MoRoCo: Topology-Adaptive Human-Fleet Coordination under Restricted Communication

Zhuoli Tian, Yanze Bao, Yuyang Zhang and Meng Guo

***Abstract*—Autonomous robot teams are increasingly deployed with human operators in communication-restricted environments for exploration and intervention tasks such as search and rescue, inspection, and reconnaissance. In these settings, short-range and unreliable links make it difficult to sustain collaborative exploration and support timely operator interaction, including status updates, video access, and teleoperation. We present MoRoCo, a topology-adaptive framework for human-multi-robot coordination under restricted communication. MoRoCo builds on a latency-bounded intermittent communication backbone that guarantees timely delivery of robot-collected information to operators, while allowing the team to reconfigure online through a detach-and-rejoin mechanism. On top of this backbone, the framework instantiates request-consistent communication topologies to support different interaction modes, including sustained operator-to-robot access and inter-team information exchange. This design enables online request handling without abandoning exploration. We validate MoRoCo in human-in-the-loop simulations and hardware experiments, showing reliable coordination, timely operator updates, and effective topology adaptation for communication-demanding tasks in unknown environments.**

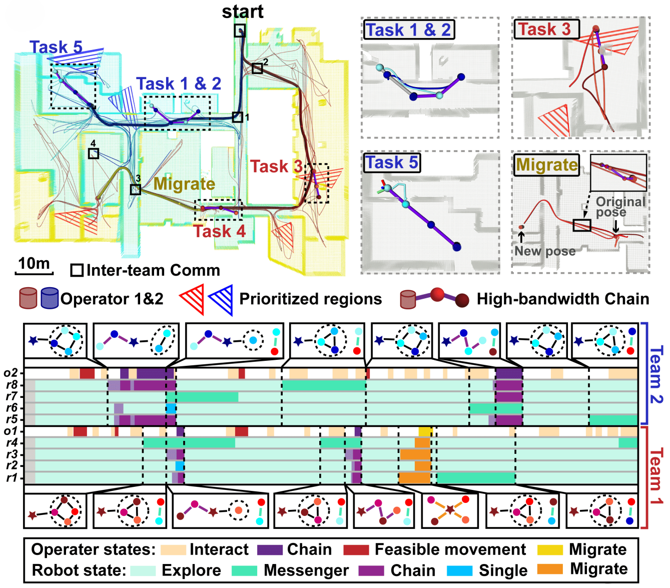


Fig. 1: Illustration of the considered scenario, where multiple operators and robots are deployed in an unknown and communication-restricted environment for exploration and task execution. In this experiment, 2 operators and 8 robots collaboratively explore a large-scale environment, while accomplishing 5 tasks via splitting and merging embedded communication graphs.

I. INTRODUCTION

Autonomous robots such as unmanned aerial vehicles (UAVs) and ground vehicles (UGVs) are increasingly deployed with human operators in hazardous and communication-degraded environments for tasks such as search and rescue, reconnaissance, and inspection [7, 13]. In these settings, robots extend human perception and action into areas that are difficult or unsafe to access directly. However, unlike standard multi-robot exploration settings, communication is often unavailable globally and instead limited to short-range local links. As a result, operator–robot and robot–robot communication cannot be assumed throughout execution, which makes it fundamentally difficult to maintain both efficient exploration and effective human supervision.

This challenge is particularly critical in human-in-the-loop deployments. During exploration, operators need timely updates about mission progress, robot status, and newly explored regions, but such information may not be available unless communication is coordinated. Operators may also require direct access to a specific robot for close-up visual inspection, video streaming, or teleoperation, while robots may need confirmation or assistance from an operator when facing uncertain observations or difficult terrain. These requests impose heterogeneous communication requirements: some only require bounded-delay information return, whereas others require sustained and reliable communication during motion. Supporting such interactions under restricted communication therefore requires the team to adapt its communication topology online rather than relying on a fixed exploration structure.

Existing work only partially addresses this setting. Many multi-robot exploration methods assume persistent or effectively global communication, so that local observations can be immediately fused and shared across the fleet [12, 6, 19, 17]. This assumption is often unrealistic in subterranean, indoor, or communication-restricted environments, where communication quality can be severely degraded by range limitations and occlusions [11, 14, 10, 1]. More recent methods explicitly consider communication-aware exploration and intermittent communication [13, 18, 4, 16], but they primarily focus on information gathering and usually treat the human as a static base station or a visualization endpoint. Conversely, studies on human–fleet interaction emphasize supervision, teleoperation, and richer operator interfaces [15, 9, 5], but typically rely on reliable operator connectivity. As a result, there remains a gap between communication-constrained exploration and communication-constrained human–fleet coordination.

The authors are with the College of Engineering, Peking University, Beijing 100871, China. Contact: `meng.guo@pku.edu.cn`, `zl.tian@stu.pku.edu.cn`.

