How can a system orchestrator utilize Large Language Models to autonomously decompose a mission into sub-tasks and allocate them based on the differing hardware constraints of multiple robotic agents?

A system orchestrator can utilize Large Language Models to autonomously decompose missions into sub-tasks and allocate them based on hardware constraints by employing hybrid architectures that combine centralized LLM reasoning with distributed execution, using dependency graphs or hierarchical structures for task decomposition, integrating robot capability information through structured prompts or ”robot resume” representations, and pairing LLM-based reasoning with classical optimization methods such as integer programming or PDDL planners to ensure allocation decisions respect hardware limitations while maintaining coordination across heterogeneous robot teams.

Abstract

This systematic review of 80 sources reveals that LLM-based system orchestrators can effectively decompose missions and allocate tasks across heterogeneous robot teams through three primary architectural approaches: centralized, decentralized, and hybrid systems. Hybrid architectures combining centralized LLM oversight with distributed execution consistently achieve the highest success rates and scalability . Mission decomposition is accomplished through dependency-aware techniques including Directed Acyclic Graphs for modeling task precedence , hierarchical tree structures , and Chain-of-Thought prompting for structured reasoning . Hardware constraints are integrated into LLM reasoning through several mechanisms: ”robot resume” approaches that generate capability descriptions from URDF files , skill-based representations encoded in structured prompts , and dynamic capability vectors enabling real-time state tracking . Systems achieve success rates of 94-100% on structured tasks when combining LLM reasoning with classical optimization methods such as integer programming or PDDL planners , outperforming pure LLM approaches in allocation optimality .

Key limitations constrain practical deployment: centralized systems face context window limitations that degrade performance beyond approximately 10 robots , LLM hallucinations compromise reliability in safety-critical scenarios , and computational demands often preclude on-robot execution, requiring cloud connectivity or model compression techniques . Effective systems address these challenges through feedback mechanisms such as the PEFA loop and digital twin synchronization that enable dynamic re-planning, human-in-the-loop verification for safety assurance , and hybrid LLM-optimization pipelines that leverage LLM flexibility for decomposition while ensuring allocation optimality through formal methods . The evidence strongly supports that autonomous mission decomposition and hardware-aware allocation is achievable with current LLM capabilities, though robust deployment requires careful integration with classical planning and optimization techniques rather than reliance on LLM reasoning alone.

Paper search

We performed a semantic search using the query ”How can a system orchestrator utilize Large Language Models to autonomously decompose a mission into sub-tasks and allocate them based on the differing hardware constraints of multiple robotic agents?” across over 138 million academic papers from the Elicit search engine, which includes all of Semantic Scholar and OpenAlex.

We retrieved the 500 papers most relevant to the query.

Screening

We screened in sources based on their abstracts that met these criteria:

- **Multi-Robot System**: Does the study involve multi-robot systems with heterogeneous agents (robots with differing hardware constraints or capabilities)?
- **Large Language Model Integration**: Does the research utilize Large Language Models for task planning, decomposition, or allocation in robotic systems?
- **Autonomous Task Decomposition**: Does the study examine autonomous or semi-autonomous mission/task decomposition and allocation (rather than exclusively human-in-the-loop approaches)?
- **Study Type Relevance**: Is this an experimental study, case study, simulation study, theoretical framework, or systematic review related to LLM applications in multi-robot coordination?
- **Publication Quality**: Is this a peer-reviewed publication (not an opinion piece, editorial, or non-peer-reviewed publication)?

We considered all screening questions together and made a holistic judgement about whether to screen in each paper.

Data extraction

We asked a large language model to extract each data column below from each paper. We gave the model the extraction instructions shown below for each column.

- **System Architecture**:

  Extract details about the overall system architecture including:

- How LLMs are integrated into the orchestration system
- Key system modules/components and their roles
- Communication flows between components
- Whether LLMs serve as central planners, distributed agents, or hybrid approaches
- Integration with classical control algorithms or optimization methods
- Multi-tier vs single-tier architectures

- **Mission Decomposition Method**:

  Extract specific techniques used to autonomously decompose missions into sub-tasks including:

- LLM prompting strategies or reasoning approaches
- Use of hierarchical structures, dependency graphs, or tree decompositions
- Natural language processing methods for mission understanding
- Handling of task dependencies and ordering constraints
- Multi-stage reasoning or iterative refinement processes
- Support for different mission specification formats (natural language, LTL, etc.)

- **Hardware Constraint Modeling**:

  Extract how the system models and accounts for different robot hardware constraints including:

- Types of hardware constraints considered (computational, sensing, mobility, payload, etc.)
- Methods for representing robot capabilities and limitations
- How capability information is integrated into the LLM reasoning process
- Approaches for handling heterogeneous robot teams
- Dynamic adaptation to changing hardware states or failures

- **Task Allocation Mechanism**:

Extract specific algorithms and methods used for assigning sub-tasks to robots including:

- Optimization algorithms or heuristics used
- Criteria for allocation decisions (time efficiency, resource usage, robot suitability)
- Real-time vs offline allocation approaches
- Handling of parallel execution and task dependencies
- Methods for load balancing and resource utilization
- Integration of LLM reasoning with mathematical optimization

- **Robot Types & Capabilities**:

  Extract details about the robotic agents used including:

- Specific types of robots (drones, ground robots, manipulators, etc.)
- Hardware capabilities and constraints addressed
- Heterogeneity in the robot team
- Computational resources available on robots
- Sensing and actuation capabilities
- Communication constraints or requirements

- **Performance Evaluation**:

  Extract quantitative and qualitative performance results including:

- Success rates, completion times, accuracy metrics
- Comparison with baseline methods or competing approaches
- Scalability results (number of robots, task complexity)
- Efficiency metrics (communication overhead, computational costs, API calls)
- Quality of task decomposition and allocation
- Robustness to dynamic changes or failures
- Any limitations or failure modes identified

- **Dynamic Adaptation**:

  Extract approaches for handling changing conditions and online adaptation including:

- Methods for detecting new tasks or environmental changes
- Re-planning and re-allocation strategies
- Human-in-the-loop mechanisms
- Event-based vs periodic updates
- Handling of unforeseen site conditions or task modifications
- Real-time feedback integration and system responsiveness

- **LLM Implementation Details**:

  Extract specific details about LLM usage including:

- Which LLM models were used (GPT variants, Llama, custom models)
- Model sizes and computational requirements
- Prompting strategies and prompt engineering approaches
- Use of multiple LLM agents (generator/supervisor, specialized roles)
- Integration with Vision-Language Models or other AI components
- Strategies for reducing LLM queries or improving efficiency

Report

Due to the limitations of the AI model, we are only able to process 80 sources while writing a report. This report was written using the 80 sources that had the highest screening scores out of the 444 sources that we screened in and extracted data from.

Characteristics of Included Studies

This systematic review synthesizes findings from 80 sources examining the use of Large Language Models for autonomous mission decomposition and task allocation in multi-robot systems. The studies span various application domains and employ diverse methodological approaches.

Study	Full text retrieved?	Study Type	Primary Focus	Robot Types
Yongdong Wang et al., 2024	Yes	Primary study	Dependency-aware task decomposition	Tracked robots, excavators
Tinging Yang et al., 2025	No	Primary study	Heterogeneous multi-agent coordination	Drones, ground robots
Zhehui Huang et al., 2025	Yes	Primary study	Compositional multi-robot coordination	Not specified
Wenhao Yu et al., 2024	Yes	Primary study	Decentralized heterogeneous collaboration	Mobile, manipulation, mobile-manipulation
Jun Wang et al., 2024	No	Primary study	Safe task planning with conformal prediction	Not specified
Yongchao Chen et al., 2023	Yes	Primary study	Centralized vs decentralized planning comparison	Robot arms, mobile manipulators
Xiaopan Zhang et al., 2024	Yes	Primary study	LLM-PDDL integration for long-horizon tasks	Ground robots/manipulators
Jun Wang et al., 2024a	Yes	Primary study	Probabilistically correct planning	Ground robots/manipulators
Min Deng et al., 2025	Yes	Primary study	Construction robotics with digital twins	Manipulators
Kehui Liu et al., 2024	Yes	Primary study	Heterogeneous multi-robot collaboration	Quadrotors, robotic dogs, arms

