IEEE Transactions on Field Robotics (T-FR) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-FR paper.

Received 1 December, 2024; revised 27 April, 2025; accepted 10 June, 2025; Date of publication 27 June, 2025.

Digital Object Identifier 10.1109/10.1109/TFR.2025.3584019

Air-Ground Collaboration for Language-Specified Missions in Unknown Environments

Fernando Cladera[*1], Zachary Ravichandran[*1], Jason Hughes[*1], Varun Murali[1], Carlos Nieto-Granda[2], M. Ani Hsieh[1], George J. Pappas[1], Camillo J. Taylor[1], and Vijay Kumar[1]

[1]GRASP Laboratory, University of Pennsylvania, Philadelphia, Pennsylvania 19104.
[2]U.S. DEVCOM Army Research Laboratory (ARL), Adelphi, Maryland 20783.
[*]Equal Contribution.

Corresponding authors: Fernando Cladera (fclad@seas.upenn.edu), Zachary Ravichandran (zacravi@seas.upenn.edu), and Jason Hughes (jasonah@seas.upenn.edu).

We gratefully acknowledge the support of ARL DCIST CRA W911NF-17-2-0181, NIFA grant 2022-67021-36856, the IoT4Ag Engineering Research Center funded by the National Science Foundation (NSF) under NSF Cooperative Agreement Number EEC-1941529, NVIDIA, and the NSF Graduation Research Fellowship Program.

ABSTRACT As autonomous robotic systems become increasingly mature, users will want to specify missions at the level of intent rather than in low-level detail. Language is an expressive and intuitive medium for such mission specification. However, realizing language-guided robotic teams requires overcoming significant technical hurdles. Interpreting and realizing language-specified missions requires advanced semantic reasoning. Successful heterogeneous robots must effectively coordinate actions and share information across varying viewpoints. Additionally, communication between robots is typically intermittent, necessitating robust strategies that leverage communication opportunities to maintain coordination and achieve mission objectives. In this work, we present a first-of-its-kind system where an unmanned aerial vehicle (UAV) and an unmanned ground vehicle (UGV) are able to collaboratively accomplish missions specified in natural language while reacting to changes in specification on the fly. We leverage a Large Language Model (LLM)-enabled planner to reason over semantic-metric maps that are built online and opportunistically shared between an aerial and a ground robot. We consider task-driven navigation in urban and rural areas. Our system must infer mission-relevant semantics and actively acquire information via semantic mapping. In both ground and air-ground teaming experiments, we demonstrate our system on seven different natural-language specifications at up to kilometer-scale navigation.

INDEX TERMS AI-Enabled Robotics; Field Robots; Human-Robot Collaboration; Multi-Robot Systems; Search and Rescue Robots; Semantic Scene Understanding; Task Planning.

I. INTRODUCTION

Exploration of unknown unstructured environments is one of the quintessential field robotics tasks, with applications to infrastructure inspection [1], search and rescue [2], disaster response [3], law enforcement [4], and crop inspection [5], among others. Heterogeneous teams of aerial and ground robots have a distinct advantage for fast exploration missions, by providing complementary capabilities compared to a team of identical robots.

For example, UAVs flying at high altitudes move in an obstacle-free space; thus, they can achieve higher speeds and provide an elevated vantage point. UAVs can trade off resolution for altitude and fly higher when a fast scene overview is required. Moreover, high-altitude UAVs can act as obstacle-free communication nodes, achieving line-of-sight with other nodes on the ground. Unfortunately, the low size, weight, and power (SWaP) constraints on UAVs restrict the payload, compute, and operation time. Contrarily, UGVs are not affected by these power limitations: they can carry heavier payloads and operate for extended periods. However, the UGV speed is limited as they have to plan trajectories to avoid static and

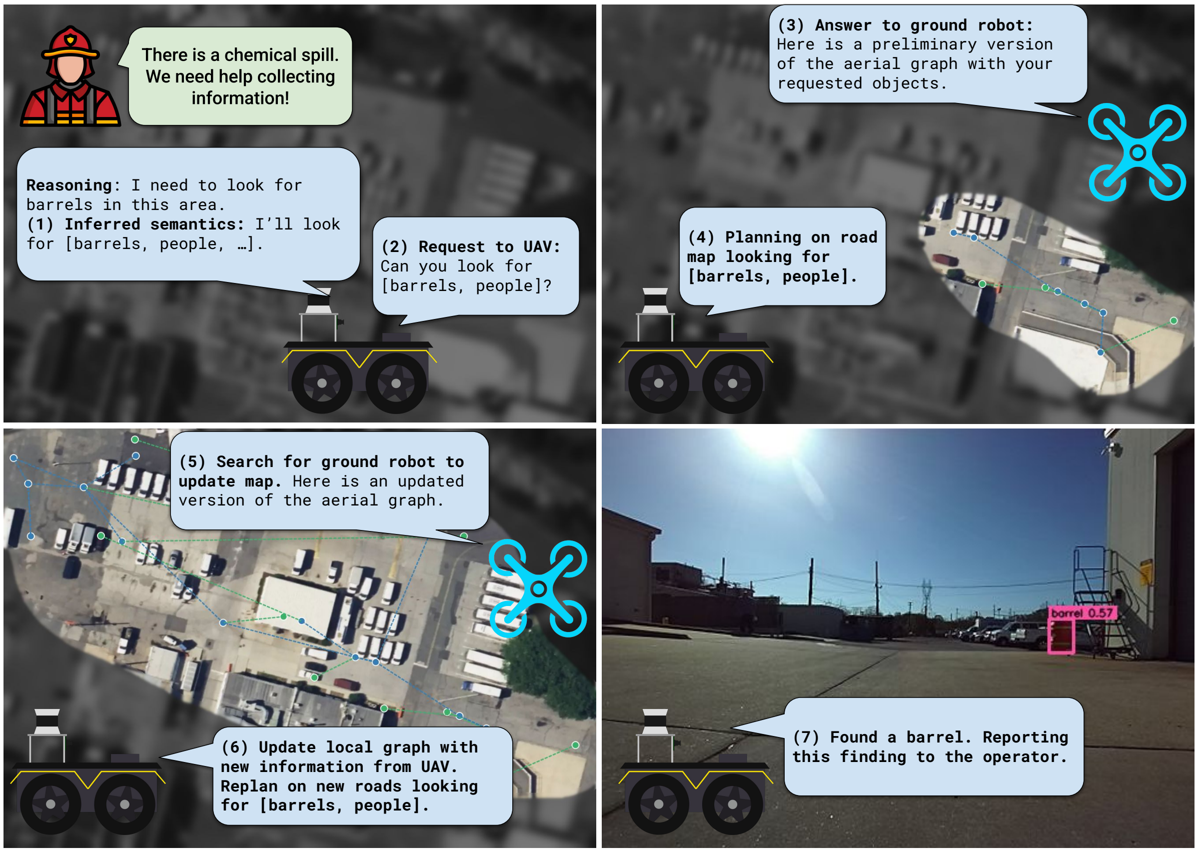

FIGURE 1. Example mission specification. The UAV and UGVs start their mission in the same location. The UGVs can task the UAV with a list of labels to look for. Once it receives a graph, the UGVs can plan on the UAV map, looking for objects of interest. Satellite image in the background is only provided for representation purposes.

dynamic obstacles and consider the traversability of the terrain where they operate.

Liu et al. [6] identified different roles UAVs and UGVs perform in a team, such as sensor, actuator, or auxiliary. **We focus on applications where UAVs and UGVs act cooperatively to perform a mission**. However, in most examples in the literature [6], the tasks are pre-defined for the robots. For instance, UAVs may be used to localize ground robots [7] or generate maps and traversability information [8]. Miller et al. [9], [10] were one of the first to use UAVs in a multi-role setting, where the UAV generated a map and acted as an active communication relay.

Still, state-of-the-art methods for air-ground robotic collaboration are far from seamless and require pre-encoded tasks. For example, leading systems in the DARPA Subterranean Challenge, including Nebula [11] and Cerberus [12], were designed for object search over a specific semantic set. Recent literature proposes planners for language-specified tasks in single-robot, small-scale, structured environments (e.g., indoors) [13]–[15]. However, such work assumes favorable conditions, including near-perfect environment knowledge and perfect communications [16].

As autonomous robotic systems become increasingly mature, we envision a heterogeneous robotic team that users can task at the level of intent rather than detailed step-by-step instructions. Given a user's mission specification, the robot team would infer the required subtasks, reason about each platform's constraints and abilities, and execute a plan suited to the environment of operation. Realizing this vision on heterogeneous robotic systems, such as our air-ground robotic team, remains a significant challenge. Existing literature generally assumes mission specifications are fixed. Creating planners that reason over dynamic mission specifications, where no task is pre-specified on either the aerial or ground vehicle, remains an open research question. Correspondingly, existing works in the literature fix the relevant semantics up front or only share metric information among robots. However, dynamically configuring semantics is a desirable property. Finally, robot mobility should not be limited by the communication network infrastructure, and robots should not need to be in communication range continuously. Creating robot teams that leverage fully opportunistic communications is an ongoing challenge.

This work builds upon our previous air-ground collaboration system [8], [9] moving past closed set semantics. We propose a language-driven air-ground system that addresses the challenges outlined in [17]. Enabled by an LLM and an open-set object detector, our work allows semantics to be specified at runtime. We include components of opportunistic communication [10] and language-driven planning [18]. To sum up, we present the following **novel contributions**:

- The *first* air-ground teaming system for language-specified missions in unknown environments.
- A *decentralized and compact* semantic mapping approach that enables the UAV and UGVs to share observations and *dynamically* select mission-relevant semantics.

We performed experimental validation and system deployment in kilometer-scale experiments in urban and rural environments. An example mission scenario is shown in Fig. 1. More details about this paper and code are available on our project page[1].

II. RELATED WORK

A. Air-ground Robot Teaming

Cooperative air-ground collaboration in robotics has more than twenty years of development [6]. The initial works of Elfes et al. used robotics blimps to transmit aerial images to ground robots, which used them for visual navigation [19]. Other early works also use aerial vehicles for localization, tracking, and obstacle avoidance [7], [20], [21]. In these works, the UAV acts as an *eye in the sky*, providing information that compliments ground robot knowledge. One of the first examples of air-ground mapping is described in [3] and [22], where a team of robots was deployed to assess the damage to historical buildings after earthquakes in Italy and Japan. These works proved the potential of air-ground teams in disaster scenarios by reducing the risk to rescue personnel.

Most existing air-ground robot teaming literature focuses on static task assignments, where the robots' tasks are predefined. For example, the UAV provides aerial information for mapping or obstacle avoidance. In some works [9], [10], the UAV has multiple roles such as a map provider and communication relay. Still, few works show explicit tasking within a UAV-UGV team. **In this work, we aim to address this issue by allowing the UGV to task the UAV as part of an integrated semantic exploration task**.

B. Semantic Representations for Navigation

Semantic planning requires representations that contain the contextual information for high-level planning and traversability information for lower-level control. The perception community has advanced online semantic planning for tasks such as active exploration and object search [9], [23]–[25]. Most of the semantic mapping literature focuses on single-robot applications. In these works, scene graphs have emerged as a popular representation for semantic planning, as they capture objects, topology, and traversable regions. Hydra provides a real-time scene graph engine [26] designed for indoor environments. Strader et al. [27] extend this work to outdoor environments. Topological maps are similar but do not include a hierarchy [28], [29]. Recent work incorporates foundation models into mapping pipelines to create open-vocabulary representations. Mappers, including Concept-Graphs [30], HOVSG [14], and Clio [31], assign semantic feature vectors to entities in the map. Semantic labels are then produced at runtime, depending on the task.