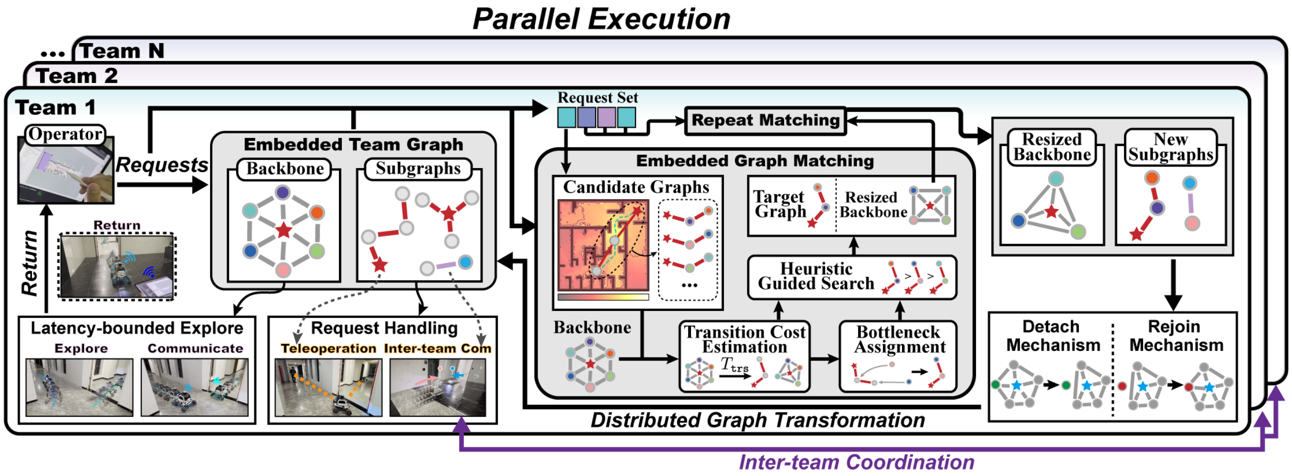


Fig. 2: Overall framework of MOROCO for multi-team human–robot coordination under restricted communication. Multiple teams of operator and robots simultaneously explore the environment and fulfill several online requests, following the embedded communication graphs under the intermittent communication protocol (**Left**); The embedded graph is optimized and matched against different targeted graphs given the current graph and the set of active requests (**Middle**); The resulting subgraphs are realized by the proposed detach-and-rejoin mechanism via only local coordination (**Right**).

To address this gap, we present **MoRoCo**, a topology-adaptive framework for multi-operator multi-robot coordination under restricted communication. MoRoCo is built on a latency-bounded intermittent communication backbone that guarantees timely delivery of robot-collected information to operators using only local communication. On top of this backbone, it enables online topology adaptation through a detach-and-rejoin mechanism, allowing subsets of robots to temporarily leave the exploration backbone, realize request-specific communication structures, and later rejoin the team. This mechanism is further coupled with a request-consistent graph instantiation procedure that supports multiple interaction modes, including bounded-latency exploration, sustained operator-to-robot access, and inter-team information exchange.

The contributions of this work are threefold. First, we formulate a human–multi-robot coordination problem that explicitly captures online operator–robot and robot–operator interactions under restricted communication. Second, we develop a topology-adaptive coordination framework that combines latency-bounded intermittent communication, collaborative exploration, and online request fulfillment through communication-topology reconfiguration. Third, we validate the proposed framework in human-in-the-loop simulations and hardware experiments, demonstrating reliable coordination and effective support for communication-demanding tasks in unknown environments.

II. PROBLEM DESCRIPTION

We consider a set of K operator–robot teams deployed in an unknown workspace $\mathcal{A} \subset \mathbb{R}^3$ for collaborative exploration and intervention. Each team $\mathcal{T}_k = \{\mathtt{h}_k\} \cup \mathcal{N}_k$ consists of one human operator $\mathtt{h}_k$ and a set of robots $\mathcal{N}_k$, with $\mathcal{N} = \cup_{k=1}^{K} \mathcal{N}_k$ the full robot set. The workspace map, free space, and obstacles are initially unknown. Each robot is equipped with onboard sensing and navigation, and incrementally builds a local map $M_i(t) \subseteq \mathcal{A}$ together with its pose $p_i(t) \in M_i(t)$. The operators do not build maps directly; instead, each operator $\mathtt{h}_k$ maintains a local situational map $M_k^h(t)$ that is updated through communication with the robots in its team.

Restricted communication. Communication is local and quality-dependent. Two agents u and v can directly communicate at time t only if their communication quality exceeds a threshold, i.e.,

$$\mathrm{Qual}(p_u(t), p_v(t), \mathcal{A}) > \delta, \tag{1}$$

where $\delta > 0$ is a minimum communication-quality requirement. This model applies to robot–robot, operator–robot, and operator–operator communication. We denote by $\mathcal{C}_i(t)$ the set of agents that robot i can directly communicate with at time t. In addition to pairwise exchanges, multiple agents can form a communication chain

$$i_0 \leftrightarrow i_1 \leftrightarrow \cdots \leftrightarrow i_L \leftrightarrow \mathtt{h}_k, \tag{2}$$

where every neighboring pair satisfies (1). Such chains are required for interaction modes that need sustained and reliable operator access, such as teleoperation or video streaming.

Online requests. During execution, both operators and robots may issue online requests that modify the communication and coordination requirements, which are grouped into three representative classes. (I) *Latency-preserving exploration*, where operators require bounded-delay updates from their team and may also specify regions to prioritize or avoid during exploration. (II) *Sustained operator–robot interaction*, where an operator or robot requests a persistent communication structure for tasks such as video inspection, teleoperation, confirmation, or direct assistance. (III) *Inter-team coordination and relocation*, where teams may request inter-team information exchange, operator movement, or temporary robot reassignment to support other teams or tasks. Let $\Xi(t)$ denote the set of currently active requests at time t. Some requests are persistent, such as bounded-latency operator updates, whereas others are completed once a target location is reached, a communication chain is established for a required duration, or an information-exchange task is fulfilled.

Latency requirement. A key requirement in our setting is that information collected by any robot in team $\mathcal{T}_k$ must reach operator $\mathtt{h}_k$ within a prescribed delay $T_{\mathrm{h}}^k > 0$. Formally, if robot $i \in \mathcal{N}_k$ observes new map information at time t, then this information should be reflected in the operator-side map no later than $t + T_{\mathrm{h}}^k$. This latency requirement captures the need for timely supervision even when persistent connectivity is unavailable.

Joint execution model. The behavior of each robot and operator can be viewed as a sequence of motion and communication events. Let Γ_i denote the execution of robot i and Γ_k^h that of operator $\mathtt{h}_k$. We write

$$\Gamma = \Big(\{\Gamma_i\}_{i\in\mathcal{N}}, \{\Gamma_k^h\}_{k=1}^K\Big) \tag{3}$$

for the joint execution of the full multi-team system. An execution is feasible if it respects motion and communication constraints, and if every active request in $\Xi(t)$ is eventually served by the evolving team topology and communication schedule.

