2025 IEEE International Conference on Robotics and Automation (ICRA)
May 19-23, 2025. Atlanta, USA

SPINE: Online Semantic Planning for Missions with Incomplete Natural Language Specifications in Unstructured Environments

Zachary Ravichandran, Varun Murali, Mariliza Tzes, George J. Pappas, and Vijay Kumar

***Abstract*— As robots become increasingly capable, users will want to describe high-level missions and have robots infer the relevant details. Because pre-built maps are difficult to obtain in many realistic settings, accomplishing such missions will require the robot to map and plan online. While many semantic planning methods operate online, they are typically designed for well specified missions such as object search or exploration. Recently, Large Language Models (LLMs) have demonstrated powerful contextual reasoning abilities over a range of robotic tasks described in natural language. However, existing LLM-enabled planners typically do not consider online planning or complex missions; rather, relevant subtasks and semantics are provided by a pre-built map or a user. We address these limitations via SPINE, an online planner for missions with incomplete mission specifications provided in natural language. The planner uses an LLM to reason about subtasks implied by the mission specification and then realizes these subtasks in a receding horizon framework. Tasks are automatically validated for safety and refined online with new map observations. We evaluate SPINE in simulation and real-world settings with missions that require multiple steps of semantic reasoning and exploration in cluttered outdoor environments of over 20,000m^2. Compared to baselines that use existing LLM-enabled planning approaches, our method is over twice as efficient in terms of time and distance, requires less user interactions, and does not require a full map. Additional resources are provided at https://zacravichandran.github.io/SPINE.**

I. Introduction

Consider an inspection robot operating after a heavy storm. A user may provide the following mission specification: *Communications are down. Why?* The robot will have to explore missing or changed regions of the map, locate relevant semantic entities (*e.g.,* communication infrastructure), and collect precise mission-relevant information to assess infrastructure damage. We refer to these mission specifications as *incomplete* because they imply subtasks and semantic targets that are not directly given to the robot; rather, they must be inferred from context. Because such missions often occur in partially-known environments, an autonomous robot must actively map its surroundings and refine its plan online.

Semantic planning methods have made progress on tasks such as object search, inspection, exploration, and mobile manipulation [1]–[9]. These methods typically maintain a semantic map of the environment such as a metric-semantic grid, object-oriented map, or scene graph, which the planner reasons over in search of its goal [4], [7], [10]. With advances in semantic mapping, these representations can be built in real-time, which enables online planning [11]–[14], and some of these approaches are robust enough to be fielded in large scale environments [15]–[17]. However, online semantic planners typically require a well-specified mission (*e.g.*, *inspect all the antennas in Zone A*). And while formal planning languages enable complex mission-level specifications [18]–[20], they still require a human operator to explicitly compose subtasks.

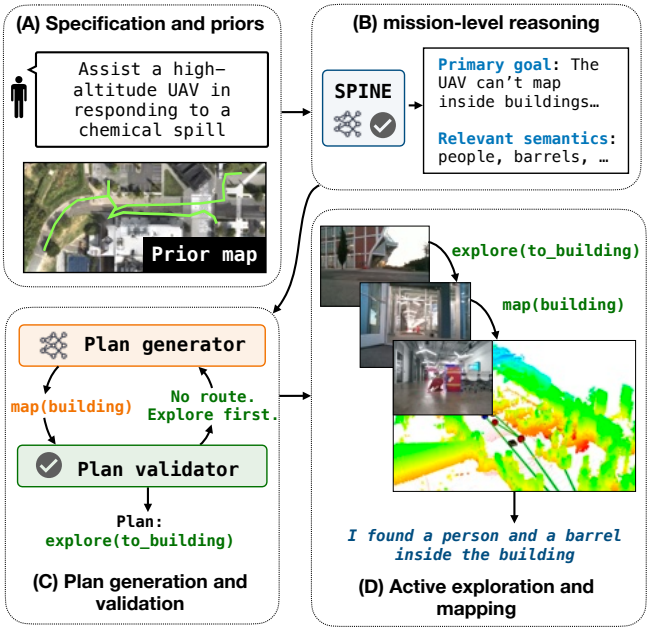


Fig. 1. (A) SPINE takes as input a mission with incomplete specifications and prior map. (B) SPINE then reasons about the goals and semantics required to achieve the mission. (C) SPINE's plan generator uses an LLM to generate subtasks, while its validation module ensures those subtasks are realizable. (D) SPINE then actively explores and reasons over acquired information in order to complete its mission.

Recent work has addressed these limitations by using Large Language Models (LLMs) – which have demonstrated powerful contextual reasoning over many domains – to plan over tasks described in natural language [21]–[24]. Researchers have applied LLM-enabled planners to problems including mobile manipulation, navigation, and fault detection [23], [25]–[30]. However, LLM-enabled planners typically require a pre-built map [25], [26], [29], [31], [32], and these methods generally consider well-specified missions [23], [29], [33]–[35]. These assumptions prevent current LLM-enabled planning methods from operating in partially-known and unstructured environments, such as large-scale outdoor settings.