Effective representations for multi-robot teaming require many of the aforementioned capabilities. One of the major additional challenges is that they require a shared frame between robots. Consistent multi-robot state estimation is addressed by works such as Kimera-multi [32]. Similarly, Hydra-Multi builds scene graphs across a robot team via loop-closure detection and relative state estimation [33]. Alternatively, the SPOMP system addresses this problem with a semantics-based relative localization module [9]. However, this approach requires an orthomap to be constructed in real time and transmitted to the ground robots. Compact representations are desirable as communication and compute may be limited.

This work introduces a sparse map representation shared by the UAV and UGV. We devise a method to build a lightweight graph with traversability priors encoded for the UGV, which is necessary due to the limited bandwidth between platforms in real world scenarios. We demonstrate how the UAV can build such map in real time and how the UGV can use this representation to execute its mission while augmenting it with newly observed information.

C. Language-specified Mission Planning

Motivated by the increasing maturity and accessibility of LLMs, the robotics community has studied the use of language for specifying tasks and missions. LLM-enabled planners have been developed for mobile manipulation [13], [16], [34], service robotics [35], autonomous driving [36], navigation [37]–[40], and fault detection [41], [42] These methods typically configure a pre-trained LLM, such as GPT-4 [43], via in-context prompts. This approach avoids fine-tuning or retraining an LLM, which is computationally expensive, but it still channels the LLM's common sense into a specific problem [44]. The LLM-enabled planner is then provided a set of behaviors such as graph navigation goals [36], predicates in a formal planning language [15], lower-level APIs for code generation [45]–[47], or learned behaviors [13]. At runtime, the LLM is given an environment map, such as a graph [16] or semantic regions [44]. A line of research plans over formal languages such as Linear Temporal

[1] http://tfr-air-ground.fcladera.com

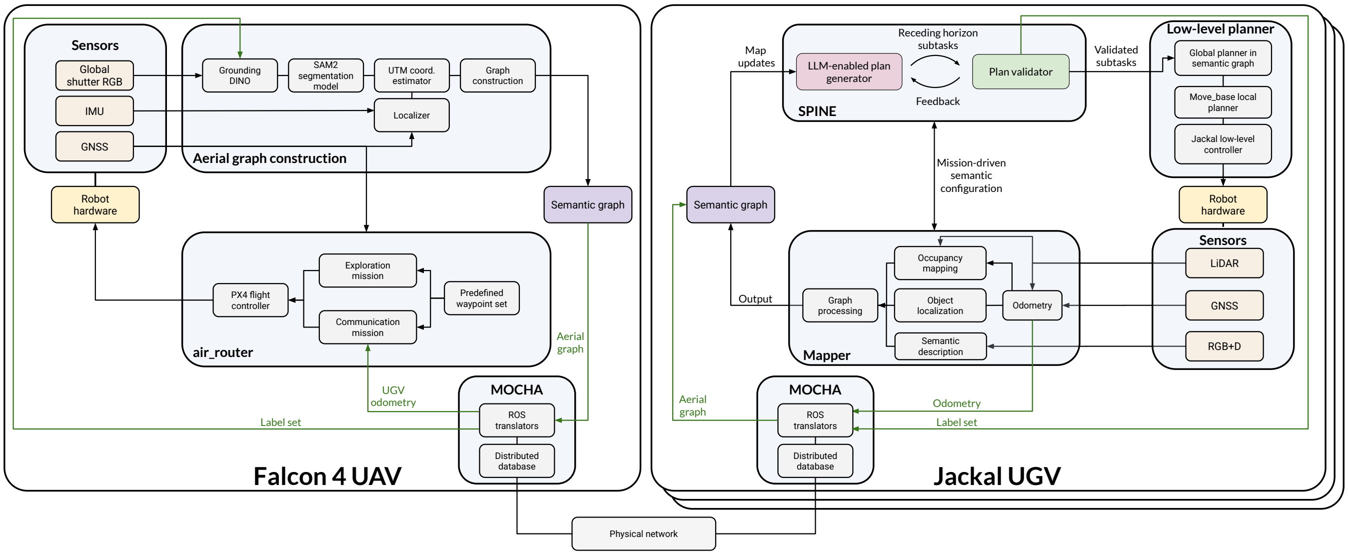

FIGURE 2. System overview. Green lines represent messages that are transmitted to and from MOCHA and thus are communicated to other robots in the system.

Logic (LTL) or Planning Domain Definition Language (PDDL) [15], [48]–[53]. Recent literature also considers language-specified missions for multi-robot systems [54]–[58]. However, these works consider controlled simulation or closed-world manipulation environments with perfect or near-perfect environment knowledge.

In the above approaches, instructions typically state the required subtasks and required semantics [59]. While works such as SayPlan consider less well-specified tasks ("find me something to drink"), they still assume pre-mapped or highly-structured indoor scenes [16], [34]. Other research relaxes the requirement of a pre-built semantic map by incorporating feedback from perception systems [34], [41], [44] or specifying semantics at runtime [59], [60]. However, perception is limited to object detection or designed for small room-centric environments where the planner can leverage clear hierarchy and natural bounds on the environment.

We recently introduced the SPINE planning architecture, which serves as the base of our UGV planner [18]. In contrast to the above works, SPINE reasons over under-specified missions where the user provides high-level intent. Furthermore, the environment is only partially-known or unknown at runtime, so the system must actively acquire task-relevant information. Compared to our previous work, we **extend SPINE to enable tasking the UAV**, allowing it to generate a better prior for the UGV mission.

D. Multi-robot Communications

Communication in air-ground teams can be broadly divided into three groups: a) ensured network connectivity, b) rendezvous approaches, and c) opportunistic communication. Ensured network communication approaches focus on deploying a group of robots such that there is guaranteed communication between all network nodes [61], [62]. Continuous communication approaches are key when a human operator is required in the loop or when robots need to report to a centralized base station continuously. The main disadvantage of this approach is that the experiment area is limited by the communication channel quality, and it can be severely limited in environments with non-line-of-sight (LOS) between robots.

Rendezvous approaches rely on established guarantees for communication as robots physically meet to exchange information [63], [64]. Alternatively, robots can form continuous communication networks and venture out to areas without communications [65], [66], with successful application in underground mapping at the DARPA Subterranean Challenge. Still, these techniques limit mobility when the number of robots is reduced.

In [9], [10], we introduced MOCHA, a fully opportunistic communication approach for air-ground teams. MOCHA is based on a gossip communication approach [67], that enables large-scale exploration, as communication constraints do not limit robot operations. To enhance collaboration by sharing information, the aerial robot acts as a *data mule*, finding ground robots and physically ferrying data between agents. We showed that MOCHA performs well in large-scale environments, both in simulation and in real-world experiments. **This work leverages MOCHA for communication between the robots, and for the communication-aware planner on the UAV**.

E. Robot Teaming with Human in the Loop

Human-in-the-loop frameworks have become pivotal in improving robot teaming by integrating human oversight and domain-specific tasking into multi-robot systems. Measuring the efficacy of human-robot teaming has been

Producer	Name	Message type	Size
UGV	`UGV_odometry`	Pose stamped	74 B
UGV	`label_set`	String (JSON)	2B
UAV	`aerial_graph`	String (JSON)	40.2 KB

TABLE 1. Messages transmitted between robots using MOCHA, our opportunistic communications framework, after 10 minutes of an experiment.

studied extensively in [68]–[70]. Specifically, Novitzky et al. uses the game "capture the flag" in [71] to assess effective ways to task robot teammates. Robot teams with humans in-the-loop have been developed in [72]–[74] for various inspection tasks. All of these rely on predefining what is to be inspected before the system is running by hard-coding in [74] or predefining semantic labels as in [73]. Therefore, these systems lack the ability for the human to re-task the robot team on the fly like our system can. Prior work has used different interfaces to interact with the robot team including map interfaces [75] that allow for `goto` commands or `search` commands, virtual reality [76], and gesture control [77], which allow for advanced teleoperation of the robot. These interfaces require more input from the human-in-the-loop, which means the robot system is making fewer decisions. **In comparison, we present the *first* heterogeneous teaming system with a human in the loop capable of interacting with the system through natural language.**

III. SYSTEM OVERVIEW

A. Mission Specification

This section describes the overall system architecture and mission specification using language as depicted in Fig. 1. A human operator requests assistance with a particular task from one or more UGVs, which infers the semantics in the scene that need to be found. The UGVs also infer a list of labels to request the UAV to search for, in addition to traversable classes like paved roads.

The UAV then produces an initial graph with traversable regions and objects of interest, which is incrementally transmitted through MOCHA to the UGVs. New labels can be added on-demand by the UGVs and sent to the UAV. The UAV also acts as a communication relay, carrying messages between the different robots and the operators as needed.

Once a graph has been received, the UGVs plan a trajectory using the graph to objects of interest. If objects are found, they are reported to the operator. Our current system can run as many ground robots as required, but no coordination occurs between them. Additionally, we can pre-generate a partial environment graph using other data sources, such as satellite images.

B. System Architecture

A system overview is presented in Fig. 2. The different subcomponents are described in Sec. IV. We use a Global

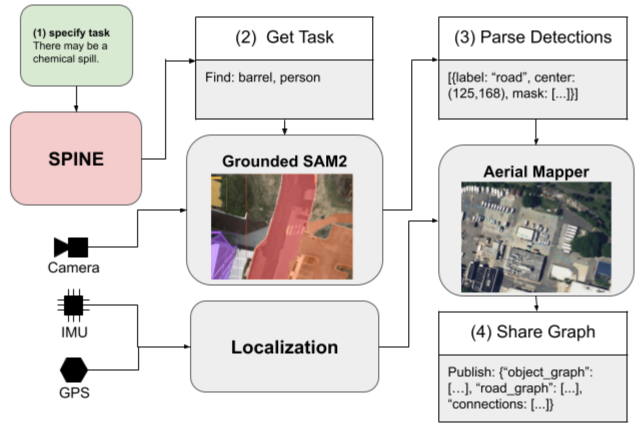

FIGURE 3. UAV mapping system with dynamic input from the UGV. (1) A user defines a task which is passed to the SPINE planner. (2) The SPINE planner assigns the UAV semantic classes to look for. (3) The UAV mapper parses the detections from Grounded SAM2 to produce a map. (4) The map is shared with the UGVs.

Navigation Satellite System (GNSS) to establish the positions of the different robots, as well as to annotate the coordinates of the nodes in the graph produced by the aerial robot.

C. Communications

The list of messages transmitted by each robot over MOCHA is described in Tab. 1, with a representative size for each compressed message 10 minutes after the start of the mission.

We used Rajant [78] breadcrumb radios for our physical communication layer which MOCHA runs on top of. Robots use the Rajant DX-2 and Cardinal radios, whereas the base station runs a FE1 radio. The Rajant Breadcrumb API is used to obtain information regarding the link quality, such as the received signal strength indicator (RSSI). This information is used to trigger a communication exchange between two robots.

D. Assumptions

Compared to our previous work [9], our system does not feature cross-view localization between UAV and UGVs.

We also require continuous communication between the UGVs and a base station with internet access so that they can forward LLM API calls for SPINE. We deployed *dummy* ME4 nodes to act as communication relays during our experiments for this. These nodes do not participate actively in the opportunistic communication process.