Objective. Our overall objective is to coordinate all teams so that the workspace is explored as efficiently as possible while satisfying the online interaction requests and communication constraints. In compact form, we seek a feasible joint execution Γ and mission completion time T such that

$$\min_{\Gamma,T} \; T \tag{4}$$

subject to

$$\mathcal{A} \subseteq M_k^h(T), \qquad \forall k \in \{1,\ldots,K\}, \tag{5}$$

$$\Gamma \models \xi, \qquad \forall \xi \in \Xi(t),\ \forall t \geqslant 0. \tag{6}$$

That is, by time T, the explored environment should be available to all operators, and the execution must satisfy all active online requests throughout the mission.

The difficulty of this problem lies in the fact that future requests, robot encounters, and communication opportunities are not known a priori. As a result, the system must make online coordination decisions that preserve bounded-latency exploration while adapting the communication topology to support emerging operator–robot and inter-team interactions.

III. PROPOSED SOLUTION

As illustrated in Fig. 2, the proposed MoRoCo framework consists of three tightly coupled components. First, each team maintains a latency-bounded intermittent communication backbone that supports collaborative exploration while guaranteeing timely operator updates. Second, when an online request requires a communication pattern different from the exploration backbone, the team performs topology adaptation through a detach-and-rejoin mechanism. Third, an online coordination layer selects and instantiates request-consistent subgraphs so that multiple requests can be served without abandoning the ongoing mission. Together, these components realize the mission objective in Sec. II through a sequence of online feasible coordination decisions.

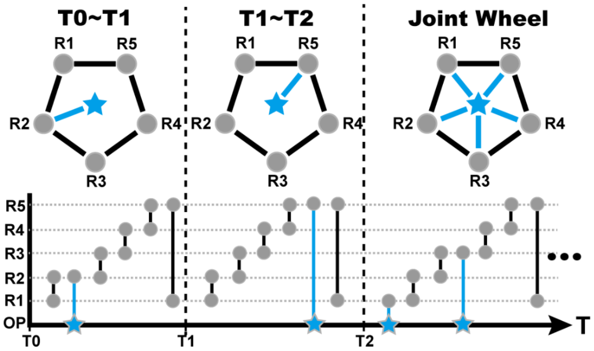


Fig. 3: Top: the joint-wheel communication graph over time; **Bottom:** the pairwise meeting events (black vertical lines) and return events (blue lines).

A. Latency-Bounded Communication Backbone

We begin with a single team $T_k = \{h_k\} \cup \mathcal{N}_k$ and omit the team index for brevity. The backbone is designed to ensure that information gathered by any robot reaches the operator within the prescribed latency bound T_h while exploration continues. Instead of assuming persistent connectivity, MoRoCo enforces this requirement through scheduled local communication events. The backbone is represented by an embedded joint-wheel graph

$$G^w = (V, E, \sigma, x), \tag{7}$$

where $V = \{h\} \cup \mathcal{N}$ contains the operator and all robots. This wheel structure is a coordination backbone rather than a continuously active communication graph: at any time only local pairwise meetings occur, but over a horizon longer than T_h they induce a wheel-like information flow that returns robot-collected data to the operator within bounded delay.

Each robot maintains a local plan Γ_i and a latency state χ_i. When two robots meet, they exchange their scheduling information, then jointly replan their next communication event while preserving the feasibility of a timely return to the operator. Denoting by t_{ij}^+ and p_{ij}^+ the next meeting time and location for robot pair (i, j), the update satisfies

$$t_{ij}^+ + T_\ell^{\mathrm{nav}}(p_{ij}^+, p_h) \leqslant T_h + \chi^*, \tag{8}$$

where $T_\ell^{\mathrm{nav}}(\cdot,\cdot)$ is the estimated travel time for a candidate return robot ℓ, p_h is the current operator position, and χ^* is the current guaranteed information timestamp after the local update. This condition ensures that after each pairwise meeting, the newly merged information still reaches the operator before the latency bound is violated. As a result, exploration and communication are coupled through local replanning, while the implementation remains distributed and the operator continues to receive periodic latency-bounded updates.

B. Topology Adaptation via Detach-and-Rejoin

The wheel backbone is suitable for bounded-latency exploration, but it cannot directly support all online interaction

