent $i$ can improve its expected return by changing its policy alone. Furthermore, a joint policy $\pi^*$ is **Pareto-optimal** if there is no other joint policy $\pi'$ that can increase the expected return for at least one agent without decreasing the expected return for any other agent.

Combining these concepts, one solution concept for our POSG is a **Pareto-optimal Nash Equilibrium (PONE)** [57]. This is a joint policy that is both stable (a Nash Equilibrium) and efficient (Pareto-optimal), representing a desirable outcome for the collaborative human-AI team. In the Dec-POMDP case, in which agents share the same rewards, this corresponds to a joint policy that maximizes the expected return of all agents.

### 3.8 Modeling the Stakeholder

There are different ways to model the stakeholder in our POSG formalism. In the most general case, a stakeholder is itself an autonomous agent that can observe partial information about the workflow state, and take actions to interact with the workflow and other agents (manager, workers). Thus, a stakeholder agent $\alpha \in I$ can be represented via its own action set $A_\alpha$, observation set $O_\alpha$, and reward function $R_\alpha$. The stakeholder's action set may include those of the Manager Agent and additional actions to enable communication with the Manager Agent and updating stakeholder preferences $U$ (Section 3.2).

If the stakeholder is passive and does not directly choose actions in the POSG, thus giving full control to the Manager Agent, we may instead model the stakeholder as part of the transition dynamics $P$ (Section 3.5). By including the time index $t$ inside state $s^t$, the stakeholder can change its preferences $U \in s^t$ over time $t$ through state transitions $P(s^{t+1}, o^{t+1} | s^t, a^t)$.

## 4 Foundational Research Challenges

The realization of an autonomous Manager Agent requires significant advances across several core areas of AI. We identify four foundational research challenges that are critical to our vision.

### 4.1 Hierarchical Task Decomposition

For a Manager Agent, mapping a high-level workflow description into a task graph $G$ with governance constraints $\{\mathcal{H}, \mathcal{S}\}$ is the bottleneck that unlocks all downstream capabilities (e.g. allocation, monitoring, adaptation). Recent LLM-based multi-agent frameworks [10, 87] show that performance gains correlate almost linearly with the quality of the induced task graph—underlining that structure learning, not raw language generation, is the critical path. Empirical studies reveal that both single-agent LRMs and multi-agent orchestration systems fail once graph depth, branching factor or novelty exceed modest thresholds. For example, large-scale audits of reasoning tasks [47, 50] find sharp phase transitions beyond which success probability collapses. Multi-agent variants inherit these limits, with recent research [79, 80] reporting substantial error cascades when sub-agents produce incompatible sub-plans, despite significant engineering for coordination. Current agents can discover local patterns but lack a hierarchical notion of causality that scales with problem complexity, prompting the question:

*How can a Manager Agent scale to robustly solving large complex planning problems in dynamic multi-agent systems?*

Why might models struggle? Recent work shows that transformer-based agents appear to rely on shallow subgraph matching over memorized patterns rather than genuine composition [29]. This shortcut is effective for in-distribution tasks but can fail when unseen combinations or long-range dependencies dominate in dynamic, heterogeneous teams. Moreover, partial observability and non-stationary interactions make these problems worse in multi-agent settings. They amplify exposure bias: when one worker deviates locally, errors propagate through the task graph. This invalidates the Manager Agent's original planning assumptions made to satisfy environment constraints $\mathcal{H}, \mathcal{S}$ and preferences $U$.

Despite this, we can find some promising directions to tackle these problems in the following areas:

(1) **Structured latent planning.** Augment LRMs with explicit symbolic planners or graph-structured controllers that operate over learned abstractions, enabling verifiable sub-plan composition in the style of neuro-symbolic planning [15, 18, 51].
(2) **Meta-adaptive decomposition.** Treat task-graph induction itself as a meta-RL problem: train the manager to iteratively propose, simulate and revise decompositions with self-consistency and outcome feedback [48].

### 4.2 Multi-Objective Optimization with Non-Stationary Preferences

An optimal policy $\pi_M^*$ for the Manager Agent must juggle multiple, often competing objectives such as *cost*, *latency* and *quality*, yet the workflow stakeholder can re-rank these objectives mid-execution. This creates a **dynamic multi-objective** problem layered on top of the usual multi-agent coordination issues.

Two separate sub-fields attack aspects of this problem, but neither address the entire setting. Multi-Objective Reinforcement Learning (MORL) provides a framework for optimizing multiple, potentially conflicting objectives via **scalarization methods** that combine objectives via weighted sums [74], and **Pareto-based methods** that learn policy sets representing different trade-offs [75]; both of which assume the objectives are fixed a priori, and break when preference weights shift online.

Conversely, modern preference-learning pipelines model *one* scalar reward and thus ignore multi-objective trade-offs. In reinforcement learning from human feedback (RLHF), a reward model is trained to score an output higher than a ranked alternative, and the policy is fine-tuned to maximize that score [64]. In contrast, reinforcement learning from verifiable rewards (RLVR) replaces this learned model with a verifiable reward, modeling preference in task execution via reward shaping around easily measured attributes such as length or clarity. [25, 48]. Both lines of work assume the reward remains fixed; recent extensions handle *non-stationary* scalar preferences [70] but still optimize a single objective—an expressivity gap formalized in [1].

Since an effective manager must pragmatically handle evolving, conflicting objectives, the open research question remains:

*How can a Manager Agent learn a robust policy that can efficiently adapt to non-stationary stakeholder preferences over multiple objectives without requiring costly retraining?*

Early answers may lie in test-time alignment and meta-learning that infer fresh weight vectors from a few stakeholder corrections [60, 83], or explicit hierarchical control where the Manager Agent re-weights objectives based on stakeholder interactions while task agents solve the resulting single-objective sub-tasks, both of which remain unexplored in verifiable-reward, multi-agent settings.

### 4.3 Coordination in Ad Hoc Teams

The Manager Agent must orchestrate collaboration in dynamic, heterogeneous teams where agents may join or leave without prior coordination. This means the Manager Agent cannot rely on pre-coordinated strategies or prior joint training with a fixed set of teammates. This defines the classic *ad hoc teamwork* (AHT) problem—a long-standing challenge in multi-agent systems [56, 72].

Key open problems in AHT that are directly relevant to our setting include generalizing to new types of teammates with their individual capabilities, expectations, and working preferences, as well as effectively collaborating with teammates who are themselves learning and adapting their behavior [56]. The Manager Agent must be able to quickly infer the skills, knowledge, and preferences of workers based on limited interactions, and flexibly adapt how it communicates and coordinates with workers [5, 13]. The open research question is therefore:

*How can the Manager Agent rapidly infer the capabilities, reliability, and intent of new teammates from limited interaction and leverage this understanding for effective, on-the-fly task delegation and coordination?*

Several approaches point toward partial solutions. Ribeiro et al. [67] use model-based reinforcement learning to learn teammate behavior and adapt policies on the fly, but their method assumes sufficient prior interaction and struggles under extreme heterogeneity. Zhang et al. [88] propose training on offline trajectories to predict teammate-aware goals, though their approach lacks the ability to reason about unobserved agent types. Wang et al. [78] embed teammate behaviors to support policy generalization in novel teams, but still assume consistent observation structures and offer limited mechanisms for dynamic role negotiation. Liu et al. [52] leverage large language models for hierarchical plan generation using interactive reasoning, though their reliance on language abstractions may not scale to low-level execution. Jin et al. [42] introduce a capability-aware ad hoc teamwork model using agent hierarchies, but this approach presumes known capability classes.

Each method tackles a facet of the problem: policy adaptation, team modeling, or high-level coordination, but none yet provide the full-stack reasoning, real-time inference, and dynamic task restructuring required for Manager Agents to operate robustly in open, evolving teams. Ad hoc coordination remains a critical and unsolved challenge for collaborative AI.

### 4.4 Governance and Compliance by Design

Manager Agent autonomy in complex organizational workflows creates a critical challenge: maintaining governance and compliance across dynamic multi-agent systems. These agents must interpret natural language constraints, adapt to evolving regulations, and demonstrate compliance across heterogeneous teams. The rapid advancement and deployment of AI to safety critical environments poses significant regulatory challenges with highly unpredictable and rapidly changing requirements [27, 46]. Solving these challenges involves tackling a series of problems, specifically:

*How can Manager Agents maintain governance and compliance in dynamic multi-agent workflows while adapting to evolving regulatory constraints without compromising operational effectiveness?*

Multi-agent constraint satisfaction requires ensuring workflow-level compliance across dynamically changing teams where agents have heterogeneous capabilities and roles. Recent works attempt to address distributed safety coordination. Gu et al. [33] make progress on safe decentralized MARL via scalable constrained policy optimization, but only in a static team composition setting. Aydeniz et al. [9] achieve team-level constraint satisfaction through joint entropy maximization but only for binary collision avoidance scenarios, not tackling the complex and ambiguous requirements found in regulatory compliance.

