

OceanPlan: Hierarchical Planning and Replanning for Natural Language AUV Piloting in Large-scale Unexplored Ocean Environments

Ruochu Yang¹, Fumin Zhang^{1,2}, Mengxue Hou³

Abstract—We develop a hierarchical LLM-task-motion planning and replanning framework to efficiently ground an abstracted human command into tangible Autonomous Underwater Vehicle (AUV) control through enhanced representations of the world. We also incorporate a holistic replanner to provide real-world feedback with all planners for robust AUV operation. While there has been extensive research in bridging the gap between LLMs and robotic missions, they are unable to guarantee success of AUV applications in the vast and unknown ocean environment. To tackle specific challenges in marine robotics, we design a hierarchical planner to compose executable motion plans, which achieves planning efficiency and solution quality by decomposing long-horizon missions into sub-tasks. At the same time, real-time data stream is obtained by a replanner to address environmental uncertainties during plan execution. Experiments validate that our proposed framework delivers successful AUV performance of long-duration missions through natural language piloting.

I. INTRODUCTION

AUVs have been actively used in a wide range of marine applications like hurricane prediction and ocean observation systems [1], [2]. However, it is usually heavy labor to pilot AUVs in real-world missions given complex mechanical manuals and mission files. It would be a relief to simplify this AUV piloting procedure with one single command, preferably in natural language. Recent upsurge of LLMs offers us a promising option to achieve this vision. While LLMs are proved to internalize rich knowledge in text formats, it raises a critical issue of leveraging such knowledge for embodied robot capabilities in the unknown and dynamically changing physical world [3]. We keep reflecting on this question: given an abstracted human command "Search the aborted warship", how can we bridge the gaps between overarching LLMs and physical AUV motions to accomplish the mission? To fulfill our vision, there are inevitable challenges in the realm of marine robotics. Underwater localization is widely recognized as a major challenge, since Global Positioning System (GPS) cannot penetrate seawater. Other terrestrial or aerial localization methods like map-based localization are unavailable in the ocean as well. Another challenge is limited information about the ocean environment. The ocean

topography is unknown *a priori* and highly unstructured. Geographical terrains and marine objects can pose unexpected collision to AUVs. Moreover, due to the vast spatial scale of the ocean and limited onboard battery capacity of the AUV, it is not enough for the planner to generate only feasible solutions. To ensure mission success, the planner has to offer energy-efficient strategies.

To address these unique challenges in marine robotics, we design a framework OceanPlan to accomplish efficient and robust AUV missions as shown in Figure 1. In general, our work belongs to the hierarchical planning category where there are many similar works. To the best of our knowledge, this is the first LLM-task-motion planning and replanning framework for natural language piloting in the AUV domain. By labeling a work as missing a (re)planner, we imply that it doesn't explicitly account for that (re)planner. For example, while every robot needs a motion planner to execute skills, some works assume that these skills are ready on hand. [4] only uses LLMs to directly call pre-defined robotic APIs. [5] integrates LLM planning with motion planning but no task planning and replanning. [6] implements replanning at both task and motion planning levels without LLMs for human interaction. [7] incorporates LLM planning to enhance task planning in specific tasks. [8] leverages replanning to deal with invalid LLM planning. [9] develops a framework of LLM-motion planning and replanning for robotic manipulators. [10] encompasses all parts of LLM-task-motion planning, but lack of replanning to handle uncertainty. Other than simply connecting planners, our framework leverages their unique advantages to achieve efficient AUV missions targeted to marine robotics challenges. For robust AUV operation in the uncertain ocean, a holistic replanner coordinates with all planners based on real-time environmental feedback. An underwater simulator is necessary because conducting empirical algorithms or train DRL algorithms on real AUVs is impractical. The main contributions of this work are summarized as follows:

- We present a hierarchical LLM-task-motion planning and replanning framework OceanPlan to pilot AUVs in natural language, specifically for long-horizon missions in large-scale unexplored ocean environments. This hierarchy is key to achieving planning efficiency with pruned search branches and grounding robotic plans with refined representations of the real world.
- We instantiate OceanPlan on a photo-realistic ocean simulator HoloEco featuring various scenes and a polished AUV model EcoMapper. Through comprehensive

This research work is supported by ONR grants N00014-19-1-2556 and N00014-19-1-2266; AFOSR grant FA9550-19-1-0283; NSF grants GCR-1934836, CNS-2016582 and ITE-2137798; and NOAA grant NA16NOS0120028.

¹ School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, USA. ² Department of Electrical and Computer Engineering, Department of Mechanical and Aerospace Engineering, The Hong Kong University of Science and Technology, Hong Kong, China. ³ School of Electrical Engineering, University of Notre Dame, Notre Dame, USA.