Finally, the UAV exploration mission is set before execution (see Sec. IV.D). No active exploration is performed on the UAV.

IV. METHODOLOGY

This section describes the components running onboard the UAV and UGVs and how they complement each

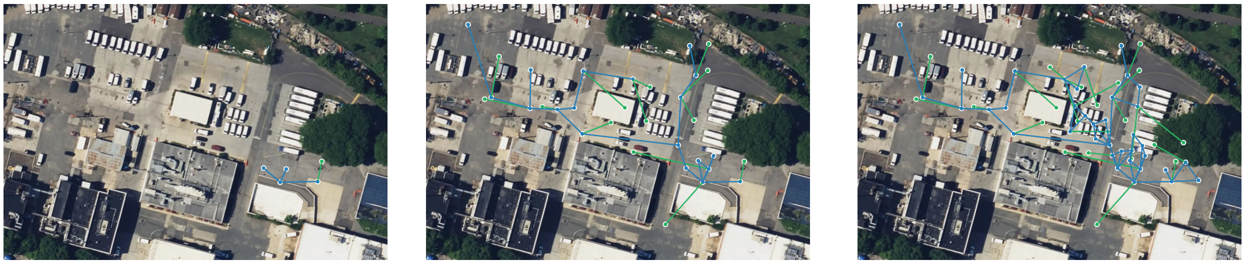

FIGURE 4. An example of the generated semantic-metric map from the aerial robot is shown at different mapping iterations overlaid on Mapbox [79] satellite imagery. The estimated traversable graph is shown in blue and the semantics (in this example `orange barrel`) are shown in green. Left: Mapping iteration=4, Center: Mapping iteration=12; Right: Mapping iteration=25.

other. A key component of the approach is to use an *open-set* semantic-metric map that can be shared between the robots (compact communication) and efficiently supply updates in a structured format to the LLM-based decision maker. The core representation used by SPINE on the UGVs is a semantic-topological graph. As is common in the semantic mapping literature, each node is either a *region* or an *object*. Regions are traversable points in free space [14], [26]. An edge between two regions indicates an obstacle-free path. Localized objects are represented as object nodes. An edge between an object and a region node indicates that the object can be observed from that region. Regions and objects may be enriched with additional semantic information (e.g., this bridge is blocked by a vehicle), which provides additional context for planning. Both UGVs and UAV mapping approaches use Grounding DINO [80] to detect objects of interest and incorporate information into the map to account for their particular visual and compute capabilities.

A. UAV Semantic Mapper

This module is designed to build an object-centric semantic map that is grounded in a common frame of reference. It also maintains a region network of the environment that the UGVs can use for global planning. The components of the UAV semantic mapper are shown in Fig. 3, and an example of such a map is shown in Fig. 4. For the purpose of this work, we assume that the traversable component for the UGVs is the semantic class `road`. Still, our system can handle other classes if desired or required by the operation environment or mission.

At the start of each mission, the set of labels is communicated to the aerial robot using MOCHA. The mapper then listens for images and their associated GNSS location and orientation from a custom GNSS/INS system.

Localization. From our past work [8], [81], a major challenge in building useful aerial maps is the quality of orientation estimation and synchronization between the imagery and the odometry. In high wind scenarios where the UAV may maintain a significant pitch, localized points in the image projected onto the ground plane may be incorrect by several meters. We use GTSAM [82] to fuse GNSS coordinates and IMU data to get a high rate state estimate between GNSS readings. This allows us to get an accurate pose for each image. Further details on the implementation can be found in the Appendix A.3.

Object detection and masking. We use Grounded SAM2 [83] for aerial object detections because of its zero-shot open-set performance, even on aerial data. The model combines the powers of Grounding DINO [80] for open-set object detection and SAM2 [84] for accurate segmentation of the objects. Grounding-DINO [80] is trained by aligning DINO [85] with text labels at internet scale. We use a base and large model, respectively, which combine to 12 GB of VRAM with a latency of 3 to 5 seconds when running onboard the UAV, depending on how many objects are detected.

Performance vs. compute tradeoffs. Our system allows for abstract task specifications. This underspecification can result in several objects of interest being identified for mapping. We notice empirically that when smaller objects that are harder to see from the air need to be detected, the memory requirements of the detection model exceed the available memory. In this case, we optionally use a smaller segmentation module SlimSAM [86] to ensure that memory stays within budget. We note that this will result in noisy locations of objects being added to the graph but the UGV can use its additional compute and memory capabilities to verify and correct the graph while still ingesting a coarse prior. The implications of this trade off are discussed in Sec. IV.B and demonstrated in Sec. VI.

Graph construction. Given the detections, the object coordinates are estimated by applying a distance transform on their associated pixel centroids. Each object is assigned the furthest point from the mask boundary as its coordinate. The pixel coordinates are then converted to UTM with respect to a known origin. To construct a region graph we use `road` as the semantic class. We first take the largest road mask in each image and only add a region node if the mask is more than 20% of the image. We also use the class `building` and `car` as negative objects, i.e., objects that we do not report as `road`. Without them the network often misclassifies buildings and cars as road, leading to untraversable edges in the graph. When there is a large road segment we compute the same distance transform as before and assign the road point to the furthest point from the mask boundary. This point is then transformed to UTM coordinates and added to the region graph network by connecting it to the nearest region node.

B. UGV Semantic Mapper

The UGV mapper also maintains a local occupancy map, which the planner uses for free space exploration. The semantic graph constructed by the UAV provides an initial map estimate (see Sec. IV A), and the UAV can iteratively provide map updates during the mission.

Our UGV semantic mapping implementation is shown in Fig. 2. The mapper takes RGB + Depth, LiDAR, and semantic configuration as inputs. LiDAR is used for odometry estimation (Faster-LIO [87]) and local occupancy map construction (GroundGrid [88]). The occupancy map is used to add and remove regions and edges from the map based on connectivity. RGB+D is used for object localization and captioning. Objects are detected using GroundingDino [80]. Detections are then clustered and localized with a multiple-hypothesis tracker. A vision-language model (LLaVA [89]) enriches the semantic information available to the planner. Outputs from these modules are used to add and remove nodes and enrich them with semantic information. A semantic configuration is provided by the planner and is used to set the labels of the object detector and provide queries to the vision language model. The detection and tracking modules run at $\sim$ 5 Hz, the vision-language model runs at $\sim$ 1 Hz, and occupancy map construction runs well over 10 Hz, all onboard.

C. SPINE Planning

The backbone planner used in this work is based on SPINE, first presented in [18]. SPINE is an LLM-enabled planner that takes as input a mission specification in natural language. During mission execution, SPINE builds a semantic map from both its onboard semantic mapper and the UAV map. SPINE may correct the UAV map, as it may contain errors. SPINE's LLM then reasons over the mission specification and the semantic map to generate a plan comprised of high-level behaviors, which is validated in real time. This section summarizes some key ideas relevant to teaming.

Mapping interface. The mapping interface provides a textual representation of the semantic graph for SPINE's LLM. To build this representation, the semantic graph is first serialized into a JSON string. At each subsequent planning iteration, map updates are provided to the LLM in-context via an interface which captures high-level graph manipulations: `add_nodes`, `remove_nodes`, `add_edges`, `remove_edges`, and `update_nodes`. The nodes are defined as a dictionary of attributes, which allows for providing nodes with rich semantic descriptions (example in Fig. 11).

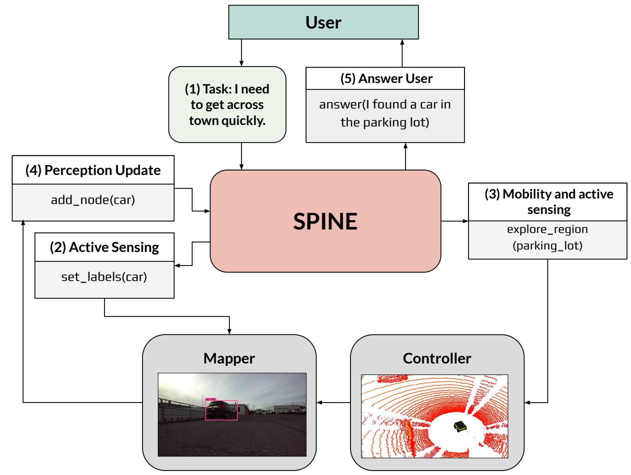

FIGURE 5. SPINE planning framework (1) User specifies a mission. (2) SPINE infers relevant semantic categories and configures perception labels. (3) SPINE then reasons about exploration targets. (4) Perception finds a car and notifies planner. (5) Planner informs user.

Plan generation. SPINE composes plans via behaviors for navigation, active mapping, and user interaction. The LLM generates a receding horizon behavior sequence at each planning iteration, and this sequence is provided to the validation module. An `answer` behavior terminates a mission and notifies the user of results, and a `clarify` behavior is used to gain further instructions, if needed.

We enforce chain-of-though reasoning by requiring the LLM to provide a justification for each action sequence, which reduces hallucinations or otherwise ungrounded behavior [90]. All inputs to the planner and action history are maintained in-context via the provided APIs. See Fig. 5 for an example action-control-perception loop.

Plan validation. To create subtasks, the planner must correctly invoke its behavior library while reasoning over constraints such as traversability. Since LLMs are prone to hallucinate this information, we filter LLM-generated plans through a validation module, which ensures all plans are syntactically correct and physically realizable. If a given task is invalid, the validator forms state-specific feedback to the LLM. While this validation procedure provides valuable planning safeguards, the spatial constraints are contingent on perception.

Online map correction. SPINE uses semantic graphs constructed from the UAV to form initial plans and accelerate replanning during mission execution. While these maps offer valuable contextual and traversability information, they may contain spurious traversability edges. Identifying such errors online so that SPINE can replan is vital to a robust autonomy solution. SPINE thus uses feedback from the downstream controllers to infer spurious edges and remove them from the graph. If SPINE sends a navigation command across an edge that

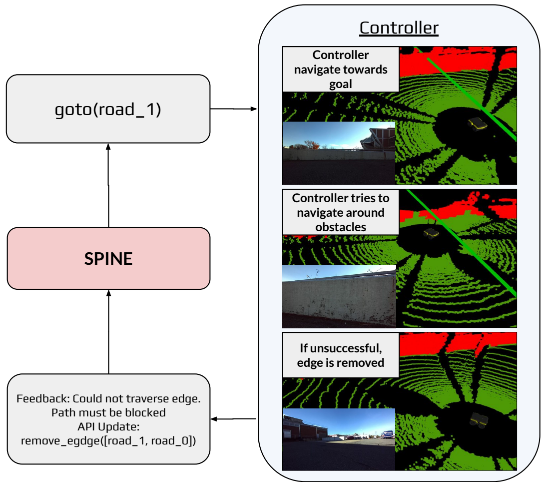

FIGURE 6. Example map correction. SPINE attempts to traverse an invalid edge (green line). Green points denote ground and red points denote obstacles. The controller cannot find a trajectory around the obstacle, and it times out after a pre-specified duration. The incorrect edge is removed from the graph, and that feedback is sent to SPINE.

the controller cannot traverse, SPINE removes that edge from the graph, as illustrated in Fig. 6.

D. Mission Execution

The UAV is tasked with an initial exploration mission over the area of interest. For our experiments, a predefined waypoint set is provided to avoid flying over unsafe regions. However, waypoints can be generated with any coverage algorithm.

After a timer t_i has elapsed, the UAV transitions into search and communication modes, looking for ground robots to transmit the graph. The last known pose is used as a heuristic for where to find the ground robot. Once the ground robot has been found and all the data has been transmitted, the UAV transitions again into exploration mode, covering all the waypoints in its mission.