The Manager Agent's need to interpret natural language constraints demands translating ambiguous regulatory text into executable policies that can guide agent behavior. Existing approaches address binary safety constraints versus nuanced regulatory requirements that are inherently ambiguous, where optimal policies depend on how organizations choose to handle uncertainty and interpretation. Yao et al. [86] achieve zero-shot adaptation to varying constraint thresholds using conditioned policy optimization, but do not address the problem of mapping from complex regulatory reasoning, focusing purely on numerical parameter adjustment.

Real-time governance adaptation requires adapting to varying safety constraints during deployment without retraining, as regulatory landscapes change post-deployment while agents must continue operating. Recent interpretability advances attempt to enable runtime safety analysis. Anthropic researchers [7, 8] achieve training-time safety analysis through mechanistic interpretability and circuit tracing, but do not address automated post-deployment adaptation to regulatory changes. This connects to the broader challenge that test-time alignment remains an open problem, and optimal helpfulness-harmlessness tradeoffs are domain-specific, often creating conflicts between compliance objectives [11, 32].

Some promising solution areas to these problems could be found in *ad-hoc constraint-aware teaming* which extends existing team constraint approaches [9, 33] to dynamic compositions, *natural language constraint grounding* that combines existing work into using control barrier function learning with LLM-based regulatory interpretation [86], and further mechanistic research to better understand the underlying traces of models inner workings [35].

## 5 Manager Agent Gym: A Simulator for Human–AI Workflow Orchestration

Progress on the four challenges in Section 4 demands a single testbed that exercises hierarchical control, dynamic multi-objective preferences, ad-hoc teaming, and governance *together.* As summarized by the comparison in Table 1 (Appendix A), existing benchmarks each cover a slice: office-workflow benchmarks emphasize long-horizon planning but assume static goals; strategic games capture partial observability and coordination but not governance or multi-objective trade-offs; safety-critical simulations stress team response yet keep objectives and rosters fixed. None evaluates the full spectrum of Manager Agent capabilities.

To fill this gap, we release **Manager Agent Gym** (MA-Gym):[1] a discrete-timestep environment where a manager operates over a graph-based multi-agent workflow and must address any or all of the four challenges per episode. We provide a high-level description of MA-Gym in Section 5.1 along with initial benchmark results in Section 5.2. Detailed specifications can be found in Appendix C and the MA-Gym code repository.

### 5.1 MA-Gym Overview and Workflows

MA-Gym instantiates the POSG formalism in Section 3 with an initial task dependency graph $G$ and agent team $I$ consisting of AI workers and simulated human workers, which are able to communicate via actions that invoke a communication store $C$. Additionally, MA-Gym includes a stakeholder agent $\alpha$ (Section 3.8) with preference weights $U$, which can choose to take actions based on its policy $\pi_\alpha$ configured in the simulator, including communicating with other agents, updating stakeholder preferences, and answering questions from the Manager Agent.

Procedurally, MA-Gym runs each workflow episode as defined in our POSG model: at each timestep, each agent $i \in I$ (including Manager and stakeholder agents) gets an observation and takes an

[1] MA-Gym download: https://anonymous.4open.science/r/manager_agent_gym-E5BA

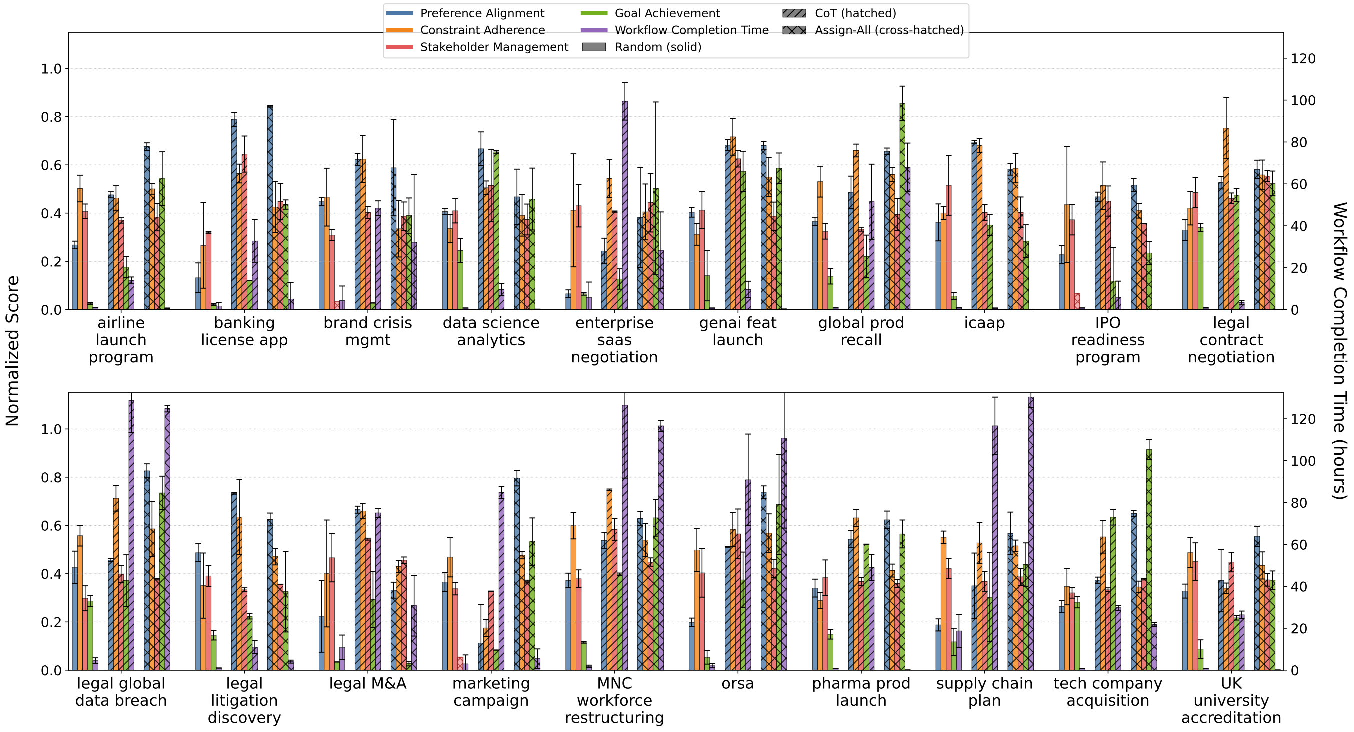

**Figure 2: Random, Chain-of-Thought (CoT) and Assign-All policy performances plotted across 20 workflows (bars show average and standard deviation across 5 random seeds per workflow). Details of workflows can be found in Appendix C.6, Table 4.**

action from its respective set of available actions $A_i$ (full action sets in Appendix C.5, Table 3) based on its policy $\pi_i$. The actions are then executed in MA-Gym, and all tasks that are ready and have been assigned to a worker are executed. Worker agents can join and leave the team at any timestep during an episode, as configured in MA-Gym.

As part of the initial release of MA-Gym, we define 20 challenging workflow scenarios representing a diverse range of real-world domains, with different stakeholder preferences, graph complexity (number of nodes and dependencies), team composition (number and types of agents), and hard/soft constraints. A full listing of these workflows can be found in Appendix C.6, Table 4.

## 5.2 Baselines and Results

We define and evaluate two baselines of LLM-based Manager Agents with OpenAI GPT-5 as a base model. **Random:** observes the dependency graph $G$ and at each timestep is restricted to taking an action chosen uniform-randomly from $A_M$ (the agent still has to specify input parameters for the sampled action); **CoT:** observes the dependency graph and uses Chain-of-Thought reasoning to choose its next action from the full action set $A_M$. A definitive list of allowed Manager Agent actions can be found in Appendix C.5. We repeat all 20 workflows across 5 random seeds, capping the maximum number of Manager Agent actions at 100 before terminating the episode, and report on Preference alignment, Constraint adherence, Goal achievement, Stakeholder management, and Workflow completion time in Figure 2; see Appendix C for definitions of these metrics.

Three key patterns emerged from our evaluation across the 20 workflows. (1) CoT managers successfully completed all tasks in 80% of workflows, while Random managers completed no workflow. (Some results in Figure 2 show non-zero goal achievement for the Random agent since the goal-achievement metric includes partial credit for achieving partial goals.) This explains CoT's increased execution times (46.9h vs. 2.7h): it prioritizes human task assignments to maximize goal achievement, at the cost of increased execution time. (2) The increased goal achievement of the CoT agent comes at the cost of a significant coordination penalty: CoT delegates 25.8% of tasks to humans versus 9.3% for Random (2.8× more), resulting in on average 17.4x±3.8 slower execution. CoT's structured decomposition and refinement behaviors thus succeed in driving workflows to completion, but also expose bottlenecks when slower human execution blocks dependency chains. (3) Stakeholder management remains systematically neglected: stakeholder-related actions account for less than 3.3% of allocations across both baselines (0.9% for CoT; 3.3% for Random), indicating a strong bias toward "doing work" rather than eliciting or adapting to evolving preferences.