All authors are with are with the GRASP Laboratory, University of Pennsylvania. Corresponding author: zacravi@seas.upenn.edu.
We acknowledge support from ARL DCIST CRA W911NF- 17-2-0181, NSF Grant CCR-2112665, and the NSF Graduate Research Fellowship.

2025 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3315-4139-2/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICRA55743.2025.11128238

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:06:42 UTC from IEEE Xplore. Restrictions apply.

To address these limitations, we present SPINE, an online semantic planner for missions with incomplete specifications described in natural language. Given a specification, SPINE uses an LLM to infer subtasks comprising navigation, active mapping, and user interaction. These LLM-generated subtasks are validated for syntactic and spatial correctness, which prevents unsafe actions from being executed by the robot. Validated subtasks are realized in a receding horizon manner and are refined online as SPINE actively builds a semantic map. By leveraging external data sources such as satellite imagery or UAV-generated maps, SPINE can operate in partially-known and unstructured environments. To summarize, the contributions of the paper are:

1. An online semantic planner for language-specified missions in partially-known, unstructured environments.
2. A plan generation module to infer subtasks from incomplete specifications and refine the subtasks online.
3. A verification module that enables an LLM to safely propose navigation and exploration goals in unstructured and partially-known environments.

We evaluate our method in large-scale outdoor simulated and real-world environments with missions that include semantic route inspection, multi-object search, and air-ground teaming. Compared to LLM-enabled planning baselines that use either fully-known prior maps or receive explicit mission specifications, SPINE achieves comparable mission success while requiring significantly less time and user input.

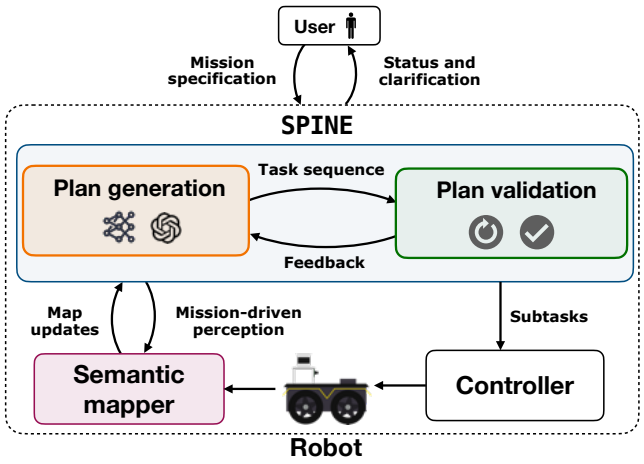


Fig. 2. SPINE architecture. A user provides SPINE with a mission specification. SPINE plans via behaviors for user interaction, active mapping, and robot control. SPINE's plan generator infers a task sequence which is validated online for correctness and feasibility; if necessary, feedback and corrections are provided in real-time. Actions are sent to the appropriate module, and the planner refines its plan as new information is acquired.

II. Related Work

Representations for Semantic Planning. Effective planning representations capture traversability, semantics, and spatial relationships needed for reasoning over contextual goals. Advances in semantic mapping have enabled online planning for tasks such as active exploration or object search [1], [4], [15], [17], [36]. Scene graphs are a common representation for semantic planning, as they concisely represent objects, topology, and traversable regions [11], [14]. Semantic topological maps also provide a graphical scene representation but do not include a hierarchy [37], [38]. Recent work incorporates foundation models into mapping pipelines in order to create open-vocabulary representations. For example, ConceptGraphs [39], HOVSG [40], and Clio [41] assign semantic feature vectors to entities in the map, then task-relevant labels are assigned at runtime. SPINE is compatible with such state of the art mapping methods. In our experiments, we use an open-vocabulary semantic-topological mapper, which allows the planner to configure mission-specific semantics at runtime and operate in unstructured outdoor environments.

Online Semantic Planning. Semantic planners reason over objects, regions, or other contextual information to solve tasks such as object search, inspection, and semantic exploration [1], [3], [5], [6], [42]. Many works address online planning in partially-known environments, the planner's initial action sequence is informed by priors and refined online with new observations [3], [3], [7]. Beyond object-level reasoning, semantic information also accelerates exploration of partially-known or unknown environments [1], [43], [44]. Fusing semantic knowledge from foundation models with classical search methods such as frontier exploration has been shown to an especially effective exploration strategy [8], [43]. Structured or formal planning languages, such as Linear Temporal Logic (LTL), may be used to compose more complex missions [10], [19], [20], [45], [46]. Notably, these methods require detailed mission specifications from a user, whereas our method infers mission details.