On the UGV, SPINE sends navigation subtasks to the low-level planner, which references the semantic graph (e.g., navigation on previously observed portions of the graph, extending the graph via exploration, etc.). Graph navigation paths are then sent to the ROS Move Base controller[2], which plans trajectories realized by the Jackal controller.

V. EXPERIMENTS

We evaluate our system over three sets of experiments. We first evaluate the incremental graph construction of the aerial autonomy component of our system (Sec. V.A). We then evaluate the ground autonomy's ability to realize language-specified missions in partially-known environments (Sec. V.B). Finally, we evaluate air-ground teaming in unknown environments given language-specified missions (Sec V.C).

[2] http://wiki.ros.org/move_base

FIGURE 7. Example images of the detection and segmentation onboard the aerial robot. Top: Pennovation. Bottom: Range 15. The red line indicates the flight path of the UAV overlaid on Mapbox [79] satellite imagery.

Platforms. We used the Falcon 4 [91] as our UAV platform. The UAV is controlled by a flight controller running the PX4 firmware [92]. The primary sensors are a Blackfly S U3-32S4C-C RGB global-shutter camera, and a VectorNav VN-100T inertial measurement unit (IMU). The platform uses a U-blox ZED F9P GNSS for localization. Finally, all the data is processed by an onboard Nvidia Jetson Orin NX.

Our UGV platform consists of a Clearpath Jackal equipped with an AMD Ryzen 3600 CPU with 32 GB of RAM and an Nvidia RTX 4000 Ada SFF GPU. An Ouster OS1-64 LiDAR is used for obstacle avoidance, path planning, and odometry, while a ZED 2i stereo camera provides sensing for object detection. We also use a U-blox ZED-F9P GNSS to record the initial position of the UGV, which establishes a common reference frame with the overhead graph. Finally, we use a VectorNav VN-100T IMU for global heading corrections. More details about the hardware platforms can be found in the Appendix.

Environments. We perform ground autonomy experiments in two rural environments termed Buckner and Range 15, shown in Fig. 10 and Fig. 9, respectively. Each environment stresses a unique component of the auton-

Specification	Total Detections	False Positives	Precision
Pennovation 1	708	127	82.1 %
Pennovation 2	388	59	84.8 %

TABLE 2. Performance of the aerial object detection module based on Grounded SAM2, zero-shot transfer.

omy stack. Range 15 is large and semantically sparse. We use this environment to evaluate the autonomy's reasoning over long-horizon missions, state estimation, and communications at scale. Buckner is smaller in scale but features richer semantics, such as bridges, gates, and parking lots, all within a few hundred meters. Buckner experiments evaluate the UGV autonomy's ability to reason over richer or ambiguous specifications.

We demonstrate the full air-ground system in Pennovation, an urban office park with fields, buildings, parking lots, and other structures. We also use data from these demonstrations to evaluate the aerial autonomy performance. All environments were uncontrolled. There were dynamic entities such as cars and people moving during experiments, and the environments contained many obstacles, both positive (fences, walls) and negative (ditches).

A. Aerial Mapping

In this section, we evaluate the components of the UAV mapping system. The important qualities of our aerial maps are: 1) Resolution, task relevance, semantic detections; 2) Small size for transmission between robots; 3) Accurate estimate of the traversable portions of the environment. We design experiments to measure the effectiveness of our proposed method for these qualities. We first test the quality of the object detector, and we then measure the size of the stored maps relative to the distance traveled.

Object detections. To evaluate our object detection module, we manually annotate the desired objects in two of our datasets and evaluate the false positive rate of the detector. The desired labels for this evaluation are task specific to these datasets and are not the same across both datasets. The results of this evaluation are shown in Tab. 2. As shown, the object detector is able to perform remarkably well with zero-shot transfer to our data. The results are surprising considering the viewpoint of the UAV camera. The qualitative evaluation of the detected objects is shown in Fig. 7.

Incremental graph construction. We show an example of the incremental semantic-metric map generated by the robot overlaid onto a satellite image in Fig. 4. An advantage of our method is the reduced size of the map. We show in Fig. 8 that the maps are roughly linear in the distance traveled, and areas of size 20,000 m^2 can be stored in just a few kilobytes.

Specification	Total Detections	False Positives	Precision
Pennovation 1	82	9	89.0 %
Pennovation 2	24	2	91.6 %

TABLE 3. Performance of the traversable edge detection module.

Compared to our previous work [8], [9], the messages transmitted between robots are significantly smaller. For instance, the size of the uncompressed graph after flying 1000 m is approximately 12 to 17 KB, compared to 530 to 699 KB of the semantic map image generated by ASOOM. These results showcase the efficiency of the aerial semantic graph generated by the UAV compared to the previously used dense maps.

Traversability estimation. Finally, to measure the quality of the estimated road network, we manually annotate the true positives and false positives in the detected roads. The results of this analysis are shown in Tab. 3. These results are encouraging: accurate road maps improve the performance of the ground autonomy stack.

B. Ground Autonomy

We evaluate the UGV autonomy platform independently of the UAV. We design experiments to assess the UGV autonomy's ability to complete missions with differing specifications, environments, and priors. In contrast to the air-ground teaming evaluation and system demonstration in Sec. V.C and VI, where the UAV provides a semantic map, the UGV is provided a semantic map from registered satellite imagery prior to mission execution. These priors contain traversability information and some relevant semantics. However, the priors are imperfect and contain irrelevant information and mistakes that the UGV must correct online. Our experiments evaluate the UGV's ability to infer navigation, exploration, and information acquisition goals from natural language spec-

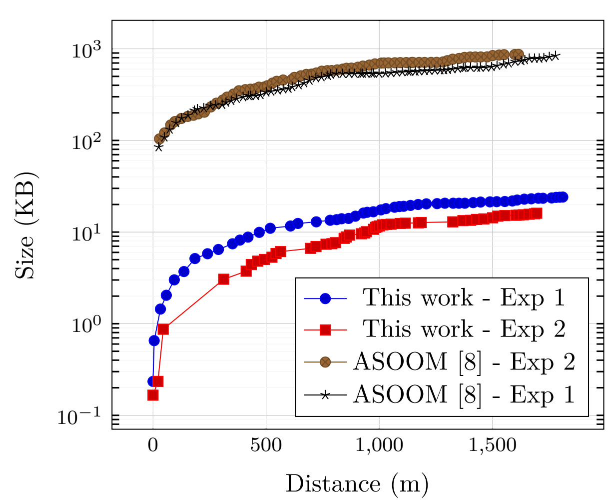

FIGURE 8. Size of the uncompressed map as a function of the distance traveled by the UAV.

ifications, react to findings, and use findings to complete the user's mission.

Experimental setup. We consider five mission specifications, as summarized in Tab. 4, and evaluate each specification one to four times. We construct priors for each environment; however, the priors are not designed for a specific task. One of the primary purposes of the prior was to keep the robot away from negative obstacles, such as ditches, which could not be detected by the UGV.

Results and discussion. We summarize the experimental results for each mission specification in Tab 5. We report mission success, distance traveled, and the number of API calls. The number of API calls indicates how many subtask steps SPINE inferred during its mission.

The UGV was successful in achieving S2 through S5. There were two unsuccessful missions during S1, each of which failed due to the large scale of the environment. In one outcome, a gust of wind blew leaves around the ground vehicle's LiDAR, which caused unrecoverable odometry drift (see Fig. 17). In the second, the UGV lost communications and could not perform the LLM queries required for planning.

We highlight two emblematic missions. Fig. 10 shows the UGV trajectory from a mission with the specification: "I heard the bridge was blocked. Can you check?". SPINE infers that the relevant semantics are `bridge`, `car`, `truck`, `bicycle`, `pedestrian`, `construction cone`, `debris`, and `barrier`. The planner then navigates to the bridge by constructing the following plan `goto(road_5), map_region(bridge_1)`. Upon mapping the bridge, SPINE discovers several obstacles, including a person, car, and construction barrier. SPINE uses the VLM to get a more detailed description, as shown in Fig. 11, and reasons over this information to respond "The path across the bridge is likely blocked due to the presence of a construction cone and a person. There is also a silver car and a woman on the bridge, which may indicate a temporary blockage."

Fig. 9 overviews a mission in which the UGV was provided the task: "I heard of suspicious activity near the red houses. Go check." The UGV had red houses on its prior map, so it infers a plan to map those. Along the way, it observes cars and people. Because those objects are not near the red houses, the UGV records those in its map but does not consider them as relevant to the task. Finally, the UGV reaches the red houses, discovers people, and reports them to the user. The user then asks the UGV to return to the starting location to offload data. Overall the mission requires the UGV to traverse 1200 meters, which test all layers of the autonomy stack including the odometry and communication network. SPINE uses active perception to resolve errors in the mission specification. Examples involving active perception are presented in the Appendix.

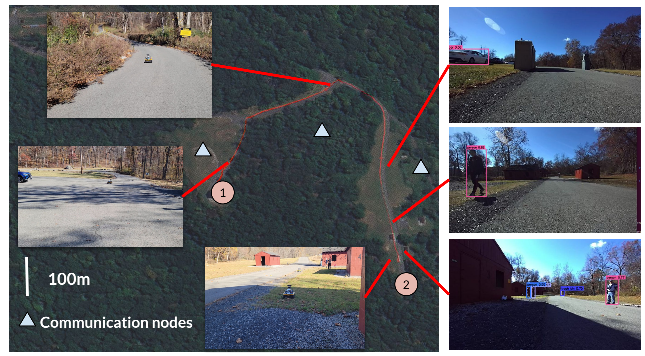

FIGURE 9. Example mission from S1. "I heard of activity near the red houses. Go check." At the start (1), the UGV identifies exploration targets (red houses, bottom) and navigates there from its start location (2). The UGV successfully identifies several people. The ground vehicle communicates its findings in real time via text. It then returns to its starting location to offload mission data.

FIGURE 10. Example mission from S3. "I got reports that the bridge was blocked. Go check." The UGV starts its mission at the bottom left of the trajectory (1). The UGV then infers a mapping target (bridge). After navigation and mapping, it identifies several obstacles and reports its findings to the user (2).

C. Air-Ground Teaming Evaluation

We evaluate the performance of our air-ground teaming system against single-UGV variants on language-specified missions in unknown environments. The first baseline ("UGV w/ GT") receives a full map constructed from satellite imagery, which estimates upper-bound performance of the mission given our UGV autonomy stack. The second baseline ("UGV") receives no prior map, and

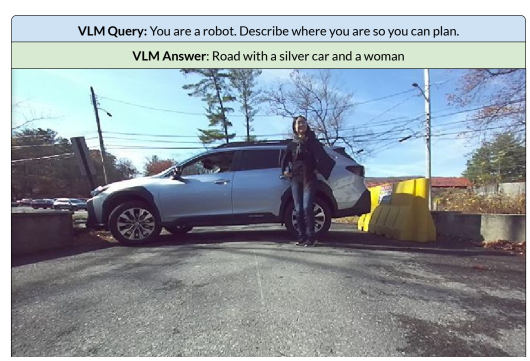

FIGURE 11. In the above example, the ground vehicle must learn if the bridge is blocked (Specification 3). The ground vehicle's planner, SPINE, uses a Vision Language Model running onboard to obtain detailed semantic information for the user.