Together, these findings sharpen the core Manager Agent challenge: stepwise reasoning enables workflows to converge, but without balancing human–AI load or aligning to stakeholder preferences, it incurs heavy coordination overhead and overlooks objectives. To probe the relationship between different manager policies $\pi_M$ and performance, we also evaluated GPT-4.1 under identical conditions as GPT-5. GPT-5 achieved consistently higher goal completion (0.6–0.7 on analytics and product-launch workflows; see

Fig. 3) and deployed richer planning operators – executing 14× more decompositions and 26× more dependency links than GPT-4.1 (see Table 2 for action usage statistics). By contrast, GPT-4.1 relied heavily on messaging and status queries, resembling a reactive communicator. More detailed analysis can be found in Appendix B. These results show that stronger reasoning capacity supports a more proactive orchestration style, but reasoning alone does not eliminate coordination bottlenecks, stakeholder neglect or ensure constraint adherence. Addressing these gaps will require new objectives and training signals beyond reasoning ability, particularly for preference adaptation and efficient collaboration.

## 6 Organizational and Ethical Considerations

The deployment of an autonomous Manager Agent, a system with the authority to delegate tasks and manage resources for a team of humans and AIs, raises significant organizational and ethical considerations. A responsible research program must proactively address these ethical, regulatory, and privacy challenges, which are central to the development of trustworthy AI [43, 46].

Considerations arise at the level of an individual task assigned to a worker, at the level of the workflow, and as a result of the changes to workflows over time. For example, a failure in an agent-managed workflow may result in the unjust attribution of blame to a human worker assigned a specific task. This scenario exemplifies the phenomenon known as the 'moral crumple zone', where responsibility for systemic failures is placed disproportionately on human operators, while the underlying complexity and opacity of the technological system obscure the accountability of other actors, such as designers, developers, and institutional stakeholders [30].

A well-designed Manager Agent, however, has the potential to mitigate these risks. The agent's architecture can be designed for transparency. For example, the communication store ($C$) and the detailed action logs can serve as an immutable audit trail. Actions like `SendMessage` and `Inspect` are not just for coordination, but are crucial mechanisms for maintaining human oversight and creating shared context. By making its reasoning process explicit and traceable, the agent can help construct clear accountability frameworks where responsibility is distributed fairly among all participants, human and AI, based on their actual control and contribution [12].

The Manager Agent's core function of assigning tasks to workers is a form of automated resource allocation. Without careful design, this process is susceptible to bias, which could perpetuate or amplify existing inequities [55]. For example, the agent might learn spurious correlations from historical data and consistently assign more desirable, high-growth tasks to one group of workers (e.g., AI agents or a specific human demographic) while relegating another group to tedious, low-impact work [31]. There will be scenarios where there is value in humans performing tasks to facilitate learning, reduce skills atrophy or promote social cohesion.

This challenge connects to the extensive body of research on fairness in AI and resource allocation [23]. To address this, fairness cannot be an afterthought. We propose that formal fairness criteria—such as those based on principles of envy-freeness or maximin share [17]—must be integrated directly into the agent's objective function. By making fairness an explicit optimization goal alongside efficiency and quality, we can guide the agent's learning process toward equitable and just allocation policies.

Effective management requires monitoring, but monitoring creates significant privacy risks, particularly for human workers whose activities, communications, and performance are being tracked [31]. The agent's access to the communication history ($C$) and task artifacts ($X$) could expose sensitive personal or proprietary information.

This necessitates research into privacy-preserving architectures [68]. For instance, the agent might use federated learning to learn from worker data without centralizing it, or employ differential privacy to gather aggregate statistics on team performance without revealing individual contributions [54]. A key challenge is that different agents (e.g., internal employees vs. external contractors) may have heterogeneous privacy requirements, demanding flexible and context-aware privacy protocols [68]. The goal is to design a system that achieves the necessary level of observability for effective management while protecting the privacy of all participants.

Ultimately, the safe and responsible deployment of a Manager Agent requires a comprehensive governance framework. This framework must synthesize the elements discussed above, establishing clear ethical principles, mandating transparency and explainability (XAI) mechanisms [2], and implementing robust protocols for continuous monitoring and auditing. By defining clear lines of human oversight and accountability, we can build systems that are not only powerful but also trustworthy and aligned with human values.

## 7 Conclusion and Future Work

This paper has outlined an ambitious vision: an autonomous Manager Agent that orchestrates dynamic teams of human and AI agents to solve complex problems. This timely research goal, enabled by advances in large language and reasoning models, integrates multiple areas of multi-agent systems research. We formalized autonomous workflow management as a structured POSG and identified four core technical challenges: compositional reasoning, multi-objective optimization with non-stationary preferences, ad hoc team coordination, and governance by design. The Manager Agent problem unifies these disparate research threads into a holistic challenge.

To support research on the Manager Agent challenge, we have released the Manager Agent Gym (MA-Gym) which implements the POSG formalism and supports algorithm design and evaluation for Manager Agents. We benchmarked LLM-based Manager Agents across a diverse set of workflows inspired by real-world tasks, showing that jointly optimizing for goal achievement, constraint adherence, and resource usage (e.g. workflow runtime) is a difficult problem for agentic AI. Our future work includes building on this initial release of MA-Gym by defining additional challenging workflow scenarios, expanding the types and capabilities (i.e. tooling) of worker agents, and supporting more robust evaluation techniques for ambiguous aspects of workflow quality [22, 69].

Our goal is to develop a new generation of Manager Agents that effectively address the core challenges presented in this paper through advances in optimal sequential decision making and collaboration in dynamic multi-agent environments. We invite the research community to join us in pursuing novel solutions to these challenges and collectively shape the future of intelligent, collaborative systems.

## References

[1] David Abel, Will Dabney, Anna Harutyunyan, Mark K. Ho, Michael L. Littman, Doina Precup, and Satinder Singh. 2021. On the Expressivity of Markov Reward. In *Neural Information Processing Systems*.

[2] A. Adadi and M. Berrada. 2018. Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access* 6 (2018), 52138–52160.

[3] M. Aghzal, E. Plaku, G. J. Stein, and Z. Yao. 2025. A Survey on Large Language Models for Automated Planning. *arXiv preprint arXiv:2502.12435* (2025).

[4] Stefano V. Albrecht, Filippos Christianos, and Lukas Schäfer. 2024. *Multi-Agent Reinforcement Learning: Foundations and Modern Approaches*. MIT Press. https://www.marl-book.com

[5] Stefano V. Albrecht, Jacob W. Crandall, and Subramanian Ramamoorthy. 2016. Belief and Truth in Hypothesised Behaviours. *Artificial Intelligence* 235 (2016), 63–94. DOI: 10.1016/j.artint.2016.02.004.

[6] Stefano V. Albrecht and Peter Stone. 2018. Autonomous Agents Modelling Other Agents: A Comprehensive Survey and Open Problems. *Artificial Intelligence* 258 (2018), 66–95. DOI: 10.1016/j.artint.2018.01.002.

[7] Anthropic. 2025. Attribution Graphs: Learning Circuits in Language Models. https://transformer-circuits.pub/2025/attribution-graphs/methods.html.

[8] Anthropic. 2025. Tracing Thoughts: Language Model Responses. https://www.anthropic.com/research/tracing-thoughts-language-model.

[9] Ayhan Alp Aydeniz, Enrico Marchesini, Robert Loftin, Christopher Amato, and Kagan Tumer. 2025. Safe Entropic Agents under Team Constraints. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*. 2411–2413.

[10] David Bai, Ishika Singh, David Traum, and Jesse Thomason. 2024. TwoStep: Multi-agent Task Planning using Classical Planners and Large Language Models. *arXiv preprint arXiv:2403.17246* (2024). https://arxiv.org/abs/2403.17246

[11] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. 2022. Constitutional AI: Harmlessness from AI Feedback. arXiv:2212.08073 [cs.CL] https://arxiv.org/abs/2212.08073

[12] M. Baldoni et al. 2022. Accountability in multi-agent organizations: from conceptual design to agent programming. *Autonomous Agents and Multi-Agent Systems* 36, 1 (2022), 23.

[13] Samuel Barrett. 2015. *Making friends on the fly: Advances in ad hoc teamwork*. Vol. 603. Springer.

[14] James Bennett and Stan Lanning. 2006. *The Netflix Prize*. Technical Report. Los Gatos, CA. Dataset release and competition announcement.

[15] Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2024. Graph of Thoughts: Solving Elaborate Problems with Large Language Models. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*. https://arxiv.org/abs/2308.09687

[16] R. Bommasani et al. 2021. On the Opportunities and Risks of Foundation Models. *arXiv preprint arXiv:2108.07258* (2021).