LLMs for Planning. Language has emerged as a powerful representation for specifying tasks, and LLM-enabled planners have been applied to domains including mobile manipulation [25], [26], [47], service robotics [48], autonomous driving [49], navigation [29], [50]–[52], and fault detection [27], [28] These methods typically configure an LLM via in-context or system prompts with a problem description and a set of action primitives such as graph navigation goals [49], lower-level APIs for code generation [23], [53], [54], or learned behaviors [25]. At runtime, the LLM is given an task specification and a map, such as a graph [26] or semantic regions [30], then produces an action sequence. A line of research develops LLM-enabled planners that translate mission specifications to a formal language such as Linear Temporal Logic (LTL) or Planning Domain Definition Language (PDDL) [31]–[33], [35], [55]–[57]. While these instructions are complex, they explicitly state subtasks and semantic referents [34]. Other research relaxes the requirement of a pre-built semantic map by incorporating feedback from perception systems [27], [30], [47] or specifying semantics at runtime [24], [34]. 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. In contrast, SPINE reasons over under-specified missions, does not require a pre-built map, and can operate in large

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:06:42 UTC from IEEE Xplore. Restrictions apply.

unstructured environments.

III. Online Semantic Planning Approach

A. Problem Statement

We consider a robot that operates in an unstructured environment and is capable of performing behaviors such point navigation or area exploration. The robot's planner is provided with a mission specification in natural language, S. Importantly, this specification is *incomplete*, meaning it implies a goal and corresponding action sequences (*i.e.,* subtasks) which are *unknown* to the planner, thus planner must infer an action sequence that fulfills that goal with minimal planning iterations. The planner is also provided with a map, M_k which is updated at each planning iteration, k, by an onboard mapper, and its previous actions, $a_{1:k}$. At each iteration, the planner provides an action sequence, $\pi(S, M_k, a_{1:k}) \rightarrow a_{k+1:H}$, where H denotes a planning horizon greater than k. This plan is realized in a receding horizon manner and refined online.

B. SPINE Overview

SPINE comprises two modules – a *plan generator* and *plan validator* – as outlined in Fig. 2. Along with the mission specification, online map, and action history, the plan generator receives online feedback, f, and proposes candidate actions $\pi_g(S, M_k, a_{1:k}, f) \rightarrow a'_{k+1:H'}$ (§III-D). The plan validator ensures that these actions are syntactically correct and physically realizable given the current map, $\pi_v(a'_{k+1:H}, M_k) \rightarrow (f, a_{k+1:H'})$, where and $a_{k+1:H}$ is a validated action sequence and f denotes feedback that may be used by the generator to correct erroneous action sequence (§III-E). We outline SPINE's inference process in Alg. 1.

C. Semantic Mapper

Our architecture assumes a topological graph-based semantic mapper, where nodes are of type *region* or *object*. Regions indicate traversable points in freespace, and objects represent semantic entities. Edges in the graph are defined between either two regions ("region edges") or an object and a region ("object edges"). Regions edges denote paths traversable by the robot, while object edges denote that an object is visible from a certain region. Nodes may be enriched with additional semantic information (*e.g.*,"this region is in a busy parking lot", "this car is damaged"), which provides SPINE with additional cues for planning. The mapper also maintains a local occupancy map which SPINE uses for action validation (§III-E). The mapper is initialized with priors from satellite imagery, UAV maps, or previous mission data and is updated at each iteration.

D. Plan Generator

The plan generator must infer action sequences that best fulfill the incomplete mission specification, which may require exploring previously unknown portions of the environment. We instantiate the plan generator with an LLM, given their contextual reasoning abilities. We configure the LLM via a system prompt that describes its role, a definition of the mapping interface, and a description of the robot's behaviors. At each iteration, the plan generator's four inputs – the specification S, map M_k, action sequence $a_{1:k}$, and feedback f – are serialized into a textual representation and provided to the LLM via in-context prompts.

Textual mapping interface. The plan generator receive the prior semantic map in the following JSON schema:

```
{"regions": [{"name": "node_name",
              "coordinates": "..."}, "..."],
"objects": ["..."],
"region_edges": [["source", "target"], "..."],
"object_edges": ["..."]}
```

Nodes are defined as a dictionary of attributes; this dictionary must contain the node's `name` and `coordinates`, but may be enriched with additional information such as a response to a mission-specify query (example in Fig. 5). Edges are simply defined by tuples of source and target nodes. At each planning iteration, all map updates are provided to the plan generator via in-context prompts which utilize the following API for high-level graph manipulation: `add_nodes`, `remove_nodes`, `add_edges`, `remove_edges`, `update_nodes`.

Reasoning for planning. The plan generator provides an action sequence drawn from a predefined library of atomic behaviors (Alg. 1 line 2). Our implementation uses behaviors for navigation, active perception, and user interaction (Tab. I), though in general these may include any feasible robot action. The generator parameterizes these behaviors with arguments that refer to the current semantic map or mission specification, and this process employs chain-of-thought (CoT) reasoning which explicitly states SPINE's *primary goal*, the *relevant semantic graph* for the mission, and a *justification* for the proposed action sequence [58]. During the first planning iteration, generator also provides *relevant semantics* which may be used used to configure an open-vocabulary semantic mapping framework. We find that enforcing reasoning at multiple levels of abstraction helps the LLM to maintain a focus on its high-level goal while iteratively planning over shorter horizons. We illustrate one step of this process with a simplified example:

Example III.1 Consider a robot provided with the mission specification *I need to cross the river.* and a semantic map containing two regions – `region_1` located at `(0, 0)` and `dock` located at `(0, 1)` – and one edge that connects these two regions. There is also a `boat` located at `(0, 2)` and connected to the `dock`, but that boat is *initially unknown*. SPINE would receive the following map representation:

```
{"regions": [
  {"name": "region_1", "coordinates": [0, 0]}
  {"name": "dock", "coordinates": [0, 1]}],
"objects": [],
"region_edges": [["region_1", "dock"]],
"object_edges": []}
```

Because a boat would enable the user to cross the river, and

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:06:42 UTC from IEEE Xplore. Restrictions apply.