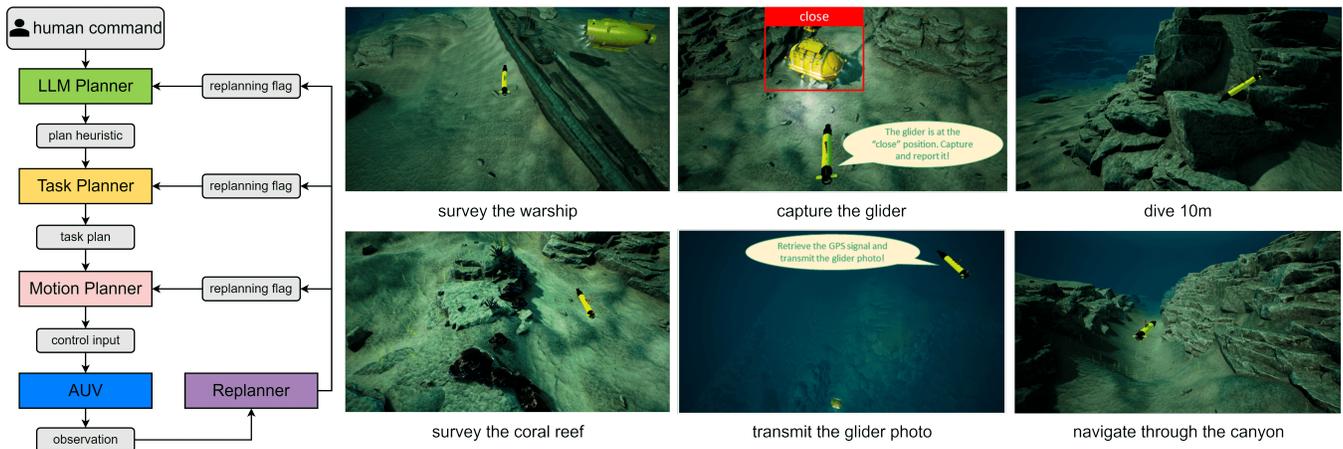


Fig. 1: OceanPlan can accomplish AUV missions through natural language commands in the large-scale unexplored ocean.

simulation, we verify efficiency (due to hierarchical architecture) and robustness (due to replanning) of our work targeting inherent AUV challenges.

II. RELATED WORKS

LLMs have exhibited considerable capabilities of interpreting natural language in the context of real world. Recently, unifying LLMs with robotics has emerged as a rapidly evolving research topic. Many works focus on LLM planning pipelines for robotic execution. One common approach relies on prompt engineering to let LLMs derive a sequential plan towards a user query [4], [7]. [11] harnesses LLM scores and RL-based affordances to select the most provable skill of completing the overall instruction. [5] utilizes LLMs to create new code policies for unseen scenarios. [12] guarantees efficiency and optimality by LLM semantic guidance and LTL consistent guidance. [13] leverages LLMs specifically for navigation tasks in unknown environments. [14] empowers LLMs with classical PDDL planners to achieve optimal planning. [15] makes a sharp point that LLMs are not really planning because they don't fully understand the physical world and their "plans" depend largely on provided prompts. It is also problematic for LLMs to generate a long-horizon plan at the very beginning without subsequent updates. [16] proposes dynamical planning to update future actions based on current and visited scenes. Taking these issues into consideration, we resort to the following well-established robotic field.

Task and Motion Planning (TAMP) is a vastly investigated field in the robotics community [17]. Classical TAMP works are established in deterministic and fully observable space, branching into topics like pick-place planning [18], manipulation planning [19], path planning [20], and rearrangement planning [21]. It is a fundamental extension to consider inevitable uncertainty in the real world [22]. [23] temporally decomposes long-horizon problems into a sequence of short horizons. [24] develops an interleaved DFS-BnB and MCTS method to achieve low computation cost and plan optimality. In light of potential failures in the real world, closed-loop replanning serves as a suitable solution [25]. Works span

from re-prompting LLMs with corrective instructions [26], receiving real-time environmental feedback [8], leveraging model-based control like MPC [9], integrating embodied modalities [27], to hierarchical replanning at both logic and motion levels [6]. Reinforcement Learning (RL) has been closely linked with motion planning, where a control policy is directly trained out of robotic action-reward datasets. [28] illustrates intrinsic connections between RL and planning. [29] proposes a real-world RL system for fine-tuning locomotion policies of legged robots. [30] guides indoor robot navigation through a DRL policy conditioned on target and current images. [31] develops generalized exploration policies over unseen environments by separately training object localization and navigation networks.

III. PROBLEM FORMULATION

In the 3D ocean world, we denote the AUV physical state as $s_t = [x, y, z, \alpha, \beta, \omega]^T \in S$, where t is the timestep, x, y, z are Cartesian coordinates, α, β, ω are roll-pitch-yaw angles, and S is the state domain. Denote $u_t = [\phi, \psi]^T \in U$ as the AUV control input at timestep t , where ϕ is the heading angle in the 2D x-y plane, ψ is the pitch angle to enable vertical movement, and U is the control domain. We denote the real-world observation as z_t obtained from AUV sensors. We denote the human command as q , which is an abstracted text specifying AUV missions. Since planning all the way down from the abstracted command q to physical AUV control u_t is extremely long-horizon, we decompose this overall problem into three sub-problems.

A. LLM Planning

To streamline planning from the robot intelligence level to the human intelligence level, we resort to LLM planning. Since the ocean environment is usually unexplored with open-set objects, we leverage LLMs' generalized inferring capabilities trained out of vast amounts of information. We denote h as a plan heuristic, which is a textual description interpreted by LLMs about the most probable way of achieving the human command q given semantic representations of the current environment. Therefore, we formulate the

following LLM planning sub-problem to command AUVs in natural language and bias the long-horizon search towards this abstracted command in the large-scale unknown ocean.