[17] E. Budish. 2011. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy* 119, 6 (2011), 1061–1103.

[18] Alessio Capitanelli and Fulvio Mastrogiovanni. 2024. A framework for neurosymbolic robot action planning using large language models. *Frontiers in Neurorobotics* 18 (06 2024). doi:10.3389/fnbot.2024.1342786

[19] Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. 2025. Why Do Multi-Agent LLM Systems Fail? arXiv:2503.13657 [cs.AI] https://arxiv.org/abs/2503.13657

[20] Matthew Chang, Gunjan Chhablani, Alexander Clegg, Mikael Dallaire Cote, Ruta Desai, Michal Hlavac, Vladimir Karashchuk, Jacob Krantz, Roozbeh Mottaghi, Priyam Parashar, Siddharth Patki, Ishita Prasad, Xavier Puig, Akshara Rai, Ram Ramrakhya, Daniel Tran, Joanne Truong, John M. Turner, Eric Undersander, and Tsung-Yen Yang. 2024. PARTNR: A Benchmark for Planning and Reasoning in Embodied Multi-agent Tasks. arXiv:2411.00081 [cs.RO] https://arxiv.org/abs/2411.00081

[21] Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, et al. 2021. FinQA: A Dataset of Numerical Reasoning over Financial Data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 3697–3711.

[22] John J. Cherian, Isaac Gibbs, and Emmanuel J. Candès. 2024. Large language model validity via enhanced conformal prediction methods. arXiv:2406.09714 [stat.ML] https://arxiv.org/abs/2406.09714

[23] A. Chouldechova. 2017. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data* 5, 2 (2017), 153–163.

[24] Gheorghe Comanici, Eric Bieber, and Mike Schaekermann et al. 2025. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. arXiv:2507.06261 [cs.CL] https://arxiv.org/abs/2507.06261

[25] DeepSeek-AI, Daya Guo, Dejian Yang, et al. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948 [cs.CL] https://arxiv.org/abs/2501.12948

[26] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 248–255. doi:10.1109/CVPR.2009.5206848

[27] Sotirios Dimitriou and George Gantzias. 2024. AI Governance in a Complex and Rapidly Changing Regulatory Landscape: A Global Perspective. *Humanities and Social Sciences Communications* 11, 1 (2024), 1–13.

[28] Prashant Doshi and Piotr J Gmytrasiewicz. 2006. On the Difficulty of Achieving Equilibrium in Interactive POMDPs. In *Proceedings of the National Conference on Artificial Intelligence*, Vol. 21. 1131.

[29] Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. 2023. Faith and Fate: Limits of Transformers on Compositionality. arXiv:2305.18654 [cs.CL] https://arxiv.org/abs/2305.18654

[30] M. C. Elish. 2016. Moral Crumple Zones: Cautionary Tales in Human-Robot Interaction. *Engaging Science, Technology, and Society* 2 (2016), 435–458.

[31] Eve Legal. 2025. Common Uses and Risks of AI in the Workplace.

[32] Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, Andy Jones, Sam Bowman, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Nelson Elhage, Sheer El-Showk, Stanislav Fort, Zac Hatfield-Dodds, Tom Henighan, Danny Hernandez, Tristan Hume, Josh Jacobson, Scott Johnston, Shauna Kravec, Catherine Olsson, Sam Ringer, Eli Tran-Johnson, Dario Amodei, Tom Brown, Nicholas Joseph, Sam McCandlish, Chris Olah, Jared Kaplan, and Jack Clark. 2022. Red Teaming Language Models to Reduce Harms: Methods, Scaling Behaviors, and Lessons Learned. arXiv:2209.07858 [cs.CL] https://arxiv.org/abs/2209.07858

[33] Shangding Gu, Jakub Grudzien Kuba, Ruiqing Chen, Ziyan Wang, Zheng Tian, Jun Wang, Alois Knoll, and Yaodong Yang. 2024. Scalable Constrained Policy Optimization for Safe Multi-agent Reinforcement Learning. In *Advances in Neural Information Processing Systems*, Vol. 37.

[34] Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, Jessica Wu, Joe Nudell, Joel Niklaus, John Nay, Jonathan H. Choi, Kevin Tobia, Margaret Hagan, Megan Ma, Michael Livermore, Nikon Rasumov-Rahe, Nils Holzenberger, Noam Kolt, Peter Henderson, Sean Rehaag, Sharad Goel, Shang Gao, Spencer Williams, Sunny Gandhi, Tom Zur, Varun Iyer, and Zehua Li. 2023. LegalBench: A Collaboratively Built Benchmark for Measuring Legal Reasoning in Large Language Models. arXiv:2308.11462 [cs.CL]

[35] Balint Gyevnar, Cheng Wang, Christopher G. Lucas, Shay B. Cohen, and Stefano V. Albrecht. 2024. Causal Explanations for Sequential Decision-Making in Multi-Agent Systems. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*.

[36] Eric A Hansen, Daniel S Bernstein, and Shlomo Zilberstein. 2004. Dynamic programming for partially observable stochastic games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 4. 709–715.

[37] Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021. CUAD: An Expert-Annotated NLP Dataset for Legal Contract Review. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[38] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. 2023. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*.

[39] Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. 2022. Therapeutics Data Commons: Machine Learning Datasets and Tasks for Drug Discovery and Development. *Proceedings of Neural Information Processing Systems, NeurIPS Datasets and Benchmarks* (2022).

[40] Jonathan Hyun, Nicholas R. Waytowich, and Boyuan Chen. 2025. CREW-WILDFIRE: Benchmarking Agentic Multi-Agent Collaborations at Scale. *arXiv preprint arXiv:2507.05178* (2025). https://arxiv.org/abs/2507.05178

[41] Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2023. SWE-bench: Can Language Models Resolve Real-World GitHub Issues? *arXiv preprint arXiv:2310.06770* (2023).

[42] Charles Jin, Zhang-Wei Hong, Farid Arthaud, Idan Orzech, and Martin Rinard. 2023. Decentralized Inference via Capability Type Structures in Cooperative Multi-Agent Systems. *arXiv preprint arXiv:2304.13957* (2023).

[43] Davinder Kaur, Suleyman Uslu, Kaley J Rittichier, and Arjan Durresi. 2022. Trustworthy artificial intelligence: a review. *ACM Computing Surveys (CSUR)* 55, 2 (2022), 1–38.

[44] Alaa Khamis, Ahmed Hussein, and Ahmed Elmogy. 2015. Multi-robot task allocation: A review of the state-of-the-art. *Cooperative Robots and Sensor Networks* (2015), 31–51.

[45] Aleksandar Krnjaic, Raul D. Steleac, Jonathan D. Thomas, Georgios Papoudakis, Lukas Schäfer, Andrew Wing Keung To, Kuan-Ho Lao, Murat Cubuktepe, Matthew Haley, Peter Börsting, and Stefano V. Albrecht. 2024. Scalable Multi-Agent Reinforcement Learning for Warehouse Logistics with Robotic and Human Co-Workers. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

[46] Anton Kuznietsov, Balint Gyevnar, Cheng Wang, Steven Peters, and Stefano V. Albrecht. 2024. Explainable AI for Safe and Trustworthy Autonomous Driving: A Systematic Review. *IEEE Transactions on Intelligent Transportation Systems (T-ITS)* (2024).

[47] Thomas Kwa, Ben West, Joel Becker, Amy Deng, Katharyn Garcia, Max Hasin, Sami Jawhar, Megan Kinniment, Nate Rush, Sydney Von Arx, Ryan Bloom, Thomas Broadley, Haoxing Du, Brian Goodrich, Nikola Jurkovic, Luke Harold Miles, Seraphina Nix, Tao Lin, Neev Parikh, David Rein, Lucas Jun Koba Sato, Hjalmar Wijk, Daniel M. Ziegler, Elizabeth Barnes, and Lawrence Chan. 2025. Measuring AI Ability to Complete Long Tasks. arXiv:2503.14499 [cs.AI] https://arxiv.org/abs/2503.14499

[48] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. 2025. Tulu 3: Pushing Frontiers in Open Language Model Post-Training. arXiv:2411.15124 [cs.CL] https://arxiv.org/abs/2411.15124

[49] Somchaya Liemhetcharat and Manuela Veloso. 2012. Modeling and learning synergy for team formation with heterogeneous agents. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. 365–374.

[50] Bill Yuchen Lin, Ronan Le Bras, Kyle Richardson, Ashish Sabharwal, Radha Poovendran, Peter Clark, and Yejin Choi. 2025. ZebraLogic: On the Scaling Limits of LLMs for Logical Reasoning. In *International Conference on Machine Learning (ICML)*.

[51] Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. 2023. LLM+P: Empowering Large Language Models with Optimal Planning Proficiency. *arXiv preprint arXiv:2304.11477* (2023). https://arxiv.org/abs/2304.11477 v3, Sep 2024.