Algorithm 1: Overall online execution of the MoRoCo framework `MoRoco`$(\cdot)$

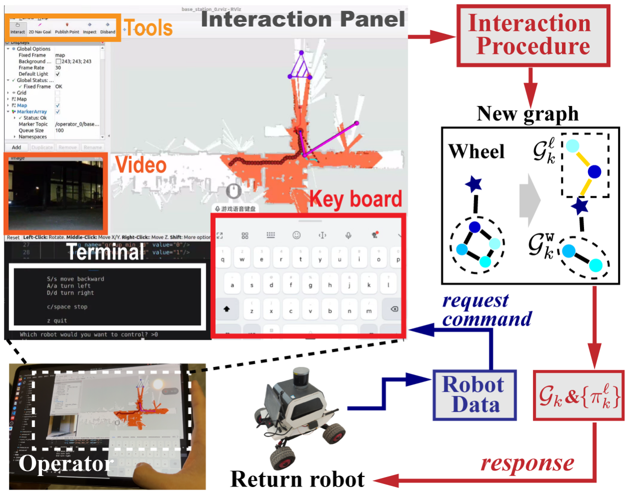
```
Input  : $\{\mathcal{G}^{\mathrm{w}}_{k,0}\}$, $\{\Gamma_i\}$.
Output: Local plans $\{\Gamma_i\}$ and evolving graphs $\{\mathcal{G}_k\}$.
while not terminated and all teams $\{\mathcal{T}_k\}$ in parallel do
    /* Within the wheel graph */
    Robot $i \in \mathcal{N}^{\mathrm{w}}_k$ executes $\Gamma_i$ in parallel;
    if meeting event $\mathbf{c}_{ij}$ occurs then
        $\Gamma_i, \Gamma_j, \mathbf{c}^{+}_{ij} \leftarrow$ PairCoord($\Gamma_i$, $\Gamma_j$, $\chi_{ij}$);
        Update embeddings of $\mathcal{G}^{\mathrm{w}}_k$;
    /* Request instantiation */
    if request bundle $\Xi_k$ received then
        Order requests in $\Xi_k$ by priority;
        foreach request $\xi^{\ell}_k \in \Xi_k$ do
            if $\xi^{\ell}_k$ admissible on $\mathcal{G}^{\mathrm{w}}_k$ then
                $\widetilde{\mathcal{G}}^{\mathrm{w}}_k$, $\widetilde{\mathcal{G}}^{\ell}_k$, $\widetilde{\pi}^{\ell}_k$, $\widetilde{\Pi}^{\ell}_k \leftarrow$
                  GraphMatch($\mathcal{G}^{\mathrm{w}}_k$, $\widehat{\mathcal{G}}^{\ell}_k$, $\xi^{\ell}_k$);
                if an executable instantiation then
                    $\mathcal{G}^{\mathrm{w}}_k \leftarrow \widetilde{\mathcal{G}}^{\mathrm{w}}_k$, add $\widetilde{\mathcal{G}}^{\ell}_k$ to $\mathcal{G}_k$;
                    Store $\widetilde{\pi}^{\ell}_k$ for detachment;
        Follow detach schedules and update $\mathcal{G}_k$;
    /* Request execution */
    foreach active request $\xi^{\ell}_k$ in parallel do
        All robots in $\mathcal{N}^{\ell}_k$ perform request $\xi^{\ell}_k$;
        if request $\xi^{\ell}_k$ is fulfilled then
            Trigger rejoin procedure and update $\mathcal{G}_k$;
```

modes. Requests such as teleoperation, sustained video streaming, direct assistance, operator relocation, or inter-team exchange may require different communication structures, such as a multi-hop line or a connectivity-preserving star. MoRoCo therefore adapts the team topology online through a detach-and-rejoin mechanism, so that communication-demanding interactions can be supported without abandoning the exploration backbone.

The key idea is to temporarily split the team into two parts. A subset of robots leaves the backbone to form a request-specific subgraph, while the remaining robots continue exploration on a resized backbone that still satisfies the latency requirement. Once the request is completed, the detached robots rejoin and the default exploration structure is restored. To make this process robust under intermittent communication, both detachment and rejoining are executed through local handoffs rather than global rewiring: before a robot departs, its neighbors establish a bypass schedule that keeps the resized backbone executable, and rejoining follows the same principle in reverse through local synchronization and successor takeover. This two-step rewiring allows request execution and exploration to proceed concurrently while preserving bounded-latency operator supervision.

C. Request-Consistent Graph Instantiation

Given the current backbone state and an incoming request ξ, MoRoCo constructs a request-consistent target topology and computes how to realize it from the current team configuration.



```
S/s move backward
A/a turn left
D/d turn right

c/space stop

z quit
Which robot would you want to control? >0
```

Fig. 4: Illustration of the procedure on how the operator can interact with its team via the panels within the GUI, and the type of data exchanged during the return event including the gathered information and human commands.

Let G^{w}_k denote the current wheel backbone of team k, and let $\mathcal{G}_k(\xi)$ denote the family of candidate target graphs for request ξ. For each candidate graph $G^{\xi}_k \in \mathcal{G}_k(\xi)$, let $\mathcal{N}^{\xi}_k$ be the robots allocated to the request, let $G^{w'}_k$ be the resized backbone after removing these robots, and let Π^{ξ}_k denote the assignment from detached robots to target roles. MoRoCo selects the executable transition that balances backbone resizing time and request-realization time:

$$\min_{G^{w'}_k, G^{\xi}_k, \Pi^{\xi}_k} \; T^{\mathrm{trs}}_k(G^{w}_k, G^{w'}_k) + w_n \max_{r \in \mathcal{N}^{\xi}_k} T^{\mathrm{nav}}_r\big(\pi^{\xi}_r, \Pi^{\xi}_k(r)\big), \quad (9)$$

where $T^{\mathrm{trs}}_k(\cdot,\cdot)$ is the time to realize the resized backbone, π^{ξ}_r is the detach event of robot r, and $T^{\mathrm{nav}}_r(\cdot,\cdot)$ is the travel time from detachment to the assigned role.

In practice, this graph-matching layer first determines which robots can be detached while keeping the resized backbone feasible, then assigns detached robots to target roles, and finally selects the candidate with the best trade-off in (9). The output is an executable plan consisting of a resized backbone, a request-specific subgraph, and the associated detach events. This representation covers the main request types in Sec. II: bounded-latency exploration remains on the wheel backbone, sustained operator–robot interaction is realized by a communication chain to a designated robot, and inter-team exchange or operator relocation is supported through detached messenger or migration subgraphs.

D. Online Coordination of Multiple Requests

During execution, requests arrive online and may overlap in time. MoRoCo handles this by maintaining a team-level embedded graph

$$G_k(t) = G^{w}_k(t) \cup \bigcup_{j=1}^{L_k} G^{j}_k(t), \quad (10)$$

where $G^{w}_k(t)$ is the current exploration backbone and each $G^{j}_k(t)$ is an instantiated subgraph serving one active request.

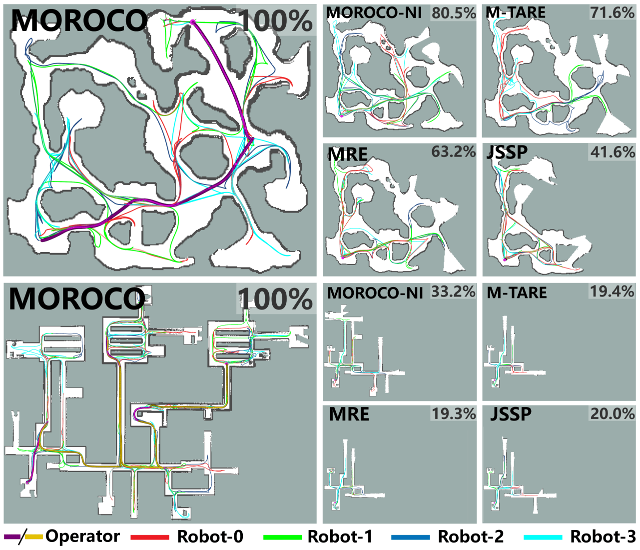


Fig. 5: Final maps obtained by the operator (**Left**), and the trajectories of the operator and all robots (**Right**) in the cave (**Top**) and indoor scenarios (**Bottom**), under the proposed method and four baselines.