Purpose	Function	Arguments	Behavior	Constraints
Navigation	`map_region`	region node	navigate to goal and find objects	syntax, reachable
	`exlore_region`	goal region, exploration radius	explore around goal	syntax, reachable, explorable
	`extend_map`	2D coordinate	add frontier at coordinate	syntax, explorable
	`goto`	region	navigate to region	syntax, reachable
Active Mapping	`inspect`	object and query	Inspect object	syntax
	`set_labels`	list of labels	Configure object detection	syntax
User interaction	`clarify`	question	ask for clarification from user	syntax
	`answer`	provides answer	denotes task is complete	syntax

TABLE I

AVAILABLE BEHAVIORS USED BY THE SEMANTIC PLANNER TO COMPOSE ACTION SEQUENCES.

boats are often found near docks, a reasonable output from the plan generator would be as follows:

```
{"primary goal": "find the user something
   with which to cross the river",
"relevant semantics": ["boat"],
"relevant graph": "dock_1"
"justification": "boats are often found near
   docks, so I should map that region",
"plan": "map_regin(dock_1), replan()"}
```

The robot's semantic mapper would be configured to look for `boats`, and the robot would then map the `dock`, upon which the semantic mapper would discover the boat and provide `SPINE` with the following update

```
add_node(boat,
 attributes={"coordinates": [0, 2]},
 aedges=["dock"])
```

`SPINE` would then provide the user with the answer "I found a boat near the dock that you could use to cross the river."

E. Plan validation

Valid action sequences require that all actions correctly invoke the behavior library while respecting constraints such as traversability; while LLMs provide powerful contextual reasoning abilities, they may hallucinate these details. The plan validation module ensures that all actions satisfy predefined constraints before being realized by the robot (Alg. 1 line 3). Any feedback provided by the validator is appended to the mission history (line 4), and `SPINE` returns its first valid plan sequence (line 5-6). If the plan generator exceeds a predefined iteration limit (MAX_ATTEMPTS), the user is notified and invited to re-task `SPINE` (lines 7-8).

The validation module defines three types of constraints: *syntax*, *reachability*, and *explorable*. Syntax constraints require that behaviors are invoked with the correct argument type. Reachability constraints apply to navigation behaviors, and they require that a path to the navigation target exist *within the current map*. These first two constraints are enforced upfront (lines 9-15); if violated, the validation module will provide specific feedback about the offending action, and the generator will produce a new task sequence. *Explorable* constraints are assigned to behaviors where the robot discovers new regions in the map. We implement this constraint via frontier-style exploration to iteratively search for a traversable path towards a given goal. The exploration terminates after reaching the goal or encountering an obstacle. For each breaking condition, semantic feedback is provided to the planner such as `exploration terminated after encountering an obstacle` (lines 16-21, see Fig. 3).

Algorithm 1: SPINE inference procedure

Input: SPECIFICATION S, MAP M_k, ACTION HISTORY $a_{1:k}$ FEEDBACK f

1 **for** ATTEMPT **in** MAX_ATTEMPTS **do**
2 $\quad a'_{k+1:H'} \leftarrow \pi_g(S, M_k, a_{1:k}, f)$ ▷ generate actions
3 $\quad (a_{k+1:H}, f_k) \leftarrow \pi_v(a'_{k+1:H'}, M_k)$ ▷ validate actions
4 $\quad f \leftarrow f + [f_k]$
5 $\quad$ **if** $a_{k+1:H} \neq \emptyset$ **then**
6 $\quad\quad$ **return** $a_{k+1:H}$ ▷ return first valid sequence
7 **return** `NotifyUser` ▷ if unable to create plan
8 **function** $\pi_v(a'_{k+1:H'}, M_k)$
9 $\quad$ VALIDATED ACTIONS $a_k = []$
10 $\quad$ GENERATOR FEEDBACK $f = []$
11 $\quad$ **for** a_i in a'_k **do**
12 $\quad\quad$ **if not** SYNTACTIALLYVALID(a_i, M_k) **then**
13 $\quad\quad\quad$ $f \leftarrow$ GETERRORFEEDBACK(a_i, M_k)
14 $\quad\quad\quad$ **return** a_k, f
15 $\quad\quad$ **for** a_i in $a'_{k+1:H'}$ **do**
16 $\quad\quad\quad$ **if *not*** EXPLORABLE(a_i) **then**
17 $\quad\quad\quad\quad$ $a_k \rightarrow a_k + [a_i]$
18 $\quad\quad\quad$ a_i, result $\leftarrow$ EXPLORE(a_i, M_k)
19 $\quad\quad\quad$ $a_k \leftarrow a_k + [a_i]$
20 $\quad\quad\quad$ $f \leftarrow f +$ [result]
21 $\quad\quad$ **return** a_k, f

IV. EXPERIMENTS

We design experiments to assess our contributions (§I):

Q1. Does `SPINE` provide time and distance savings compared to offline LLM-enabled planning approaches?