Specification	Location	Priors	Desired outcome
(S1) I heard of activity near the red houses. Go check.	Range 15	Red houses. No objects.	Identity and report several people near two red houses.
(S2) Go inspect the black vehicle at the end of the driveway.	Range 15	Black vehicle is at the beginning of the driveway.	Planner identifies misspecification and correctly finds car.
(S3) I got reports that the bridge was blocked. Go check.	Buckner	Bridge, no blockage.	Planner identities vehicle blocking the bridge.
(S4) Is there activity near the gate.	Buckner	Gate, objects.	Planner identifies and reports a person.
(S5) Identify the parking lot 30 meters east of the bridge.	Buckner	Bridge, but no parking lot.	Planner explore correct area and identifies parking lot.

TABLE 4. Mission specifications, locations, and priors used to test the ground autonomy.

this baseline must explore to accomplish the mission. We consider a triaging mission where the system must respond to a chemical spill by inferring relevant semantics (*i.e.,* chemical barrels, people), map the relevant area, and report its findings. The system is given the specification "triage the chemical spill 100 meters X," where X is a location hint, such as southeast. We repeat this experiment three times for each baseline and six times for our system, with location hints to the north and northeast.

We report results in Tab. 6, as measured by success rate, distance traveled by the UGV, and the percentage of time spent in autonomous mode. Unsurprisingly, the UGV obtains a 100% success rate when given a full prior map of the environments. Our system achieves an 83.3% success rate while spending a compatible amount of time in autonomous mode. Our system also travels a similar distance, indicating that the prior map provided by the UAV offers efficient paths. Without a prior map, the UGV is unable to find the goal. We find that the UGV increasingly struggles with long-scale exploration, as indicated in Fig. 12.

VI. SYSTEM DEMONSTRATIONS

Through the previous Sec. V, we demonstrated the use of the system in relatively controlled settings. In this section, we show qualitative demonstrations on underspecified missions that require reasoning and involve sub-tasks that have to be inferred by the system. As mentioned in Sec. IV.A, a smaller model is used in the UAV to generate the object masks, to keep the memory within budget. We demonstrate the full system through two experiments in the Pennovation environment. Large-scale demonstrations stress the autonomy stack of the ground robots, the opportunistic communication, and the behavior of the air-ground team. Note that in addition to the components evaluated previously, we stress the human-in-the-loop aspect of our system. In particular, the operator is constantly receiving updates and retasking the robot.

Spec.	Outcomes	Dist. (m)	API Calls	Failure modes
S1	1/3	1200	2	Odom., Comm.
S2	2/2	231	4	N/A
S3	4/4	265	2	N/A
S4	1/1	132	2	N/A
S5	1/1	450	3	N/A

TABLE 5. UGV autonomy outcomes. SPINE completed all missions except for two during the first specification, which failed due to odometry drift and communication loss.

Method	Success (%)	Dist. (m)	Autonomous (%)
UGV	0	54.9	94.6
UAV-UGV	83.3	104.9	99.3
UGV w/ GT	100	100.7	99.7

TABLE 6. Experiments with air-ground teaming compared to UGV-only methods. GT corresponds to a pre-computed map from satellite imagery.

We report the total distance traveled by the UGV, mission duration, number of user interactions, number of LLM API calls, number of edges in the UAV-derived semantic map that were removed by the UGV's planner, and percent of the time that the UGV was in autonomous mode in Tab. 7. We also report the number of nodes in the UGV's semantic graph over the duration of the mission in Fig. 13.

System demonstration 1: Mapping and inspection. The first mission was specified by the query "I heard of construction around the eastern roads. Can you check?." The planner inferred that the following classes were relevant: `crane`, `bulldozer`, `cement mixer`, `construction sign`, `scaffolding`, `excavator`, `hard hat`, `construction worker`, `truck`, and `barrier`. The UAV incrementally built a semantic graph and provided it to the UGV. Throughout the mission, the UGV reported relevant findings to the human operator, who would then retask the UGV based on those findings. Overall, the UGV traveled 756 meters and received 21 updates from the operator. The UGV spent nearly 90% of its time in autonomous mode. The manual takeovers came primarily from dynamic obstacles, such as buses or trucks, or small positive obstacles that the UGV obstacle detection could not identify, such as curbs, and negative obstacles such as puddles or potholes. The ratio of API calls to user interactions is lower than in the UGV-only experiments (see Tab. 5). This is because, in the UGV-only experiments, the ground robot inferred all decisions autonomously, whereas in this experiment the UGV would reach back to the human operator. Throughout the mission, the UGV provides this information via detected objects and scene captions associated with the graph. Importantly, the UGV also provides negative information (e.g., "this is a parking lot with no construction activity").

System demonstration 2: Triaging. The second mission was specified by the query "you are working with a high-altitude UAV to search for people." During mission execution, the user provided additional details. For example, the user first instructed the UGV to look 30 meters south and 50 meters east, in order to complement the UAV's flight path. SPINE infers the relevant classes are `person`, `tree`, `car`, `bicycle`, `bench`, `backpack`, `streetlight`, `trash can`, and `umbrella`. Following a similar mission concept, the UAV is provided with the same list of classes to generate and provide a semantic map. SPINE also provides an interpretable explanation for the chosen classes: For instance, `bench`, `backpack`, and `trash can` indicate places where people might be found or have left their belongings. `Streetlights` provide "structural context to the environment". The UGV works with the user to explore the environment. Throughout the mission, the UGV provides natural language description of the previously unexplored area, which provides situational awareness to the user and aids in planning. Snapshots of the experiment are shown in Fig. 14. Overall, the UGV correctly identified five people. The detection system reported one false positive, and one person was in the UGV field of view, but the tracker failed to register that person.

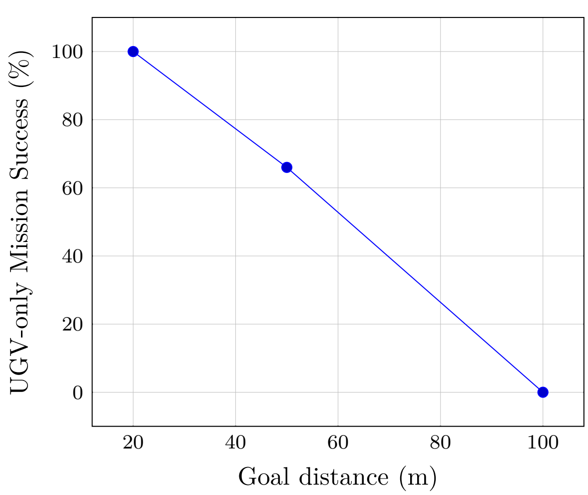

FIGURE 12. UGV-only mission success as goal distance increases. Without UAV-generated semantic maps or priors from satellite imagery, the UGV must build a map entirely from exploration, which it struggles to do as the goal distance increases.

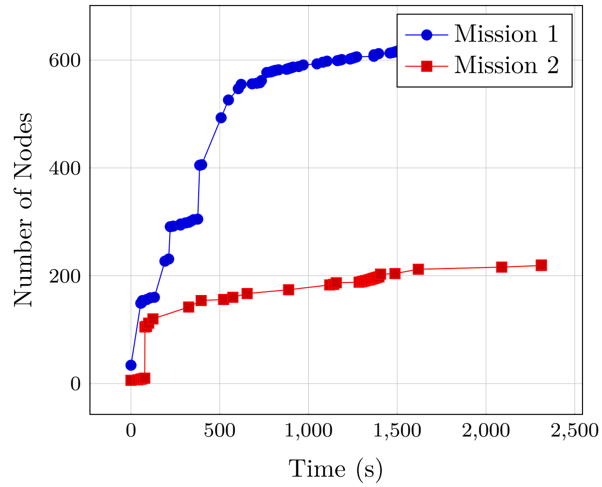

FIGURE 13. Size of graph as a function of the mission time for the ground vehicles. Ground vehicles receive graph updates from onboard mapping and the aerial vehicle, thus the map grows substantially over the mission.

VII. DISCUSSION, CONCLUSION, AND FUTURE WORK

We conclude the paper by discussing our results, summarizing our key contributions, and outlining promising directions for future work.

A. Discussion

UGV traversability estimation was a primary system bottleneck. The UGV struggled to identify negative and small positive obstacles, limiting the area that the UGV could safely explore. For example, Fig. 15 shows a portion from the second system demonstration where the UGV failed to identify a curb. While the semantic graph from the UAV provided valuable traversability information, it contained false positives such as buildings marked as roads. The UGV often identified these errors online and corrected its path. However, the safety operator had to take over when false positives brought the UGV through curbs or other challenging obstacles. This is particularly apparent in Sec. VI, where the specificity of the task is lower, and a smaller model is used to generate the object masks in the UAV, leading to harder trajectories for the UGV to follow.

The UAV and UGV both take advantage of open-set object detectors to locate task relevant objects. These models often perform worse than tuned closed-set detects and are more prone to false positives as shown in Fig. 16.

The system used GNSS to register maps between robots, but the UGV used LiDAR odometry only for state estimation. While LiDAR odometry is typically a robust state estimation solution, it suffers from drift in highly dynamic scenes, as shown in Fig. 17.

Finally, we observed that the LLM-enabled planner was unable to perform robust exploration, as evidenced by our UGV-only exploration experiments (see Fig 12), which made the UGV dependent on a strong prior from the UAV.

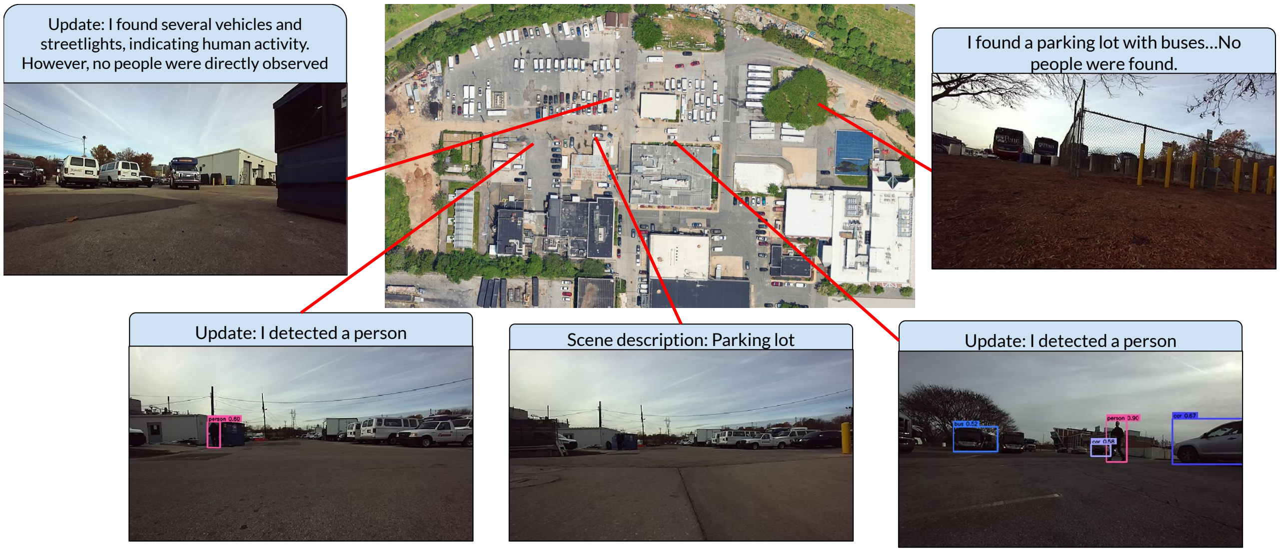

FIGURE 14. Snapshots of the second system demonstration. The air-ground team is tasked with finding people in the Pennovation environment. The ground vehicle provides updates in natural language about mission findings (people) and broader context (scene description, etc.). Because the environment is unknown at runtime, such context provides valuable situational awareness to the user.