Study	Full text retrieved?	Study Type	Primary Focus	Robot Types
Zhaoxing Li et al., 2025	Yes	Primary study	Human-in-the-loop multi-robot framework	Rovers, robotic dogs
Zhao Mandi et al., 2023	Yes	Primary study	Dialectic multi-robot collaboration	Manipulators (UR5E)
Fernando Cladera et al., 2025	No	Primary study	Air-ground collaboration	UAV, UGV
Abhinav Rajvanshi et al., 2025	Yes	Primary study	Decentralized navigation	Heterogeneous robots
Yuwei Wu et al., 2024	Yes	Primary study	Hierarchical optimization for target tracking	Drones, ground robots
Dan BW Choe et al., 2025	Yes	Primary study	LLM-to-temporal-logic framework	Drones, forklifts
Ziyao Wang et al., 2025	Yes	Primary study	Role-adaptive UAV navigation	UAVs (quadrotors)
Peihan Li et al., 2025	Yes	Survey	Comprehensive MRS-LLM survey	Various
Junting Chen et al., 2024	Yes	Primary study	Embodiment-aware heterogeneous MRS	Drones, wheeled/legged robots
Tengchao Zhang et al., 2025	Yes	Primary study	Coordination field for UAV task allocation	UAVs
Haolin Li et al., 2024	No	Primary study	Heterogeneous agent team games	Reconnaissance, grabbing, blocking robots
Kaushik Kannan et al., 2025	Yes	Primary study	Search and rescue task allocation	Heterogeneous team
Abdelhaleem Saad et al., 2025	No	Primary study	Multi-ROV aquaculture inspection	ROVs
Haokun Liu et al., 2025	No	Primary study	Aerial-ground semantic navigation	Aerial, ground robots
Chaoran Wang et al., 2025	No	Primary study	Dynamic behavior tree construction	Robotic arm, wheeled-legged robot
Wen Zhao et al., 2024	No	Primary study	Multi-robot control with GPT	UAVs, UGVs
Xinzhu Liu et al., 2024	No	Primary study	Ad hoc heterogeneous teamwork	Not specified
Nan Li et al., 2025	No	Primary study	Obstacle-aware task planning	Not specified

Study	Full text retrieved?	Study Type	Primary Focus	Robot Types
A. Khan et al., 2025	Yes	Primary study	Safety-aware task planning	Robot arms, quadrotor, robotic dog
Seoyeon Choi et al., 2025	No	Primary study	MARL with foundation models	Quadruped, manipulators
Ike Obi et al., 2025	Yes	Primary study	Safety enhancement framework	Not specified
Ruiyang Wang et al., 2025	No	Primary study	Multi-robot exploration and search	LiDAR-equipped robots
Daniel Weiner et al., 2025	No	Primary study	Modular construction task assignment	Not specified
Yoshiki Yano et al., 2025	No	Primary study	Instruction-conditioned coordination	Not specified
Piyush Gupta et al., 2025	No	Primary study	Hierarchical tree mission planning	Heterogeneous team
Kento Murata et al., 2025	No	Primary study	Multi-object retrieval planning	Mobile manipulators
Yuxiao Zhu et al., 2025	Yes	Primary study	Dynamic explainable coordination	Drones, carts
Kazuma Obata et al., 2024	Yes	Primary study	Linear programming with LLM	Arm robots, mobile robots
Siddharth Nayak et al., 2024	Yes	Primary study	Long-horizon multi-agent planning	Ground robots
Alkesh K. Srivastava et al., 2025	Yes	Primary study	Voronoi-based relay planning	TurtleBot3
Xiangkun Deng et al., 2025	No	Primary study	General multi-agent task planning	Not specified
Weizheng Wang et al., 2025	Yes	Primary study	Social robot navigation	Mobile robots, robot dogs, drones
S. S. Kannan et al., 2023	Yes	Primary study	SMART-LLM framework	Aerial, ground robots
Artem Lykov et al., 2023	Yes	Primary study	Behavior tree generation	Ground robots
Bin Zhang et al., 2023	Yes	Primary study	Actor-critic multi-agent control	Not specified
Jinqiang Cui et al., 2024	No	Primary study	Multi-UAV task planning	UAVs
Hongxin Zhang et al., 2023	Yes	Primary study	Cooperative embodied agents	Ground robots/manipulators

Study	Full text retrieved?	Study Type	Primary Focus	Robot Types
Marcos Abel Zuzu'arregui et al., 2025	No	Primary study	Precision agriculture planning	Wheeled robots, manipulators
Kartik Nagpal et al., 2025	Yes	Primary study	Multi-agent credit assignment	Not specified
Hao Sha et al., 2023	Yes	Primary study	Autonomous driving MPC	Autonomous vehicles
William Hunt et al., 2024	Yes	Primary study	Conversational multi-robot coordination	TurtleBot3
Mukund Mitra et al., 2025	No	Primary study	Heterogeneous system scheduling	Not specified
Zhiwei Liu et al., 2023	Yes	Primary study	LLM-augmented autonomous agents	Not robotic (software agents)
Rui Yang et al., 2025	No	Primary study	Distributed MAPF with LLMs	Not specified
Huibo Zhang et al., 2024	No	Primary study	In-context learning for task allocation	Not specified
Enrico Saccon et al., 2025	Yes	Primary study	Temporal planning with knowledge base	UR3e, UR5e manipulators
Zachary Ravichandran et al., 2024	No	Primary study	Online semantic planning	Drones, ground robots
Marc Glocker et al., 2025	Yes	Primary study	Memory-augmented household robotics	Embodied robots
Oleg Sautenkov et al., 2025	Yes	Primary study	Multi-UAV mission generation	UAVs
Peihan Li et al., 2025a	Yes	Primary study	Decentralized flocking control	Crazyflie drones
Hsu-Shen Liu et al., 2024	No	Primary study	Language-guided pattern formation	Not specified
Michael Ahn et al., 2024	Yes	Primary study	Large-scale robot orchestration	Mobile manipulators
Xihe Qiu et al., 2024	Yes	Primary study	Intention propagation for coordination	Functional agents
Ziqi Jia et al., 2025	Yes	Primary study	LLM-graph MARL integration	Ground robots
Harisankar Babu et al., 2025	Yes	Primary study	Adaptive domain modeling	Not specified
Shaojun Xu et al., 2024	No	Primary study	Hierarchical LTL specifications	Not specified
Shuai Jia et al., 2024	No	Primary study	Multi-UAV adversarial/cooperative tasks	Multi-UAV systems

Study	Full text retrieved?	Study Type	Primary Focus	Robot Types
Guobin Zhu et al., 2025	Yes	Primary study	LLM-aided MARL	Omnidirectional ground robots
Steven D. Morad et al., 2024	Yes	Primary study	Language-conditioned offline RL	DJI RoboMaster
Minghong Geng et al., 2025	Yes	Primary study	Hierarchical LLM-MARL framework	Not specified
Volker Strobel et al., 2024	No	Primary study	LLM2Swarm robot swarms	Robot swarms
Yibo Qiu et al., 2025	Yes	Primary study	Biological experiment automation	Dual-arm manipulators
Junwei Yu et al., 2025	Yes	Primary study	Dynamic task graph framework	Not specified
Wenjie Lin et al., 2025	No	Primary study	Metacognitive learning for planning	Not specified
Michele Grimaldi et al., 2025	No	Primary study	Underwater multi-agent autonomy	Not specified
Kun Chu et al., 2025	No	Primary study	Bimanual robot planning	Bimanual manipulators
Huaiyuan Yao et al., 2024	Yes	Primary study	Mixed-autonomy traffic coordination	Autonomous vehicles
Wenkang Ji et al., 2025	No	Primary study	Scalable code-policy generation	Not specified
Lillian Wassim et al., 2024	No	Primary study	Drone-as-a-Service operations	Drones
Jiabao Ji et al., 2025	No	Primary study	Collision-aware multi-robot control	Not specified

The studies predominantly represent primary research contributions, with one comprehensive survey . Approximately 50% of studies provided full text access, while the remainder were analyzed through abstracts. The research covers diverse robotic platforms including UAVs, ground robots, manipulators, and heterogeneous multi-robot teams operating across domains such as household robotics, construction, search and rescue, agriculture, and autonomous transportation.

System Architectures for LLM-Based Multi-Robot Orchestration

The integration of LLMs into multi-robot systems follows three primary architectural paradigms: centralized, decentralized, and hybrid approaches. Each architecture presents distinct trade-offs between coordination efficiency, scalability, and robustness.

Centralized Architectures

Centralized systems employ a single LLM as the primary decision-maker for mission decomposition and task allocation. The SMART-LLM framework uses LLMs as central planners to autonomously decompose missions into sub-tasks and allocate them based on robot skills and environment details . Similarly, LaMMA-P integrates LLMs as a central component for task decomposition and allocation, generating PDDL problem descriptions for downstream planning . The COHERENT framework implements a centralized hierarchical structure where a centralized task assigner decomposes tasks and assigns them to distributed robot executors .