Q2. Can `SPINE` achieve missions competitively compared to methods that are explicitly given a full prior map and mission specifications?

Q3. How important is validation for online planning?

We use simulation and real robot experiments to answer **Q1** and **Q2**, and we design an ablation study to answer **Q3**.

A. Implementation Details

Both simulated and real robot experiments assume a mobile robot equipped with a Lidar and RGB-D camera.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:06:42 UTC from IEEE Xplore. Restrictions apply.

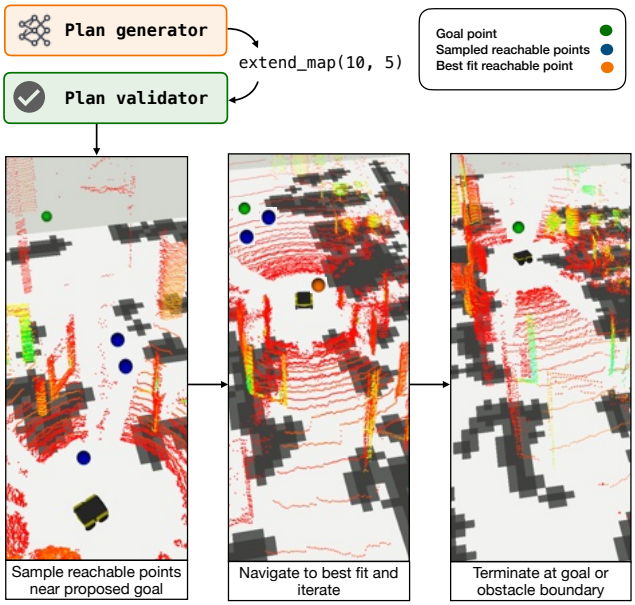


Fig. 3. Online validation enables exploration. SPINE's plan generator may produce exploration goal outside robot's obstacle map, which may not be reachable. Spatial validation iteratively finds best reachable fit, and the robot navigates to that point. Procedure terminates once robot reaches its goal

We implement the behaviors from §III-D using ROS Move-Base [59], and the plan generator uses the base GPT-4 model [60]. We implement a semantic mapper that provides the graph-base representation described in §III-C. The mapper uses Faster-LIO to estimate odometry [61] and GroundGrid [62] to estimate free-space and establish region nodes. GroundingDino [62] provides open-vocabulary object detections, and these detections are associated using a multiple-hypothesis tracker. The mapper uses the LLaVA vision-language model to enrich the map with mission-relevant semantic descriptions [63]. All autonomy runs on-board, except for the LLM which makes API queries over an internet connection. We conduct simulation experiments in photorealistic Unity testbed, which provides sensor and control feeds for a ClearPath Husky. We then perform real robot experiments using a Clearpath Jackal equipped with a Ouster Lidar, Realsense RGB-D Camera, Nvidia RTX 4000 GPU, and Ryzen 5 3600 CPU.

B. Experimental Setup

Baselines. We compare against two baselines: *Explicit Tasking* and *LLM-as-planner*. *Explicit Tasking* receives step-by-step instructions and resembles existing LLM-enabled planning methods where the user provides explicit mission instructions, such as those using formal methods, [33], [55]. *LLM-as-planner* resembles approaches which receive a full map upfront along with a mission specification [23], [25], [26], [34]; following previous work, this baseline can still discover new objects in the scene. [25], [30].

Metrics. We report five metrics: *mission success*, *time* required for mission completion, *distance* traveled for mission completion, LLM *queries*, and user *interactions* during the mission. User interactions capture the complexity of the mission, as more complex missions will require more instructions from the user.

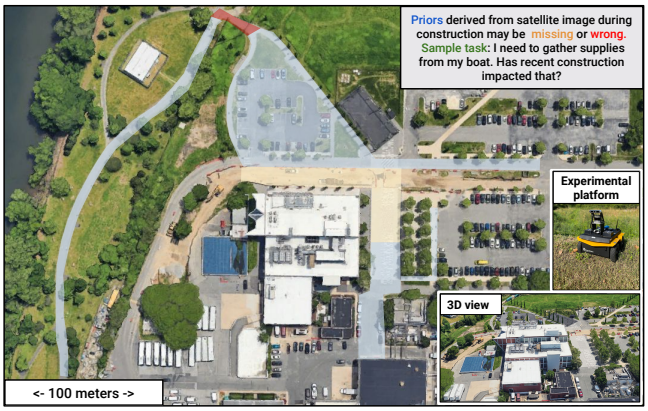


Fig. 4. Experimental platform, 3D view of environment, and example prior and corresponding task used for real-world experiments. The prior map is derived from outdated satellite imagery or obstructed due to trees and other coverings. The prior map is thus incomplete and partially incorrect, which requires the planner to reason about information acquired online.