Specification	UGV Distance (m)	Time (s)	User Interactions	API calls	Removed Edges	(%) Autonomous
(S6) I heard of construction around the eastern roads. Can you check?	765	2097	21	41	8	90
(S7) You are working with a high-altitude UAV to search for people.	695	2390	19	37	15	93

TABLE 7. System demonstration results over two different specifications. Both missions require the air ground system to acquire information in the Pennovation environment, thus the resulting metrics are similar.

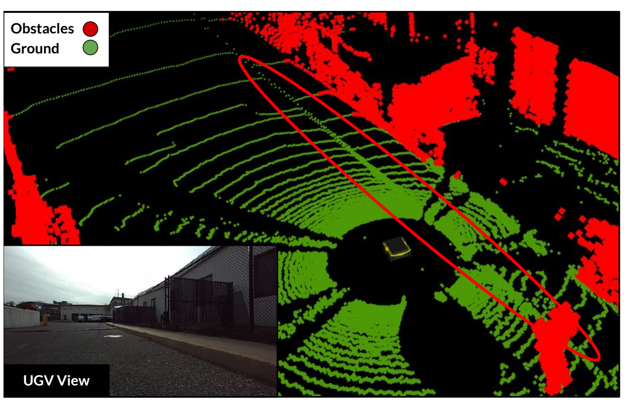

FIGURE 15. Example reason for manual takeover. Curbs (circled) are not detected by traversability estimation, thus the UGV tries to drive over them.

B. Conclusion

In summary, we present an air-ground teaming system for natural-language missions in unknown environments. The system infers relevant semantics given a specification. An aerial vehicle then incrementally builds a mission-relevant semantic map, which is relayed to a ground vehicle. The ground vehicle then uses an LLM-enabled planner to infer and realize subtasks that realize the mission, and the planner leverages online semantic mapping to augment and correct the semantic map received from the aerial vehicle. During the mission, the aerial vehicle intermittently provides a map update to the ground vehicle via an opportunistic communication network. We evaluate our system over seven different specifications in urban and rural environments in kilometer-scale navigation missions.

C. Future Work & Open Challenges

We see several interesting directions for future work. We use LLMs to translate and act upon instructions in natural language and leverage a structured map format for in-context updates. While this empirically works, we lose information in the construction of the maps and are explicitly reliant on an uplink to communicate with the language model API.

Onboard language models. In this work, we use the OpenAI GPT 4o model, which makes us reliant on a continuous communication infrastructure. In our ideal system, we would be able to decompose the main task from the user into subtasks that onboard LLMs can handle. Keeping the interface between the robots as natural language allows us to maintain compact, text

FIGURE 16. **Example misclassifications made by open-vocabulary object detection. Top: benign misclassification, falsely detected street sign. Bottom: detrimental misclassification, falsely detected person.**

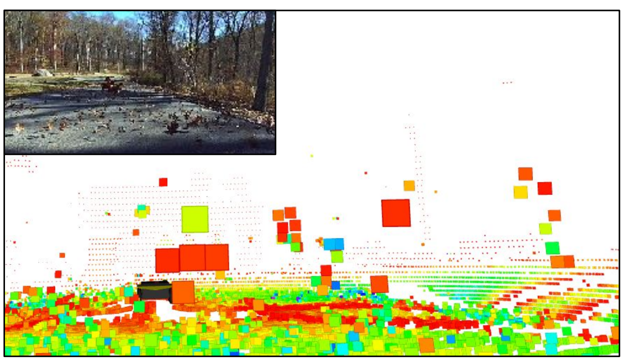

FIGURE 17. **Example of odometry failure. A gust of wind blew leaves around the ground vehicle's LiDAR. The spurious returns caused unrecoverable odometry drift.**

based communication between the robots making it interpretable by humans in the loop.

Traversability estimation. While segmenting aerial images appears to have good zero-shot performance, imperfections in the detections can lead to catastrophic failure if the UGV uses a naive local planner. Failure cases such as minor localization errors, inability to perceive thin objects like fences, and the change in depth between roads and curbs may result in collisions.

Additionally, while a coarse traversability estimation can be computed by segmenting roads, paths, and grass, these are not always meaningful for all robot modalities. For example, smaller robots will have difficulty traversing potholes, speed bumps, or curbs, which may be included in a road segmentation. Incorporating a robot-based traversability would be a valuable addition to this work. For instance, in addition to the pre-defined `road` label, we can task the aerial robot to find semantic classes that align with the robot's experience or claimed capability. Finally, in future work we plan to use dense depth images to find smaller objects like tree roots or potholes that robots can avoid based on their capabilities.

In this framework, we can also imagine re-tasking UAVs to regions in the environment that are semantically meaningful. Our pipeline is able to accommodate the geometric clustering of semantically meaningful objects into regions. This, in turn, could be used to interactively ask the aerial robot to survey desired locations in the environment with tasks such as "Map all the `cars` in `parklot_01`".

ACKNOWLEDGMENTS

We would like to acknowledge Katherine Mao, Frank Gonzalez, Dexter Ong, and Kashish Garg for their help during the field experiments for this work. We would like to acknowledge Dr. Michael Novitzky and Tyler Errico for their help and support during field experiments at USMA West Point.

REFERENCES

[1] D. Lattanzi and G. Miller, "Review of robotic infrastructure inspection systems," *Journal of Infrastructure Systems*, vol. 23, no. 3, p. 04017004, 2017.

[2] T. H. Chung, V. Orekhov, and A. Maio, "Into the robotic depths: analysis and insights from the darpa subterranean challenge," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 6, no. 1, pp. 477–502, 2023.

[3] G.-J. M. Kruijff, F. Pirri, M. Gianni, P. Papadakis, M. Pizzoli, A. Sinha, V. Tretyakov, T. Linder, E. Pianese, S. Corrao *et al.*, "Rescue robots at earthquake-hit mirandola, Italy: A field report," in *2012 IEEE international symposium on safety, security, and rescue robotics (SSRR)*. IEEE, 2012, pp. 1–8.

[4] H. G. Nguyen and J. P. Bott, "Robotics for law enforcement: Applications beyond explosive ordnance disposal," in *Enabling Technologies for Law Enforcement and Security*, vol. 4232. SPIE, 2001, pp. 433–454.

[5] S. Fountas, N. Mylonas, I. Malounas, E. Rodias, C. Hellmann Santos, and E. Pekkeriet, "Agricultural robotics for field operations," *Sensors*, vol. 20, no. 9, p. 2672, 2020.

[6] C. Liu, J. Zhao, and N. Sun, "A review of collaborative air-ground robots research," *Journal of Intelligent & Robotic Systems*, vol. 106, no. 3, p. 60, 2022.

[7] L. Chaimowicz, B. Grocholsky, J. F. Keller, V. Kumar, and C. J. Taylor, "Experiments in multirobot air-ground coordination," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 4. IEEE, 2004, pp. 4053–4058.

[8] I. D. Miller, F. Cladera, T. Smith, C. J. Taylor, and V. Kumar, "Stronger together: Air-ground robotic collaboration using semantics," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9643–9650, 2022.

[9] ——, "Air-ground collaboration with SPOMP: Semantic panoramic online mapping and planning," *IEEE Transactions on Field Robotics*, vol. 1, pp. 93–112, 2024.

[10] F. Cladera, Z. Ravichandran, I. D. Miller, M. A. Hsieh, C. Taylor, and V. Kumar, "Enabling large-scale heterogeneous collaboration with opportunistic communications," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 2610–2616.

[11] A. Agha, K. Otsu, B. Morrell, D. D. Fan, R. Thakker, A. Santamaria-Navarro, S.-K. Kim, A. Bouman, X. Lei, J. Ed-

lund *et al.*, "Nebula: Team costar's robotic autonomy solution that won phase ii of darpa subterranean challenge," *Field robotics*, vol. 2, pp. 1432–1506, 2022.

[12] M. Tranzatto, T. Miki, M. Dharmadhikari, L. Bernreiter, M. Kulkarni, F. Mascarich, O. Andersson, S. Khattak, M. Hutter, R. Siegwart *et al.*, "Cerberus in the darpa subterranean challenge," *Science Robotics*, vol. 7, no. 66, p. eabp9742, 2022.

[13] A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," in *Conference on robot learning*. PMLR, 2023, pp. 287–318.

[14] A. Werby, C. Huang, M. Büchner, A. Valada, and W. Burgard, "Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation," in *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*, 2024.

[15] J. X. Liu, Z. Yang, I. Idrees, S. Liang, B. Schornstein, S. Tellex, and A. Shah, "Lang2ltl: Translating natural language commands to temporal robot task specification," in *Conference on Robbot Learning (CoRL)*, 2023. [Online]. Available: https://arxiv.org/abs/2302.11649

[16] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. Reid, and N. Suenderhauf, "Sayplan: Grounding large language models using 3d scene graphs for scalable task planning," in *7th Annual Conference on Robot Learning*, 2023. [Online]. Available: https://openreview.net/forum?id=wMpOMO0Ss7a

[17] F. Cladera, I. D. Miller, Z. Ravichandran, V. Murali, J. Hughes, M. A. Hsieh, C. Taylor, and V. Kumar, "Challenges and opportunities for large-scale exploration with air-ground teams using semantics," *arXiv preprint arXiv:2405.07169*, 2024.

[18] Z. Ravichandran, V. Murali, M. Tzes, G. J. Pappas, and V. Kumar, "Spine: Online semantic planning for missions with incomplete natural language specifications in unstructured environments," *To appear in ICRA 2025*, 2024. [Online]. Available: https://arxiv.org/abs/2410.03035

[19] A. Elfes, M. Bergerman, J. R. H. Carvalho, E. C. de Paiva, J. Ramaos, and S. S. Bueno, "Air-ground robotic ensembles for cooperative applications: Concepts and preliminary results," in *Proceedings of the International Conference on Field and Service Robotics*, vol. 1999, 1999.

[20] N. Vandapel, R. R. Donamukkala, and M. Hebert, "Unmanned ground vehicle navigation using aerial ladar data," *The International Journal of Robotics Research*, vol. 25, no. 1, pp. 31–51, 2006.

[21] M. A. Hsieh, A. Cowley, J. F. Keller, L. Chaimowicz, B. Grocholsky, V. Kumar, C. J. Taylor, Y. Endo, R. C. Arkin, B. Jung *et al.*, "Adaptive teams of autonomous aerial and ground robots for situational awareness," *Journal of Field Robotics*, vol. 24, no. 11-12, pp. 991–1014, 2007.

[22] N. Michael, S. Shen, K. Mohta, V. Kumar, K. Nagatani, Y. Okada, S. Kiribayashi, K. Otake, K. Yoshida, K. Ohno *et al.*, "Collaborative mapping of an earthquake damaged building via ground and aerial robots," in *Field and service robotics: results of the 8th international conference*. Springer, 2014, pp. 33–47.

[23] A. Asgharivaskasi and N. Atanasov, "Semantic octree mapping and shannon mutual information computation for robot exploration," *IEEE Transactions on Robotics*, 2023. [Online]. Available: https://arashasgharivaskasi-bc.github.io/SSMI_webpage/

[24] Y. Tao, X. Liu, I. Spasojevic, S. Agarwal, and V. Kumar, "3d active metric-semantic slam," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2989–2996, 2024.

