

Collaborative Planning with Concurrent Synchronization for Operationally Constrained UAV-UGV Teams

Author Names Omitted for Anonymous Review. Paper-ID 802.

Abstract—Collaborative planning under operational constraints is an essential capability for heterogeneous robot teams tackling complex large-scale real-world tasks. Unmanned Aerial Vehicles (UAVs) offer rapid environmental coverage, but flight time is often limited by energy constraints, whereas Unmanned Ground Vehicles (UGVs) have greater energy capacity to support long-duration missions, but movement is constrained by traversable terrain. Individually, neither can complete tasks such as environmental monitoring. Effective UAV-UGV collaboration therefore requires energy-constrained multi-UAV task planning, traversability-constrained multi-UGV path planning, and crucially, synchronized concurrent co-planning to ensure timely in-mission recharging. To enable these capabilities, we propose *Collaborative Planning with Concurrent Synchronization (CoPCS)*, a learning-based approach that integrates a heterogeneous graph transformer for operationally constrained task encoding with a transformer decoder for joint, synchronized co-planning that enables UAVs and UGVs to act concurrently in a coordinated manner. CoPCS is trained end-to-end under a unified imitation learning paradigm. We conducted extensive experiments to evaluate CoPCS in both robotic simulations and physical robot teams. Experimental results demonstrate that our method provides the novel multi-robot capability of synchronized concurrent co-planning and substantially improves team performance.

More details are provided on the (anonymous) project website: <https://copcs.github.io/CoPCS>.

I. INTRODUCTION

Multi-robot systems are a promising paradigm in robotics because of their advantages such as redundancy [1], parallelism [2], and scalability [3]. These characteristics make multi-robot systems indispensable for addressing complex, large-scale tasks that are usually infeasible for a single robot. Collaborative multi-robot planning is a fundamental capability that allows robots to collaborate on their actions, share resources, and complete interdependent tasks under operational constraints. Effective collaborative planning in multi-robot systems is necessary not only for achieving operational efficiency but also for ensuring the reliability of robot teams when deployed in a variety of real-world applications, including disaster response [4]–[6] and environmental monitoring [7]–[10].

Multi-robot systems often consist of heterogeneous robots with diverse capabilities and operational constraints. For example, Unmanned Aerial Vehicles (UAVs) excel at providing rapid environmental coverage and reaching locations that are inaccessible or difficult for ground robots. However, their endurance is severely constrained by limited energy capacity, which restricts both the scale and duration of their missions. In contrast, Unmanned Ground Vehicles (UGVs) are often slower and subject to terrain traversability constraints; however, they can carry larger energy reserves and function as mobile recharging stations for UAVs. Figure 1 illustrates a scenario in

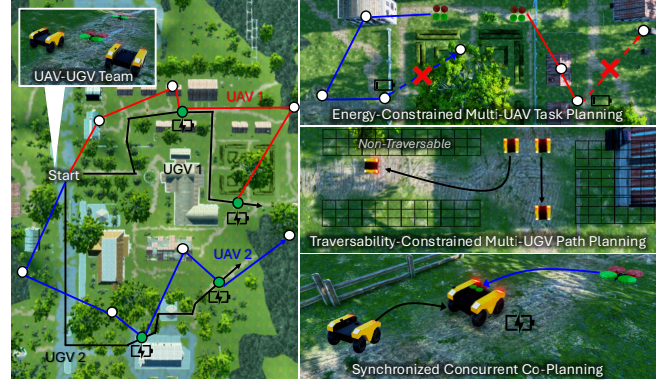


Fig. 1. A motivating scenario of a UAV-UGV team performing collaborative planning under operational constraints to visit all task points for environmental monitoring. UAVs are flight time limited given energy constraints, while UGVs are constrained by terrain traversability, preventing either from completing the mission alone. To overcome this, UGVs act as mobile recharging stations, carrying energy supplies and synchronizing with UAVs to sustain their operations. Task points are shown as white circles and recharging points as green circles, while colored lines denote UAV trajectories and black lines denote UGV paths.

which a UAV-UGV team collaborates to visit all task points for environment monitoring. Neither UAVs nor UGVs alone can accomplish the mission due to energy and traversability limitations; however, UGVs can collaborate with UAVs and sustain their operations through in-mission recharging. In such cases, UAVs must determine which task locations to visit and the visit sequence given their energy constraints (i.e., *energy-constrained multi-UAV task planning*). UGVs must plan paths along traversable road networks that support UAV operations (i.e., *traversability-constrained multi-UGV path planning*). Most importantly, these plans must be jointly optimized and time-synchronized to guarantee timely in-mission recharging (i.e., *synchronized concurrent co-planning*).

Due to the importance of UAV-UGV collaborative planning, various methods have been explored. Traditional approaches, including Mixed Integer Programming [11], Dynamic Programming [12], and metaheuristics [13], can usually identify optimal solutions. However, these methods often demand substantial domain knowledge and can become computationally expensive for large-scale problems, limiting their applicability for real-time execution. Learning-based methods address these limitations and have shown promising results. For example, graph neural networks are used to capture the relationships between robots and tasks [14], imitation learning allows systems to mimic expert demonstrations [15], and reinforcement learning improves decision-making through environmental

feedback [16]. However, most existing learning-based methods fail to adequately address operational constraints, and often fail to achieve synchronized concurrent co-planning between UAVs and UGVs, which can result in longer mission times or even mission failure.

To address these challenges, we propose the method of *Collaborative Planning with Concurrent Synchronization* (CoPCS) for operationally constrained UAV-UGV teams, which enables a UAV-UGV team to work collaboratively: energy-constrained UAVs plan their task-point visits, traversability-constrained UGVs plan supportive routes, and the full team achieves concurrent synchronization to ensure timely in-mission recharging. Specifically, we design a heterogeneous graph representation to encode entities, including tasks, path points, and heterogeneous robots. Each node represents an entity with its attributes, and each edge captures the spatial relationships among the entities. Given this graph, CoPCS uses a heterogeneous graph transformer to generate constraint-aware embeddings that facilitate synchronized collaboration. Using these embeddings, CoPCS employs a transformer decoder to generate a joint action sequence for both UAVs and UGVs through co-planning under operational constraints, covering task visits, path movements, and recharging decisions. The unified framework is trained end-to-end using imitation learning, and UAV and UGV actions are executed simultaneously, resulting in synchronized concurrent co-planning.