[52] Xinzhu Liu, Peiyan Li, Wenju Yang, Di Guo, and Huaping Liu. 2024. Leveraging Large Language Models for Heterogeneous Ad Hoc Teamwork Collaboration. *arXiv preprint arXiv:2406.12224* (2024). https://arxiv.org/abs/2406.12224

[53] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2023. AgentBench: Evaluating LLMs as Agents. arXiv:2308.03688 [cs.AI] https://arxiv.org/abs/2308.03688

[54] B. McMahan et al. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics (AISTATS)*.

[55] N. Mehrabi et al. 2021. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)* 54, 6 (2021), 1–35.

[56] Reuth Mirsky, Ignacio Carlucho, Arrasy Rahman, Elliot Fosong, William Macke, Mohan Sridharan, Peter Stone, and Stefano V. Albrecht. 2022. A Survey of Ad Hoc Teamwork Research. In *European Conference on Multi-Agent Systems (EUMAS)*.

[57] E. Munoz de Cote, A. Lazaric, and M. Restelli. 2006. Learning to Cooperate in Multi-Agent Social Dilemmas. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

[58] John F Nash Jr. 1950. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences* 36, 1 (1950), 48–49.

[59] Samer Nashed and Shlomo Zilberstein. 2022. A survey of opponent modeling in adversarial domains. *Journal of Artificial Intelligence Research* 73 (2022), 277–327.

[60] Duy Nguyen, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2025. Multi-Attribute Steering of Language Models via Targeted Intervention. arXiv:2502.12446 [cs.CL] https://arxiv.org/abs/2502.12446

[61] Frans A Oliehoek and Christopher Amato. 2016. *A concise introduction to decentralized POMDPs*. Springer.

[62] OpenAI. 2024. Introducing OpenAI o1. OpenAI Blog. https://openai.com/index/introducing-openai-o1-preview/

[63] Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative Agents: Interactive Simulacra of Human Behavior. arXiv:2304.03442 [cs.HC] https://arxiv.org/abs/2304.03442

[64] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. arXiv:2305.18290 [cs.LG] https://arxiv.org/abs/2305.18290

[65] Arrasy Rahman, Ignacio Carlucho, Niklas Höpner, and Stefano V. Albrecht. 2023. A General Learning Framework for Open Ad Hoc Teamwork Using Graph-based Policy Learning. *Journal of Machine Learning Research* 24, 298 (2023), 1–74. http://jmlr.org/papers/v24/22-099.html

[66] R. Reddy. 1988. Foundations and Grand Challenges of Artificial Intelligence. *AI Magazine* 9, 4 (1988), 9.

[67] João G. Ribeiro, Gonçalo Rodrigues, Alberto Sardinha, and Francisco S. Melo. 2023. TEAMSTER: Model-based reinforcement learning for ad hoc teamwork. *Artificial Intelligence* 324 (2023), 104013.

[68] Z. Shi et al. 2025. Privacy-Enhancing Paradigms within Federated Multi-Agent Systems. *arXiv preprint arXiv:2503.08175* (2025).

[69] Ola Shorinwa, Zhiting Mei, Justin Lidard, Allen Z. Ren, and Anirudha Majumdar. 2025. A Survey on Uncertainty Quantification of Large Language Models: Taxonomy, Open Research Challenges, and Future Directions. arXiv:2412.05563 [cs.CL] https://arxiv.org/abs/2412.05563

[70] Seongho Son, William Bankes, Sayak Ray Chowdhury, Brooks Paige, and Ilija Bogunovic. 2025. Right Now, Wrong Then: Non-Stationary Direct Preference Optimization under Preference Drift. https://openreview.net/forum?id=PabAln0jjB

[71] Evangelia Spiliopoulou, Riccardo Fogliato, Hanna Burnsky, Tamer Soliman, Jie Ma, Graham Horwood, and Miguel Ballesteros. 2025. Play Favorites: A Statistical Method to Measure Self-Bias in LLM-as-a-Judge. arXiv:2508.06709 [cs.CL] https://arxiv.org/abs/2508.06709

[72] Peter Stone, Gal A Kaminka, Sarit Kraus, and Jeffrey S Rosenschein. 2010. Ad hoc autonomous agent teams: Collaboration without pre-coordination. *Proceedings of the AAAI Conference on Artificial Intelligence* 24, 1 (2010), 1504–1509.

[73] Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D. Nguyen. 2025. Multi-Agent Collaboration Mechanisms: A Survey of LLMs. arXiv:2501.06322

[74] Peter Vamplew, John Yearwood, Richard Dazeley, and Adam Berry. 2008. On the Limitations of Scalarisation for Multi-objective Reinforcement Learning of Pareto Fronts. 372–378. doi:10.1007/978-3-540-89378-3_37

[75] Kristof Van Moffaert and Ann Nowé. 2014. Multi-objective reinforcement learning using sets of Pareto dominating policies. *The Journal of Machine Learning Research* 15, 1 (2014), 3483–3512.

[76] V. Vats et al. 2024. A Survey on Human-AI Teaming with Large Pre-Trained Models. *arXiv preprint arXiv:2403.04931* (2024).

[77] Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, John Quan, Stephen Gaffney, Stig Petersen, Karen Simonyan, Tom Schaul, Hado van Hasselt, David Silver, Timothy Lillicrap, Kevin Calderone, Paul Keet, Anthony Brunasso, David Lawrence, Anders Ekermo, Jacob Repp, and Rodney Tsing. 2017. StarCraft II: A New Challenge for Reinforcement Learning. arXiv:1708.04782 [cs.LG] https://arxiv.org/abs/1708.04782

[78] Caroline Wang, Arrasy Rahman, Ishan Durugkar, Elad Liebman, and Peter Stone. 2024. N-Agent Ad Hoc Teamwork. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

[79] Qian Wang, Tianyu Wang, Zhenheng Tang, Qinbin Li, Nuo Chen, Jingsheng Liang, and Bingsheng He. 2025. MegaAgent: A large-scale autonomous LLM-based multi-agent system without predefined SOPs. In *Findings of the Association for Computational Linguistics: ACL 2025*. 4998–5036.

[80] Yaoxiang Wang, Zhiyong Wu, Junfeng Yao, and Jinsong Su. 2025. TDAG: A Multi-Agent Framework Based on Dynamic Task Decomposition and Agent Generation. *Neural Networks* 185 (2025). https://arxiv.org/abs/2402.10178

[81] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh Pappu, Karl Leswing, and Vijay Pande. 2018. MoleculeNet: a benchmark for molecular machine learning. *Chemical Science* 9, 2 (2018), 513–530.

[82] Frank F. Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Z. Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, Mingyang Yang, Hao Yang Lu, Amaad Martin, Zhe Su, Leander Maben, Raj Mehta, Wayne Chi, Lawrence Jang, Yiqing Xie, Shuyan Zhou, and Graham Neubig. 2025. TheAgentCompany: Benchmarking LLM Agents on Consequential Real World Tasks. arXiv:2412.14161 [cs.CL] https://arxiv.org/abs/2412.14161

[83] Yuancheng Xu, Udari Madhushani Sehwag, Alec Koppel, Sicheng Zhu, Bang An, Furong Huang, and Sumitra Ganesh. 2025. GenARM: Reward Guided Generation with Autoregressive Reward Model for Test-Time Alignment. In *The Thirteenth International Conference on Learning Representations*. https://openreview.net/forum?id=J0qTpmbSbh

[84] Yingxuan Yang, Qiuying Peng, Jun Wang, Ying Wen, and Weinan Zhang. 2024. LLM-based Multi-Agent Systems: Techniques and Business Perspectives. arXiv:2411.14033 [cs.AI] https://arxiv.org/abs/2411.14033

[85] Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024. $\tau$-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. arXiv:2406.12045 [cs.AI] https://arxiv.org/abs/2406.12045

[86] Yihang Yao, Zuxin Cen, Zhepeng Liu, Vladislav Isenbaev, Wei Liu, Zhenghai Steven Wu, Bo Li, and Ding Liu. 2023. Constraint-Conditioned Policy Optimization for Versatile Safe Reinforcement Learning. *arXiv preprint arXiv:2310.03718* (2023).

[87] Junwei Yu, Yepeng Ding, and Hiroyuki Sato. 2025. DynTaskMAS: A Dynamic Task Graph-driven Framework for Asynchronous and Parallel LLM-based Multi-Agent Systems. *arXiv preprint arXiv:2503.07675* (2025). https://arxiv.org/abs/2503.07675

[88] Yuchen Zhang, Vivian Lee, and Peter Stone. 2025. TAGET: Offline Goal-Conditioned Policy Learning for Ad Hoc Teamwork. *Proceedings of the International Conference on Machine Learning (ICML)* (2025).

[89] Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, et al. 2024. SOTOPIA: Interactive evaluation for social intelligence in language agents. In *Twelfth International Conference on Learning Representations*.