[25] M. S. Kurtz, S. Prentice, Y. Veys, L. Quang, C. Nieto-Granda, M. Novitzky, E. Stump, and N. Roy, "Real-world deployment of a hierarchical uncertainty-aware collaborative multiagent planning system," 2024. [Online]. Available: https://arxiv.org/abs/2404.17438

[26] N. Hughes, Y. Chang, S. Hu, R. Talak, R. Abdulhai, J. Strader, and L. Carlone, "Foundations of spatial perception for robotics: Hierarchical representations and real-time systems," *The International Journal of Robotics Research*, 2024. [Online]. Available: https://doi.org/10.1177/02783649241229725

[27] J. Strader, N. Hughes, W. Chen, A. Speranzon, and L. Carlone, "Indoor and outdoor 3d scene graph generation via language-enabled spatial ontologies," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 4886–4893, 2024.

[28] S. Garg, K. Rana, M. Hosseinzadeh, L. Mares, N. Sünderhauf, F. Dayoub, and I. Reid, "Robohop: Segment-based topological map representation for open-world visual navigation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 4090–4097.

[29] H.-T. L. Chiang, Z. Xu, Z. Fu, M. G. Jacob, T. Zhang, T.-W. E. Lee, W. Yu, C. Schenck, D. Rendleman, D. Shah, F. Xia, J. Hsu, J. Hoech, P. Florence, S. Kirmani, S. Singh, V. Sindhwani, C. Parada, C. Finn, P. Xu, S. Levine, and J. Tan, "Mobility vla: Multimodal instruction navigation with long-context vlms and topological graphs," 2024. [Online]. Available: https://arxiv.org/abs/2407.07775

[30] Q. Gu, A. Kuwajerwala, S. Morin, K. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa, C. Gan, C. de Melo, J. Tenenbaum, A. Torralba, F. Shkurti, and L. Paull, "Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning," *International Conference on Robotics and Automation*, 2024.

[31] D. Maggio, Y. Chang, N. Hughes, M. Trang, D. Griffith, C. Dougherty, E. Cristofalo, L. Schmid, and L. Carlone, "Clio: Real-time task-driven open-set 3d scene graphs," *IEEE Robotics and Automation Letters*, 2024.

[32] Y. Tian, Y. Chang, F. Herrera Arias, C. Nieto-Granda, J. P. How, and L. Carlone, "Kimera-multi: Robust, distributed, dense metric-semantic slam for multi-robot systems," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2022–2038, 2022.

[33] Y. Chang, N. Hughes, A. Ray, and L. Carlone, "Hydra-multi: Collaborative online construction of 3d scene graphs with multi-robot teams," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 10 995–11 002.

[34] D. Honerkamp, M. Büchner, F. Despinoy, T. Welschehold, and A. Valada, "Language-grounded dynamic scene graphs for interactive object search with mobile manipulation," *IEEE Robotics and Automation Letters*, 2024.

[35] Z. Hu, F. Lucchetti, C. Schlesinger, Y. Saxena, A. Freeman, S. Modak, A. Guha, and J. Biswas, "Deploying and evaluating llms to program service mobile robots," *IEEE Robotics and Automation Letters*, 2024.

[36] S. Sharan, F. Pittaluga, V. Kumar B G, and M. Chandraker, "Llm-assist: Enhancing closed-loop planning with language-based reasoning," *arXiv preprint arXiv:2401.00125*, 2023.

[37] Q. Xie, T. Zhang, K. Xu, M. Johnson-Roberson, and Y. Bisk, "Reasoning about the unseen for efficient outdoor object navigation," 2023.

[38] D. Shah, M. R. Equi, B. Osiński, F. Xia, B. Ichter, and S. Levine, "Navigation with large language models: Semantic guesswork as a heuristic for planning," in *Proceedings of The 7th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Tan, M. Toussaint, and K. Darvish, Eds., vol. 229. PMLR, 06–09 Nov 2023, pp. 2683–2699. [Online]. Available: https://proceedings.mlr.press/v229/shah23c.html

[39] D. Shah, B. Osiński, b. ichter, and S. Levine, "Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action," in *Proceedings of The 6th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, K. Liu, D. Kulic, and J. Ichnowski, Eds., vol. 205. PMLR, 14–18 Dec 2023, pp. 492–504. [Online]. Available: https://proceedings.mlr.press/v205/shah23b.html

[40] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 10 608–10 615.

[41] R. Sinha, A. Elhafsi, C. Agia, M. Foutter, E. Schmerling, and M. Pavone, "Real-time anomaly detection and reactive planning with large language models," in *Robotics: Science and Systems*, 2024.

[42] A. Tagliabue, K. Kondo, T. Zhao, M. Peterson, C. T. Tewari, and J. P. How, "Real: Resilience and adaptation using large

language models on autonomous aerial robots," *Conference on Robot Learning (CoRL)*, 2023.

[43] OpenAI et al., "Gpt-4 technical report," 2024. [Online]. Available: https://arxiv.org/abs/2303.08774

[44] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar, P. Sermanet, N. Brown, T. Jackson, L. Luu, S. Levine, K. Hausman, and B. Ichter, "Inner monologue: Embodied reasoning through planning with language models," in *arXiv preprint arXiv:2207.05608*, 2022.

[45] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 9493–9500.

[46] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar, "Eureka: Human-level reward design via coding large language models," *arXiv preprint arXiv: Arxiv-2310.12931*, 2023.

[47] Y. J. Ma, W. Liang, H. Wang, S. Wang, Y. Zhu, L. Fan, O. Bastani, and D. Jayaraman, "Dreureka: Language model guided sim-to-real transfer," in *Robotics: Science and Systems (RSS)*, 2024.

[48] Z. Dai, A. Asgharivaskasi, T. Duong, S. Lin, M.-E. Tzes, G. Pappas, and N. Atanasov, "Optimal scene graph planning with large language model guidance," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 14 062–14 069.

[49] J. X. Liu, Z. Yang, I. Idrees, S. Liang, B. Schornstein, S. Tellex, and Shah, "Grounding complex natural language commands for temporal tasks in unseen environments," in *Proceedings of The 7th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Tan, M. Toussaint, and K. Darvish, Eds., vol. 229. PMLR, 06–09 Nov 2023, pp. 1084–1110. [Online]. Available: https://proceedings.mlr.press/v229/liu23d.html

[50] Y. Chen, R. Gandhi, Y. Zhang, and C. Fan, "Nl2tl: Transforming natural languages to temporal logics using large language models," *arXiv preprint arXiv:2305.07766*, 2023.

[51] Y. Chen, J. Arkin, C. Dawson, Y. Zhang, N. Roy, and C. Fan, "Autotamp: Autoregressive task and motion planning with llms as translators and checkers," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 6695–6702.

[52] K. Garg, J. Arkin, S. Zhang, N. Roy, and C. Fan, "Large language models to the rescue: Deadlock resolution in multi-robot systems," 2024. [Online]. Available: https://arxiv.org/abs/2404.06413

[53] B. Liu, Y. Jiang, X. Zhang, Q. Liu, S. Zhang, J. Biswas, and P. Stone, "Llm+p: Empowering large language models with optimal planning proficiency," *arXiv preprint arXiv:2304.11477*, 2023.

[54] Y. Chen, J. Arkin, Y. Zhang, N. Roy, and C. Fan, "Scalable multi-robot collaboration with large language models: Centralized or decentralized systems?" in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 4311–4317.

[55] S. S. Kannan, V. L. N. Venkatesh, and B.-C. Min, "Smart-llm: Smart multi-agent robot task planning using large language models," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 12 140–12 147.

[56] Z. Mandi, S. Jain, and S. Song, "Roco: Dialectic multi-robot collaboration with large language models," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 286–299.

[57] J. Wang, G. He, and Y. Kantaros, "Safe task planning for language-instructed multi-robot systems using conformal prediction," *arXiv preprint arXiv:2402.15368*, 2024.

[58] K. Liu, Z. Tang, D. Wang, Z. Wang, B. Zhao, and X. Li, "Coherent: Collaboration of heterogeneous multi-robot system with large language models," *arXiv preprint arXiv:2409.15146*, 2024.

[59] B. Quartey, E. Rosen, S. Tellex, and G. Konidaris, "Verifiably following complex robot instructions with foundation models," *arXiv*, vol. abs/2402.11498, 2024.

[60] B. Chen, F. Xia, B. Ichter, K. Rao, K. Gopalakrishnan, M. S. Ryoo, A. Stone, and D. Kappler, "Open-vocabulary queryable scene representations for real world planning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 11 509–11 522.

[61] M. A. Hsieh, A. Cowley, V. Kumar, and C. J. Taylor, "Maintaining network connectivity and performance in robot teams," *Journal of field robotics*, vol. 25, no. 1-2, pp. 111–131, 2008.

[62] E. Stump, N. Michael, V. Kumar, and V. Isler, "Visibility-based deployment of robot formations for communication maintenance," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 4498–4505.

[63] N. Roy and G. Dudek, "Collaborative robot exploration and rendezvous: Algorithms, performance bounds and observations," *Autonomous Robots*, vol. 11, pp. 117–136, 2001.

[64] X. Yu and M. A. Hsieh, "Synthesis of a time-varying communication network by robot teams with information propagation guarantees," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1413–1420, 2020.

[65] M. Saboia, L. Clark, V. Thangavelu, J. A. Edlund, K. Otsu, G. J. Correa, V. S. Varadharajan, A. Santamaria-Navarro, T. Touma, A. Bouman, H. Melikyan, T. Pailevanian, S.-K. Kim, A. Archanian, T. S. Vaquero, G. Beltrame, N. Napp, G. Pessin, and A.-a. Agha-mohammadi, "ACHORD: Communication-Aware Multi-Robot Coordination With Intermittent Connectivity," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 184–10 191, 2022.

[66] M. F. Ginting, K. Otsu, J. A. Edlund, J. L. Gao, and A. akbar Agha-mohammadi, "CHORD: Distributed Data-Sharing via Hybrid ROS 1 and 2 for Multi-Robot Exploration of Large-Scale Complex Environments," *IEEE Robotics and Automation Letters*, vol. 6, pp. 5064–5071, 2021.

[67] K. Birman, "The promise, and limitations, of gossip protocols," *ACM SIGOPS Operating Systems Review*, vol. 41, no. 5, pp. 8–13, 2007.

[68] M. Novitzky, R. Semmens, and P. Robinette, "Toward measures of human-robot teaming effectiveness," *Transactions on Human-Robot Interaction*, 2021.

[69] T. Kaupp and A. Makarenko, "Measuring human-robot team effectiveness to determine an appropriate autonomy level," in *2008 IEEE International Conference on Robotics and Automation*, 2008, pp. 2146–2151.

[70] L. M. Ma, M. Ijtsma, K. M. Feigh, and A. R. Pritchett, "Metrics for human-robot team design: A teamwork perspective on evaluation of human-robot teams," *J. Hum.-Robot Interact.*, vol. 11, no. 3, Sep. 2022. [Online]. Available: https://doi.org/10.1145/3522581

[71] M. Novitzky, P. Robinette, M. R. Benjamin, C. Fitzgerald, and H. Schmidt, "Aquaticus: Publicly available datasets from a marine human-robot teaming testbed," in *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2019, pp. 392–400.

[72] C. Reardon and J. Fink, "Air-ground robot team surveillance of complex 3d environments," in *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2016, pp. 320–327.