Our key contribution is the proposal of the CoPCS approach for collaborative planning with concurrent synchronization for operationally constrained UAV-UGV teams. Specific novelties of this work are as follows:

- We enable a new multi-robot capability for constrained heterogeneous co-planning, which simultaneously addresses energy-constrained multi-UAV task planning, traversability-constrained multi-UGV path planning, and most importantly, synchronized concurrent co-planning for the entire team.
- We introduce CoPCS as one of the first unified learning-based solutions that integrates heterogeneous graphs for operationally constrained task encoding with transformer decoders for sequential and synchronized action generation, which is trained end-to-end under a unified imitation learning paradigm.

II. RELATED WORK

A. Classic Methods for UAV-UGV Collaborative Planning

A variety of traditional approaches have been implemented to address the collaborative multi-robot planning problem, and can be broadly categorized into optimization-based and heuristic-based approaches. Optimization methods formulate UAV-UGV collaborative planning as NP-hard problems such as the Vehicle Routing Problem with Drones (VRP-D) or the Traveling Salesman Problem with Drones (TSP-D). Mixed Integer Programming (MIP) employs binary and continuous variables to encode task assignments and vehicle routes [11], [17]–[19], while Dynamic Programming (DP) recursively decomposes the problem into subproblems (e.g., shortest paths

over subsets of tasks) and combines their solutions using the principle of optimality [12], [20], [21]. Although these methods provide exact solutions, their computational cost grows rapidly with problem size, which limits scalability.

To address this, heuristic and metaheuristic methods have been widely adopted. Guided Local Search (GLS) [13], Tabu Search (TS) [22], and Simulated Annealing (SA) [23] iteratively improve candidate solutions by exploring neighborhoods while using different strategies to escape local optima. In UAV-UGV collaborative planning, they are often implemented in multi-level frameworks, where UGV routes are planned first and UAV task assignments optimized afterward [24]–[26]. While this decomposition can yield optimal solutions within each level, it cannot guarantee global optimality across the mission. Compared to optimization methods, heuristics and metaheuristics reduce computational costs and scale better to large problem instances, but they typically cannot ensure global optimality and often remain too slow for real-time execution.

B. Learning for UAV-UGV Collaborative Planning

Learning-based approaches have recently shown strong potential to address the computational limitations of optimization methods. Graph Neural Networks (GNNs) capture relational structures between tasks and agents through message passing [14], [27], [28]. Imitation Learning (IL) complements this by mimicking expert demonstrations to generalize solutions [15], [29]. Reinforcement Learning (RL) learns decision-making policies via interactions with the environments. An attention-based network has been applied to solve the TSP [30] and extended to multi-TSP [31], [32]. However, these methods neglect practical constraints in real-world applications, such as UAV energy. To address this, several studies introduce energy-constrained UAV planning with fixed recharging stations [33]–[35], but they mainly focus on homogeneous UAV teams without considering collaboration with UGVs.

To improve efficiency, recent efforts explore UAV-UGV co-planning, where UGVs act as mobile recharging stations to eliminate UAV detours to fixed stations [36], [37]. Yet, these studies are limited to single UAV-UGV settings. More recently, co-planning multiple energy-constrained UAVs and UGVs has introduced the capability to coordinate larger teams [16]. However, these approaches often rely on greedy strategies to assign UAVs to UGVs and lack concurrent synchronization across the team. As a result, UAVs and UGVs cannot execute their plans simultaneously, leading to unsuccessful missions or longer completion times, since UAVs may be forced to wait for UGVs to arrive before recharging.

III. APPROACH

A. Problem Definition

We discuss our CoPCS approach that enables the new multi-robot capabilities of synchronized concurrent co-planning for heterogeneous robot teams. An overview of CoPCS is illustrated in Figure 2. We denote the set of task points as $\mathcal{M} = \{m_0, m_1, \dots, m_{|\mathcal{M}|-1}\}$, which must be visited by a team of energy-constrained UAVs, defined as $\mathcal{U}^a = \{u_0^a, u_1^a, \dots, u_{|\mathcal{U}^a|-1}^a\}$. In addition, we denote the team of

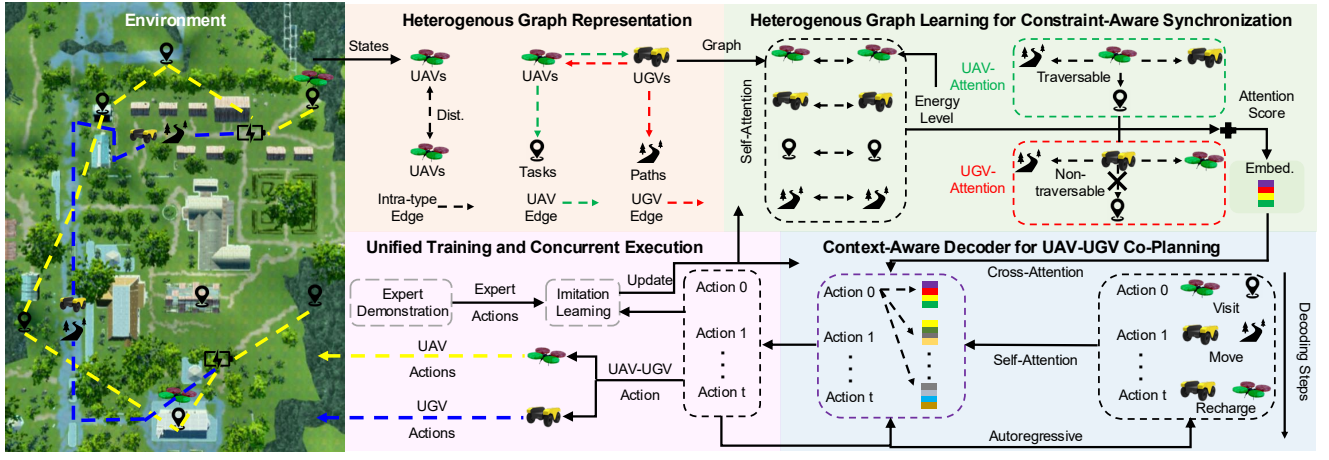


Fig. 2. Overview of CoPCS (*Collaborative Planning with Concurrent Synchronization*), a unified learning-based method that integrates a heterogeneous graph transformer for synchronized collaboration among operationally constrained UAV-UGV teams and a transformer decoder for joint action co-planning. The CoPCS approach is trained end-to-end, and its co-planning policy is executed concurrently to coordinate all the robots, which enables concurrent synchronization.