[90] Changxi Zhu, Mehdi Dastani, and Shihan Wang. 2024. A survey of multi-agent deep reinforcement learning with communication. *Autonomous Agents and Multi-Agent Systems* 38, 1 (2024), 4.

[91] Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A Question Answering Benchmark on a Hybrid of Tabular and Textual Content in Finance. *arXiv preprint arXiv:2105.07624* (2021).

[92] Kunlun Zhu, Hongyi Du, Zhaochen Hong, Xiaocheng Yang, Shuyi Guo, Zhe Wang, et al. 2025. MultiAgentBench: Evaluating the Collaboration and Competition of LLM Agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL 2025)*. Bangkok, Thailand, 8580–8622. https://aclanthology.org/2025.acl-long.421

# A Multi-Agent Benchmarks

Table 1 compares existing multi-agent benchmarks and limitations for Manager Agent evaluation. Existing multi-agent benchmarks cover a range of task domains, but fail to study all of the key problems of the management setting in one single unified environment.

# B GPT-4.1 vs GPT-5 (Reasoning Performance)

While aggregate scores reveal GPT-5's advantage in goal achievement, they do not explain how differences in reasoning capacity translate into distinct workflow management strategies.

To test whether improvements in base model reasoning capacity translate into stronger Manager Agent performance, we repeated the full evaluation from Section 5.1 with the only change being the underlying model: Chain-of-Thought action selection was driven by GPT-4.1 rather than GPT-5. Workflows, validators, prompts, and metrics were otherwise identical, and each workflow was run across five random seeds with temperature fixed at 1.0. GPT-4o was used as an impartial judge to avoid biasing evaluations [71].

Figure 3 compares GPT-4.1 (solid) and GPT-5 (hatched) across all metrics. The two models behave similarly on preference alignment, constraint adherence, and stakeholder management, which remain in the low-to-moderate range across workflows. The clearest divergence lies in goal achievement (green), where GPT-5 consistently outperforms GPT-4.1 in workflows such as data science analytics, genai feature launch, and pharma product launch. This aligns with GPT-5's enhanced reasoning capacity, where RLVR-style training [48] supports more coherent decomposition and dependency tracking. However, absolute levels remain modest: even GPT-5 rarely exceeds 0.6–0.7 in normalized goal achievement, showing that while reasoning helps, neither model reliably solves full workflows.

Aggregate metrics alone do not explain how policies differ in practice. To analyze execution style, we measured action usage frequencies under the CoT policy (Table 2). GPT-5 executes ~13.5% more actions overall and relies heavily on planning operators: it performs 14.5× more task decompositions, 7.8× more refinements, and 26× more dependency additions than GPT-4.1. By contrast, GPT-4.1 uses 2.4× more messaging, 10× more status queries, and nearly 9× more no-ops. Both assign tasks at similar rates, indicating the difference is not raw activity but rather the style of orchestration.

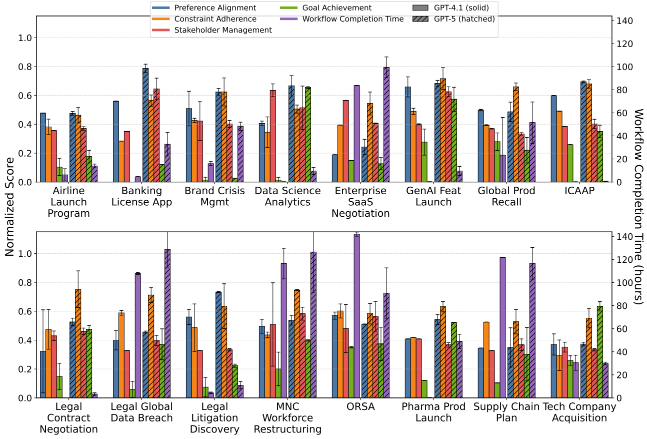

**Figure 3: GPT-4.1 vs. GPT-5 on Manager Agent performance. GPT-5 achieves consistently higher goal achievement through improved reasoning, but absolute levels remain modest and other metrics show little difference.**

Examining action sequences reveals further divergence. GPT-5 frequently builds structured chains such as `decompose` → `refine` → `assign` and `get_agents` → `assign`, reflecting proactive orchestration. GPT-4.1, in contrast, clusters `send_message` → `send_message` and `assign` → `status_check`, reflecting a reactive style centered on communication and monitoring rather than long-horizon planning. These patterns suggest GPT-5 acts as a "proactive orchestrator," while GPT-4.1 behaves more like a "reactive communicator."

These results demonstrate that stronger reasoning models provide a measurable advantage for long-horizon task execution in our environment, but also underscore their limitations. GPT-5's gains in goal achievement are tied to its ability to deploy a richer and more diverse set of planning operators—using decomposition, refinement, and dependency management in structured chains that explore the workflow state space more proactively. GPT-4.1, in contrast, falls back on narrow reactive loops centered on messaging and status checks, reflecting under-exploration of available actions. This suggests that while stronger reasoning models support more proactive orchestration, reasoning alone is insufficient: critical gaps remain in stakeholder alignment and coordination efficiency. RLVR training appears well-suited to sequential decision making, yielding tangible improvements in deliverable completion, but the absence of progress on preference adaptation and stakeholder engagement shows that current reasoning-focused training objectives are not aligned out-of-the-box with the demands of multi-agent workflow management. Reinforcement learning may be a critical ingredient for this setting, but new objectives and signals are required to close the gap between improved reasoning and effective orchestration. In this sense, the observed limitations directly surface the core challenges we have outlined, providing a natural setting in which to investigate them further.

**Table 1: Comparison of existing multi-agent benchmarks and limitations for Manager Agent evaluation.**

| Benchmark | Primary Focus | Capabilities Tested | Limitations for Manager Agent |
|---|---|---|---|
| TheAgentCompany [82] | Real-world office & software workflows | Long-horizon hierarchical planning, mixed tool use, multi-agent collaboration | Does not assess hierarchical task decomposition, dynamic multi-objective optimization, coordination across ad hoc teams, or built-in governance/compliance mechanisms. |
| CREW-Wildfire [40] | Wildfire-response simulation with heterogeneous agents | Dynamic multi-objective optimization, partial observability, large-scale ad-hoc team coordination | Domain-specific to disaster response; emphasizes embodied coordination under uncertainty but omits hierarchical task decomposition, dynamic multi-objective trade-offs, governance/compliance mechanisms, and flexible ad-hoc team formation. |
| MultiAgentBench [92] | LLM collaboration & competition suite | Medium complexity task completion, MAS communication and team topologies | Agent teams are fixed in size with full observability of skills and aptitude, Evaluates single objective fixed tasks without any environmental constraints (cost, fixed agent capacity). |
| StarCraft II [77] | Strategy game with macro and micro management of units and resources | Incomplete information and limited views, dynamic resource management, long-horizon planning and coordination of units, adversarial environment | No mixed human-AI teams, no notion of input/output resources for tasks, no stakeholder communication, no governance/compliance aspects. |
| $\tau$-bench [85] | Evaluation of agent behavior in dynamic, tool-mediated human-agent conversations under domain-specific rules | Human-in-loop interaction; tool/API integration; domain-policy compliance; consistency across trials (via $pass^k$ metric) | Focuses on single-agent tool-use and conversational consistency; does not cover hierarchical task decomposition, multi-objective optimization, ad hoc team coordination, or governance/compliance in multi-agent workflows. |
| SOTOPIA [89],<br>Generative Agents [63] | Social intelligence of LLM-based agents in multi-agent role-play scenarios | Social reasoning; negotiation; collaboration vs. competition; strategic communication; performance in challenging social interaction scenarios (e.g., SOTOPIA-hard) | Not focused on structured workflow orchestration, hierarchical task decomposition, explicit task allocation, dynamic multi-objective optimization, or governance mechanisms. |
| PARTNR [20] | Household planning and embodied human–robot collaboration defined via natural language instructions | Embodied multi-agent planning; spatial, temporal, and heterogeneous capability constraints; human–AI coordination | Lacks hierarchical task decomposition/allocation, multi-objective optimization, and scalable ad hoc team coordination. |
| SoftwareDev [38],<br>ProgramDev [19] | Collaborative software engineering through structured multi-agent workflows | Workflow decomposition via SOP-guided task breakdown; role-specialization in multi-agent team; modular communication | Emphasizes single-domain (software engineering); lacks dynamic multi-objective optimization, ad hoc team formation, governance/compliance constraints, and hierarchical decomposition in multi-agent workflows. |

# C MA-Gym Simulator API and Metric Specifications

## C.1 Core API Components

The simulator in MA-Gym is implemented as a discrete-timestep partially observable stochastic game (POSG) with modular evaluation interfaces. The key components are:

- **WorkflowExecutionEngine:** Central controller managing timestep progression, action execution, and state transitions.
- **ManagerAgent:** Abstract base class for orchestration policies. Our main implementations for baselines are `ChainOfThoughtManager` and `RandomActionManager`.
- **ValidationEngine:** Stateless evaluator applying evaluation rubrics to score workflow runs.
- **Workflow:** Encodes tasks, dependencies, resources, constraints, and communication history.
- **AgentRegistry:** Maintains worker pools, including simulated humans and AI agents.
- **CommunicationService:** Handles inter-agent messaging and stakeholder communication channels.