Problem 1 (LLM Planning): Given a human command q and the current observation z_t , compute a plan heuristic h which is structured into a symbolic goal state s_{goal} and a symbolic initial state s_{init} .

B. Task Planning

We consider that the AUV can perform a set of pre-defined actions, each of which is described by a set of preconditions and effects. To describe preconditions and effects using logical predicates, we abstract the AUV physical state s_t into a symbolic state. We also define a set of abstracted actions \mathcal{A} . Each action $\{a_i\}_{i=1}^N \in \mathcal{A}$ is instantiated with its preconditions and effects and executed by a series of AUV control inputs u_t computed by a control policy π_{a_i} . Additionally, considering that the LLM planner is essentially a semantics-based planner which struggles to understand the physical world, the plan directly generated by the LLM may not be executable in the real world. Especially when the AUV is navigating in a vast environment, it is impractical to factorize continuous numerical state-action space into textual prompts. Therefore, we formulate the task planning sub-problem as follows.

Problem 2 (Task Planning): Given a plan heuristic h and a set of actions \mathcal{A} , compute a task plan $\Pi = \{a_k\}_{k=1}^T \in \mathcal{A}$, which transitions the initial state s_{init} to the goal state s_{goal} .

C. Motion Planning

We rely on motion planning to execute physical control on the AUV in a collision-free manner. Due to the unknown ocean environment, the AUV dynamics will contain uncertainty. Hence we represent the AUV dynamics as an unknown MDP $M = \langle S, U, P \rangle$, where S is a set of states s_t , U is a set of control inputs u_t , and P is the unknown state transition function. Further because of no localization system under water, the AUV has no access to its true state s_t . We assume that the AUV is equipped with an on-board camera providing RGB images as observations z_t of its surrounding environment. Therefore, our goal is to find a policy based on the observations, denoted as $\pi(u_t|z_t)$, that maximizes the expected return $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$, where R_k represents the k -step reward and γ is the discount factor. Since we have no explicit distribution of reward R due to the unknown target position and ocean map, we sample AUV trajectories (action-observation pairs) in the simulator to learn the reward distribution and thus find the optimal policy.

Problem 3 (Motion Planning): Given an action a_k in the task plan Π , learn its associated control policy $\pi_{a_k}(\cdot|z_t)$ from the sampled trajectories. Given the current observation z_t , compute the control input u_t from the learned control policy $u_t \sim \pi_{a_k}(\cdot|z_t)$.

IV. METHODOLOGY

The overall framework is supposed to bridge the gaps from an abstracted human command to physical AUV control

while addressing specific challenges in the ocean. As shown in Figure 2, the framework is composed of four modules: i) A high-level LLM planner composes a plan heuristic of the human command to guide subsequent planning; ii) A middle-level task planner creates a feasible plan given the plan heuristic and predefined actions; iii) A low-level motion planner computes control inputs to execute the plan based on real-world observations; iv) A holistic replanner evaluates AUV status and reports unexpected situations to the corresponding planner for robust execution. This hierarchy decomposes planning complexity of the overall problem as high-level planners simplify planning for their low-level partners.

A. Generalized LLM Planner

The LLM planner retains the basic functionality of interpreting the command, but emphasizes more on its generalized knowledge to guide the task planner as a human-like brain, which offers two advantages: i) It can adapt into open ocean worlds based on its generalized inference rather than manually designed scene-specific prompts; ii) A reasonable plan heuristic efficiently biases search directions of achieving the human command in the large-scale ocean. Considering AUV challenges, we employ the following specific strategies.

Semantic Map: Identifiable objects can be sparse in the large-scale ocean, so the LLM planner must enrich its knowledge of the world given new observations. The LLM planner maintains an internal semantic map \mathcal{M}_t to memorize the explored environment so far. We utilize Vision-Language Models (VLMs) to convert image observations into texts, so that the LLM planner will be progressively grounded in the mission and reinforce the future plan heuristic to avoid repeated exploration. For example when searching the warship, the AUV detects a glider next to it. In the future given a new command "Survey the glider", the LLM planner will prioritize the warship area.

Augmenting XML File: Unlike indoor robotic scenes, ocean environments may not possess sufficient semantic information for the LLM commonsense reasoning. We use an XML file f_c to augment marine knowledge of the LLM planner. For instance, it can provide hints like "Coral reefs usually grow in areas with ample sunlight". The plan heuristic could be "The bright plain on the left is more likely to grow the coral reef than the dark hill on the right". Note that we still don't hands-on teach the LLM planner how to achieve AUV missions but just provide related marine knowledge.