UGVs as $\mathcal{U}^g = \{u_0^g, u_1^g, \dots, u_{|\mathcal{U}^g|-1}^g\}$, which must plan their paths over traversable terrains to reach a set of points $\mathcal{P} = \{p_0, p_1, \dots, p_{|\mathcal{P}|-1}\}$ in order to recharge the UAVs.

Formally, we represent the environmental context as a graph $\mathcal{G} = \{\mathcal{T}, \mathcal{E}^{intra}, \mathcal{E}^{uav}, \mathcal{E}^{ugv}\}$. The node set \mathcal{T} is defined as $\mathcal{T} = \mathcal{T}^m \cup \mathcal{T}^p \cup \mathcal{T}^a \cup \mathcal{T}^g$. The task set $\mathcal{T}^m = \{\mathbf{t}_i^m\}$ encodes the locations of all tasks, where $\mathbf{t}_i^m = [\mathbf{d}_i^m, \delta_i]$, with \mathbf{d}_i^m denoting the two-dimensional coordinates of the task location along the x and y directions, and $\delta_i \in \{0, 1\}$ indicating whether the task location has been visited or not. The path set $\mathcal{T}^p = \{\mathbf{t}_i^p\}$ contains the central locations of all paths, where \mathbf{t}_i^p denotes the position of the central location of the i -th path. The UAV set $\mathcal{T}^a = \{\mathbf{t}_i^a\}$ contains the attributes of UAVs, where $\mathbf{t}_i^a = [\mathbf{d}_i^a, f_i]$. The UAV position is denoted as \mathbf{d}_i^a , and f_i indicates the remaining energy. The UGV set $\mathcal{T}^g = \{\mathbf{t}_i^g\}$ contains the attributes of UGVs, where $\mathbf{t}_i^g = [\mathbf{d}_i^g, \mathbf{d}^a]$, with \mathbf{d}_i^g denoting the UGV position, and \mathbf{d}^a indicating the position of the UAVs.

We define three types of edge connectivity based on relative distance. The intra-type edge set is denoted as $\mathcal{E}^{intra} = \{e_{i,j}^{intra}\}$, which connects nodes of the same type. The UAV edge set is defined as $\mathcal{E}^{uav} = \{e_{i,j}^{uav}\}$, which connects UAV nodes with their related nodes, including UGVs and tasks. Similarly, the UGV edge set is denoted as $\mathcal{E}^{ugv} = \{e_{i,j}^{ugv}\}$, which connects UGV nodes with their related nodes, including UAVs and paths. In all cases, an edge $e_{i,j} = 1$ if the i -th and j -th nodes are within a predefined radius; otherwise $e_{i,j} = 0$.

Given the environmental context graph \mathcal{G} , we aim to collaboratively plan the actions of UAVs and UGVs. Formally, the UAV action is defined as $\mathbf{A}^{uav} = \{a_{i,j}\}_{|\mathcal{U}^a| \times |\mathcal{M}|}$, where $a_{i,j} = 1$ indicates that the i -th UAV visits the j -th task. The UGV action is defined as $\mathbf{A}^{ugv} = \{a_{i,j}\}_{|\mathcal{U}^g| \times |\mathcal{P}|}$, where $a_{i,j} = 1$ indicates that the i -th UGV selects the j -th path to traverse. In addition, the recharging action is defined as $\mathbf{A}^{recharge} = \{a_{i,j}\}_{|\mathcal{U}^a| \times |\mathcal{U}^g|}$, where $a_{i,j} = 1$ indicates that the i -th UAV is recharging on the j -th UGV. Our goal is to address the following UAV-UGV collaborative planning problems under operational constraints:

- **Synchronized concurrent co-planning:** The joint plan-

ning of UAV and UGV actions to achieve synchronized movement, which ensures simultaneous arrival at charging locations, as well as concurrent operation to enable UAVs and UGVs to operate in parallel for continuous and efficient task execution.

- **Energy-constrained multi-UAV task planning:** The coordination of multiple UAVs with limited energy so that all task points are visited exactly once while minimizing the overall mission completion time.
- **Traversability-constrained multi-UGV path planning:** The coordination of multiple UGVs to plan feasible and efficient routes over a road network, subject to terrain traversability constraints.

B. Heterogeneous Graph Learning for Constraint-Aware Synchronization

Given the environmental context graph \mathcal{G} , we develop a heterogeneous graph transformer network [38] ψ to compute the node embedding as $\mathcal{H} = \{\mathbf{h}_i\} = \psi(\mathcal{G})$, where \mathbf{h}_i denotes the embedding of the i -th node. These embeddings capture the relationships among UAVs, UGVs, tasks, and paths, which enable constraint-aware synchronization that further facilitates collaborative planning.

The network ψ first projects the node attributes \mathbf{t}_i into a feature vector \mathbf{z}_i of the i -th node using linear layers by $\mathbf{z}_i = \mathbf{W}^z \mathbf{t}_i$, where \mathbf{W}^z denotes the type-specific learnable weight matrix of the linear layer. Then, we compute the query, key, and value embeddings of each node by:

$$\mathbf{q}_i^l = \mathbf{W}_q^l \mathbf{z}_i^l, \quad \mathbf{k}_i^l = \mathbf{W}_k^l \mathbf{z}_i^l, \quad \mathbf{v}_i^l = \mathbf{W}_v^l \mathbf{z}_i^l \quad (1)$$

where \mathbf{q}_i^l , \mathbf{k}_i^l , and \mathbf{v}_i^l denote query, key, and value at the l -th layer. \mathbf{W}_q^l , \mathbf{W}_k^l , and \mathbf{W}_v^l denote their weight matrices.

To encode relationships among nodes of the same type (e.g., within the UAV team), we apply a self-attention mechanism that aggregates features across nodes of that type. The self-attention score from node j to connected node i in \mathcal{E}^{intra} at

layer l is computed as follows:

$$\alpha_{i,j}^l = \text{SoftMax} \left(\frac{(\mathbf{q}_i^l)^\top \mathbf{k}_j^l}{\sqrt{d}} \right) \quad (2)$$

where $\text{SoftMax}(\cdot)$ is the softmax function and d is the dimension of the $\mathbf{q}_i^l, \mathbf{k}_j^l$.