## C.2 Metric Definitions

We report results across five metrics:

*Preference alignment.* Weighted linear sum of stakeholder preferences, normalized to $[0, 1]$:

$$\text{Preference alignment} = \sum_i w_i \cdot \text{norm}(s_i)$$

**Table 2: Action usage frequencies (CoT policy). GPT-5 emphasizes diverse planning operators (decomposition, refinement, dependency management), forming proactive orchestration chains. GPT-4.1 relies more heavily on messaging, status checks, and no-ops, reflecting a reactive style with narrower exploration of the action space.**

| Action | GPT-4.1 | GPT-5 | Ratio |
|---|---:|---:|---:|
| `assign_task` | 2,594 | 2,882 | 1.1× |
| `decompose_task` | 15 | 217 | 14.5× |
| `refine_task` | 36 | 281 | 7.8× |
| `add_dependency` | 9 | 234 | 26.0× |
| `send_message` | 509 | 213 | 0.4× |
| `get_status/checks` | 126 | 15 | 0.1× |
| `noop` | 107 | 12 | 0.1× |

where $w_i$ is the preference weight and $s_i$ the rubric score. Workflows define 5–7 preferences (e.g., quality, cost, speed, compliance), each with multiple rubrics.

*Constraint Adherence.* A normalized score in $[0, 1]$ designed to measure the Manager Agent's adherence to both soft and hard workflow constraints over the episode (higher is better). If any hard constraint fails, then the overall score is 0, and the workflow is terminated. If no hard constraint violations are present, the score is calculated by checking each task and resource (output of a task) in the workflow to check for soft constraint adherence, removing points for each violation found. Rubrics include constraint coverage, deadline guardrails, prohibited action checks and then workflow specific rubrics such as sign-off verification, and data governance evidence.

*Goal achievement.* Workflow-specific evaluator defined by 10–25 deliverables (critical, major, supporting), with point values reflecting business criticality: critical deliverables (12–18 points), major deliverables (8–12 points), and supporting tasks (3–8 points). Each deliverable is assessed via LLM rubrics using primarily binary prompts (true/false for completion) or graduated scoring with proportional credit for partial fulfillment. Individual rubric scores are combined to a total metric by accumulating the scores across all deliverables. Total scores are normalized to $[0, 1]$, ensuring goal achievement reflects actual deliverable completion rather than planning effort (higher is better).

*Stakeholder management.* Normalized score in $[0, 1]$ designed to quantify how responsive, proactive, and clear the Manager Agent's communication with the stakeholder is (higher is better). Fixed evaluator across all workflows comprising 6 universal rubrics: communication penalties (engagement frequency, assignment load, response latency), coordination quality (graph complexity), and LLM-assessed interaction effectiveness (preference clarification, negotiation). Deterministic rubrics use penalty functions (e.g., $\max(0, 10 - \text{manager\_messages})$ for engagement), while LLM rubrics assess qualitative aspects like preference elicitation and stakeholder input utilization on 8–30 point scales. Aggregation uses zeroing gate: returns 0 if no manager-stakeholder communication occurs; otherwise computes mean of normalized rubric scores. Total scores normalized to $[0, 1]$.

*Workflow Completion Time.* Total simulated hours until workflow has been completed or the timestep cap (100) has been reached. Reported as average across random seeds.

## C.3 Workflow-Specific Preferences

Each workflow defined in MA-Gym defines a unique stakeholder preference structure. For example (preference weights shown in brackets):

- **Legal:** governance (35%), compliance (25%), quality (20%), speed (10%), cost (10%).
- **Finance:** compliance (25–30%), quality (20–25%), speed (15%).
- **Technology:** quality (25%), speed (15–20%), cost (10–15%).

## C.4 Exemplar LLM Rubrics

To make clear the structure of the LLM grading rubrics, we include exemplar rubrics from two workflows: one of setting up a marketing campaign, and another executing a data science project.

*Binary Deliverable Check.*

```
WorkflowRubric(
  name="brand_tracking_framework_operational",
  llm_prompt=(
   "Does an operational brand tracking framework exist with:"
   "measurement approach documented, key metrics defined, "
    "tracking methodology outlined,
    "and reporting framework established? "
    "Return true if all components
    "are documented and ready for deployment, "
    "false otherwise."
  ),
  max_score=18.0,
  run_condition=RunCondition.ON_COMPLETION,
)
```

*Partial Credit Scoring.*

```
WorkflowRubric(
  name="evaluation_rigor",
  llm_prompt=(
    "Evaluate evaluation methodology rigor:\n"
    "- train/validation/test splits with rationale\n"
   "- calibration analysis with quantitative metrics\n"
    "- leakage detection with validation checks\n"
    "- statistically justified thresholds\n"
   "- uncertainty quantification with multiple methods\n"
    "- independent peer review present\n"
   "PENALTY: Deduct 2 points for each missing requirement. "
    "Return score [0,10]."
  ),
  max_score=10.0,
  run_condition=RunCondition.ON_COMPLETION,
)
```

## C.5 Full List of Manager Agent Actions

Table 3 provides a complete list of the allowed actions for all Manager Agent baselines, and their intents. For a full description of arguments, returned observations and mechanics, please refer to the code repository.

## C.6 Evaluation Set Taxonomy

Table 4 lists all 20 workflows defined in the initial release of MA-Gym.

*What “preference change-points” means.* For each workflow with dynamic preferences, we specify the timesteps $t$ at which the preference vector $\mathbf{U}$ changes, and the new weights thereafter. In plain terms: *when* the stakeholder re-prioritizes (the change-point) and *how* the objectives are re-weighted. Unless otherwise noted, “S→Q→C standard pattern” means: early Speed/Time emphasis, mid-run Quality emphasis, late Compliance emphasis, with two change-points at approximately one-third and two-thirds of the action budget (e.g., $t \approx 35$ and $t \approx 70$ on a 100-step horizon; scale proportionally for shorter runs). We list exact change-points where they are encoded in the workflow files; for others we annotate “standard pattern.” Team churn lists worker joins/leaves by timestep.

## C.7 Example Action Buffer (25 Timesteps)

Table 5 shows a representative trajectory for the *Legal Contract Negotiation* workflow scenario. We show 20 steps from the same run and leave 5 placeholders to be filled with the remaining actions.

**Table 3: Detailed set of all actions the Manager Agent is allowed to take, including names, inputs, and a brief description of the intent of the action.**

| Action | Rationale (when to use) |
|---|---|
| (1) `assign_task(task_id, agent_id)` | Route a specific READY task to a capacity/skill-matched agent; avoid for approvals/sign-offs or human-only items. |
| (2) `assign_all_pending_tasks([agent_id])` | Fast triage for demos or low-stakes phases: bulk-assign unassigned, non-completed tasks to one agent (auto-picks a deterministic agent if omitted). |
| (3) `create_task(name, description, est_hrs, est_cost)` | Add concrete work items (artifacts, reviews, approvals) when pipeline is empty, evaluators require evidence, or you need explicit human steps. |
| (4) `remove_task(task_id)` | Prune scope: delete out-of-scope/duplicate/obsolete tasks to reduce clutter and protect the critical path. |
| (5) `send_message(content, [receiver_id])` | Coordinate: solicit tradeoffs, request approvals, clarify requirements, or broadcast instructions; incurs communication/oversight costs in evaluators. |
| (6) `noop()` | Observe without changing state when no safe/productive action exists or you are waiting for information. |
| (7) `get_workflow_status()` | Snapshot health: task status histogram, ready set size, and available agents to inform next scheduling/creation moves. |
| (8) `get_available_agents()` | Inspect who is idle/available and their capability summaries before (re)allocation. |
| (9) `get_pending_tasks()` | Triage backlog: list PENDING tasks and a name preview for quick selection. |
| (10) `refine_task(task_id, new_task_instructions)` | Tighten scope and clarity: rename, update description/estimates, and inject/replace `MANAGER_INSTRUCTIONS:` in execution notes. |
| (11) `add_task_dependency(prereq_id, dep_id)` | Enforce sequencing; guards against self-links and detects circular dependencies before linking. |
| (12) `remove_task_dependency(prereq_id, dep_id)` | Remove obsolete/incorrect prerequisite edges when ordering is no longer needed. |
| (13) `inspect_task(task_id)` | Read-only deep dive into a task's current status/details/outputs; no state changes. |
| (14) `decompose_task(task_id)` | Split a broad task into subtasks using AI, given full workflow context and the workflow seed; skips if already decomposed. |
| (15) `request_end_workflow([reason])` | Signal termination once value is saturated or deliverables accepted; requires a communication service. |
| (16) `failed_action(metadata)` | Record a provider/system failure while leaving the workflow unchanged (diagnostic breadcrumb). |