AutoRT demonstrates large-scale centralized orchestration, with LLMs generating tasks based on visual observations for deployment across over 20 robots . The EMOS framework employs a hierarchical centralized approach where a leader LLM agent coordinates with robot-specific agents through a star topology .

Decentralized Architectures

Decentralized systems distribute LLM reasoning across individual robots, enabling autonomous local decision-making. The MHRC framework supports decentralized collaboration where each robot type has independent planning capabilities through LLM-driven decision modules . S-ATLAS implements a decentralized LLM-based planner where each robot uses its own LLM agent to select actions based on context and previous decisions .

The HMCF framework equips each robot with an LLM agent capable of understanding its capabilities and converting tasks into executable instructions . LLM-Flock provides each robot with its own LLM for local planning, combined with an influence-based consensus protocol for coordination . The SAMALM framework employs a decentralized multi-agent LLM actor-critic structure where parallel LLM actors generate control signals independently .

Hybrid Architectures

Hybrid architectures combine centralized oversight with distributed execution, emerging as the dominant approach across studies. The comparative study by Chen et al. demonstrates that hybrid frameworks (HMAS-1 and HMAS-2) achieve better task success rates and scale better to more agents than purely centralized or decentralized approaches .

AutoHMA-LLM implements a multi-tier architecture utilizing a cloud-based LLM as the central planner alongside device-specific LLMs . The hierarchical LLM framework by Wu et al. integrates LLMs into multiple feedback loops with conventional optimizers, where an outer loop provides strategic guidance and an inner loop handles reactive adaptations . DART-LLM uses a hybrid approach combining LLMs with classical control algorithms for navigation and robot-specific skills .

The RoCo framework employs LLMs for both high-level communication and low-level path planning, with centralized RRT-based motion planning for parallel execution . The DELIVER framework combines high-level LLM-based planning with low-level FSM-based execution through Voronoi tessellation for spatial decomposition .

Architecture Type	Representative Systems	Key Characteristics	Scalability
Centralized	SMART-LLM , LaMMA-P , AutoRT	Single LLM decision-maker, simplified coordination	Limited by context window
Decentralized	MHRC , S-ATLAS , HMCF	Independent robot LLMs, local decision-making	Better with more agents

Architecture Type	Representative Systems	Key Characteristics	Scalability
Hybrid	HMAS-2 , AutoHMA-LLM , DART-LLM	Central oversight with distributed execution	Best overall

The hybrid HMAS-2 structure achieves the highest success rate while maintaining reasonable token efficiency . This architectural pattern is particularly effective for long-horizon, heterogeneous multi-robot planning where both global coordination and local autonomy are essential.

Mission Decomposition Methods

The autonomous decomposition of high-level missions into executable sub-tasks represents a core capability enabled by LLMs in multi-robot systems. The reviewed literature reveals several distinct approaches to mission decomposition.

Hierarchical and Graph-Based Decomposition

Dependency graphs and hierarchical structures are widely employed to represent task relationships. DART-LLM uses Directed Acyclic Graphs (DAGs) to model subtask dependencies, enabling parallel execution of independent subtasks . Similarly, LiP-LLM constructs dependency graphs using LLMs to map sequential constraints among skills, with linear programming optimizing task allocation .

The LAN2CB framework parses missions into task graphs with dependencies through a Mission Decomposition component, capturing execution topologies through behavior tree structures . DEXTER-LLM employs directed acyclic graphs for task representation with temporal constraints derived through multi-stage LLM prompting . The Nl2Hltl2Plan framework transforms natural language into Hierarchical Task Trees capturing logical and temporal relations before converting sub-tasks into flat LTL formulas .

Gupta et al. propose using hierarchical trees systematically constructed by LLMs to break down complex missions into manageable sub-tasks, with specialized APIs facilitating tree construction .

Prompting Strategies for Decomposition

Chain-of-Thought (CoT) prompting emerges as a dominant technique for structured reasoning during task decomposition. The GMATP-LLM framework uses CoT prompting to transform high-level task instructions into sets of sub-tasks . RALLY implements a two-stage structured prompting approach with local intention generation and neighborhood consensus refinement .

Few-shot prompting strategies are employed across multiple systems. LaMMA-P uses few-shot filling prompts for task decomposition and allocation . Murata et al. design a novel few-shot prompting strategy enabling LLMs to infer required objects from ambiguous commands and decompose them into appropriate subtasks . SMART-LLM uses Pythonic prompts with detailed comments to guide task decomposition .

The Zero-shot-CoT procedure combined with cue word iteration methods enhances decision-making efficiency for adversarial multi-robot games . SafePlan employs Prompt Sanity COT Reasoner and Invariant COT Reasoner to evaluate task prompts through multiple verification stages .

Multi-Stage Reasoning Processes

Iterative refinement processes characterize many decomposition approaches. The PEFA (Proposal-Execution-Feedback-Adjustment) mechanism in COHERENT enables iterative plan adjustment based on execution feedback . RoCo validates plans step-by-step with environmental feedback until valid plans are achieved .

LLaMAR implements a plan-act-correct-verify framework where the Planner suggests subtasks, the Actor predicts actions, the Corrector self-corrects based on failures, and the Verifier assesses completion . The CRAFT framework uses VLM-guided reward-refinement loops for iterative task decomposition .

Decomposition Approach	Representative Methods	Task Dependency Handling	Mission Formats Supported
DAG-based	DART-LLM , LiP-LLM , DEXTER-LLM	Explicit graph constraints	Natural language
Hierarchical trees	LAN2CB , Gupta et al.	Tree structure ordering	Natural language
CoT prompting	GMATP-LLM , RALLY	Implicit through reasoning	Natural language
LTL integration	Nl2Hltl2Plan , SafePlan	Temporal logic formulas	Natural language, LTL
PDDL integration	LaMMA-P , PLANTOR	PDDL problem constraints	Natural language, PDDL

Support for Multiple Mission Specification Formats

Several frameworks support translation between mission specification formats. Nl2Hltl2Plan translates natural language commands into hierarchical Linear Temporal Logic . SafePlan uses LTL for formalizing safety properties and supports natural language inputs . The PLANTOR framework converts plans into Behaviour Trees for direct use in ROS2 .

Dan BW Choe et al. transform natural language requests into Signal Temporal Logic (STL) specifications using BNF grammar, which are then solved as Mixed Integer Linear Programs . DEXTER-LLM translates mission objectives into verifiable LTL formulas for abstraction of tasks and temporal relations .

Hardware Constraint Modeling

Effective task allocation in heterogeneous multi-robot systems requires accurate representation of robot capabilities and limitations. The reviewed literature presents diverse approaches to modeling and integrating hardware constraints into LLM reasoning processes.

Types of Hardware Constraints Addressed

The studies address multiple categories of hardware constraints:

Mobility constraints are most commonly considered, including movement capabilities, traversability, and navigation skills . The HMCF framework tracks robot specifications including traversability and the ability to ascend/descend stairs .

Payload and manipulation constraints are addressed through skill-based representations. Chen et al. define different lifting capabilities for robot arms . SMART-LLM includes payload constraints such as maximum mass a robot can pick up . The SAFER framework considers joint position limits, velocity limits, and torque limits .

Sensing constraints are incorporated through observation specifications. The MHRC framework includes sensing capabilities in its observation module . CoordField represents patrol UAVs with wide-area scanning and tracking UAVs with precise target following capabilities .

Computational constraints affect LLM deployment strategies. LLM-MARS notes that robots' hardware is often not capable of running LLMs locally, requiring remote server execution . RALLY addresses resource-constrained architectures through lightweight LLM versions .

Methods for Representing Robot Capabilities

Robot Resume Approach : The EMOS framework introduces "Robot Resume," where agents comprehend robot URDF files and call robot kinematics tools to generate descriptions of physics capabilities . This self-prompted approach creates textual summaries and numerical representations of mobility, perception, and manipulation capabilities .

Skill-Based Representations : DART-LLM defines skill sets for each robot and team, with task assignment based on individual robot skills . S-ATLAS represents capabilities as textual skills (e.g., 'take a picture', 'grab', 'go to') . SMART-LLM encodes robot skills as Python dictionaries with specific constraints .

Capability Vectors : CoordField uses a dynamic capability vector c_i(t) representing each UAV's current capabilities . The digital twin framework by Deng et al. employs a capability model defining task requirement constraints based on available and required capabilities .

Natural Language Descriptions : The LLM-to-TL framework represents capabilities in natural language (e.g., "can lift pallets" or "max speed 1 m/s") . SayCoNav shares background information about each robot's skills and operational constraints through text .