We instantiate the current observation z_t as agent-view RGB images of the surrounding environment. Then we use a VLM to associate each image with a text descriptor like "a rocky hill on the left". We score these text descriptors with the probability of how likely they will complete the command given the XML file and the current semantic map. The one with the highest probability is selected as the plan heuristics h . By providing query-response pairs with the LLM planner, the plan heuristic is eventually structured into

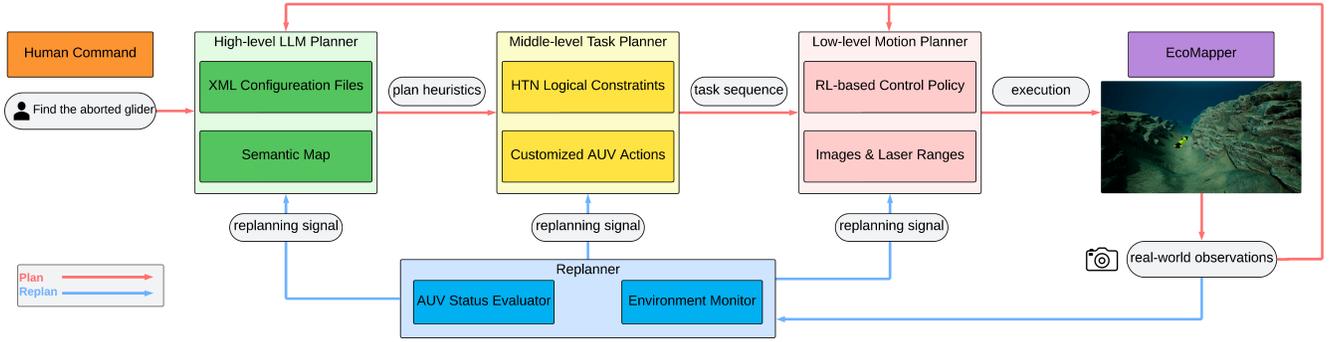


Fig. 2: Hierarchical framework of LLM-task-motion planning and replanning.

a formal task consisting of symbolic initial state s_{init} and goal state s_{goal} for the task planner.

B. HTN Task Planner

We choose HTN planner [32] as the symbolic task planner and center the actions \mathcal{A} on marine autonomy. For example, given the task "Survey the warship", the task planner should generate an action sequence Π to guide the AUV to the warship without collision in the complex ocean. As such, we define the following Boolean-valued predicates:

- `navigated ?auv` - AUV: if the AUV is navigated towards a certain area.
- `env_sensed`: if a certain area is sensed.
- `detected ?target - object`: if the target object is detected.
- `captured ?target - object`: if the target object is taken photo of.
- `approached ?target - object`: if the target object is approached at a close distance.
- `reported ?target - object`: if photos of the target object is sent to the pilot.
- `replanned`: if the replanning signal is sent to the task planner.

The AUV actions are defined as follows:

- `navigate (AUV)`: navigate the AUV towards a certain area.
- `sense`: take images of a certain area.
- `approach (target)`: move close to the target object.
- `capture (target)`: take photos of a target object.
- `report (target)`: surface to transmit photos and the GPS position of the target object.
- `rescue`: surface and send a rescue signal.

Detailed preconditions and effects of each action are presented in Appendix. We would like to delve deeper into the abstraction nature of our defined actions. There exist large-scale and unstructured landscapes like canyons and hills in the ocean, all of which pose collision risks to AUVs. However, these landscapes will not differ much across different ocean regions and may appear quite similar in high-dimensional representations like images. Furthermore, there are much less semantic objects in the ocean compared to indoor environments. In this sense, the unstructured ocean

topography with sparse semantic information constitutes a challenging factor of controlling AUVs. By abstracting motions like "swim through a canyon" or "bypass a hill" into a unified action `navigate (AUV)`, we only need to capture the overall landscape around the AUV through visual observations. In this way, we sidestep impractical ocean map modeling or precise object detection along the AUV trajectory.

C. DQN Motion Planner

The motion planner determines control inputs for the AUV to execute in the physical world. As presented in Section III-C, we are motivated to leverage DRL methods to guide AUV motions based on these visual inputs. We select Deep Q Network (DQN) [33] as our solution, which aims at finding a direct mapping (represented by a DNN θ) from the current observation z_t to the Q-function $Q_\pi(z_t, u_t|\theta)$ and then uses greedy algorithm to generate a control policy

$$\pi_{a_i}(u_t|z_t) \leftarrow \arg \max_{u_t} Q_\pi(z_t, u_t|\theta) \quad (1)$$

associated with an action $a_i \in \mathcal{A}$. The control input u_t is drawn from the control policy $u_t \sim \pi_{a_i}(\cdot|z_t)$ given the current observation z_t . We instantiate z_t as the agent-view RGB image and focus on learning the control policy associated with the `navigate (AUV)` action. The control policy plans a primitive motion like "move forward" or "turn left" given the current image, enabling the AUV to safely explore the specified region.

As mentioned in Section IV-B, we generalize the control policy of the `navigate (AUV)` action across diverse landscapes. By leveraging spatial representations embedded in images, the same control policy can be trained without differentiating between canyons or hills. DRL approaches typically take millions of steps to learn composite tasks. Our proposed planning hierarchy tremendously simplifies the motion planner into short-range movements. Moreover, our simulator HoloEco enables safe interaction with the environment so we can freely collect training datasets.

