SLEI3D：有限通信条件下通过异构机器人集群实现同步探索与检查

Junfeng Chen[1], Yuxiao Zhu[2],Xintong Zhang[2],Bing Luo[2],和 Meng Guo[1]

arXiv:2601.00163v1 [cs.RO] 1 Jan 2026

摘要——**无人机和地面车辆等机器人集群已广泛用于静态环境的例行检查，此类环境中的感兴趣区域已知且预先规划。然而，在许多应用中，感兴趣区域并不明确，需要在探索过程中在线识别。因此，本文研究在未知环境中同步探索、检查，并随后实时向移动地面控制站通信以报告发现结果的问题。异构机器人配备了不同的传感器，例如用于快速探索的远距离激光雷达和用于详细检查的近距离摄像机。此外，此类环境中通常无法进行全局通信，机器人只能在彼此距离较近且无遮挡时通过自组无线网络相互通信。本文提出一种新颖的规划与协调框架（SLEI3D），整合了协同三维探索、自适应检查和及时通信（通过间歇式或主动式协议）的在线策略。为应对特征数量和位置方面的不确定性，本文为机器人子群之间及子群内部开发了一种多层多速率规划机制，使其能够主动会合并协调局部规划。通过对众多大规模任务进行高保真仿真，对所提框架进行了全面验证，任务规模最大达48 台机器人和384 千立方米。此外，还开展了7 台机器人的硬件实验。**

致从业者——**本文的研究动机源于大型建筑物和基础设施巡检中，大规模异构机器人集群的协同挑战：异构无人机必须协作探索未知环境、识别感兴趣区域，更重要的是检查特定特征（如裂缝、泄漏及其他异常）。此外，这些特征还必须传回控制站，以供进一步分析。现有方法主要关注探索任务，往往忽视近距离检查的需求。为此，本文提出一种分层灵活框架，可在线协调一组异构无人机，高效完成探索、检查和通信任务，而特征的数量和位置均未知。该方法不依赖全对全通信网络，而是通过间歇式主动通信协议应对有限的通信范围和带宽，使机器人能够在在线执行过程中交换局部规划、已探索区域及检测到的特征。通过在高保真仿真器中进行大量仿真，结果表明，所提框架能够在各种场景中高效、可靠地执行大规模同步探索与检查任务。此外，实验还展示了该框架应对机器人故障和通信中断的鲁棒性。针对无人机地面车辆和无人机开展的硬件实验验证了该框架的实际应用价值。**

索引词—**异构多机器人系统、三维探索、协同任务规划、间歇性通信。**

作者任职于 [1]北京大学先进制造与机器人学院，北京 100871，中国；以及 [2]昆山杜克大学自然与应用科学部，中国苏州 215316。
meng.guo@pku.edu.cn

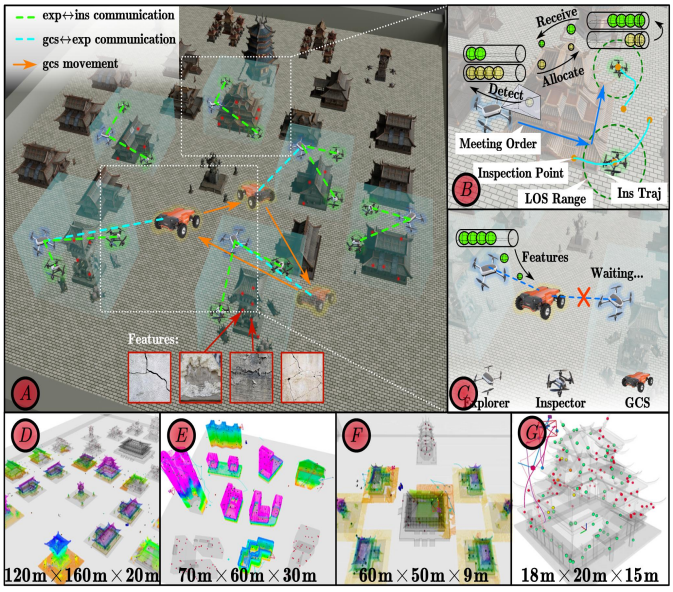

图 1：(a)6 台探索机器人和 12 台检查员被安排同时探索并检查数量众多且未知的特征；**(b)** 同一子组中的探索者和检查员协同执行巡检任务；**(c)**移动地面站主动与探索机器人会合，以接收最新特征；**(d)-(g)** 在四个不同大规模场景中在线执行的快照。