Constraint Type	Representation Method	Example Systems
Mobility	Skill sets, capability vectors	DART-LLM , CoordField
Payload	Maximum limits, skill constraints	SMART-LLM , Chen et al.
Sensing	Observation specifications	MHRC , RALLY
Computational	Remote execution, model compression	LLM-MARS , RALLY
Communication	Protocol specifications, connectivity	DELIVER , LLM-Flock

Integration with LLM Reasoning

Hardware constraints are integrated into LLM reasoning through several mechanisms:

Structured Prompts : The hierarchical LLM framework provides comprehensive information about robot capabilities and current status to the task LLM through structured prompts . LaMMA-P integrates capability information through generation of PDDL problem descriptions for each robot's domain .

Context Aggregation : COHERENT fuses robot capabilities with observations and task requirements into long text prompts for the task assigner LLM . EMOS uses robot resumes in prompts for embodiment-aware reasoning .

Feedback Mechanisms : The digital twin framework enables closed-loop feedback for real-time updates and task allocation refinement based on changing hardware states . HMCF implements regular status updates from robots to LLM agents for task reallocation .

Handling Heterogeneous Robot Teams

Approaches for managing heterogeneity include:

Coalition Formation : SMART-LLM addresses skill gaps by involving additional robots through coalition formation policies . EMOS assigns tasks based on embodiment-aware reasoning using robot resumes .

Dynamic Role Assignment : CoordField divides UAVs into patrol and tracking types with differing sensing capabilities . SayCoNav automatically generates collaboration strategies leveraging diverse robot skills .

Capability Matching : The digital twin framework ensures collective possession of required capabilities by assigned robot teams . LaMMA-P uses a modular design for flexible task decomposition based on varying robot skills .

Dynamic Adaptation to Hardware States

Several systems implement mechanisms for adapting to changing hardware conditions:

Real-time Status Monitoring : HMCF requests regular updates from robots regarding status and task progress . The digital twin framework provides real-time synchronization between physical and digital models .

Failure Detection and Compensation : AquaChat++ incorporates thruster fault detection and compensation mechanisms with event-triggered replanning . SayCoNav re-generates collaboration strategies when a robot's physical condition changes during navigation .

Task Allocation Mechanisms

Task allocation represents the critical bridge between mission decomposition and execution. The reviewed literature presents diverse algorithmic approaches ranging from optimization-based methods to pure LLM reasoning.

Optimization-Based Approaches

Integer and Linear Programming : The digital twin framework by Deng et al. uses an Integer Programming model solved with a CP-SAT solver, optimizing makespan while penalizing robot assignment counts . LiP-LLM employs linear programming for task allocation based on skill feasibility and distance metrics . PLANTOR uses mixed-integer linear programming for resource allocation and temporal planning .

Mathematical Optimization Integration : Dan BW Choe et al. solve Signal Temporal Logic specifications as Mixed Integer Linear Programs using Gurobi . The hierarchical LLM framework formulates a bi-level optimization problem with LLMs modifying both levels .

Search-Based Methods : DEXTER-LLM employs branch-and-bound search with integer programming to minimize maximum ending time while respecting constraints . LAN2CB uses a minimal conflict strategy to minimize trajectory intersections during goal allocation .

Heuristic and Learning-Based Approaches

Auction Mechanisms : MTU-LLM implements K-means clustering for task grouping with distance-based bidding for matching victim requirements to robot capabilities . The OATH framework uses weighted auctions within a cluster-auction-selection framework .

Reinforcement Learning Integration : RALLY employs an RMIX-based credit-assignment mechanism combining LLM offline priors with MARL online policies . ICCO uses Multi-Agent Reinforcement Learning jointly trained with LLM-generated instructions .

Consensus Protocols : LLM-Flock uses an influence-based consensus protocol where robots negotiate and adopt plans based on influence scores . RALLY implements a two-stage consensus generation strategy for distributed decision-making .

LLM-Native Allocation

Several systems rely primarily on LLM reasoning for allocation decisions:

Direct LLM Allocation : SMART-LLM generates executable code for task planning through Pythonic prompts without separate optimization . RoCo uses dialog-based planning with feasibility checks through LLM reasoning .

Hybrid LLM-Optimization : LaMMA-P integrates LLM-based task decomposition with PDDL planners for detailed action sequences . GMATP-LLM combines LLM reasoning capabilities with intelligent PDDL planners .

Allocation Method	Optimization Approach	Real-time Capability	Dependency Handling
Integer Programming	IP/MILP solvers	Limited	Explicit constraints
Linear Programming	LP optimization	Moderate	Skill feasibility
Auction-based	Distance bidding	Good	Clustering
MARL-integrated	RL optimization	Good	Learned
LLM-native	Prompt-based	Good	LLM reasoning
Search-based	Branch-and-bound	Limited	Graph constraints

Real-Time vs Offline Allocation

Real-time Systems : SayCoNav continuously updates plans based on shared information during navigation . COHERENT uses real-time allocation through the PEFA mechanism . The DELIVER framework implements real-time allocation with finite-state machines for coordination .

Offline/Batch Processing : HVBTA uses pre-trained LLMs for suitability assessment in modular construction . MTU-LLM focuses on offline allocation ensuring workload balance .

Hybrid Approaches : The hierarchical LLM framework updates the task LLM every 20-30 seconds while the action LLM updates every 5 seconds . DEXTER-LLM dynamically adapts to new tasks through event-based module retriggering .

Parallel Execution and Load Balancing

Parallel Task Management : DART-LLM enables parallel execution of independent subtasks through DAG-based decomposition . LAN2CB generates code simultaneously for multiple action nodes in behavior trees . DynTaskMAS uses sophisticated scheduling algorithms to maximize parallelism while respecting dependencies .

Load Balancing : DELIVER reduces per-agent workload by up to 55% compared to single-agent systems while maintaining low coordination overhead . MTU-LLM demonstrates better workload balance compared to baseline approaches . DynTaskMAS employs a greedy allocation strategy to balance load across agents .

Performance Evaluation

The reviewed studies present diverse performance metrics demonstrating the effectiveness of LLM-based multi-robot orchestration systems.

Success Rates and Accuracy

Success rates vary significantly across systems and task complexity levels:

System	Task Type	Success Rate	Baseline Comparison
DART-LLM	L1/L2/L3 tasks	100%/97%/94%	Outperforms SMART-LLM
LaMMA-P	Household tasks	105% higher than baselines	vs. SMART-LLM, CoT
S-ATLAS	Multi-robot planning	94.59-96.57%	vs. CMAS, DMAS
HMCF	Heterogeneous tasks	92.4%	4.76% improvement
LLaMAR	Long-horizon tasks	30% higher	vs. state-of-the-art
DEXTER-LLM	All scenarios	100%	3x more tasks than baselines
SafePlan	Safety verification	84-95%	90.5% reduction in harmful prompts

SMART-LLM achieves 70% success in compound and complex tasks with GPT-4 and 100% in elemental tasks . The Murata et al. framework achieves 47/50 successful assignments compared to 28/50 for random assignment . LLM-MARS demonstrates 79.28% average task execution accuracy for compound commands, exceeding 90% for commands with up to two tasks .

Efficiency Metrics

Communication Efficiency : AutoHMA-LLM achieves a 46% reduction in communication steps and 31% decrease in token usage . S-ATLAS requires fewer LLM queries compared to centralized approaches . DEXTER-LLM requires 62% fewer LLM queries during adaptation .

Computational Efficiency : MTU-LLM demonstrates a tenfold reduction in average computation time (0.033s vs 0.596s) . DynTaskMAS achieves 21-33% reduction in execution time across task complexities with 35.4% improvement in resource utilization . LAMARL improves sample efficiency by 185.9% through prior policy generation .

Planning Time : LiP-LLM shows shorter process times through combinatorial optimization . The average runtime for S-ATLAS to design a plan is 0.4 minutes .

Scalability Results

Scalability assessments reveal both capabilities and limitations:

Positive Scalability Findings : HMAS-2 scales better to more agents than CMAS . DynTaskMAS achieves near-linear throughput scaling up to 16 concurrent agents . LLaMAC is applied to tasks involving more than 50 agents . S-ATLAS shows increased advantage over baselines with larger robot teams .

Scalability Challenges : HMCF identifies scalability as a challenge for large-scale deployments involving dozens or hundreds of robots . DynTaskMAS shows diminishing returns at 32 agents . Centralized approaches face context window limitations that constrain scalability .