Evaluation Environments. We perform simulation experiments in a rural outdoor space of over 40,000m^2 where missions require the robot to travel over 400m (example in Fig 5). Real experiments are conducted in an office park of over 20,000m^2 where missions require the robot to travel over 250m (Example in Fig 4). We construct prior maps for both environments based on information that could be acquired from satellite imagery or UAV-generated maps.

Mission Specifications. We consider missions with the following linguistic specifications:

1. There was a storm last night. I am worried that impacted logistics, because I need to drop off supplies today. Can I still do that?
2. I sent a robot out to collect supplies from an incoming boat. I have not heard back. What happened?
3. Communications are down, why?
4. I need to gather supplies from my boat. Has recent construction impacted that?
5. You're assisting a UAV in response to a chemical spill. Triage regions not visible from the air.

Each mission requires completing 2-8 subtasks of semantic reasoning and exploration. We run each mission one to three times and vary the prior map and initial conditions. See Fig. 1, Fig. 4, and Fig. 5 for examples.

C. Simulation Results

As reported in Tab. II, Explicit Tasking completes 100% of missions while taking on average 532s, traveling 292m, making 8.6 API calls, and 4 user interacting. Despite receiving only a partial map and incomplete specification, SPINE achieves a 94.3% mission success while requiring similar time and distance. Notably, SPINE makes a similar number of LLM queries, which indicates it's iteratively inferring and realizing the subtasks given to Explicit Tasking

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:06:42 UTC from IEEE Xplore. Restrictions apply.

TABLE II

SIMULATION EXPERIMENT RESULTS.

Method	Metrics				
	Success	Time	Distance	Interactions	Queries
SPINE	94.3%	536.6s	312.4m	1	6.6
LLM-as-planner	100%	1244.7	677.4	1	1.7
Explicit Tasking	100%	523s	292m	4	8.6

TABLE III

REAL-WORLD EXPERIMENT RESULTS.

Method	Metrics				
	Success	Time	Distance	Interactions	Queries
SPINE	100%	1126.0s	224.0m	1	8.3
LLM-as-planner	100%	3701.1s	570.8m	1	1.42
Explicit Tasking	100%	1035.0s	202m	5	8.6

(**Q2**). Imperfect success rate comes from the third mission, where SPINE must inspect multiple communication towers for damage. After finding that the first tower is damaged, instead of inspecting the next tower, SPINE declares the mission complete. While the LLM-as-planner approach is competitive in terms of success, because it must fully map the environment it requires over twice the time and traversal distance required (**Q1**). This method does require less LLM queries as compared to SPINE; because it receives a full map, this planner can generate a near-complete task sequence at the first planning iteration.

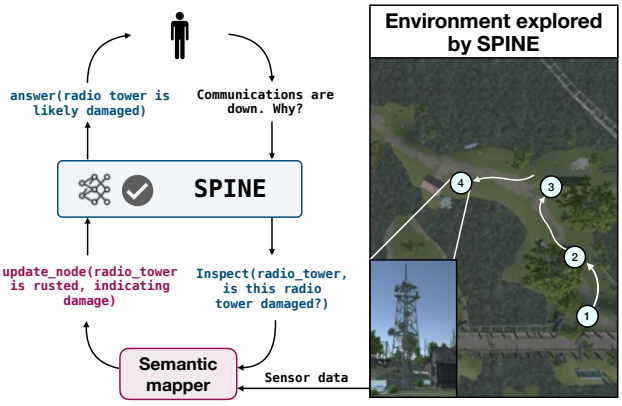


Fig. 5. Given a mission and prior map, SPINE must (2) explore and (3, 4) visit and inspect communication infrastructure. SPINE then forms an appropriate inspection query for the mapper's vision language model (VLM), and it uses the acquired information to solve the mission.

D. Real Robot Results

As reported in Tab. III, Explicit Tasking takes 1035s, travels 202m, 8.6 API calls, and requires 5 user interactions to complete a mission on average. SPINE still compares favorably to Explicit Tasking in terms of time, distance, and user interactions required (**Q2**). Interestingly, SPINE's mission success rate was higher than in simulation, which is likely due to the increased scale of the simulated environment. Also due to increased environmental scale, there is a comparatively larger gap between the LLM-as-Planner approach and SPINE (**Q1**); SPINE requires less than one third of the time and 2.5 less distance to complete a mission. We noticed that average robot speed was slower across all methods, which was due to more complex perception input, greater actuation noise, and increased obstacles as compared to simulation. See Fig. 1 for an example mission.