To facilitate synchronization among heterogeneous robots, we employ a cross-attention mechanism that aggregates features across selected node types based on their edge connectivity. For UAVs, message passing is performed along all UAV edges \mathcal{E}^{uav} , and the cross-attention score from the connected k -th node to the i -th UAV node at layer l is defined as:

$$\beta_{i,k}^{l,a} = \text{SoftMax} \left(\frac{(\mathbf{q}_i^l)^\top \mathbf{k}_k^l}{\sqrt{d}} \right) \quad (3)$$

For UGVs, we explicitly encode traversability constraints by restricting message passing to UGV edges \mathcal{E}^{ugv} that exclude connections to task nodes, since tasks may lie in non-traversable regions. Formally, the cross-attention score from the connected n -th node to the i -th UGV node in at layer l is defined as:

$$\beta_{i,n}^{l,g} = \text{SoftMax} \left(\frac{(\mathbf{q}_i^l)^\top \mathbf{k}_n^l}{\sqrt{d}} \right) \quad (4)$$

Finally, given the scores from both self-attention and cross-attention mechanisms, we update the embedding \mathbf{h}_i of each node by aggregating both scores, which is defined as follows:

$$\mathbf{h}_i^{l+1} = \text{LN} \left(\mathbf{h}_i^l + \sum_{e_{i,j}^{intra}=1} \alpha_{i,j}^l \mathbf{v}_j^l + \sum_{e_{i,k}^{uav}=1} \beta_{i,k}^{l,a} \mathbf{v}_k^l + \sum_{e_{i,n}^{ugv}=1} \beta_{i,n}^{l,g} \mathbf{v}_n^l \right) \quad (5)$$

where $\text{LN}(\cdot)$ denotes the layer normalization operation. The updated embedding \mathbf{h}_i^{l+1} for the i -th node encodes both intra-type features (second term) and inter-type features (third and fourth term). This explicitly enables synchronization because robots are able to exchange information with one another, while remaining constraint-aware through the inclusion of UAV energy and UGV traversability in their respective embeddings.

C. Context-Aware Decoder for UAV-UGV Co-Planning

Given the embedding set \mathcal{H} generated by the heterogeneous graph transformer network ψ , we develop a transformer-based decoder [39] ϕ to generate joint action sequences for both UAVs and UGVs in an auto-regressive manner, thereby allowing co-planning. The decoder predicts one action at each decoding step t , it generates action a_t conditioned on the action history $\{a_1, \dots, a_{t-1}\}$ and the graph embedding \mathcal{H} , where $a_t \in \mathbf{A}^{uav}$ if it is a UAV visiting action, $a_t \in \mathbf{A}^{ugv}$ if it is a UGV moving action, and $a_t \in \mathbf{A}^{recharge}$ if it is a recharge action. This autoregressive process begins with an empty sequence and continues generating actions until mission completion, enabling coherent co-planning through modeling of temporal dependencies and environmental context.

Specifically, at each decoding step t , the decoder network ϕ first projects each action a_i in the current action sequence and its positional index i into a feature vector \mathbf{x}_i , where $i \leq t-1$. This process is defined as $\mathbf{x}_i = \mathbf{E}^a[a_i] + \mathbf{E}^i[i]$, where \mathbf{E}^a and \mathbf{E}^i are learnable embedding matrices for actions and positional

indices, respectively. To generate coherent actions for the robot team, we apply a causal self-attention mechanism to model temporal dependencies, which ensures that the generation of action a_t only attends to the previous actions $\{a_1, \dots, a_{t-1}\}$. The network ϕ projects the feature vector \mathbf{x}_i into query, key, and value embeddings, which are defined as follows:

$$\mathbf{q}_i^{l,o} = \mathbf{W}_q^{l,o} \mathbf{x}_i^l, \mathbf{k}_i^{l,o} = \mathbf{W}_k^{l,o} \mathbf{x}_i^l, \mathbf{v}_i^{l,o} = \mathbf{W}_v^{l,o} \mathbf{x}_i^l \quad (6)$$

where $\mathbf{q}_i^{l,o}, \mathbf{k}_i^{l,o}$, and $\mathbf{v}_i^{l,o}$ denote query, key, and value at the l -th layer for each action a_i , $\mathbf{W}_q^{l,o}, \mathbf{W}_k^{l,o}$, and $\mathbf{W}_v^{l,o}$ denote their learnable weight matrices. Given these embeddings, the self-attention score is computed as follows:

$$\gamma_{i,j}^l = \text{SoftMax} \left(\frac{(\mathbf{q}_i^{l,o})^\top \mathbf{k}_j^{l,o}}{\sqrt{d}} \right) \quad (7)$$

where $\gamma_{i,j}^l$ denotes the attention score from action a_j to a_i .

To enable context-aware action generation for collaborative planning, we condition each action on the graph embeddings from \mathcal{H} , which includes the contextual information over the environment. Specifically, the network ϕ projects the action feature \mathbf{x}_i into query and graph embedding \mathbf{h}_k into key and value, which is defined as follows:

$$\mathbf{q}_i^{l,c} = \mathbf{W}_q^{l,c} \mathbf{x}_i^l, \mathbf{k}_k^{l,c} = \mathbf{W}_k^{l,c} \mathbf{h}_k, \mathbf{v}_k^{l,c} = \mathbf{W}_v^{l,c} \mathbf{h}_k \quad (8)$$

where $\mathbf{q}_i^{l,c}, \mathbf{k}_k^{l,c}$, and $\mathbf{v}_k^{l,c}$ denote query, key, and value at the l -th layer, $\mathbf{W}_q^{l,c}, \mathbf{W}_k^{l,c}$, and $\mathbf{W}_v^{l,c}$ denote their learnable weight matrices. Given these embeddings, the cross-attention score is computed as follows:

$$\delta_{i,k}^l = \text{SoftMax} \left(\frac{(\mathbf{q}_i^{l,c})^\top \mathbf{k}_k^{l,c}}{\sqrt{d}} \right) \quad (9)$$