Comparison with Baseline Methods

Studies consistently demonstrate LLM-based approaches outperforming traditional methods:

vs. Rule-Based Methods : CoELA driven by GPT-4 surpasses MCTS-based and rule-based hierarchical planners by more than 40% in efficiency . COHERENT achieves the highest success rate compared to primitive MCTS and LLM-MCTS .

vs. Pure Optimization : LiP-LLM outperforms RoCo and SMART-LLM in success rates and process times . SAFER reduces safety violations by 77.5% compared to state-of-the-art LLM planners .

vs. Reinforcement Learning : LGC-MARL achieves higher success rates with lower normalized token costs compared to centralized LLM methods . L2M2 requires less than 20% of training samples compared to baseline MARL methods .

Identified Limitations

Several limitation categories emerge across studies:

LLM-Specific Limitations : CoELA shows unstable performance on complex reasoning tasks due to limited 3D spatial reasoning . LLaMAR performance is limited by the underlying VLM's spatial reasoning capabilities . EMOS faces challenges with hallucinations in LLMs .

Computational Constraints : LLM-Flock identifies computational demands as a significant limitation requiring model compression or hardware acceleration . Higher computational costs are noted due to multiple LLM queries .

Environmental Assumptions : LaMMA-P assumes fully observable, static environments . RoCo is limited by assumptions of accurate perception and open-loop execution .

Dynamic Adaptation

Real-world multi-robot deployments require systems capable of responding to changing conditions, failures, and evolving mission requirements.

Methods for Detecting Changes

Trigger-Based Detection : LAN2CB uses trigger conditions to detect changes during execution, updating dependency analysis accordingly . DEXTER-LLM re-triggers modules in response to new events detected during operation .

Continuous Monitoring : The digital twin framework provides real-time synchronization between physical operations and digital representations . SayCoNav detects changes in robot physical conditions during navigation . CoordField continuously monitors UAV status reports for environmental changes .

Feedback Integration : COHERENT's PEFA mechanism enables continuous feedback and adjustment based on execution results . MHRC uses textual feedback mechanisms and CoT prompts for change detection .

Re-Planning and Re-Allocation Strategies

Iterative Re-Planning : RoCo re-plans based on environmental feedback until valid plans are achieved or maximum attempts reached . LLaMAR's Corrector module adjusts actions based on failures . The hierarchical LLM framework implements frequent updates with the task LLM triggered every 20-30 seconds .

Dynamic Behavior Trees : The LLM-HBT framework enables dynamic extension of behavior trees and invocation of a centralized coordinator for subtask reassignment upon failure . LLM-MARS allows rapid reorganization to accommodate new tasks through behavior tree regeneration .

Optimization-Based Re-Allocation : The digital twin framework implements replanning optimization minimizing makespan while penalizing deviations from original plans . DEXTER-LLM uses search-based optimization for task reassignment upon robot failure .

Human-in-the-Loop Mechanisms

Multiple frameworks incorporate human oversight for enhanced safety and adaptability:

Direct Intervention : HMCF enables real-time human oversight and intervention through commands to robots . William Hunt et al. allow human advisors to interrupt and check plans with agents . DEXTER-LLM implements online verification and confirmation by human operators .

Supervisory Control : The hierarchical LLM framework incorporates real-time human input for feedback on performance and environmental hazards . The digital twin framework includes a user command receiver module for interventions .

Narrative-Based Adaptation : The construction robotics framework enables narrative-driven schedule adaptation using LLMs to interpret natural language inputs for constraint updates .

Adaptation Mechanism	Update Trigger	Response Time	Human Involvement
PEFA loops	Execution feedback	Real-time	Optional
Behavior tree extension	Failure detection	Real-time	None
Digital twin sync	Status change	Real-time	Command receiver
Trigger conditions	Mission events	Event-based	None
Confidence thresholds	Uncertainty levels	On-demand	Help requests

Event-Based vs Periodic Updates

Event-Based Systems : LAN2CB updates based on trigger conditions during execution . DEXTER-LLM employs event-based module retriggering . The conformal prediction approach in S-ATLAS enables robots to seek help when uncertainty exceeds thresholds .

Periodic Updates : The hierarchical LLM framework refreshes the outer LLM every 8-10 steps and the action LLM every 5 seconds . LLM-Flock implements periodic position updates between plan consensus and motion execution .

Hybrid Approaches : The digital twin framework combines event-based updates through user interventions with periodic real-time synchronization . BioMARS uses WebSocket protocols for event-based communication while maintaining continuous monitoring .

LLM Implementation Details

The practical implementation of LLM-based multi-robot orchestration involves careful selection of models, prompting strategies, and efficiency optimizations.

LLM Models Employed

GPT Family : GPT-4 and variants are most commonly employed, including GPT-4o , GPT-3.5-turbo , and GPT-4-vision . AutoHMA-LLM uses GPT-4o-mini alongside GPT-4o .

Open-Source Models : Llama models are widely used including Llama-3.1-8B , Llama-2-7b/13b , and LLaMA3 . Qwen models appear across multiple studies including Qwen-72B, Qwen-32B, and Qwen-7B . DeepSeek-R1 and Claude variants are also employed .

Specialized Models : LLM-MARS fine-tunes the Falcon 7B model with LoRA adapters . REMALIS uses a custom 7 billion parameter model . GTE-Base is used for language-conditioned navigation .

Model Category	Examples	Typical Use Cases
GPT-4 variants	GPT-4o , GPT-4-vision	Central planning, complex reasoning
Llama family	Llama-3.1-8B , Llama-2-70b	Local deployment, fine-tuning
Qwen models	Qwen-72B , Qwen2.5-32b	Balanced performance
Specialized	Falcon-7B , custom 7B	Domain-specific tasks

Prompting Strategies

Chain-of-Thought : CoT prompting is widely employed for structured reasoning . GMATP-LLM uses CoT for task decomposition and assignment . Zero-shot chain-of-thought prompting is used in CoELA to encourage more reasoning .

Few-Shot Learning : LaMMA-P uses few-shot filling prompts . PLANTOR employs few-shot prompting combined with CoT . The digital twin framework uses few-shot learning for constraint extraction .

Structured Prompts : DART-LLM uses structured prompts including instruction, environment, robot set, skills, and few-shot examples . SMART-LLM employs Pythonic prompts with line-by-line comments . Hierarchical prompts integrate essential information in multi-stage structures .

Multi-Agent LLM Configurations

Specialized Roles : SAFER employs separate Task Planning LLM and Safety Planning LLM agents . LLaMAR uses four specialized modules (Planner, Actor, Corrector, Verifier) . BioMARS uses distinct Biologist Agent and Technician Agent .

Hierarchical Agents : The hierarchical LLM framework uses outer and inner loop LLMs with different responsibilities . COHERENT implements a centralized task assigner with individual robot executors each having independent LLMs .

Decentralized Agents : HMCF equips each robot with its own LLM agent . LLM-Flock provides each robot with its own LLM for local planning .

Vision-Language Model Integration

Several frameworks integrate VLMs for enhanced perception:

Object Detection : DART-LLM uses VLM-based object detection for object map database updates . RoCo integrates with perception systems for environment understanding .

Scene Understanding : Dan BW Choe et al. use VLMs for conflict detection and description . The aerial-ground framework integrates GridMask-enhanced fine-tuned VLM . UAV-CodeAgents fine-tunes Qwen2.5VL-7B on annotated satellite images .

Visual Monitoring : BioMARS employs VLMs for hierarchical visual monitoring with geometric and semantic validation .

Efficiency Optimization Strategies

Query Reduction : LiP-LLM uses linear programming to reduce reliance on LLM inference . BOLAA orchestrates specialized agents to reduce individual query loads . S-ATLAS employs parallel querying of GPT-3.5 while sequentially querying Llama models .

Model Compression : LLM-Flock suggests model compression, distillation, or hardware acceleration for deployment . DynTaskMAS uses INT8 quantization for efficient inference . RALLY fine-tunes smaller models using LLaMA-Factory with LoRA .

Caching and Memory : HMCF uses Retrieval-Augmented Generation to reduce redundant queries . The digital twin framework implements narrative-driven updates to minimize unnecessary re-computation .

Synthesis

The heterogeneity in findings across the reviewed literature can be explained through several key factors that influence when different approaches succeed or fail.

Architectural Trade-offs Across Deployment Contexts

The choice between centralized, decentralized, and hybrid architectures is not simply a design preference but reflects fundamental trade-offs that apply differently across contexts. Centralized approaches like SMART-LLM and AutoRT excel in scenarios with stable communication infrastructure and moderate robot counts (typically under 10), achieving high coordination efficiency through unified reasoning. However, their performance degrades with increasing team size due to context window limitations and communication bottlenecks.