E. Validation module ablation

We assess the importance of online validation by comparing our method to a variant without validation (**Q3**). We provide an identical specification to each method, and we measure mission success rate as we randomly remove portions of the prior map, averaged over four trials. Results, shown in Fig. 6, indicate that verification is increasingly important as the environment becomes less certain. Qualitatively, the LLM is prone to hallucinate connections and exploration goals. Validation prevents hallucinated goals from being realized on the robot and offers an alternative plan instead (See Fig. 3).

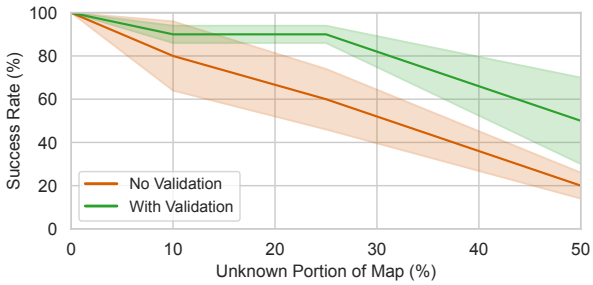


Fig. 6. Validation ablation experiment results (mean and variance).

V. CONCLUSION

We present SPINE, an online planner for missions with incomplete specifications in partially-known and unstructured environments. SPINE uses an LLM to decompose specifications in natural language into a sequence of subtasks, comprising navigation, active mapping, and user interaction, which are automatically validated and refined online. Simulation and real-world experiments demonstrate that SPINE performs comparably to methods that receive step-by-step instructions from an expert user. SPINE is also more efficient in terms of distance and time required to complete a mission as compared to the two step process of first mapping and then using an LLM-enabled planner, and does not require full *a priori* knowledge of the environment.

Future work may take several directions. SPINE requires an internet connection for LLM queries, which requires a network infrastructure. Going forward, we would like to mitigate this limitation by adapting smaller, open-sourced LLMs such as Llama [64] or Gemma [65] variants that are efficient enough to run onboard a robot. And while this paper focuses on single-robot planning, we believe that extending SPINE for online and distributed multi-robot planning applications a natural extension this work.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:06:42 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] 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.

[2] S.-K. Kim, A. Bouman, G. Salhotra, D. D. Fan, K. Otsu, J. W. Burdick, and A. akbar Agha-mohammadi, "Plgrim: Hierarchical value learning for large-scale exploration in unknown environments," *ArXiv*, vol. abs/2102.05633, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:231861864

[3] M. F. Ginting, S.-K. Kim, D. D. Fan, M. Palieri, M. J. Kochenderfer, and A. akbar Agha-Mohammadi, "SEEK: Semantic reasoning for object goal navigation in real world inspection tasks," 2024. [Online]. Available: https://arxiv.org/abs/2405.09822

[4] 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.

[5] 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.

[6] M. F. Ginting, S.-K. Kim, O. Peltzer, J. Ott, S. Jung, M. J. Kochenderfer, and A. akbar Agha-mohammadi, "Safe and efficient navigation in extreme environments using semantic belief graphs," 2023. [Online]. Available: https://arxiv.org/pdf/2304.00645.pdf

[7] V. Vasilopoulos, G. Pavlakos, S. L. Bowman, J. D. Caporale, K. Daniilidis, G. J. Pappas, and D. E. Koditschek, "Reactive Semantic Planning in Unexplored Semantic Environments Using Deep Perceptual Feedback," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4455–4462, July 2020.

[8] N. Yokoyama, S. Ha, D. Batra, J. Wang, and B. Bucher, "Vlfm: Vision-language frontier maps for zero-shot semantic navigation," in *International Conference on Robotics and Automation (ICRA)*, 2024.

[9] M. Tzes, V. Vasilopoulos, Y. Kantaros, and G. J. Pappas, "Reactive informative planning for mobile manipulation tasks under sensing and environmental uncertainty," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 7320–7326.

[10] A. Ray, C. Bradley, L. Carlone, and N. Roy, "Task and motion planning in hierarchical 3d scene graphs," 2024. [Online]. Available: https://arxiv.org/abs/2403.08094

[11] 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

[12] X. Liu, J. Lei, A. Prabhu, Y. Tao, I. Spasojevic, P. Chaudhari, N. Atanasov, and V. Kumar, "Slideslam: Sparse, lightweight, decentralized metric-semantic slam for multi-robot navigation," *arXiv preprint arXiv:2406.17249*, 2024.

[13] F. Furrer, T. Novkovic, M. Fehr, A. Gawel, M. Grinvald, T. Sattler, R. Siegwart, and J. Nieto, "Incremental Object Database: Building 3D Models from Multiple Partial Observations," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018, pp. 6835–6842.

[14] 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.

[15] I. D. Miller, F. Cladera, T. Smith, C. J. Taylor, and V. Kumar, "Air-ground collaboration with spomp: Semantic panoramic online mapping and planning," *IEEE Transactions on Field Robotics*, vol. 1, pp. 93–112, 2024.