We further define the hidden representation \mathbf{f}_i^l for each action a_i , which serves as its contextual embedding at the l -th decoder layer. Starting with $\mathbf{f}_i^0 = \mathbf{x}_i$, the hidden representation is iteratively updated by aggregating both self-attention and cross-attention, defined as follows:

$$\mathbf{f}_i^{l+1} = \text{LN} \left(\mathbf{f}_i^l + \sum_{j \leq i} \gamma_{i,j}^l \mathbf{v}_j^{l,o} + \sum_{k \in \mathcal{H}} \delta_{i,k}^l \mathbf{v}_k^{l,c} \right) \quad (10)$$

where the second term captures temporal dependencies among actions through self-attention, and the third incorporates contextual information from the graph embeddings through cross-attention. Finally, the network ϕ projects the final hidden representation \mathbf{f}_{t-1}^L from the last layer into action logits by $\mathbf{u}_t = \mathbf{W}^u \mathbf{f}_{t-1}^L$, where \mathbf{W}^u denotes the learnable matrix. The action probability distribution conditioned on the graph embedding is then computed as $P(a_t | a_1, a_2, \dots, a_{t-1}, \mathcal{H}) = \text{SoftMax}(\mathbf{u}_t)$ and the action is then selected greedily as:

$$a_t = \arg \max_{a_t} P(a_t | a_1, a_2, \dots, a_{t-1}, \mathcal{H}) \quad (11)$$

The selected action a_t is appended to the action history, forming $\{a_1, a_2, \dots, a_t\}$, and the decoder uses this new sequence to generate the next action a_{t+1} . This autoregressive process continues until all tasks are completed.



Fig. 3. Qualitative results of CoPCS in a map-based simulator. The first four figures illustrate a mission with 15 tasks and 5 paths, while the last four figures show a mission with 45 tasks and 10 paths. The experiments evaluate different team configurations, ranging from 1 UAV and 1 UGV to 4 UAVs and 2 UGVs. Task points are depicted as white circles, recharging points as green circles, UAV trajectories as colored lines, and UGV paths as black lines.

TABLE I

QUANTITATIVE RESULTS UNDER DIFFERENT MISSION REQUIREMENTS ACROSS OPERATIONALLY CONSTRAINED UAV-UGV TEAM CONFIGURATIONS.

Mission	Method	1UAV-1UGV			2UAV-1UGV			2UAV-2UGV			4UAV-2UGV		
		Makespan	UAV energy	UGV energy	Makespan	UAV energy	UGV energy	Makespan	UAV energy	UGV energy	Makespan	UAV energy	UGV energy
T15-P5	GLS [13]	8662.71	1050.57	13262.04	7108.87	1115.01	13472.43	6319.81	1068.90	21972.07	5522.37	1322.88	24697.03
	TS [22]	9086.49	1010.59	13906.42	6390.69	1059.31	12872.67	8119.87	1328.26	19737.92	6770.71	1389.42	28746.32
	SA [23]	9078.41	1032.61	14639.16	6860.22	1177.55	14502.63	7125.12	1320.02	17222.68	5238.74	1321.54	18715.35
	MLP [40]	6936.44	934.34	10927.01	4043.61	1070.81	8512.61	4983.65	1167.34	12353.34	3182.47	1327.13	11051.53
	CoPCS	6617.75	875.92	10016.21	3477.51	934.75	8251.61	3465.41	965.06	11511.28	2614.08	1180.29	8569.27
T45-P10	GLS [13]	15073.27	1836.16	22369.79	12628.09	2385.66	23615.11	8884.95	2136.77	22014.14	8143.29	2728.93	29281.86
	TS [22]	16160.50	1978.89	24510.74	10543.42	2387.07	21277.91	11110.28	2251.52	22086.83	9617.13	2901.18	32365.98
	SA [23]	18075.23	2116.41	25632.18	16719.52	2692.45	37677.11	10507.05	2444.45	21901.16	9536.52	2626.84	33265.91
	MLP [40]	14326.98	2295.39	12793.25	8213.82	1886.78	11854.86	8351.97	1808.43	16052.93	6394.29	2224.99	17242.63
	CoPCS	11900.02	1744.71	9707.79	6659.41	1719.32	11249.41	6379.37	1769.12	11742.68	5973.65	2098.53	15077.62

D. Unified Training and Concurrent Execution

Since our approach requires learning complex collaborative policies for heterogeneous robots, we need substantial training data covering diverse mission scenarios. Manual labeling of optimal action sequences for task visiting, path movement, and recharging across all scenarios is impractical. Therefore, we adopt imitation learning using demonstrations from a Mixed Integer Programming (MIP) solver that generates optimal solutions for each training scenario. The MIP solver formulates the UAV-UGV collaboration as a constrained optimization problem that enables multi-UAV task planning and multi-UGV path planning. The MIP solver therefore generates optimal action sequences as ground truth actions $\{a_1^*, a_2^*, \dots, a_t^*\}$. See Appendix for details on the MIP mathematical formulations. While the MIP solver provides optimal demonstrations, it has exponential time complexity and cannot execute in real time for mission planning.

Given the demonstration as ground truth, we train the network ψ and ϕ in a unified way using an autoregressive training algorithm with teacher forcing. Specifically, the algorithm receives a 2D occupancy map of the environment including the task points \mathcal{M} and path points \mathcal{P} , as well as the UAVs \mathcal{U}^a and UGVs \mathcal{U}^g along with their attributes \mathbf{t}_i . The network ψ processes the \mathbf{t}_i into embeddings \mathcal{H} . The network ϕ generates the action sequence conditioned on the embedding \mathcal{H} . At each step t , the decoder receives the ground truth previous action $\{a_1^*, a_2^*, \dots, a_{t-1}^*\}$ to predict the probability distribution over all possible actions, and then generates the action a_t . The training objective is to minimize the cross-entropy loss between predicted and ground truth actions, which is defined as follows:

$$\mathcal{L} = - \sum_{t=1}^T \log P(a_t = a_t^* | a_1^*, a_2^*, \dots, a_{t-1}^*, \mathcal{H}) \quad (12)$$

where the loss is minimized using Adam as the optimization solver [41]. The gradients computed from this loss are

backpropagated through the transformer decoder ϕ to the heterogeneous graph transformer ψ , which enables unified end-to-end training of all learnable parameters for context-aware embedding generation and sequential action prediction.

During execution, our approach assumes a 2D occupancy map of the environment, including tasks, paths, UAVs, and UGVs, consistent with the training setup. The heterogeneous graph transformer network generates embeddings for each object, providing constraint-aware representations that enable synchronized collaboration within the UAV-UGV team. Based on these embeddings, the context-aware transformer decoder produces a joint action sequence that ensures coherent co-planning of UAVs and UGVs. With these preconditions, this joint sequence is then partitioned into UAV-specific and UGV-specific action sequences, which are executed concurrently, allowing both vehicle types to operate in parallel and achieve synchronized concurrent co-planning.