D. Holistic Replanner

Ocean environments are fairly uncertain and dynamic, so it is essential to adjust AUV behaviour through replanning. We

design a holistic replanner to trigger replanning at respective planners by simultaneously considering the plan heuristic, action effects, and AUV states. This hierarchical replanning at all planners is an intuitive yet effective approach. Low-level issues like motion drift can be directly addressed by the motion planner without altering the entire plan. High-level problems should be addressed by the task planner injecting a corrective action or by the LLM planner re-assessing the unfinished mission. Given real-world feedback, the symbolic replanning flag is designed as follows:

$$f = \begin{cases} \emptyset, & \text{normal but mission not done} \\ 0, & \text{mission done} \\ 1, & \text{LLM replanning required} \\ 2, & \text{task replanning required} \\ 3, & \text{motion replanning required} \end{cases} \quad (2)$$

We instantiate feedback to the replanner as three sensors: an IMU sensor, a forward laser sensor, and a velocity sensor. Every time the AUV executes the current control input u_t in the real world, the updated observation z' is obtained to analyze the numerical AUV state. Specifically, the replanner comprises two components: an AUV status evaluator and an environment monitor.

AUV Status Evaluator: The AUV status evaluator tracks abnormal AUV behavior. A significant reduction in AUV velocity indicates degeneration of the AUV mobility. With the replanning flag marked as $f = 2$, the task planner immediately terminates the current plan and implements the `rescue` action for assistance. If the IMU sensor detects radical AUV accelerations, the replanning flag will be triggered as $f = 3$, and the motion planner generates a corrective control input to restore the original moving direction.

Environment Monitor: The environment monitor keeps assessing the surroundings. If the AUV hasn't achieved the mission after the current plan heuristic, the replanning flag is set as $f = 1$. The LLM planner re-assesses the environment, updates the semantic map, and generates a new plan heuristic. Since the `navigate(AUV)` action is trained towards unstructured landscapes without accounting for objects, we employ a forward laser sensor in case of any collision in front of the AUV. With the replanning flag as $f = 3$, the motion planner controls the AUV away from the collision direction.

V. EXPERIMENTS

We evaluate in HoloEco simulator that given one single human command, if the AUV can efficiently and safely navigate towards the target in the large-scale unexplored ocean environment.

A. Experimental Setup

To evaluate our system, we build a marine simulator HoloEco upon HoloOcean [34], which provides high scalability and fidelity for AUV activities in a 3D ocean environment. It also ensures no damage to AUVs during the DQN training process. We instantiate three objects "coral reef, glider, warship" and three unstructured landscapes

"canyon, hill, plain". We provide the VLM with reference images of objects and landscapes so that it can accurately interpret images for the LLM planner. We instantiate the AUV model as a precise prototype EcoMapper [35]. This detailed dynamics of the scene and the AUV presents how real missions can be similarly performed using our method.

B. Mission 1 - Search Aborted Warship

After we issue an abstracted command "Search the aborted warship", OceanPlan directs EcoMapper to accomplish this comprehensive mission across the wide unknown ocean. The entire process is shown in Figure 3. Through seven phases of planning and replanning, EcoMapper successfully locates and reports the warship after exploring a wide range of the ocean. For example in phase 5, considering the command and the detected warship image, the LLM planner generates a plan heuristic "Directly survey the warship" and structures it into a formal task as

- `s_init = (not (approached warship)) (detected warship) (not (reported warship))`
- `s_goal = (reported warship)`

To achieve the task, the HTN task planner formulates a plan [`approach(warship)`, `capture(warship)`, `report(warship)`] and the motion planner sequentially executes the actions in the plan. Once the current plan is executed, the replanner evaluates both mission progress and the AUV status. A full video is available at the project website <https://sites.google.com/view/oceanplan>.

C. Mission 2 - Search Glider near Coral Reef

In this mission, there is a glider working around the coral reef. The AUV pilot would like to check the glider by issuing an abstracted command "Search the glider near coral reef". In phase 1, the VLM detects a warship and a glider. The LLM planner updates the semantic map with "A glider is near the warship on the right". The plan heuristic is "Search the left area for the other glider near the coral reef". In phase 2 with few objects, environment semantic information is extremely sparse. The LLM planner turns to the XML file in Section V-A for more hints and generates a plan heuristic "Based on the coral reef attribute in the XML file, it is likely to grow in the front plain with ample sunlight". A full video is available at the project website <https://sites.google.com/view/oceanplan>.

D. Mission 3 - Replan

Through three unexpected situations where replanning is initiated at the corresponding planner, we present robust AUV operation in the unpredictable ocean. In Mission 1, replanning at the LLM planner has been extensively presented, so we focus on presenting replanning at task planner and motion planner. A full video is available at the project website <https://sites.google.com/view/oceanplan>.

Situation 1: The replanner detects a significant reduction in AUV speed and sends a replanning signal to the task planner. The task planner immediately replaces the current