[16] Y. Chang, K. Ebadi, C. E. Denniston, M. F. Ginting, A. Rosinol, A. Reinke, M. Palieri, J. Shi, A. Chatterjee, B. Morrell, A. akbar Agha-mohammadi, and L. Carlone, "Lamp 2.0: A robust multi-robot slam system for operation in challenging large-scale underground environments," 2022. [Online]. Available: https://arxiv.org/abs/2205.13135

[17] 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

[18] W. Gosrich, S. Mayya, S. Narayan, M. Malencia, S. Agarwal, and V. Kumar, "Multi-robot coordination and cooperation with task precedence relationships," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5800–5806.

[19] S. Kalluraya, G. J. Pappas, and Y. Kantaros, "Multi-robot mission planning in dynamic semantic environments," 2023. [Online]. Available: https://arxiv.org/pdf/2304.00645.pdf

[20] C. K. Verginis, Y. Kantaros, and D. V. Dimarogonas, "Planning and control of multi-robot-object systems under temporal logic tasks and uncertain dynamics," 2022. [Online]. Available: https://arxiv.org/abs/2204.11783

[21] M. Omama, P. Inani, P. Paul, S. C. Yellapragada, K. M. Jatavallabhula, S. Chinchali, and M. Krishna, "Alt-pilot: Autonomous navigation with language augmented topometric maps," 2023. [Online]. Available: https://arxiv.org/abs/2310.02324

[22] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar, "Voyager: An open-ended embodied agent with large language models," 2023. [Online]. Available: https://arxiv.org/abs/2305.16291

[23] 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 *arXiv preprint arXiv:2209.07753*, 2022.

[24] B. Chen, F. Xia, B. Ichter, K. Rao, K. Gopalakrishnan, M. S. Ryoo, A. Stone, and y. Daniel Kappler booktitle=arXiv preprint arXiv:2209.09874, "Open-vocabulary queryable scene representations for real world planning."

[25] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, E. Jang, R. J. Ruano, K. Jeffrey, S. Jesmonth, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, K.-H. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. Reyes, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, M. Yan, and A. Zeng, "Do as i can and not as i say: Grounding language in robotic affordances," in *arXiv preprint arXiv:2204.01691*, 2022.

[26] 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

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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," 2024. [Online]. Available: https://arxiv.org/abs/2309.09182

[33] 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

[34] B. Quartey, E. Rosen, S. Tellex, and G. Konidaris, "Verifiably following complex robot instructions with foundation models," vol. 1, 2024.

[35] Y. Chen, J. Arkin, Y. Zhang, N. Roy, and C. Fan, "Autotamp: Autoregressive task and motion planning with llms as translators and checkers," *arXiv preprint arXiv:2306.06531*, 2023.

[36] 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/

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:06:42 UTC from IEEE Xplore. Restrictions apply.

[37] S. Garg, K. Rana, M. Hosseinzadeh, L. Mares, N. Suenderhauf, F. Dayoub, and I. Reid, "Robohop: Segment-based topological map representation for open-world visual navigation," *arXiv*, 2023.

[38] 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

[39] 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.

[40] A. Werby, C. Huang, M. Büchner, A. Valada, and W. Burgard, "Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation," *Robotics: Science and Systems*, 2024.

[41] 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," 2024.

[42] M. Ryll, J. Ware, J. Carter, and N. Roy, "Semantic trajectory planning for long-distant unmanned aerial vehicle navigation in urban environments," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 1551–1558.

[43] K. Zhou, K. Zheng, C. Pryor, Y. Shen, H. Jin, L. Getoor, and X. E. Wang, "Esc: Exploration with soft commonsense constraints for zero-shot object navigation," in *Proceedings of the 40th International Conference on Machine Learning*, ser. ICML'23. JMLR.org, 2023.

[44] B. Yu, H. Kasaei, and M. Cao, "Frontier semantic exploration for visual target navigation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 4099–4105.

[45] A. Pacheck and H. Kress-Gazit, "Physically feasible repair of reactive, linear temporal logic-based, high-level tasks," *IEEE Transactions on Robotics*, 2023.

[46] C. Menghi, C. Tsigkanos, P. Pelliccione, C. Ghezzi, and T. Berger, "Specification patterns for robotic missions," 2019. [Online]. Available: https://arxiv.org/abs/1901.02077

[47] D. Honerkamp, M. Büchner, F. Despinoy, T. Welschehold, and A. Valada, "Language-grounded dynamic scene graphs for interactive object search with mobile manipulation," *arXiv preprint arXiv:2403.08605*, 2024.

[48] 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.

[49] 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.

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

[51] 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

[52] 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

[53] 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.

[54] 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.

[55] 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

[56] 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

[57] 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.

[58] 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

[59] M. Quigley, B. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," *ICRA workshop on open source software*, vol. 3, no. 3.2, p. 5, 2009.

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

[61] 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.

[62] 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.

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

[64] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," 2023. [Online]. Available: https://arxiv.org/abs/2302.13971

[65] Gemma Team et al., "Gemma: Open models based on gemini research and technology," 2024. [Online]. Available: https://arxiv.org/abs/2403.08295

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:06:42 UTC from IEEE Xplore. Restrictions apply.