I. 引言

无人机（UAVs）和无人地面车辆（UGVs）组成的机器人群已被部署，用于对发电厂 [1], [2], 桥梁 [3], [4] 和工业场所 [5], [6]等大型偏远基础设施执行日常维护和巡检任务。此类任务通常是静态且重复的，需要访问和检查的感兴趣区域序列已预先给定。然而，正如图 1 所示，在许多应用中，环境和感兴趣区域（AoI）事先均未知，这就要求机器人同时探索环境、识别感兴趣区域，然后检查其中的特征。现有大多数协同探索工作 [7]–[10] 仅关注探索任务，以快速获取全局地图，却忽视了对探索过程中发现的某些特征进行近距离检查的需求 [11],[12], ，例如行星洞穴中的裂缝 [13], 、搜救过程中的生命迹象 [14], 以及考古场所的放大图像 [15]。更重要的是，这些工作通常假设机器人之间存在全局全对全通信，但在许多上述场景中，通信设施可能不可用 [16]或严重退化 [17], 因此这一假设并不现实。在这种情况下，机器人只能通过受视距（LOS） [18] 和邻近约束 [19]限制的自组网络交换信息。这对机器人集群的协调提出了巨大

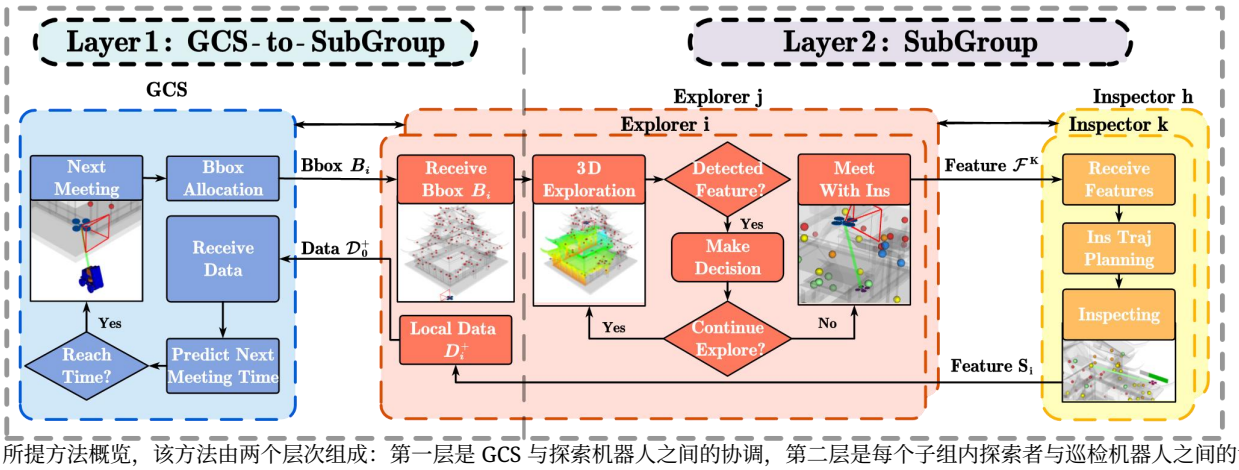

图 2： 所提方法概览，该方法由两个层次组成：第一层是 GCS 与探索机器人之间的协调，第二层是每个子组内探索者与巡检机器人之间的协调，所有协调均在有限通信条件下进行。

挑战，因为通信事件、探索和巡检任务如今紧密相关，因此应当同时进行规划 [20]–[23]。

此外，通常需要及时向移动地面控制站（GCS）更新探索进度和已检查特征的情况，例如，以便规划维护和修理等后续动作 [24]。在没有全局通信网络的情况下，这尤其具有挑战性，因为机器人分散开来进行探索时，GCS 会失去连接 [25]。换言之，机器人与 GCS 应主动、频繁地协调通信，包括通信时间、位置和内容。如何为 GCS 与机器人集群设计灵活高效的协调框架，仍未得到解决。

如图 2 所示，本文提出了一种同时进行大规模探索、检查和通信的框架（SLEI3D），适用于在有限通信条件下于未知环境中作业的异构机器人队伍。给定包围感兴趣区域（AoI）的边界框后，将机器人划分为多个由探索者和巡检机器人组成的子组。作为关键组成部分，本文为配备远距离激光雷达的探索者设计了一种三维协同探索策略，通过几何感知前沿生成高效检测感兴趣区域。随后，将已识别的感兴趣区域分配给配备近距离摄像机的巡检机器人，并将其建模为三维约束路径规划问题，以最大限度地提高巡检效率并确保安全距离。接着，设计了一种任务完成时间预测算法，使 GCS 能够快速收集已检查特征的结果。更重要的是，在 GCS 与各子组之间设计了一种间歇通信协议，以便根据更新后的地图和特征进行在线数据交换与调整。相比之下，在每个子组内部采用主动通信协议，由探索者根据检测到的特征动态确定与巡检机器人通信的时间和位置。本文对所提框架的效率和可靠性进行了理论分析，并通过大量大规模仿真和硬件实验进行了验证。最多部署 48 台机器人，对多个包含 150 多项巡检任务的大规模场景进行探索和检查。