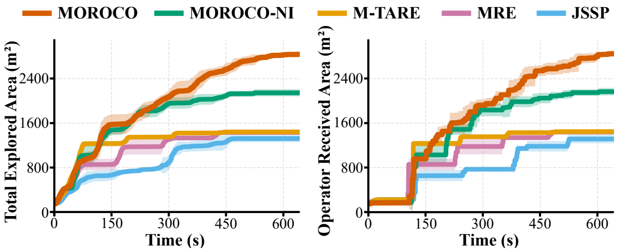


Fig. 6: Evolution of the explored area (**Left**) and the operator-received area (**Right**), via the proposed method and baseline methods in the building complex scenarios, averaged over 5 runs.

When a new request arrives, the coordination layer checks whether a feasible resized backbone and a feasible request subgraph can both be realized. If so, the request is instantiated by the graph-matching procedure above. When multiple requests are pending simultaneously, MoRoCo processes them sequentially on the current embedded graph, updating the available backbone after each accepted request. In this way, the fleet is incrementally decomposed into multiple active subgraphs while preserving an exploration-capable backbone for the remaining robots.

Once a request is completed, its subgraph is dissolved and the involved robots rejoin the backbone through the rejoining mechanism in Sec. III-B. The backbone is then expanded and resynchronized so that the team recovers its default exploration structure. This online decomposition-and-composition process is central to MoRoCo: instead of switching globally between exploration and interaction, the team continuously reallocates robots between the latency-preserving backbone and task-specific communication structures according to the active request set. As a result, MoRoCo unifies collaborative exploration, bounded-latency information return, and online support for communication-demanding interactions within a single coordination framework.

TABLE I: Single-team results averaged over 5 runs.

Environment	**Method**	**CA**[1]	**ER**[2]	**Eff**[3]	**LU**[4]	**RR**[5]	**Int**[6]
Building $60m \times 50m$	CMRE	1433	50.5	2.4	432.8	3.8	no
	M-TARE	1442	50.8	2.4	445.5	3.7	no
	JSSP	1323	46.6	2.2	425.5	2.9	no
	MOROCO-NI	2146	75.6	3.6	449.6	2.1	no
	MOROCO	**2838**	**100**	**4.7**	**573.5**	**1.4**	**yes**
Cave $65m \times 57m$	CMRE	1233	63.1	1.1	433.6	3.8	no
	M-TARE	1397	71.5	1.3	556.0	4.0	no
	JSSP	813	28.0	0.7	500.8	2.5	no
	MOROCO-NI	1572	80.5	1.4	875.2	1.75	no
	MOROCO	**1953**	**100**	**1.8**	**1027.1**	**1.1**	**yes**
Indoor $120m \times 90m$	CMRE	562	19.3	0.2	543.2	3.6	no
	M-TARE	564	19.4	0.2	484.5	3.5	no
	JSSP	580	19.9	0.2	495.6	2.9	no
	MOROCO-NI	966	33.2	0.3	473.1	2.1	no
	MOROCO	**2908**	**100**	**1.0**	**2818.0**	**1.4**	**yes**

[1] CA: explored area in m^2. [2] ER: exploration rate in %.
[3] Eff: exploration efficiency in m^2/s.
[4] LU: last update time in s. [5] RR: return rate in #/T_h.
[6] Int: whether operator–robot interaction is supported.

IV. EXPERIMENTS

For validation, extensive numerical simulations and hardware experiments are presented in this section. All components are implemented in `Python` and `ROS` and tested on a laptop with an Intel Core i7-1280P CPU.

A. *Simulation Setup*

Each robot builds a local 2D occupancy-grid map and navigates using the default differential-drive ROS stack. The simulations are conducted in ROS-Stage with `gmapping`. The maximum linear and angular speeds are set to 1.0 m/s and 1.0 rad/s, respectively, and the sensing range is 15 m. Communication is intermittent and enabled only when the measured link quality exceeds a threshold of 50 dB. Local maps are merged whenever connectivity is available, and operators interact with the fleet through a customized RViz interface to issue online requests and update the latency bounds at return events, as shown in Fig. 4.

We consider three benchmark environments for quantitative comparisons: a building of size 60m×50m, a cave of size 65m×57m, and a larger indoor of size 120m×90m. In the single-team comparison, one operator and four robots are deployed under the latency request ξ_1^k with $T_h = 160$s, and each method is repeated over five runs. In the multi-team comparison, two teams are used, each consisting of one operator and two robots, with $T_h = 160$s and $T_c = 360$s. In addition, we evaluate a request-rich two-team setting in a 90m×70m office environment, where each team contains one operator and four robots and must handle multiple online requests during exploration. For the single-team comparison, we compare MoRoCo against CMRE [2], M-TARE [3], JSSP [8], and an ablated version denoted as MOROCO-NI, which disables online human–robot interaction. Since the baseline methods do not explicitly enforce bounded operator-update latency, they are minimally adapted by inserting return events while preserving their original exploration policies.