Decentralized systems like MHRC and S-ATLAS scale more gracefully to larger teams (demonstrated with 50+ agents in LLaMAC) but face coordination challenges in tightly coupled tasks requiring precise synchronization. The hybrid HMAS-2 approach achieves the best success rates across task types because it combines centralized oversight for global coordination with distributed execution for scalability, a pattern applicable when both coordination precision and scalability matter.

Task Complexity and Decomposition Strategy Matching

The effectiveness of decomposition methods depends critically on task characteristics. DAG-based approaches (DART-LLM , LiP-LLM) excel for tasks with clear precedence relationships and independent subtasks enabling parallel execution, achieving 94-100% success rates on structured tasks . However, these approaches require well-defined dependency structures that may not exist for exploratory or adaptive missions.

CoT prompting methods (GMATP-LLM , RALLY) handle more ambiguous task specifications but show higher variability in outcomes, particularly for complex reasoning requiring multiple reasoning steps. The few-shot prompting strategy by Murata et al. achieves 94% success (47/50) on ambiguous commands because it provides exemplars that ground the LLM's reasoning in domain-appropriate patterns.

LTL integration (Nl2Hltl2Plan , SafePlan) provides formal guarantees for safety-critical applications but requires users to specify or accept temporal logic constraints, limiting accessibility for non-expert operators.

Hardware Constraint Modeling Depth and Allocation Accuracy

Studies employing rich capability representations consistently outperform those with minimal constraint modeling. The ”robot resume” approach (EMOS) achieves higher success rates in embodiment-aware tasks because it explicitly captures physics capabilities that pure skill-based representations miss. The digital twin framework with dynamic capability modeling enables adaptive task allocation that responds to real-time hardware state changes.

Conversely, systems assuming homogeneity (S-ATLAS initially assumes shared skill sets) or using only coarse capability categories show degraded performance when hardware heterogeneity is high. The MTU-LLM framework achieves better workload balance by explicitly encoding capability-requirement matching in its prompt structure.

Real-Time Adaptation Capability and Environmental Dynamics

Systems with continuous feedback integration (PEFA mechanism , digital twin synchronization) maintain higher success rates in dynamic environments compared to open-loop planners. RoCo's re-planning approach achieves success through iterative validation , but introduces latency that may be unacceptable for time-critical applications.

The hierarchical LLM framework's multi-rate update strategy (task LLM every 20-30 seconds, action LLM every 5 seconds) represents a principled trade-off between adaptation responsiveness and computational cost. This tiered approach explains why it handles real-time target tracking while systems with single-rate updates struggle with rapidly changing conditions.

Model Selection and Computational Context

GPT-4 variants consistently achieve higher success rates than smaller models (e.g., GPT-4o achieving 0.75-0.92 success rates vs. GPT-3.5-turbo's lower performance), but at significantly higher computational cost and latency. For edge deployment scenarios, fine-tuned smaller models like Llama-3.1-8B with LoRA provide viable alternatives when combined with structured prompting.

The success of Falcon 7B fine-tuned with LoRA for behavior tree generation demonstrates that domain-specific fine-tuning can compensate for reduced model capacity when task scope is well-defined. Similarly, RALLY's capacity-migration augmentation enables lightweight models to achieve comparable performance on resource-constrained platforms.

Convergent Findings

Despite heterogeneity, several findings converge across studies:

1. **Hybrid architectures provide the best balance** of coordination and scalability for most multi-robot scenarios .
2. **Structured prompting with domain context** (skill definitions, environmental constraints) consistently improves decomposition quality over generic prompting .
3. **Integration of classical optimization with LLM reasoning** (LiP-LLM , PLANTOR) achieves better allocation optimality than pure LLM approaches.
4. **Feedback mechanisms are essential** for maintaining performance in dynamic environments, with success rates dropping significantly without them .
5. **Hardware constraint modeling depth correlates with allocation accuracy** , particularly for heterogeneous teams .

References

A. Khan, Michael Andrev, M. Murtaza, Sergio Aguilera, Rui Zhang, Jie Ding, Seth Hutchinson, and Ali Anwar. "Safety Aware Task Planning via Large Language Models in Robotics." *arXiv.org*, 2025.

Abdelhaleem Saad, Waseem Akram, and Irfan Hussain. "AquaChat++: LLM-Assisted Multi-ROV Inspection for Aquaculture Net Pens with Integrated Battery Management and Thruster Fault Tolerance." *arXiv.org*, 2025.

Abhinav Rajvanshi, Pritish Sahu, Tixiao Shan, Karan Sikka, and Han-Pang Chiu. "SayCoNav: Utilizing Large Language Models for Adaptive Collaboration in Decentralized Multi-Robot Navigation." *arXiv.org*, 2025.

Alkesh K. Srivastava, Jared Michael Levin, Alexander Derrico, and Philip Dames. "DELIVER: A System for LLM-Guided Coordinated Multi-Robot Pickup and Delivery Using Voronoi-Based Relay Planning." *arXiv.org*, 2025.

Artem Lykov, Maria Dronova, Nikolay Naglov, Mikhail Litvinov, S. Satsevich, A. Bazhenov, Vladimir Berman, Aleksei Shcherbak, and Dzmitry Tsetserukou. "LLM-MARS: Large Language Model for Behavior Tree Generation and NLP-Enhanced Dialogue in Multi-Agent Robot Systems." *arXiv.org*, 2023.

Bin Zhang, Hangyu Mao, Jingqing Ruan, Ying Wen, Yang Li, Shao Zhang, Zhiwei Xu, et al. "Controlling Large Language Model-Based Agents for Large-Scale Decision-Making: An Actor-Critic Approach." *arXiv.org*, 2023.

Chaoran Wang, Jingyuan Sun, Yanhui Zhang, Mingyu Zhang, and C. Wu. "LLM-HBT: Dynamic Behavior Tree Construction for Adaptive Coordination in Heterogeneous Robots," 2025.

Dan BW Choe, Sundhar Vinodh Sangeetha, Steven Emanuel, Chih-Yuan Chiu, Samuel Coogan, and Shreyas Kousik. "Seeing, Saying, Solving: An LLM-to-TL Framework for Cooperative Robots." *arXiv.org*, 2025.

Daniel Weiner, and Raj Korpan. "Hybrid Voting-Based Task Assignment in Modular Construction Scenarios." *arXiv.org*, 2025.

Enrico Saccon, Ahmet Tikna, Davide De Martini, Edoardo Lamon, Luigi Palopoli, and M. Roveri. "A Temporal Planning Framework for Multi-Agent Systems via LLM-Aided Knowledge Base Management." *arXiv.org*, 2025.

Fernando Cladera, Zachary Ravichandran, Jason Hughes, Varun Murali, Carlos Nieto-Granda, M. Ani Hsieh, George J. Pappas, C. Taylor, and Vijay Kumar. "Air-Ground Collaboration for Language-Specified Missions in Unknown Environments." *IEEE Transactions on Field Robotics*, 2025.

Guobin Zhu, Rui Zhou, Wenkang Ji, and Shiyu Zhao. "LAMARL: LLM-Aided Multi-Agent Reinforcement Learning for Cooperative Policy Generation." *IEEE Robotics and Automation Letters*, 2025.

Hao Sha, Yao Mu, Yuxuan Jiang, Li Chen, Chenfeng Xu, Ping Luo, S. Li, Masayoshi Tomizuka, Wei Zhan, and Mingyu Ding. "LanguageMPC: Large Language Models as Decision Makers for Autonomous Driving." *arXiv.org*, 2023.

Haokun Liu, Zhaoqi Ma, Yunong Li, Junichiro Sugihara, Yicheng Chen, Jinjie Li, and Moju Zhao. "Hierarchical Language Models for Semantic Navigation and Manipulation in an Aerial-Ground Robotic System." *arXiv.org*, 2025.

Haolin Li, Peng Yi, Dixiao Wei, and Wenyan Bai. "Seek-and-Take Games of Heterogeneous Agent Teams with Large Language Model." *ACM Cloud and Autonomic Computing Conference*, 2024.

Harisankar Babu, Philipp Schillinger, and Tamim Asfour. "Adaptive Domain Modeling with Language Models: A Multi-Agent Approach to Task Planning." *2025 IEEE 21st International Conference on Automation Science and Engineering (CASE)*, 2025.

Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, J. Tenenbaum, Tianmin Shu, and Chuang Gan. "Building Cooperative Embodied Agents Modularly with Large Language Models." *International Conference on Learning Representations*, 2023.