[73] I. D. Miller, F. Cladera, A. Cowley, S. S. Shivakumar, E. S. Lee, L. Jarin-Lipschitz, A. Bhat, N. Rodrigues, A. Zhou, A. Cohen, A. Kulkarni, J. Laney, C. J. Taylor, and V. Kumar, "Mine tunnel exploration using multiple quadrupedal robots," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2840–2847, 2020.

[74] J. Hughes, D. Kim, S. Henderson, and P. Manjunath, "Towards collaborative aerial swarming architecture for military applications," in *2023 23rd International Conference on Control, Automation and Systems (ICCAS)*, 2023, pp. 1049–1054.

[75] D. Larkin, M. Novitzky, J. Kim, and C. M. Korpela, "Atak integration through ros for autonomous air-ground team," in *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2021, pp. 1116–1122.

[76] D. Kim and P. Y. Oh, "Aerial manipulation using a human-embodied drone interface," in *2022 IEEE International Conference on Advanced Robotics and Its Social Impacts (ARSO)*, 2022, pp. 1–7.

[77] K. N. Lavanya, D. R. Shree, B. R. Nischitha, T. Asha, and C. Gururaj, "Gesture controlled robot," in *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*, 2017, pp. 465–469.

[78] "Breadcrumb Wireless Network Nodes | Rajant Networks — rajant.com," https://rajant.com/products/breadcrumb-wireless-nodes/, [Accessed 29-11-2024].

[79] Mapbox, "Mapbox," 2025, accessed: 2025-04-27. [Online]. Available: https://www.mapbox.com/

[80] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," in *Computer Vision – ECCV 2024*, A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, Eds. Cham: Springer Nature Switzerland, 2025, pp. 38–55.

[81] F. Cladera, K. Chaney, M. A. Hsieh, C. J. Taylor, and V. Kumar, "Evmapper: High altitude orthomapping with event cameras," *arXiv preprint arXiv:2409.18120*, 2024.

[82] F. Dellaert, "Factor graphs and gtsam: A hands-on introduction," *Georgia Institute of Technology, Tech. Rep*, vol. 2, no. 4, 2012.

[83] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang, "Grounded sam: Assembling open-world models for diverse visual tasks," 2024. [Online]. Available: https://arxiv.org/abs/2401.14159

[84] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer, "Sam 2: Segment anything in images and videos," 2024. [Online]. Available: https://arxiv.org/abs/2408.00714

[85] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[86] Z. Chen, G. Fang, X. Ma, and X. Wang, "Slimsam: 0.1% data makes segment anything slim," in *Proceedings of the 38th International Conference on Neural Information Processing Systems*, ser. NIPS '24. Red Hook, NY, USA: Curran Associates Inc., 2025.

[87] C. Bai, T. Xiao, Y. Chen, H. Wang, F. Zhang, and X. Gao, "Faster-lio: Lightweight tightly coupled lidar-inertial odometry using parallel sparse incremental voxels," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4861–4868, 2022.

[88] N. Steinke, D. Goehring, and R. Rojas, "Groundgrid: Lidar point cloud ground segmentation and terrain estimation," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 420–426, 2024.

[89] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *NeurIPS*, 2023.

[90] J. Wei, X. Wang, D. Schuurmans, M. Bosma, b. ichter, F. Xia, E. Chi, Q. V. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 24 824–24 837. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf

[91] X. Liu, G. V. Nardari, F. Cladera, Y. Tao, A. Zhou, T. Donnelly, C. Qu, S. W. Chen, R. A. F. Romero, C. J. Taylor, and V. Kumar, "Large-scale autonomous flight with real-time semantic slam under dense forest canopy," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5512–5519, 2022.

[92] L. Meier, D. Honegger, and M. Pollefeys, "Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 6235–6240.

[93] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2016.

[94] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping using the bayes tree," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

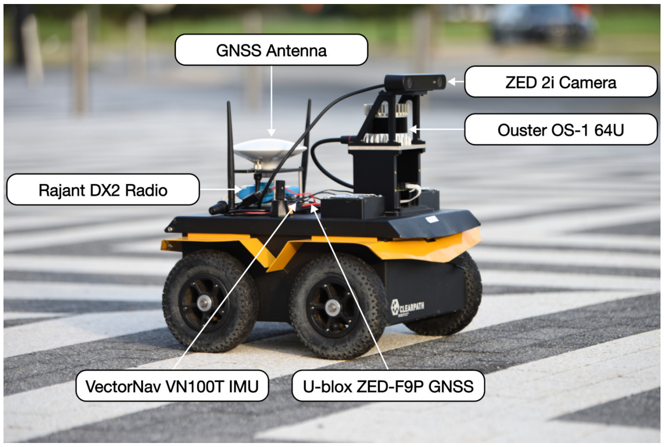

FIGURE A.1. Jackal UGV platform used for ground experiments.

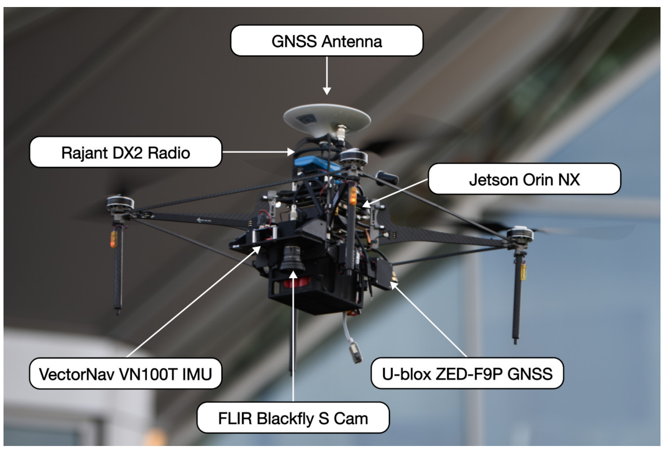

FIGURE A.2. Falcon 4 platform configured for high-altitude semantic mapping missions.

APPENDIX

A.1. Platforms

We use a custom-built quadcopter and Clearpath Jackals for the experiments in this work. Figs. A.2 and A.1 show these robots. Both platforms are outfitted with compute capabilities to perform semantic mapping onboard and run their respective autonomy algorithms.

One of the major limitations of these platforms is the power and weight budgets for compute. For example, the power budget allowed for the Jackal is 200 W, and the UAV power budget is 100 W. Moreover, the UAV payload is limited to 1.2 kg, including perception and compute. Both the UGV and UAV have a battery life of approximately 30 minutes.

Both platforms were fitted with Rajant mesh radios. We also used *dummy* nodes (Fig. A.4) as communication relays for SPINE, as described in Sec. III.D.

A.2. UAV State Machine

We relied on `air_router` [10] to task the UAV during the experiments. The high-level state machine, illustrated in Fig. A.3, transitions the UAV between an exploration and a communication mission. It also looks for ground robots based on the last known position or last known goal, to deliver messages if communication has not occurred recently.

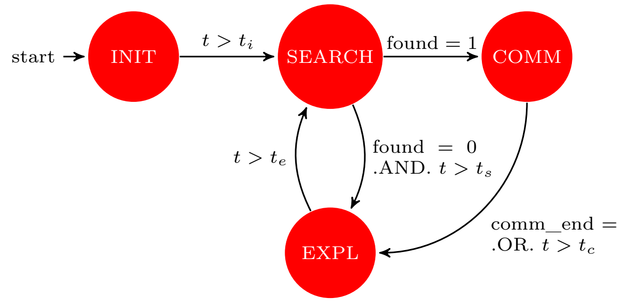

FIGURE A.3. UAV state machine executed during the air-ground experiments. Figure from [10].

FIGURE A.4. Communication nodes used for field experiments. Left: FE1 base station node. Right: *dummy* ME4 relay node used for LLM API calls.

A.3. UAV Localization

In order to localize objects within images we need an accurate pose estimate for when each image was taken. To this end, we use GTSAM [82] to fuse GNSS position estimates with IMU readings. The VectorNav VN-100T runs its own Kalman Filter to fuse magnetometer with gyroscope and accelerometer readings. This gives us north aligned orientation data at 400 Hz. We get 5 Hz global position readings from the GNSS which we model as unary pose factors [82] and use the preintegrated IMU measurements [93] to constrain the pose graph. As the UAV can cover several kilometers, we use the iSAM2 [94] optimizer to maintain a reasonable latency over large distances. We run the optimizer at each GNSS measurement. We filter erroneous GNSS measurements by ensuring that they lie within 2σ of the estimated pose and covariance. We also interpolate between poses at 100 Hz by integrating the IMU measurements from the last optimized pose to the current time. This allows us to get accurate pose estimates for each image. We note that while we use this particular set of sensors in this work, the framework could be extended to include other sensors such as barometers (for height) and raw magnetometer readings.

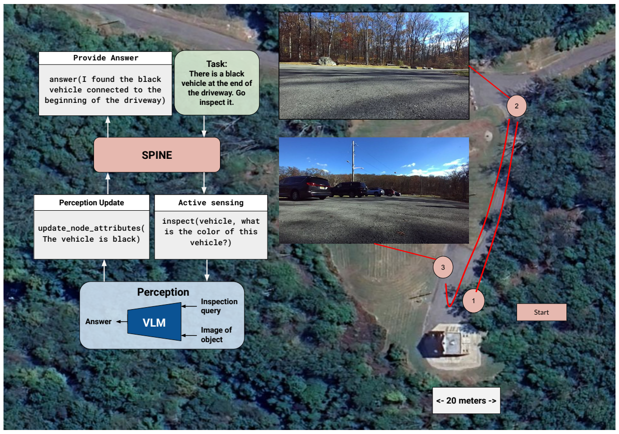

FIGURE A.5. Example of SPINE using active perception to correct mission misspecifications. User tasks UGV with inspecting a black vehicle at the end of the driveway (1). However, there is no ground vehicle at the end of the driveway (2). The UGV identifies several cars and trucks at the start of driveway, and inspects those (3). The ground vehicle forms a mission-relevant perception query "is this vehicle black" in order to resolve the mission.

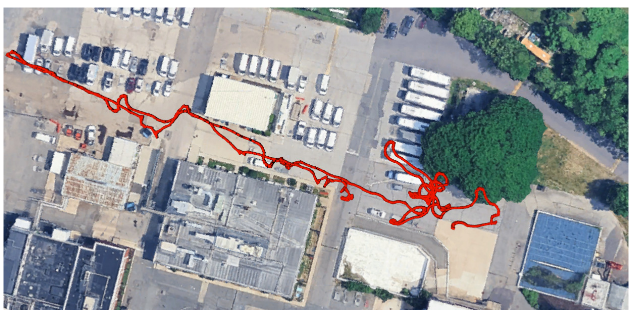

FIGURE A.6. UGV trajectory from the first system demonstration. Using the UAV-generated graph, the UGV travels over 700 m while finding objects and describing regions of the environment.

A.4. Active Perception

SPINE uses active perception to resolve errors in the mission specification. Fig. A.5 shows a mission where the UGV is provided the task: "There is a black vehicle at the end of the driveway. Go inspect it." The UGV goes to the end of the driveway but does not find a vehicle. During navigation, it observes the vehicle at the beginning of the driveway. So, it doubles back to observe those, and queries perception to find the color of the vehicle.

A.5. Large-scale Demonstrations

We plot the UGV path from the first system demonstration, as reported in Sec VI, in Fig. A.6. Please note that the trajectory is overlaid on a Google Earth image, so the vehicles displayed may not be present in the actual experiment.