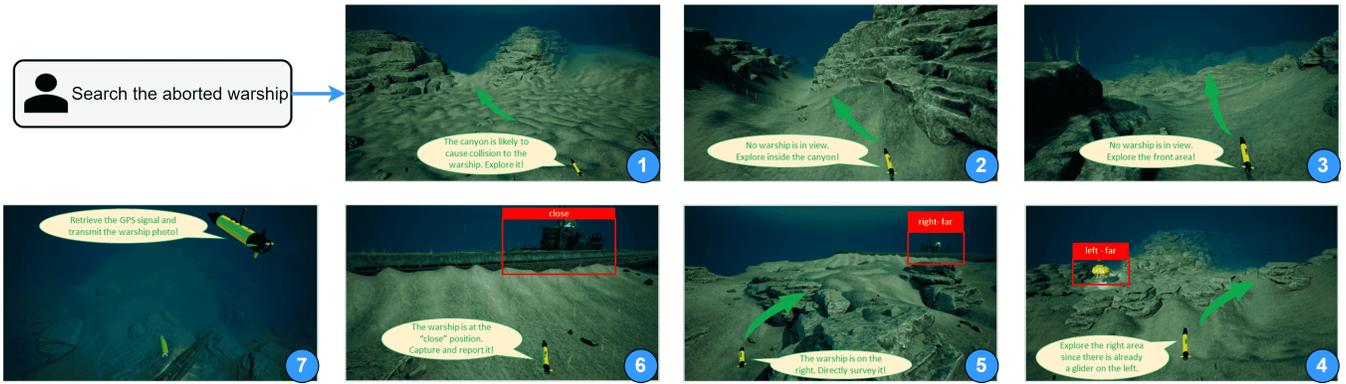


Fig. 3: The entire process of EcoMapper searching the aborted warship given an abstracted human command. Each numbered picture corresponds to a specific phase of the process.

plan with the `rescue` action so that EcoMapper surfaces and transmits the rescue signal in time.

Situation 2: The replanner identifies a radical leftward acceleration of AUV, triggering the replanning flag of the motion planner. The motion planner responds with a corrective control input "turn right" to offset the unexpected leftward acceleration.

Situation 3: The replanner detects that EcoMapper is at a high risk of colliding with a glider in front of it. The motion planner controls EcoMapper to move away from the glider.

E. Ablation Studies

We perform two ablation studies of Mission 1 and Mission 2 to evaluate importance of LLM planner and task planner during long missions in the unexplored ocean. We carry out 10 simulation runs and average both the completion time and the success rate. We exclude response time of LLM and VLM. We aim to claim through quantitative comparison that lack of any component in our proposed framework will largely depreciate the AUV performance.

Ablation of LLM Planner: In this study, we only use task planner and motion planner to achieve the same command with the same actions. In each phase, we manually evaluate the images and form a task for the task planner. We only evaluate if the target object is in view and don't consider semantic information about other objects. If not in view, we randomly choose an exploring direction. As shown in Figure 4, it takes around three times longer than the proposed method to achieve Mission 1 and around twice to achieve Mission 2. We can conclude that it is inefficient to rely solely on the task planner and motion planner to perform long-horizon planning given an abstracted command. In absence of heuristics from the LLM planner, the task planner will take much more time to randomly explore the ocean space in a brute-force pattern.

Ablation of Task Planner: In this study, we only use LLM planner and motion planner. We provide the LLM planner with the same actions, but don't provide their preconditions and effects or illustrative prompts. The LLM planner relies on itself to organize the actions to achieve the heuristic. As shown in Figure 4, the

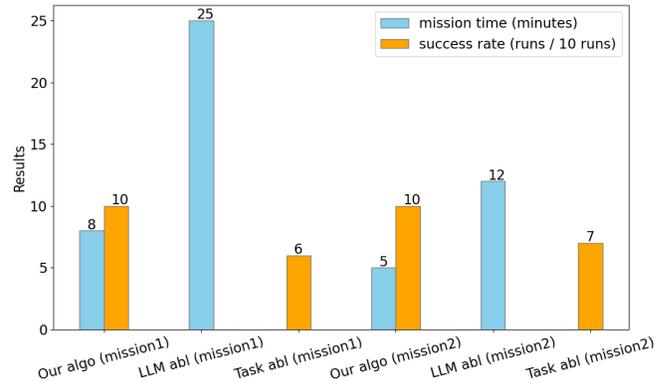


Fig. 4: Quantitative results of ablation studies demonstrate that our method achieves a good balance between efficiency and validity of planning a long-horizon mission given an abstracted command.

LLM planner generates invalid plans 6 out of 10 runs in Mission 1 and 7 out of 10 runs in Mission 2. For example in Mission 1, for the heuristic "Directly survey the warship", a wrong plan `[navigate(ecomapper), sense, approach(warship), navigate(ecomapper), sense, capture(warship), navigate(ecomapper), sense, report(warship)]` is generated, where the second `navigate(ecomapper)` action causes EcoMapper to collide with the warship. We can conclude that without logical connections introduced by the task planner, the LLM planner cannot guarantee the plan quality.

VI. CONCLUSION

We propose a hierarchical planning and replanning framework to pilot AUVs through natural language in the large-scale unknown ocean given major marine robotics challenges. Given a human command, an LLM planner generates a plan heuristic, a task planner guarantees a valid plan, a motion planner executes the plan in the real world, and a holistic replanner ensures robust AUV operation. In a marine simulator HoloEco, OceanPlan is validated to ground human commands for effective and safe AUV missions.