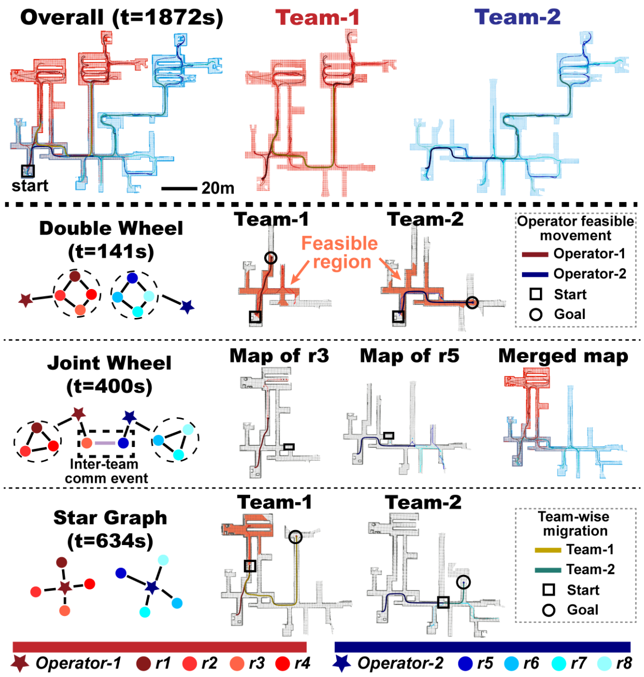


Fig. 7: Simulation results of two teams deployed in a large subterranean environment for Scenario-2, where the latency $T_\mathrm{h} = 120s$ and $T_\mathrm{c} = 300s$. **Top:** the explored area of each team within the final map for Team-1 (red) and Team-2 (blue), including the trajectories for robots and operators; **Bottom:** the evolution of embedded graphs for both teams over time.

B. Single-Team Comparisons

Table I and Fig. 6 summarize the single-team results. Across all three environments, MoRoCo is the only method that simultaneously achieves complete exploration, bounded-latency operator updates, and online operator interaction. In the building, cave, and indoor environments, MoRoCo reaches 100% coverage with explored areas of 2838, 1953, and 2908 m^2, respectively. By comparison, MOROCO-NI reaches only 75.6%, 80.5%, and 33.2%, while CMRE, M-TARE, and JSSP degrade substantially once periodic returns are enforced. Since all methods are evaluated under the same latency requirement, the observed gap reflects the effectiveness of the underlying coordination strategy rather than differences in communication assumptions. Fig. 6 shows a consistent trend: although CMRE and M-TARE expand quickly at the beginning, their operator-received maps stagnate early under the return constraint, leading to incomplete coverage; JSSP performs worse due to inefficient rendezvous and waiting.

MoRoCo also yields the lowest return rate in all three environments, with 1.4, 1.1, and 1.4 returns per T_h, respectively. MOROCO-NI improves over the baselines, showing that the latency-aware backbone itself is beneficial, but it still fails to reliably complete larger environments without online operator guidance. Overall, these results show that bounded-latency exploration cannot be achieved by simply inserting compliant return events. Instead, it requires joint coordination of exploration progress, communication scheduling, and operator interaction, as evidenced by the smaller gap between explored area and operator-received area achieved by MoRoCo in Fig. 6.

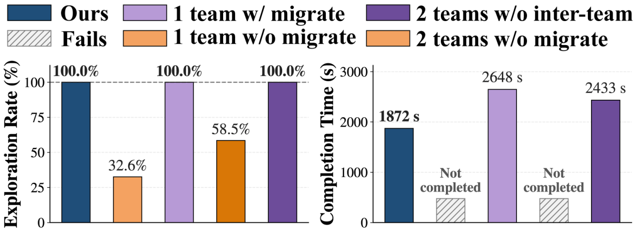


Fig. 8: Comparison of exploration rate (**Left**) and completion time (**Right**) for the fleet of single and two teams with and without "migrate" requests.

TABLE II: Summary of the Contingent Tasks, along with the Associated Online Requests.

Task	Type	Request	Robots	$T_{\mathtt{tra}}$[s][1]	$T_{\mathtt{sta}}$[s][2]	$T_{\mathtt{dur}}$[s][3]
Task-1	Inspection	ξ_4^2	r5, r8	44	356	39
Task-2	Inspection	ξ_4^2	r5, r8	16	411	152
Task-3	Assistance	ξ_6^1	r1, r3	43	572	24
Task-4	Push door	ξ_4^1	r1, r2, r3	35	1231	25
Task-5	Inspection	ξ_4^2	r5, r7, r8	28	1755	99

[1] Transition time from the joint wheel to chain graph.
[2] Starting time of the task execution. [3] Duration of the task.

C. Multi-Team Results

1) Migrate-enabled coordination in branched environments: Scenario-2 evaluates inter-team latency constraints in a large subterranean environment with two teams, each consisting of one operator and four robots, and $T_\mathrm{c} = 300s$ in Figs. 7 and 8. With `migration` and inter-team coordination enabled, both operators obtain the complete map at $1872s$ while achieving complementary branch coverage: team-1 and team-2 explore 59.2% and 67.4% of the environment with only 26.6% overlap, and operators travel $139.4m$ and $145.0m$, respectively. These results highlight two factors for scalability in this branched topology. First, `migration` is necessary when feasible-region motion alone cannot reach new frontiers; without it, exploration stalls early and the final exploration rate drops to 32.6% for one team of eight robots and 58.5% for two teams of four robots. Second, explicit inter-team coordination under latency constraints is critical for efficiency rather than mere completion: removing it still allows completion under `migration`, but overlap rises from 26.6% to 84.0% and completion slows from $1872s$ to $2433s$. Moreover, although `migration` enables reliable completion across configurations, a single team of eight robots remains slower and finishes at $2648s$, underscoring the benefit of multi-team parallelism for covering multiple branches simultaneously.