IV. EXPERIMENTS

A. Experimental Setups

We comprehensively evaluate CoPCS in three setups: a map-based 2D simulator, a Unity 3D multi-robot simulator with ROS1, and physical robot teams with ROS2. Evaluations span multiple team configurations (from 1 UAV-1 UGV to 4 UAV-2 UGV) and varying mission complexities, defined by the number of tasks and available UGV paths (15 tasks/5 paths and 45 tasks/10 paths). Simulations experiments are conducted within a $10\text{km} \times 10\text{km}$ operational area. UAVs operate at a constant speed of $q_a = 10$ m/s, while UGVs traverse road networks at $q_g = 4.5$ m/s. Each UAV has an energy capacity of 287.7 kJ, while each UGV has an energy capacity of 36,810 kJ. The energy consumption of UAVs and UGVs follows the robot profiles defined in [42], expressed as $0.0461(q_a)^3 - 0.5834(q_a)^2 - 1.8761q_a + 229.6$



Fig. 4. Qualitative results of co-planning with concurrent synchronization for a team of two Warthog UGVs and two quadrotor UAVs in a high-fidelity Unity-based 3D simulator integrated with ROS1. The mission includes 45 tasks and 10 traversable paths. The first column of the two subfigures illustrates the team starting configuration and a synchronized recharging event during execution, while the remaining subfigures depict trajectories at different timestamps.



Fig. 5. Qualitative results illustrating the generalizability of CoPCS in unseen environments. The large-scale suburban and urban maps are not used during training and also contain variations in starting points and road network layouts. In the top environment, a 2-UAV and 2-UGV team is required to visit 15 tasks, while in the bottom environment, a 4-UAV and 2-UGV team is required to visit 15 tasks.

and $464.8q_g + 356.3$, respectively. In each scenario, the UAV-UGV team begins from a common depot. Energy-constrained UAVs are tasked with visiting the designated task points, while traversability-constrained UGVs provide recharging support along a road network. All video demonstrations are provided on our project website. See Appendix for implementation details of CoPCS.

We generate synthetic datasets to train and evaluate our approach under varying team configurations and mission requirements. The differences between datasets lie in the number of robots, the number of tasks to be completed, and the number of paths available. Each dataset contains 1,000 problem instances, with task positions randomly generated. Within each dataset, 800 instances are used for training, while the remaining 200 instances are reserved for testing and evaluation.

We compare our CoPCS approach with three metaheuristic methods which focus on fixed UGV path and a baseline method: (1) Guided Local Search (GLS) [13] that improves UAV task routes by penalizing frequently used task assignments to escape local optima and explore alternative routing strategies, (2) Tabu Search (TS) [22] that explores UAV task routes while maintaining memory of recent moves to prevent cycling back to previous solutions, (3) Simulated Annealing (SA) [23] that initially accepts suboptimal UAV task routes with high probability, then gradually becomes more selective as the search converges, and (4) Multilayer Perceptron (MLP)

[40] that uses the same transformer network ϕ for sequential planning, but uses a multilayer perceptron to generate the embeddings \mathcal{H} . To quantitatively evaluate and compare our approach with baseline methods, we employ four metrics for collaborative planning. (1) **Makespan** [43] is defined as the total mission completion time (i.e., the time required to visit all task points) measured in seconds (s), (2) **UAV energy** is the cumulative energy consumption of all UAVs measured in kilojoules (kJ), (3) **UGV energy** is the cumulative energy consumption of all UGVs measured in kilojoules (kJ).

B. Results in Map-Based Robotics Simulations

We first evaluate CoPCS in map-based simulations that mimic real-world rural environments, exploring its capability for collaborative planning under different mission requirements and UAV-UGV team configurations.

The qualitative results in the map-based 2D simulator are shown in Figure 3. The results demonstrate that CoPCS enables operationally constrained UAV-UGV teams to collaborate effectively and accomplish the environmental monitoring task. We noticed that CoPCS introduces synchronized concurrent co-planning, allowing UAVs and UGVs to operate in parallel for continuous execution while ensuring synchronization at recharging points, as illustrated by the recharging behavior at the 210-second timestamp. It also supports multi-UAV task planning by determining which tasks each UAV should visit and in what order, reducing makespan under energy constraints.

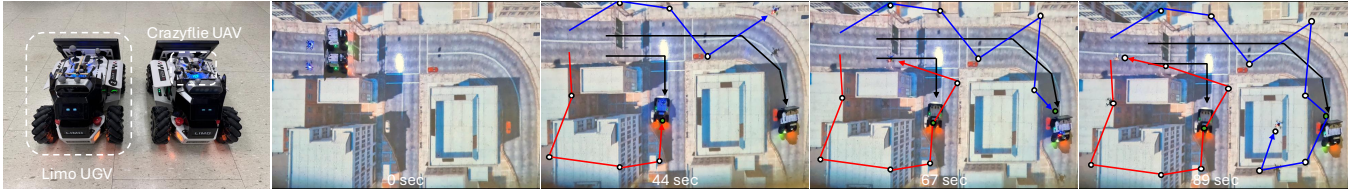


Fig. 6. Qualitative results of CoPCS in real-world mixed-reality experiments using two Crazyfly UAVs and two Limo UGVs controlled through ROS2.

In addition, CoPCS coordinates multi-UGV path planning over road networks while respecting traversability constraints.

The quantitative results are presented in Table I. For the mission with 15 tasks and 5 paths, the three metaheuristic baselines deliver the weakest performance, particularly in makespan and UGV energy consumption, as they optimize only UAV task planning while relying on fixed UGV paths, neglecting synchronized collaboration. This leads to inefficient UGV routes, higher energy usage, while UAVs require more time for recharging. The MLP baseline performs better, as it considers UAV-UGV co-planning, but its simple architecture limits its ability to capture relationships among robots, resulting in longer makespan and higher energy consumption. In contrast, CoPCS jointly plans UGV paths with UAV trajectories under synchronized collaboration, leading to the lowest energy consumption and the shortest makespan. When the mission scales to 45 tasks and 10 paths, the performance gap between CoPCS and the baselines widens substantially. The metaheuristic methods perform the worst, the MLP baseline improves slightly but remains limited. CoPCS consistently outperforms all baselines across all team configurations, achieving shorter makespan and lower energy consumption in this larger and more complex mission by enabling synchronized concurrent planning.