REFERENCES

- [1] F. Zhang, D. M. Fratantoni, D. A. Paley, J. M. Lund, and N. E. Leonard, "Control of coordinated patterns for ocean sampling," *International Journal of Control*, vol. 80, no. 7, pp. 1186–1199, 2007.
- [2] D. A. Paley, F. Zhang, and N. E. Leonard, "Cooperative control for ocean sampling: The glider coordinated control system," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 4, pp. 735–744, 2008.
- [3] S. Tellex, N. Gopalan, H. Kress-Gazit, and C. Matuszek, "Robots that use language," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 25–55, 2020.
- [4] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," in *International Conference on Machine Learning*. PMLR, 2022, pp. 9118–9147.
- [5] 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)*. IEEE, 2023, pp. 9493–9500.
- [6] T. Lin, C. Yue, Z. Liu, and X. Cao, "Generalized multi-level replanning tamp framework for dynamic environment," *arXiv preprint arXiv:2310.14816*, 2023.
- [7] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, "Progprompt: program generation for situated robot task planning using large language models," *Autonomous Robots*, pp. 1–14, 2023.
- [8] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar *et al.*, "Inner monologue: Embodied reasoning through planning with language models," in *Conference on Robot Learning*. PMLR, 2023, pp. 1769–1782.
- [9] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," in *Conference on Robot Learning*. PMLR, 2023, pp. 540–562.
- [10] D. Shah, M. R. Equi, B. Osiński, F. Xia, S. Levine *et al.*, "Navigation with large language models: Semantic guesswork as a heuristic for planning," in *7th Annual Conference on Robot Learning*, 2023.
- [11] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.
- [12] 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," *arXiv preprint arXiv:2309.09182*, 2023.
- [13] B. Yu, H. Kasaei, and M. Cao, "L3mvm: Leveraging large language models for visual target navigation," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 3554–3560.
- [14] 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.
- [15] K. Valmeekam, A. Olmo, S. Sreedharan, and S. Kambhampati, "Large language models still can't plan (a benchmark for llms on planning and reasoning about change)," in *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022.
- [16] A. Rajvanshi, K. Sikka, X. Lin, B. Lee, H.-P. Chiu, and A. Velasquez, "Saynav: Grounding large language models for dynamic planning to navigation in new environments," *arXiv preprint arXiv:2309.04077*, 2023.
- [17] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez, "Integrated task and motion planning," *Annual review of control, robotics, and autonomous systems*, vol. 4, pp. 265–293, 2021.
- [18] M. Gualtieri and R. Platt, "Robotic pick-and-place with uncertain object instance segmentation and shape completion," *IEEE robotics and automation letters*, vol. 6, no. 2, pp. 1753–1760, 2021.
- [19] H. Zhang, S.-H. Chan, J. Zhong, J. Li, P. Kolapo, S. Koenig, Z. Agioutantis, S. Schafrik, and S. Nikolaidis, "Multi-robot geometric task-and-motion planning for collaborative manipulation tasks," *Autonomous Robots*, pp. 1–22, 2023.
- [20] M. Burke, K. Lu, D. Angelov, A. Straižys, C. Innes, K. Subr, and S. Ramamoorthy, "Learning rewards from exploratory demonstrations using probabilistic temporal ranking," *Autonomous Robots*, vol. 47, no. 6, pp. 733–751, 2023.
- [21] Y. Ding, X. Zhang, C. Paxton, and S. Zhang, "Task and motion planning with large language models for object rearrangement," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 2086–2092.
- [22] T. Lozano-Pérez, M. T. Mason, and R. H. Taylor, "Automatic synthesis of fine-motion strategies for robots," *The International Journal of Robotics Research*, vol. 3, no. 1, pp. 3–24, 1984.
- [23] A. Curtis, X. Fang, L. P. Kaelbling, T. Lozano-Pérez, and C. R. Garrett, "Long-horizon manipulation of unknown objects via task and motion planning with estimated affordances," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 1940–1946.
- [24] M. Hou, Y. Li, F. Zhang, S. Sundaram, and S. Mou, "An interleaved algorithm for integration of robotic task and motion planning," in *2023 American Control Conference (ACC)*. IEEE, 2023, pp. 539–544.
- [25] C. R. Garrett, C. Paxton, T. Lozano-Pérez, L. P. Kaelbling, and D. Fox, "Online replanning in belief space for partially observable task and motion problems," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 5678–5684.
- [26] P. Sharma, B. Sundaralingam, V. Blukis, C. Paxton, T. Hermans, A. Torralba, J. Andreas, and D. Fox, "Correcting Robot Plans with Natural Language Feedback," in *Proceedings of Robotics: Science and Systems*, New York City, NY, USA, June 2022.
- [27] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence, "Palme: An embodied multimodal language model," 2023.
- [28] R. Munos *et al.*, "From bandits to monte-carlo tree search: The optimistic principle applied to optimization and planning," *Foundations and Trends® in Machine Learning*, vol. 7, no. 1, pp. 1–129, 2014.
- [29] L. Smith, J. C. Kew, X. B. Peng, S. Ha, J. Tan, and S. Levine, "Legged robots that keep on learning: Fine-tuning locomotion policies in the real world," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 1593–1599.
- [30] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 3357–3364.
- [31] A. Devo, G. Mezzetti, G. Costante, M. L. Fravolini, and P. Valigi, "Towards generalization in target-driven visual navigation by using deep reinforcement learning," *IEEE Transactions on Robotics*, vol. 36, no. 5, pp. 1546–1561, 2020.
- [32] D. Nau, Y. Cao, A. Lotem, and H. Munoz-Avila, "Shop: Simple hierarchical ordered planner," in *Proceedings of the 16th international joint conference on Artificial intelligence-Volume 2*, 1999, pp. 968–973.
- [33] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [34] E. Potokar, S. Ashford, M. Kaess, and J. Mangelson, "HoloOcean: An underwater robotics simulator," in *Proc. IEEE Intl. Conf. on Robotics and Automation, ICRA*, Philadelphia, PA, USA, May 2022.
- [35] C. Wang, F. Zhang, and D. Schaefer, "Dynamic modeling of an autonomous underwater vehicle," *Journal of Marine Science and Technology*, vol. 20, pp. 199–212, 2015.