2) Request-rich topology adaptation: We next evaluate MoRoCo in a request-rich multi-team scenario that requires repeated transitions among different communication topologies. Two teams, each with four robots and one operator, are deployed in a $90\mathrm{m}\times70\mathrm{m}$ office environment. In addition to exploration, five online tasks are distributed in the environment, including inspection, direct assistance, and remote door

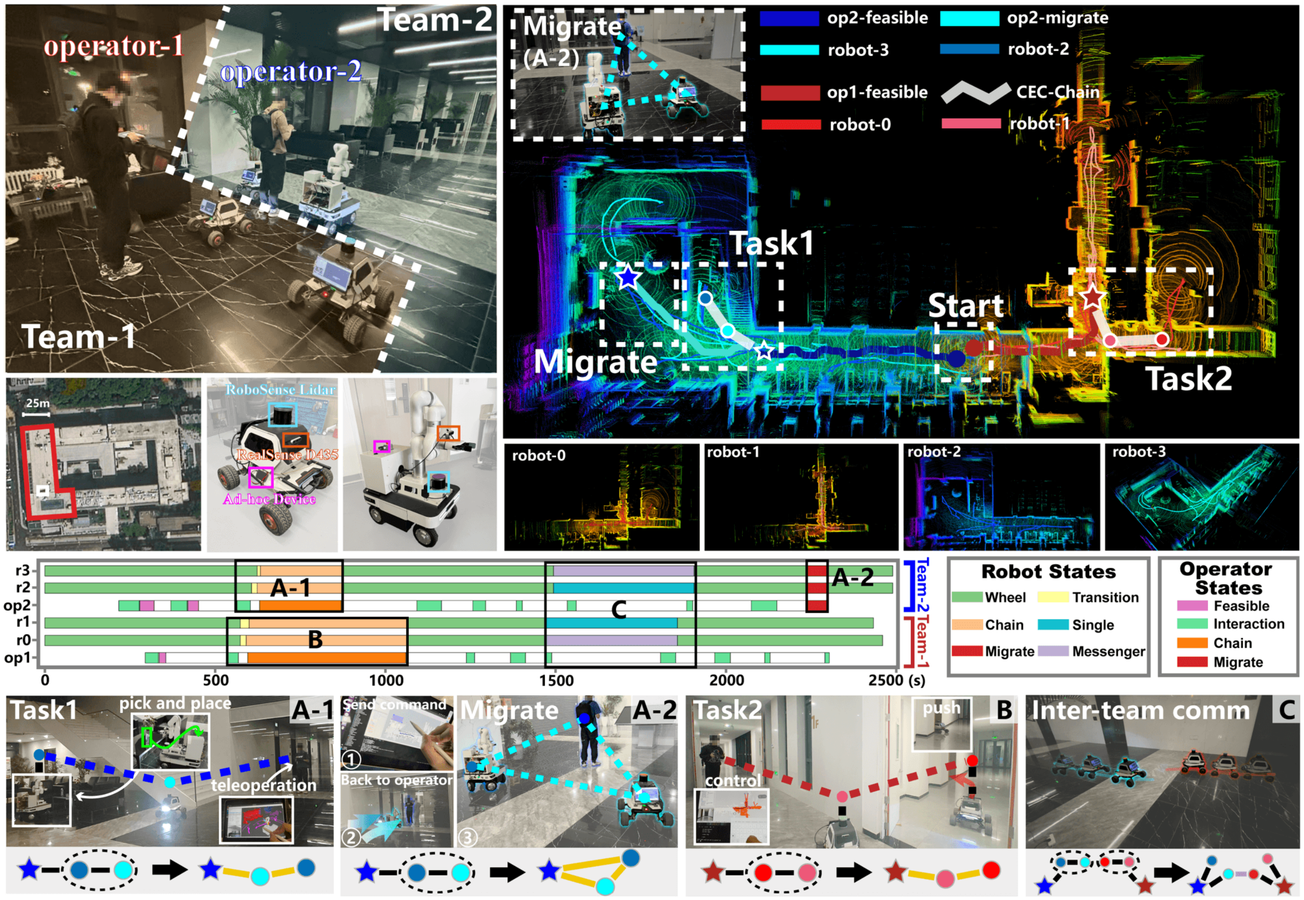


Fig. 9: Top-left: the experimental environment and the heterogeneous robot teams. Team-1 consists of 2 ground robots, while team-2 consists of 1 ground robot and 1 mobile manipulator. Local communication is enabled via the FT-Dlink ad-hoc device. **Top-right**: the final merged map at the end of the mission, and the local maps explored by each robot. Trajectories of the operators are indicated by the thick lines, starting from the initial location and ending at the solid stars, while the thin lines represent the trajectories of the robots. **Middle**: The evolution of communication graphs in both teams during the whole mission. **Bottom**: Snapshots of key moments and graph transitions during the mission.

opening, all of which require temporary communication chains and therefore trigger topology adaptation during execution. As summarized in Table II and Fig. 1, all five tasks are completed successfully by 1854s, including four operator-issued requests of type ξ_4 and one robot-issued request of type ξ_6. This result shows that MoRoCo can support both human-initiated and robot-initiated contingent tasking without interrupting the overall exploration mission.

The computational overhead of topology adaptation remains small relative to physical task execution. Constructing the task-specific target graph set takes 0.3s on average, and graph matching requires only 0.1s, whereas the actual task executions last between 25s and 152s. Across the experiment, each team performs 11 graph transitions, with wheel-to-chain transitions completed within 34–42s and the fastest direct task-to-task transition requiring only 16s, indicating that MoRoCo can exploit topological similarity across consecutive requests to reduce reconfiguration delay. Despite repeated graph changes, the system also maintains bounded inter-team information exchange, with only four external communication events occurring at 574s, 1167s, 1683s, and 2224s, while both operators still receive complete maps at 2242s and 2290s. These results demonstrate that MoRoCo scales to frequent online tasking and remains practical because planning time stays well below execution time.

D. Hardware Validation

We finally validate MoRoCo in real-world indoor experiments under range-limited ad-hoc communication. The hardware platform consists of three Scout-mini ground robots and one Cobot-kit mobile manipulator, deployed as two teams with two robots and one operator per team in a $100\text{m}\times 30\text{m}$ office environment without pre-installed communication infrastructure. The Scout-mini robots are equipped with lidar, IMU, onboard computation, and RGB-D cameras, while the mobile manipulator supports remote manipulation and close-range perception. All robots communicate through local ad-hoc devices, and navigation is implemented using the standard ROS navigation stack with a maximum speed of 0.5 m/s.