C. Results Using High-Fidelity 3D Simulations with ROS1

We validate CoPCS in a high-fidelity Unity-based 3D simulator integrated with ROS1 for robot control. Qualitative results are shown in Figure 4. The simulator replicates outdoor field environments for environmental monitoring tasks. The experiments involve two wheeled Warthog UGVs constrained by terrain features such as water and buildings, and two quadrotor UAVs limited by energy capacity. The team configuration is illustrated in the first subfigure. CoPCS enables UAVs and UGVs to operate in parallel: the UAV team allocates tasks and determines visit sequences, while the UGVs coordinate to plan feasible routes. More importantly, the UAV-UGV team achieves concurrent synchronization, allowing timely recharging and ensuring successful mission completion. One such recharging event is shown in the second subfigure. These results demonstrate the effectiveness of CoPCS in handling collaborative planning for UAV-UGV teams.

We further evaluate the generalizability of CoPCS in unseen environments, as illustrated in Figure 5. CoPCS is trained on datasets generated from the rural environment shown in Figure 4, and its generalization is assessed on two additional maps. The first map represents a suburban setting where a 2UAV-2UGV team is required to visit 15 tasks, while the

second map depicts an urban environment where a 4UAV-2UGV team must complete 15 tasks. These maps present distinct scenarios with different starting points and road network layouts. In both cases, CoPCS enables the team to successfully and efficiently visit all task points under operational constraints. Specifically, in the suburban map, the two UGVs coordinate to plan supportive paths in different directions to recharge UAVs, while in the urban map, the UAVs effectively plan their task visits to maximize efficiency. Overall, the results highlight the capability of our approach to generalize to diverse mission scenarios without the need for retraining.

D. Case Studies on Physical UAV-UGV Teams with ROS2

We validate CoPCS in real-world case studies using a UAV-UGV team composed of two Crazyfly UAVs and two Limo UGVs. The team operates in a mixed-reality setup, where a projector renders a simulated outdoor city scene onto the physical environment and a motion capture system tracks robot poses as well as task and path locations. The robot team is centrally controlled through ROS2 by a master workstation equipped with an NVIDIA RTX 4090 GPU and 64 GB RAM. Qualitative results are shown in Figure 6. The task points are distributed across the map, including non-traversable regions such as buildings for the Limo robots. The third and fourth subfigures illustrate recharge events of the team. In this setup, the CoPCS policy achieves an inference rate of 9 Hz, which is sufficient for our real-time planning requirements. Overall, the results demonstrate the effectiveness and applicability of our approach in enabling physical robot teams to achieve collaborative planning with concurrent synchronization.

V. CONCLUSION

In this paper, we propose CoPCS, a framework for collaborative planning with concurrent synchronization in operationally constrained UAV-UGV teams. CoPCS builds on a unified learning framework that integrates a heterogeneous graph transformer to encode robots and tasks under constraints for synchronization, and a context-aware transformer decoder for co-planning, enabling UAVs and UGVs to act concurrently in a coordinated manner. The framework is trained end-to-end under an imitation learning paradigm. Results from comprehensive experiments validate the effectiveness of CoPCS, demonstrate that our CoPCS approach enables the new multi-robot capability for constrained UAV-UGV teams of synchronized concurrent co-planning, and significantly outperforms existing methods.