Hsu-Shen Liu, So Kuroki, Tadashi Kozuno, Wei-Fang Sun, and Chun-Yi Lee. "Language-Guided Pattern Formation for Swarm Robotics with Multi-Agent Reinforcement Learning." *IEEE/RJS International Conference on Intelligent RObots and Systems*, 2024.

Huaiyuan Yao, Longchao Da, Vishnu Nandam, J. Turnau, Zhiwei Liu, Linsey Pang, and Hua Wei. "CoMAL: Collaborative Multi-Agent Large Language Models for Mixed-Autonomy Traffic." *SDM*, 2024.

Huibo Zhang, Shengkang Chen, Huan Yin, and Fumin Zhang. "An In-Context Learning Approach for LLM-Enabled Multi-Robot Task Allocation." *IEEE International Conference on Robotics and Biomimetics*, 2024.

Ike Obi, Vishnunandan L. N. Venkatesh, Weizheng Wang, Ruiqi Wang, Dayoon Suh, T. I. Amosa, Wonse Jo, and Byung-Cheol Min. "SafePlan: Leveraging Formal Logic and Chain-of-Thought Reasoning for Enhanced Safety in LLM-Based Robotic Task Planning." *arXiv.org*, 2025.

Jiabao Ji, Yongchao Chen, Yang Zhang, R. Kompella, Chuchu Fan, Gaowen Liu, and Shiyu Chang. "Collision- and Reachability-Aware Multi-Robot Control with Grounded LLM Planners." *arXiv.org*, 2025.

Jinqiang Cui, Guocai Liu, Hui Wang, Yue Yu, and Jiankun Yang. "TPML: Task Planning for Multi-UAV System with Large Language Models." *2024 IEEE 18th International Conference on Control & Automation (ICCA)*, 2024.

Jun Wang, Guocheng He, and Y. Kantaros. "Probabilistically Correct Language-Based Multi-Robot Planning Using Conformal Prediction." *IEEE Robotics and Automation Letters*, 2024.

———. "Safe Task Planning for Language-Instructed Multi-Robot Systems Using Conformal Prediction." *arXiv.org*, 2024.

Junting Chen, Checheng Yu, Xunzhe Zhou, Tianqi Xu, Yao Mu, Mengkang Hu, Wenqi Shao, Yikai Wang, Guohao Li, and Lin Shao. "EMOS: Embodiment-Aware Heterogeneous Multi-Robot Operating System with LLM Agents." *International Conference on Learning Representations*, 2024.

Junwei Yu, Yepeng Ding, and Hiroyuki Sato. "DynTaskMAS: A Dynamic Task Graph-Driven Framework for Asynchronous and Parallel LLM-Based Multi-Agent Systems." *Proceedings of the ... International Conference on Automated Planning and Scheduling*, 2025.

Kartik Nagpal, Dayi Dong, Jean-Baptiste Bouvier, and Negar Mehr. "Leveraging Large Language Models for Effective and Explainable Multi-Agent Credit Assignment." *Adaptive Agents and Multi-Agent Systems*, 2025.

Kaushik Kannan, and Jungyun Bae. "MTU-LLM: LLM-Based Multi-Robot Task Allocation and Path Planning for Heterogeneous Robots in Search and Rescue Operations." *AI Computer Science and Robotics Technology*, 2025.

Kazuma Obata, Tatsuya Aoki, Takato Horii, Tadahiro Taniguchi, and Takayuki Nagai. "LiP-LLM: Integrating Linear Programming and Dependency Graph With Large Language Models for Multi-Robot Task Planning." *IEEE Robotics and Automation Letters*, 2024.

Kehui Liu, Zixin Tang, Dong Wang, Zhigang Wang, Bin Zhao, and Xuelong Li. "COHERENT: Collaboration of Heterogeneous Multi-Robot System with Large Language Models." *IEEE International Conference on Robotics and Automation*, 2024.

Kento Murata, Shoichi Hasegawa, Tomochika Ishikawa, Y. Hagiwara, Akira Taniguchi, Lotfi El Hafi, and T. Taniguchi. "Multi-Robot Task Planning for Multi-Object Retrieval Tasks with Distributed On-Site Knowledge via Large Language Models." *arXiv.org*, 2025.

Kun Chu, Xufeng Zhao, C. Weber, and Stefan Wermter. "LLM+MAP: Bimanual Robot Task Planning Using Large

Language Models and Planning Domain Definition Language." *arXiv.org*, 2025.
Lillian Wassim, Kamal Mohamed, and Ali Hamdi. "LLM-DaaS: LLM-Driven Drone-as-a-Service Operations from Text User Requests." *arXiv.org*, 2024.
Marc Glocker, Peter Hönig, Matthias Hirschmanner, and Markus Vincze. "LLM-Empowered Embodied Agent for Memory-Augmented Task Planning in Household Robotics." *arXiv.org*, 2025.
Marcos Abel Zuzu'arregui, Mustafa Melih Toslak, and Stefano Carpin. "One For All: LLM-Based Heterogeneous Mission Planning in Precision Agriculture." *arXiv.org*, 2025.
Michael Ahn, Debidatta Dwibedi, Chelsea Finn, Montse Gonzalez Arenas, K. Gopalakrishnan, Karol Hausman, Brian Ichter, et al. "AutoRT: Embodied Foundation Models for Large Scale Orchestration of Robotic Agents." *arXiv.org*, 2024.
Michele Grimaldi, Carlo Cernicchiaro, Sebastian Realpe Rua, Alaaeddine El-Masri-El-Chaarani, Markus Buchholz, Loizos Michael, Pere Ridao Rodriguez, I. Carlucho, and Yvan R. Petillot. "Advancing Shared and Multi-Agent Autonomy in Underwater Missions: Integrating Knowledge Graphs and Retrieval-Augmented Generation." *arXiv.org*, 2025.
Min Deng, Bo Fu, Lingyao Li, and Xi Wang. "Integrating LLMs and Digital Twins for Adaptive Multi-Robot Task Allocation in Construction." *arXiv.org*, 2025.
Minghong Geng, Shubham Pateria, Budhitama Subagdja, Lin Li, Xin Zhao, and Ah-Hwee Tan. "L2M2: A Hierarchical Framework Integrating Large Language Model and Multi-Agent Reinforcement Learning." *International Joint Conference on Artificial Intelligence*, 2025.
Mukund Mitra, Yashaswi Sinha, Arushi Khokhar, Sairam Jinkala, and Pradipta Biswas. "LMD-FISH: Language Model Driven - Framework for Intelligent Scheduling of Heterogenous Systems." *IUI Companion*, 2025.
Nan Li, Jiming Ren, Haris Miller, Samuel Coogan, K. Feigh, and Ye Zhao. "Adaptive Obstacle-Aware Task Assignment and Planning for Heterogeneous Robot Teaming," 2025.
Oleg Sautenkov, Yasheerah Yaqoot, Muhammad Ahsan Mustafa, Faryal Batool, Jeffrin Sam, Artem Lykov, Chih-Yung Wen, and Dzmitry Tsetserukou. "UAV-CodeAgents: Scalable UAV Mission Planning via Multi-Agent ReAct and Vision-Language Reasoning." *arXiv.org*, 2025.
Peihan Li, and Lifeng Zhou. "LLM-Flock: Decentralized Multi-Robot Flocking via Large Language Models and Influence-Based Consensus." *arXiv.org*, 2025.
Peihan Li, Zijian An, Shams Abrar, and Lifeng Zhou. "Large Language Models for Multi-Robot Systems: A Survey." *arXiv.org*, 2025.
Piyush Gupta, David Isele, Enna Sachdeva, Pin-Hao Huang, Behzad Dariush, Kwonjoon Lee, and Sangjae Bae. "Generalized Mission Planning for Heterogeneous Multi-Robot Teams via LLM-Constructed Hierarchical Trees." *IEEE International Conference on Robotics and Automation*, 2025.
Rui Yang, and Rajiv Gupta. "DREAM: Distributed Regional Efficient Agent Management with LLMs for Online Multi-Agent Pathfinding." *ACM SIGOPS Operating Systems Review*, 2025.
Ruiyang Wang, Hao-Lun Hsu, David Hunt, Shaocheng Luo, Jiwoo Kim, and Miroslav Pajic. "LLM-MCoX: Large Language Model-Based Multi-Robot Coordinated Exploration and Search," 2025.
S. S. Kannan, Vishnunandan L. N. Venkatesh, and Byung-Cheol Min. "SMART-LLM: Smart Multi-Agent Robot Task Planning Using Large Language Models." *IEEE/RJS International Conference on Intelligent RObots and Systems*, 2023.
Seoyeon Choi, Kanghyun Ryu, Jonghoon Ock, and Negar Mehr. "CRAFT: Coaching Reinforcement Learning Autonomously Using Foundation Models for Multi-Robot Coordination Tasks." *arXiv.org*, 2025.
Shaojun Xu, Xusheng Luo, Yutong Huang, Letian Leng, Ruixuan Liu, and Changliu Liu. "Nl2Hltl2Plan: Scaling Up Natural Language Understanding for Multi-Robots Through Hierarchical Temporal Logic Task Specifications." *IEEE Robotics and Automation Letters*, 2024.
Shuai Jia, Zhe Cui, Kai Wang, Yujie Ding, Dongming Han, and Tianyi Ma. "Adversarial and Cooperation Tasks in

Multi- Agent System with Large Language Models." *2024 IEEE International Conference on Unmanned Systems (ICUS)*, 2024.