本工作的主要贡献有三点：（I）针对异构机器人队伍在有限通信条件下的同步探索与检查，提出了一种新颖的问题表述；（II）提出多层多速率协调框架，对探索任务、检查任务和机器人间通信进行联合优化；（III）通过大规模仿真验证了该方法在实际场景中的性能。据我们所知，这是首个提供如此全面解决方案的工作。

II. 相关工作

A. 多机器人协同探索

自主探索在机器人技术领域有着悠久的历史 [26], [27], 例如， [7] 提出了一种直观而强大的基于前沿的方法来引导探索。通过将这些前沿分配给不同机器人进行并发探索，该方法已扩展至多机器人团队，例如采用分布式拍卖 [28], 多车辆路径规划 [9],信息增益优化 [8], [10],以及拓扑图动态优化 [29]。另一方面， [30] 提出了一种泛洪算法，确保多个机器人能够探索整个环境而不遗漏任何区域。 [31] 提出了一种多机器人深度优先搜索（MR-DFS）方法，通过并行搜索来探索以图表示的未知环境。然而，这些工作通常假设所有机器人都能即时通信并随时交换信息，即它们始终能够访问同一张全局地图。但对于机器人间通信范围受限的未知环境而言，这往往不切实际。除上述经典方法外，基于强化学习（RL）的方法也已应用于多机器人探索，见 [32]–[34]。这些方法主要关注定制观测空间的设计，以纳入部分可观测性约束，并设计新颖的奖励函数来提高探索效率。因此，许多近期工作将机器人间通信规划与自主探索相结合 [35]。 [36],中的“分区与 Snearkernet”中继算法、 [37], 中的四状态策略、 [38],中针对会合点的集中式整数优化、 [39],中基于施泰纳树的部署，以及 [40] 中的圆形通信模型，均采用基于事件的通信方案。另一方面， [41] 采用全连接网络

并始终保持通信，而 [21] 则利用无线电投放器作为机器人之间的扩展通信中继。尽管如此，这些工作仅考虑协同探索任务，并未涉及对特定特征的巡检任务。

B. 自主巡检

借助不同传感器，自主巡检已应用于多种场景，例如 [14], [15]。这类任务需要根据三维结构和传感器内参生成一组视点，再将这些视点分配给机器人进行检查。 [42] 提出了一种基于骨架的空间分解方法，并结合旅行商问题（TSP）算法。 [43] 通过体素膨胀或减法生成航路点和路径基元，并采用基元覆盖图（PCG）优化整体路径。然而，这些工作主要关注单机器人探索策略，并不直接适用于多机器人系统。

[44] 将随机采样与势场相结合，生成候选视点，并通过求解整数优化问题将其分配给无人机编队。此外，[45], 中的多无人机巡检覆盖路径规划（MU-CPPI）算法将视点分配问题建模为集合覆盖车辆路径问题（SC-VRP）。该方法采用先探索后检查 框架：无人机首先全面探索环境以构建先验地图，随后根据获取的数据专门进行巡检路径规划。这种解耦通常会导致巡检效率低下。为解决这一问题， [46],提出了同步探索与拍摄框架（SOAR）。SOAR使用配备激光雷达的探索机器人检测未覆盖区域，并在表面边界处生成巡检视点；同时，通过求解一致多仓库多旅行商问题 （Consistent-MDMTSP），将配备相机的拍摄机器人分配到这些视点。然而，该方法依赖无人机之间以及无人机与静态地面控制站（GCS）之间持续的全对全通信，以进行全局任务分配。此外，该框架指定单个探索者作为唯一的边界检测器，导致其在大规模环境中成为计算瓶颈。与本文最相关的工作 [47]称为CARIC，它为多机器人系统中的同步探索与检查引入了一种分层策略。CARIC将机器人划分为专业化团队并分配至各子区域，由团队在当地执行任务，并通过视距（LOS）通信将检查结果传回静态地面控制站。然而，CARIC通过预先在矩形包围盒上定义通信点来维持机器人间的连通性；这种方法不适用于具有非凸内部结构的不规则包围盒，也不适用于动态地面控制站与多个异构无人机团队交互的场景。

C. 有限通信下的多机器人协调

通信约束下多机器人协调的关键挑战在于确定机器人间通信应在何时何地进行，以及消息内容如何针对不同目的进行定制。这一问题不仅存在于协同探索 [48],, 也存在于协同巡逻 [49], 和覆盖规划 [50] 中。研究者提出了不同的通信协议。例如， [38]提出了适用于所有机器人的会合—合并—任务协议，通过构建约束整数优化问题来确定会合点和会合时间。相关研究 [25] 通过迭代求解多个带时间窗的多车辆路径问题（ MVRP-TW），提出了分布式间歇通信协议。最相关的研究 [47]通过在规则边界框上选择通信点来处理视距通信约束，但这种方法不适用于不规则结构。然而，这些研究主要假设任务目的单一，例如协作、探索或覆盖。此外，上述研究均未考虑移动地面控制站作为信息枢纽的作用，也未考虑中继实时信息的需求。