APPENDIX

ACTION PRECONDITIONS AND EFFECTS

Detailed preconditions and effects of predefined actions are presented in Figure 5.

ACTION IMPLEMENTATION

We illustrate detailed implementation of the control policies associated with the `navigate(AUV)` and `approach(target)` actions.

```

(:predicates
navigated ?auv - AUV
env_sensed
detected ?target - object
captured ?target - object
approached ?target - object
reported ?target - object
replanned
)

(:action: approach
:parameters (?target - object)
:preconditions (and
(not (approached ?target))
(detected ?target))
:effect (and (approached ?target))
)

(:action: capture
:parameters (?target - object)
:preconditions (and
(approached ?target)
(not (captured ?target))
:effect (and (captured ?target))
)

(:action: report
:parameters (?target - object)
:preconditions (and
(captured ?target)
(not (reported ?target))
:effect (and (reported ?target))
)

(:action: rescue
:parameters ()
:preconditions (and (replanned))
:effect ()
)

(:action: navigate
:parameters (?auv - AUV)
:preconditions (and
(not (navigated ?auv))
(not (detected ?target)))
:effect (and (navigated ?auv))
)

(:action: sense
:parameters ()
:preconditions (and
(not (env_sensed))
(navigated ?auv))
:effect (and (env_sensed))
)

```

Fig. 5: Preconditions and effects of predefined AUV actions.

A. *navigate (AUV) action*

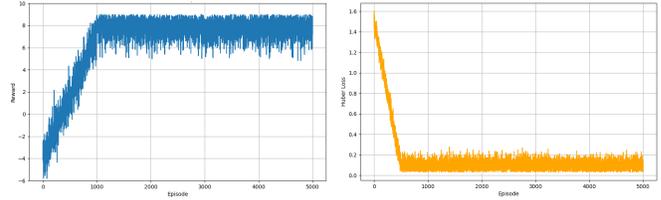
The control policy of the `navigate (AUV)` action is a learned stochastic policy function which takes the current image as input and outputs a primitive action. We utilize DQN to train the control policy, which requires training two policies: the behaviour policy and the target policy with the same network parameters. The behaviour policy collects experiences through interaction with the simulated environment, while the target policy learns from them to update its own network. We follow the training pipeline of [31] and instantiate the training scenario as EcoMapper navigating through a canyon, hill, and plain safely and quickly.

- Control inputs: turn left, turn right, move forward, move up.
- Observation: the current agent-view RGB image.
- Immediate reward: time penalty -0.1 to encourage shorter trajectories; collision penalty -0.1 to discourage collision. Goal-reaching reward +10 upon action completion.
- Terminated condition: The AUV safely navigates through the area. Truncated condition: Control inputs exceed 30.
- Network architecture: We use Convolutional Neural Network to extract features of the current image. Next we flatten and feed the features to a Feedforward Network, which returns the Q value of all control inputs.

The training results in Figure 6 present convergence of both the loss and the reward. The training hyperparameters are shown in Table I.

TABLE I: DQN training hyperparameters.

Hyperparameters	Value
discount factor	0.95
replay memory total size	60000
relay memory batch size	64
learning rate	0.005
initial training samples	2000
target policy updating frequency	500
training episodes	5000
epsilon limit of behavior policy	0.05



(a) Reward over episode.

(b) Huber loss over episode.

Fig. 6: DQN training results.

B. *approach (target) action*

The control policy of the `approach (target)` action is to move the AUV close to the target object by tracking the target in the image center. The VLM outputs the relative position of the target with respect to the AUV by comparing the current image and the target images. The VLM first identifies if the target object is 'close' or 'far' to the AUV. If deemed 'far', the position is discretized into five categories: 'left-far', 'right-far', 'center-far', 'top-far', and 'bottom-far'. We take into account three classes of objects 'glider', 'warship', and 'coral reef'. Following question-answer samples, the VLM can identify the object's relative position in the image. The `approach (target)` action will not end until the target is detected as 'close'. Since there is no training for this action, we use a forward laser sensor to avoid collision with the target object. An illustrative detection result is shown in Figure 7.

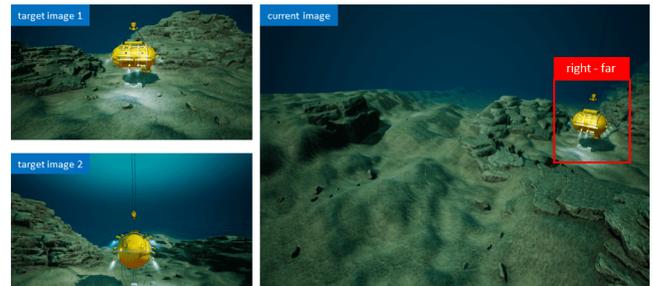


Fig. 7: On the left side, two target images show the target object 'glider' at the 'center-close' position. On the right side, the current image is identified with the 'glider' at the 'right-far' position.