Siddharth Nayak, Adelmo Morrison Orozco, M. T. Have, Vittal Thirumalai, Jackson Zhang, Darren Chen, Aditya Kapoor, et al. "Long-Horizon Planning for Multi-Agent Robots in Partially Observable Environments." *Neural Information Processing Systems*, 2024.

Steven D. Morad, Ajay Shankar, Jan Blumenkamp, and Amanda Prorok. "Language-Conditioned Offline RL for Multi-Robot Navigation." *IEEE International Conference on Robotics and Automation*, 2024.

Tengchao Zhang, Yonglin Tian, Fei Lin, Jun Huang, Patrik P. Süli, Rui Qin, and Fei-Yue Wang. "CoordField: Coordination Field for Agentic UAV Task Allocation In Low-Altitude Urban Scenarios." *arXiv.org*, 2025.

Tinging Yang, Ping Feng, Qixin Guo, Jindi Zhang, Xiufeng Zhang, Jiahong Ning, Xinghan Wang, and Zhongyang Mao. "AutoHMA-LLM: Efficient Task Coordination and Execution in Heterogeneous Multi-Agent Systems Using Hybrid Large Language Models." *IEEE Transactions on Cognitive Communications and Networking*, 2025.

Volker Strobel, Marco Dorigo, and Mario Fritz. "LLM2Swarm: Robot Swarms That Responsively Reason, Plan, and Collaborate Through LLMs." *arXiv.org*, 2024.

Weizheng Wang, Ike Obi, and Byung-Cheol Min. "Multi-Agent LLM Actor-Critic Framework for Social Robot Navigation." *arXiv.org*, 2025.

Wen Zhao, Liqiao Li, Hanwen Zhan, Yingqi Wang, and Yiqi Fu. "Applying Large Language Model to a Control System for Multi-Robot Task Assignment." *Drones*, 2024.

Wenhao Yu, Jie Peng, Yueliang Ying, Sai Li, Jianmin Ji, and Yanyong Zhang. "MHRC: Closed-Loop Decentralized Multi-Heterogeneous Robot Collaboration with Large Language Models." *arXiv.org*, 2024.

Wenjie Lin, and Jin Wei-Kocsis. "Think, Reflect, Create: Metacognitive Learning for Zero-Shot Robotic Planning with LLMs." *arXiv.org*, 2025.

Wenkang Ji, Huaben Chen, Mingyang Chen, Guobin Zhu, Lufeng Xu, Roderich Gross, Rui Zhou, Ming Cao, and Shiyu Zhao. "GenSwarm: Scalable Multi-Robot Code-Policy Generation and Deployment via Language Models." *arXiv.org*, 2025.

William Hunt, Toby Godfrey, and Mohammad Divband Soorati. "Conversational Language Models for Human-in-the-Loop Multi-Robot Coordination." *Adaptive Agents and Multi-Agent Systems*, 2024.

Xiangkun Deng, Gang Tao, Chenxu Wen, Xi Zhang, Zhiyang Ju, and Jianwei Gong. "GMATP-LLM: A General Multi-Agent Task Dynamic Planning Method Using Large Language Models." *Cybersecurity and Cyberforensics Conference*, 2025.

Xiaopan Zhang, Hao Qin, Fuquan Wang, Yue Dong, and Jiachen Li. "LaMMA-P: Generalizable Multi-Agent Long-Horizon Task Allocation and Planning with LM-Driven PDDL Planner." *IEEE International Conference on Robotics and Automation*, 2024.

Xihe Qiu, Haoyu Wang, Xiaoyu Tan, Chao Qu, Yujie Xiong, Yuan Cheng, Yinghui Xu, Wei Chu, and Yuan Qi. "Towards Collaborative Intelligence: Propagating Intentions and Reasoning for Multi-Agent Coordination with Large Language Models." *arXiv.org*, 2024.

Xinzhu Liu, Peiyan Li, Wenju Yang, Di Guo, and Huaping Liu. "Leveraging Large Language Model for Heterogeneous Ad Hoc Teamwork Collaboration." *Robotics: Science and Systems*, 2024.

Yibo Qiu, Zan Huang, Zhiyu Wang, Handi Liu, Yiling Qiao, Yifeng Hu, Shu'ang Sun, Hangke Peng, Ronald X Xu, and Mingzhai Sun. "BioMARS: A Multi-Agent Robotic System for Autonomous Biological Experiments." *arXiv.org*, 2025.

Yongchao Chen, Jacob Arkin, Yang Zhang, Nicholas Roy, and Chuchu Fan. "Scalable Multi-Robot Collaboration with Large Language Models: Centralized or Decentralized Systems?" *IEEE International Conference on Robotics and Automation*, 2023.

Yongdong Wang, Runze Xiao, Junichi Kasahara, Ryosuke Yajima, Keiji Nagatani, Atsushi Yamashita, and Hajime Asama. "DART-LLM: Dependency-Aware Multi-Robot Task Decomposition and Execution Using Large Lan-

guage Models." *arXiv.org*, 2024.
Yoshiki Yano, Kazuki Shibata, Maarten Kokshoorn, and Takamitsu Matsubara. "ICCO: Learning an Instruction-Conditioned Coordinator for Language-Guided Task-Aligned Multi-Robot Control." *arXiv.org*, 2025.
Yuwei Wu, Yuezhan Tao, Peihan Li, Guangyao Shi, Gaurav S. Sukhatme, Vijay Kumar, and Lifeng Zhou. "Hierarchical LLMs In-the-Loop Optimization for Real-Time Multi-Robot Target Tracking Under Unknown Hazards." *arXiv.org*, 2024.
Yuxiao Zhu, Junfeng Chen, Xintong Zhang, Meng Guo, and Zhongkui Li. "DEXTER-LLM: Dynamic and Explainable Coordination of Multi-Robot Systems in Unknown Environments via Large Language Models." *arXiv.org*, 2025.
Zachary Ravichandran, Varun Murali, Mariliza Tzes, George J. Pappas, and Vijay Kumar. "SPINE: Online Semantic Planning for Missions with Incomplete Natural Language Specifications in Unstructured Environments." *IEEE International Conference on Robotics and Automation*, 2024.
Zhao Mandi, Shreeya Jain, and Shuran Song. "RoCo: Dialectic Multi-Robot Collaboration with Large Language Models." *IEEE International Conference on Robotics and Automation*, 2023.
Zhaoxing Li, Wenbo Wu, Yue Wang, Yanran Xu, William Hunt, and Sebastian Stein. "HMCF: A Human-in-the-Loop Multi-Robot Collaboration Framework Based on Large Language Models." *arXiv.org*, 2025.
Zhehui Huang, Guangyao Shi, Yuwei Wu, Vijay Kumar, and Gaurav S. Sukhatme. "Compositional Coordination for Multi-Robot Teams with Large Language Models." *arXiv.org*, 2025.
Zhiwei Liu, Weiran Yao, Jianguo Zhang, Le Xue, Shelby Heinecke, Rithesh Murthy, Yihao Feng, et al. "BOLAA: Benchmarking and Orchestrating LLM-Augmented Autonomous Agents." *arXiv.org*, 2023.
Ziqi Jia, Junjie Li, Xiaoyang Qu, and Jianzong Wang. "Enhancing Multi-Agent Systems via Reinforcement Learning with LLM-Based Planner and Graph-Based Policy." *IEEE International Conference on Robotics and Automation*, 2025.
Ziyao Wang, Rongpeng Li, Sizhao Li, Yuming Xiang, Haiping Wang, Zhifeng Zhao, and Honggang Zhang. "RALLY: Role-Adaptive LLM-Driven Yoked Navigation for Agentic UAV Swarms." *arXiv.org*, 2025.