REFERENCES

- [1] P. Gao, S. Siva, A. Micciche, and H. Zhang, "Collaborative scheduling with adaptation to failure for heterogeneous robot teams," in *IEEE International Conference on Robotics and Automation*, 2023.
- [2] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, et al., "Argos: A modular, parallel, multi-engine simulator for multi-robot systems," *Swarm Intelligence*, vol. 6, pp. 271–295, 2012.
- [3] D. Portugal and R. P. Rocha, "Multi-robot patrolling algorithms: examining performance and scalability," *Advanced Robotics*, vol. 27, no. 5, pp. 325–336, 2013.
- [4] M. Erdelj and E. Natalizio, "Uav-assisted disaster management: Applications and open issues," in *International Conference on Computing, Networking and Communications*, 2016.
- [5] A. Khan, S. Gupta, and S. K. Gupta, "Emerging uav technology for disaster detection, mitigation, response, and preparedness," *Journal of Field Robotics*, vol. 39, no. 6, pp. 905–955, 2022.
- [6] M. Erdelj, M. Król, and E. Natalizio, "Wireless sensor networks and multi-uav systems for natural disaster management," *Computer Networks*, vol. 124, pp. 72–86, 2017.
- [7] S. Asadzadeh, W. J. de Oliveira, and C. R. de Souza Filho, "Uav-based remote sensing for the petroleum industry and environmental monitoring: State-of-the-art and perspectives," *Journal of Petroleum Science and Engineering*, vol. 208, p. 109633, 2022.
- [8] K. Liu and J. Zheng, "Uav trajectory optimization for time-constrained data collection in uav-enabled environmental monitoring systems," *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 24 300–24 314, 2022.
- [9] A. Fascista, "Toward integrated large-scale environmental monitoring using wsn/uav/crowdsensing: A review of applications, signal processing, and future perspectives," *Sensors*, vol. 22, no. 5, p. 1824, 2022.
- [10] R. Eskandari, M. Mahdianpari, F. Mohammadimanesh, B. Salehi, B. Brisco, and S. Homayouni, "Meta-analysis of unmanned aerial vehicle (uav) imagery for agro-environmental monitoring using machine learning and statistical models," *Remote Sensing*, vol. 12, no. 21, p. 3511, 2020.
- [11] C. C. Murray and A. G. Chu, "The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 86–109, 2015.
- [12] Z. Liang, Y. Dong, and D. Zhang, "An algorithm combining deep reinforcement learning and dynamic programming for solving tsp-d," in *IEEE Smart World Congress*, 2024.
- [13] F. Arnold and K. Sörensen, "Knowledge-guided local search for the vehicle routing problem," *Computers & Operations Research*, vol. 105, pp. 32–46, 2019.
- [14] J. Luo, C. Li, Q. Fan, and Y. Liu, "A graph convolutional encoder and multi-head attention decoder network for tsp via reinforcement learning," *Engineering Applications of Artificial Intelligence*, vol. 112, p. 104848, 2022.
- [15] M. A. Javarone, "An evolutionary strategy based on partial imitation for solving optimization problems," *Physica A: Statistical Mechanics and its Applications*, vol. 463, pp. 262–269, 2016.
- [16] M. S. Mondal, S. Ramasamy, L. Russo, J. D. Humann, J. M. Dotterweich, and P. Bhounsule, "How to coordinate uavs and ugvs for efficient mission planning? optimizing energy-constrained cooperative routing with a drl framework," 2025.
- [17] S. G. Manyam, K. Sundar, and D. W. Casbeer, "Cooperative routing for an air-ground vehicle team—exact algorithm, transformation method, and heuristics," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 1, pp. 537–547, 2019.
- [18] L. Di Puglia Pugliese and F. Guerriero, "Last-mile deliveries by using drones and classical vehicles," in *International Conference on Optimization and Decision Science*, 2017.
- [19] G. Climaco, L. Simonetti, and I. Rosseti, "A branch-and-cut and mip-based heuristics for the prize-collecting travelling salesman problem," *RAIRO-Operations Research*, vol. 55, pp. S719–S726, 2021.
- [20] W. Kool, H. van Hoof, J. Gromicho, and M. Welling, "Deep policy dynamic programming for vehicle routing problems," in *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, 2022.
- [21] G. Lera-Romero, J. J. Miranda Bront, and F. J. Soullignac, "Dynamic programming for the time-dependent traveling salesman problem with time windows," *INFORMS Journal on Computing*, vol. 34, no. 6, pp. 3292–3308, 2022.
- [22] M. Gmira, M. Gendreau, A. Lodi, and J.-Y. Potvin, "Tabu search for the time-dependent vehicle routing problem with time windows on a road network," *European Journal of Operational Research*, vol. 288, no. 1, pp. 129–140, 2021.
- [23] E. Yağmur and S. E. Kesen, "Multi-trip heterogeneous vehicle routing problem coordinated with production scheduling: Memetic algorithm and simulated annealing approaches," *Computers & Industrial Engineering*, vol. 161, p. 107649, 2021.
- [24] P. Maini and P. Sujit, "On cooperation between a fuel constrained uav and a refueling ugv for large scale mapping applications," in *International Conference on Unmanned Aircraft Systems*, 2015.
- [25] F. Ropero, P. Munoz, and M. D. R-Moreno, "Terra: A path planning algorithm for cooperative ugv-uav exploration," *Engineering Applications of Artificial Intelligence*, vol. 78, pp. 260–272, 2019.
- [26] M. S. Mondal, S. Ramasamy, J. D. Humann, J.-P. F. Reddinger, J. M. Dotterweich, M. A. Childers, and P. Bhounsule, "Optimizing fuel-constrained uav-ugv routes for large scale coverage: Bilevel planning in heterogeneous multi-agent systems," in *International Symposium on Multi-Robot and Multi-Agent Systems*, 2023.
- [27] M. Prates, P. H. Avelar, H. Lemos, L. C. Lamb, and M. Y. Vardi, "Learning to solve np-complete problems: A graph neural network for decision tsp," in *AAAI Conference on Artificial Intelligence*, 2019.
- [28] M. Boffa, Z. B. Houidi, J. Krolkowski, and D. Rossi, "Neural combinatorial optimization beyond the tsp: Existing architectures under-represent graph structure," *arXiv*, 2022.
- [29] P. Hu, "Travel the same path: A novel tsp solving strategy," *arXiv*, 2022.
- [30] W. Kool, H. Van Hoof, and M. Welling, "Attention, learn to solve routing problems!" *arXiv*, 2018.
- [31] Z. Zhang, G. Qi, and W. Guan, "Coordinated multi-agent hierarchical deep reinforcement learning to solve multi-trip vehicle routing problems with soft time windows," *IET Intelligent Transport Systems*, vol. 17, no. 10, pp. 2034–2051, 2023.
- [32] D. Fuertes, C. R. del Blanco, F. Jaureguizar, J. J. Navarro, and N. Garcia, "Solving routing problems for multiple cooperative unmanned aerial vehicles using transformer networks," *Engineering Applications of Artificial Intelligence*, vol. 122, p. 106085, 2023.
- [33] M. L. Betalo, S. Leng, A. M. Seid, H. N. Abishu, A. Erbad, and X. Bai, "Dynamic charging and path planning for uav-powered rechargeable wsns using multi-agent deep reinforcement learning," *IEEE Transactions on Automation Science and Engineering*, 2025.
- [34] M. Fan, Y. Wu, T. Liao, Z. Cao, H. Guo, G. Sartoretti, and G. Wu, "Deep reinforcement learning for uav routing in the presence of multiple charging stations," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 5, pp. 5732–5746, 2022.
- [35] S. Brotee, F. Kabir, M. A. Razzaque, P. Roy, M. Mamun-Or-Rashid, M. R. Hassan, and M. M. Hassan, "Optimizing uav-ugv coalition operations: A hybrid clustering and multi-agent reinforcement learning approach for path planning in obstructed environment," *Ad Hoc Networks*, vol. 160, p. 103519, 2024.
- [36] M. S. Mondal, S. Ramasamy, R. Rownak, L. Russo, J. D. Humann, J. M. Dotterweich, and P. Bhounsule, "Risk-aware energy-constrained uav-ugv cooperative routing using attention-guided reinforcement learning," in *IEEE International Conference on Robotics and Automation*, 2025.
- [37] M. S. Mondal, S. Ramasamy, J. D. Humann, J. M. Dotterweich, J.-P. F. Reddinger, M. A. Childers, and P. Bhounsule, "An attention-aware deep reinforcement learning framework for uav-ugv collaborative route planning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2024.
- [38] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *International World Wide Web Conference*, 2020.
- [39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [40] J. Park, C. Kwon, and J. Park, "Learn to solve the min-max multiple traveling salesmen problem with reinforcement learning," in *International Conference on Autonomous Agents and Multiagent Systems*, 2023.
- [41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv*, 2017.
- [42] A. M. Hurwitz, J. M. Dotterweich, and T. A. Rocks, "Mobile robot battery life estimation: battery energy use of an unmanned ground vehicle," in *Energy Harvesting and Storage: Materials, Devices, and Applications XI*, 2021.
- [43] F. Kudo and K. Cai, "A tsp-based online algorithm for multi-task multi-agent pickup and delivery," *IEEE Robotics and Automation Letters*, vol. 8, no. 9, pp. 5910–5917, 2023.