**Table 4: Workflow taxonomy and validator coverage. One row per workflow with goal, preferences, team churn (i.e. when agents enter/leave), and key validators.**

| Workflow | Domain | Goal (1-line) | Preferences | Team churn | Key validators |
|---|---|---|---|---|---|
| Airline Launch Program | Aviation | New route feasibility → launch plan | quality, cost, speed | PMO joins @ $t \approx 15$; analyst leaves @ $t \approx 45$ | operational readiness; market analysis; safety compliance (LLM-judge) |
| Banking License Application | Finance | Regulatory compliance → license approval | quality, compliance, governance | compliance officer joins @ $t \approx 20$; consultant leaves @ $t \approx 50$ | regulatory completeness; documentation quality; risk assessment |
| Brand Crisis Management | Marketing | Crisis response → reputation recovery | quality, compliance, governance, speed, cost, reputation_recovery | crisis team joins @ $t \approx 5$; PR consultant leaves @ $t \approx 25$ | response timeliness; stakeholder coverage; message consistency (LLM-judge) |
| Data Science & Analytics | Analytics | Explore → model → report | quality, speed, cost | data engineer joins @ $t \approx 20$ | quality; metric sanity; notebook hygiene |
| Enterprise SaaS Negotiation | Sales | Pipeline → proposal → contract | quality, speed, cost | sales engineer joins @ $t \approx 25$; legal counsel leaves @ $t \approx 55$ | contract coverage; pricing validation; compliance checks |
| GenAI Feature Launch | Technology | Feature dev → testing → release | quality, speed, cost | ML engineer joins @ $t \approx 30$; QA leaves @ $t \approx 65$ | feature completeness; safety validation; performance metrics |
| Global Product Recall | Manufacturing | Crisis response → market reentry | consumer_safety, regulatory_compliance, crisis_management, operational_execution, brand_recovery, financial_risk_management, speed | crisis team joins @ $t \approx 0$; recovery team joins @ $t \approx 25$ | safety protocols; regulatory coordination; completion tracking (LLM-judge) |
| ICAAP | Risk | Capital adequacy report draft | quality, compliance, governance, speed, cost | reviewer joins @ $t \approx 40$ | governance completeness; section coverage; risk type coverage |
| IPO Readiness Program | Finance | Regulatory compliance → public listing | sec_compliance, governance, financial_readiness, legal_regulatory, speed, cost | legal counsel joins @ $t \approx 15$; auditor leaves @ $t \approx 35$ | SEC compliance; board independence; audit quality (LLM-judge) |
| Legal Contract Negotiation | Legal | Clause redlines + summary | quality, compliance, governance, speed, cost | counsel joins @ $t \approx 25$; paralegal leaves @ $t \approx 60$ | clause coverage; prohibited terms; summary quality |
| Legal Global Data Breach | Legal | Incident response → report + briefing | quality, compliance, governance, speed, cost | incident response team joins @ $t \approx 0$; external counsel joins @ $t \approx 12$ | evidence preservation; regulatory notifications; privilege protection |
| Legal Litigation e-Discovery | Legal | Collection → culling → memo | quality, compliance, governance, speed, cost | vendor joins @ $t \approx 30$ | source provenance; privilege filters; data validation |
| Legal M&A | Legal | SPA review + risk notes | quality, compliance, governance, speed, cost | associate joins @ $t \approx 35$ | clause coverage; change-of-control checks; due diligence completeness |
| Marketing Campaign | Marketing | Brief → assets → plan | quality, speed, cost | designer joins @ $t \approx 30$ | brand compliance; asset checklist; campaign effectiveness |
| MNC Workforce Restructuring | HR | Strategy → implementation → monitoring | quality, compliance, governance, speed, cost | HR specialist joins @ $t \approx 20$; consultant leaves @ $t \approx 50$ | legal compliance; employee relations; change management |
| ORSA | Risk | Own risk & solvency draft | quality, compliance, governance, speed, cost | actuary joins @ $t \approx 35$ | governance checklist; cross-ref integrity; risk assessment |
| Pharmaceutical Product Launch | Healthcare | Regulatory approval → market launch | quality, speed, cost | regulatory specialist joins @ $t \approx 25$; CMO leaves @ $t \approx 60$ | cGMP compliance; ICH guidelines; safety validation |
| Supply Chain Planning | Operations | Plan → simulate → report | quality, speed, cost | analyst joins @ $t \approx 20$ | SLA conformance; cost envelope; logistics coordination |
| Tech Acquisition Integration | Technology | Due diligence → integration → validation | quality, compliance, governance, speed, cost | integration team joins @ $t \approx 10$; consultant leaves @ $t \approx 45$ | regulatory approval; system integration; talent retention |
| UK University Accreditation | Education | Application → review → certification | quality, compliance, governance, speed, cost | external reviewer joins @ $t \approx 30$; admin leaves @ $t \approx 55$ | academic standards; documentation quality; regulatory compliance |

**Table 5: Action sequence (first 30 actions, numbered 1–30) — ICAAP (GPT-5, `run_id=seed_44`).**

| # | Action | Rationale (short) | Est. hrs | Est. cost | Task ID (prefix) |
|---|---|---|---|---|---|
| 1 | DecomposeTask: 52a0f9e0 (→ 5 subtasks) | Expose parallelism; unblock downstream assignments | — | — | 52a0f9e0 |
| 2 | AssignTask → credit_risk_modeler | Skill-fit allocation to credit capital workstream | — | — | ef0dc5fc |
| 3 | AddDependency: Define Gross Metric Requirements $\prec$ Source Mapping and Extract Quarter-End Data from Authoritative Systems | Enforce ordering; prevent premature starts | — | — | — |
| 4 | AssignTask → credit_risk_modeler | Skill-fit allocation to credit capital workstream | — | — | daf763f1 |
| 5 | AssignTask → documentation_lead | Parallelize drafting and evidence linking | — | — | 00000000 |
| 6 | AssignTask → credit_risk_modeler | Skill-fit allocation to credit capital workstream | — | — | 6de77323 |
| 7 | AssignTask → documentation_lead | Parallelize drafting and evidence linking | — | — | ef574e2d |
| 8 | AssignTask → credit_risk_modeler | Skill-fit allocation to credit capital workstream | — | — | 8916d422 |
| 9 | AssignTask → op_risk_analyst | Advance Operational Risk quant/scenario track | — | — | bda70785 |
| 10 | AssignTask → irrbb_analyst | Advance IRRBB measurement and limit checks | — | — | ede266dd |
| 11 | AssignTask → documentation_lead | Parallelize drafting and evidence linking | — | — | 2fe81eb1 |
| 12 | GetAvailableAgents | Check capacity/skills before next allocations | — | — | — |
| 13 | AssignTask → op_risk_analyst | Advance Operational Risk quant/scenario track | — | — | 00000000 |
| 14 | AssignTask → irrbb_analyst | Advance IRRBB measurement and limit checks | — | — | 00000000 |
| 15 | AssignTask → op_risk_analyst | Advance Operational Risk quant/scenario track | — | — | 00000000 |
| 16 | AssignTask → irrbb_analyst | Advance IRRBB measurement and limit checks | — | — | 00000000 |
| 17 | AssignTask → op_risk_analyst | Balance workload across verticals | — | — | 00000000 |
| 18 | AssignTask → irrbb_analyst | Balance workload across verticals | — | — | 00000000 |
| 19 | AssignTask → stress_testing_designer | Initiate scenario set (baseline/adverse/severe) | — | — | 00000000 |
| 20 | AssignTask → stress_testing_designer | Initiate scenario set (baseline/adverse/severe) | — | — | 00000000 |
| 21 | AssignTask → stress_testing_designer | Initiate scenario set (baseline/adverse/severe) | — | — | 00000000 |
| 22 | AssignTask → documentation_lead | Parallelize drafting and evidence linking | — | — | 00000000 |
| 23 | AssignTask → stress_testing_designer | Initiate scenario set (baseline/adverse/severe) | — | — | 00000000 |
| 24 | AssignTask → credit_risk_modeler | Skill-fit allocation to credit capital workstream | — | — | 00000000 |
| 25 | AssignTask → credit_risk_modeler | Skill-fit allocation to credit capital workstream | — | — | 00000000 |
| 26 | GetAvailableAgents | Check capacity/skills before next allocations | — | — | — |
| 27 | GetAvailableAgents | Check capacity/skills before next allocations | — | — | — |
| 28 | AssignTask → documentation_lead | Parallelize drafting and evidence linking | — | — | 00000000 |
| 29 | GetAvailableAgents | Check capacity/skills before next allocations | — | — | — |
| 30 | AssignTask → capital_planner | Kick off normative capital planning work | — | — | 00000000 |