As shown in Fig. 9, we evaluate two representative tasks that require topology adaptation. In the first, Operator 1 requests remote access for a "push door and inspect" task; graph matching takes 3.1s, the participating robots reach their assigned locations in 27s, and a stable video stream is established for teleoperation and inspection. In the second, Operator 2 teleoperates the Cobot-kit manipulator for a "pick and place" task; graph matching takes 2.8s, the topology transition takes 26s, and the task is completed in 240s through a tablet-based interface with camera, point-cloud, and manipulation feedback. After these tasks, Team 2 performs a team-wise migration as a star graph at 2243s and resumes exploration. Overall, the hardware results confirm that bounded-latency intermittent updates keep operators sufficiently informed for online decision-making, while graph decomposition and rejoining enable communication-demanding tasks such as teleoperation and remote inspection under restricted communication.

V. CONCLUSION

This work presented MoRoCo, an online topology-adaptive framework for multi-operator multi-robot coordination under restricted communication. Built on a latency-bounded intermittent communication backbone, MoRoCo enables collaborative exploration and online request servicing through dynamic graph reconfiguration using only local communication. Simulations and hardware experiments demonstrated its effectiveness in communication-restricted environments.

REFERENCES

[1] Mohammad Al Khawaldah and Andreas Nüchter. Enhanced frontier-based exploration for indoor environment with multiple robots. *Adv. Robot.*, 29(10):657–669, 2015.

[2] Wolfram Burgard, Mark Moors, Cyrill Stachniss, and Frank E Schneider. Coordinated multi-robot exploration. *IEEE Trans. Robot.*, 21(3):376–386, 2005.

[3] C. Cao, H. Zhu, Z. Ren, H. Choset, and J. Zhang. Representation granularity enables time-efficient autonomous exploration in large, complex worlds. *Sci. Robot.*, 8(80): eadf0970, 2023.

[4] Kyle Cesare, Ryan Skeele, Soo-Hyun Yoo, Yawei Zhang, and Geoffrey Hollinger. Multi-uav exploration with limited communication and battery. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 2230–2235, 2015.

[5] Jiaqi Chen, Boyang Sun, Marc Pollefeys, and Hermann Blum. A 3d mixed reality interface for human-robot teaming. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 11327–11333, 2024.

[6] Rafael Gonçalves Colares and Luiz Chaimowicz. The next frontier: Combining information gain and distance cost for decentralized multi-robot exploration. In *Proc. ACM Symp. Appl. Comput.*, pages 268–274, 2016.

[7] Micael Santos Couceiro. An overview of swarm robotics for search and rescue applications. *Artif. Intell.: Concepts, Methodol., Tools, Appl.*, pages 1522–1561, 2017.

[8] Alysson Ribeiro Da Silva, Luiz Chaimowicz, Thales C. Silva, and M. Ani Hsieh. Communication-constrained multi-robot exploration with intermittent rendezvous. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pages 3490–3497, 2024.

[9] Abhinav Dahiya, Alexander M Aroyo, Kerstin Dautenhahn, and Stephen L Smith. A survey of multi-agent human–robot interaction systems. *Robot. Auton. Syst.*, 161:104335, 2023.

[10] Niklas Dahlquist, Samuel Nordström, Nikolaos Stathoulopoulos, Björn Lindqvist, Akshit Saradagi, and George Nikolakopoulos. Deployment of an aerial multiagent system for automated task execution in large-scale underground mining environments. *IEEE Trans. Field Robot.*, 2:465–484, 2025. doi: 10.1109/TFR.2025.3580068.

[11] Joel M Esposito and Thomas W Dunbar. Maintaining wireless connectivity constraints for swarms in the presence of obstacles. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 946–951, 2006.

[12] Ahmed Hussein, Mohamed Adel, Mohamed Bakr, Omar M Shehata, and Alaa Khamis. Multi-robot task allocation for search and rescue missions. In *J. Phys.: Conf. Ser.*, volume 570, page 052006, 2014.

[13] Filip Klaesson, Petter Nilsson, Tiago Stegun Vaquero, Scott Tepsuporn, Aaron D. Ames, and Richard M. Murray. Planning and optimization for multi-robot planetary cave exploration under intermittent connectivity constraints. In *Proc. ICAPS Workshop Planning Robot.*, 2020.

[14] Michael T Ohradzansky, Eugene R Rush, Danny G Riley, Andrew B Mills, Shakeeb Ahmad, Steve McGuire, Harel Biggie, Kyle Harlow, Michael J Miles, Eric W Frew, et al. Multi-agent autonomy: Advancements and challenges in subterranean exploration. *Field Robot.*, 2:1068–1104, 2022.

[15] Danny G. Riley and Eric W. Frew. Fielded human-robot interaction for a heterogeneous team in the darpa subterranean challenge. *ACM Trans. Human-Robot Interact.*, 12:1 – 24, 2023.

[16] Zhuoli Tian, Yuyang Zhang, Jinsheng Wei, and Meng Guo. iHERO: Interactive human-oriented exploration and supervision under scarce communication. In *Proc. Robot.: Sci. Syst.*, 2024.

[17] Ekaterina Tolstaya, James Paulos, Vijay Kumar, and Alejandro Ribeiro. Multi-robot coverage and exploration using spatial graph neural networks. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pages 8944–8950, 2021. doi: 10.1109/IROS51168.2021.9636675.

[18] Tiago Vaquero, Martina Troesch, and Steve Chien. An approach for autonomous multi-rover collaboration for mars cave exploration: Preliminary results. In *Proc. Int. Symp. Artif. Intell., Robot. Autom. Space*, 2018.

[19] Boyu Zhou, Hao Xu, and Shaojie Shen. Racer: Rapid collaborative exploration with a decentralized multi-uav system. *IEEE Trans. Robot.*, 39(3):1816–1835, 2023. doi: 10.1109/TRO.2023.3236945.