III. 问题描述

A. 工作空间与机器人模型

考虑一组 N机器人 $\mathcal{N} \triangleq \{1, \cdots, N\}$ 它们在一个共同、未知且有界的工作空间 $\mathcal{W} \subset \mathbb{R}^3$中协同工作。假设工作空间的边界尺寸和形状是先验已知的。每个机器人 $i \in \mathcal{N}$ 具有一个状态 $x_i \in \mathbb{R}^3$ ，系统状态由堆叠向量 $X \triangleq [x_i] \in \mathcal{X}$给出。由于存在机器人间避碰和机器人-障碍物碰撞规避等安全约束，系统状态被限制在安全集 $\mathcal{X} \subset \mathcal{X}$内。机器人分为两种类型 $\mathcal{N} \triangleq \mathcal{N}_{\text{e}} \cup \mathcal{N}_{\text{o}}$，即探索机器人 $\mathcal{N}_{\text{e}}$ 和巡检机器人 $\mathcal{N}_{\text{o}}$。每个机器人 $i \in \mathcal{N}$ 都能在本地执行同步定位与建图，并在其地图内安全导航，避免与其他机器人或检测到的障碍物发生碰撞。记

$$\mathbf{p}_{sg} \triangleq \texttt{Navi}_i(p_i, p_g, \mathcal{M}_i), \tag{1}$$

导航模块 [51]通过位于其局部工作空间地图 $\mathcal{M}_i$内的路径$\mathbf{p}^{sg}$，引导机器人 i 从当前位姿 $p_i \in \mathcal{W}$ 前往目标位姿 $p_g \in \mathcal{W}$，同时确保$\mathbf{p}_{sg} \subset \mathcal{X}_{\text{。}}$

每对机器人都可以通过自组织无线通信网络交换数据，但须满足视距（LOS）和有限通信范围约束。记

$$(D_i^+, D_j^+) \triangleq \texttt{Comm}_{ij}(D_i, D_j), \tag{2}$$

作为通信模块，如果机器人 i， $j \in \mathcal{N}$会通过通信更新其局部数据 D_i 和 D_j ，前提是其视距未被 $\mathcal{W}$中的障碍物阻挡，且其相对距离在通信范围 $r_{ij} > 0$内。

此外，还有一个移动地面控制站，其唯一索引为 0，仅负责发送和接收数据，不承担通信管理者的角色。它遵循（1）中的相同导航模块，并与（2）中的其他机器人具有相同的通信约束，即通信范围由 $r_{0i} > 0$、 $\forall i \in \mathcal{N}$ 给出。类似地，地面控制站的局部地图表示为 $\mathcal{M}_0$ ，局部数据表示为 D_0。为简洁起见，记为 $\mathcal{N}^+ \triangleq \{0\} \cup \mathcal{N}$。

B. 探索与检查

每个探索者 $i \in \mathcal{N}_{\text{e}}$ 都可以通过探索模块更新其局部地图

$$\mathcal{M}_i^+ \triangleq \texttt{Explore}_i(p_i, \mathcal{M}_i, \mathcal{W}), \tag{3}$$

其中， $\mathcal{M}_i^+$ 是更新后的局部地图。例如，可通过机载激光雷达在线构建八叉树地图 [52] 。此外，工作空间中有 $Q > 0$ 个感兴趣特征，记为 $\mathcal{F} \triangleq \{f_q, \forall q \in Q\}$。基于更新后的地图，每个探索者 $i \in \mathcal{N}_{\text{e}}$ 都可以通过多模态感知系统，在其局部地图中识别出一组可能包含若干特征的感兴趣区域（AoI）。该系统整合了由融合视觉深度图像进行的三维重建，并利用实时语义分割和加速目标检测加以增强 [53]，[54]。该检测过程通过特征融合模块形式化表示:

$$\left\{(f_q, \phi_q)\right\} \triangleq \texttt{Fit}_i(\mathcal{M}_i), \tag{4}$$

其中， $\phi_q \subset \mathcal{M}_i$ 是可能包含特征 $f_q \in \mathcal{F}$ 的感兴趣区域（AoI）。

每个巡检机器人 $i \in \mathcal{N}_{\text{o}}$ 都可以近距离检查感兴趣区域（AoI） ϕ_q ，即通过检查模块进一步确定特征 f_q 是否存在于 ϕ_q 内及其状态:

$$(D_i^+, s_q, t_q) \triangleq \texttt{Inspect}_i(p_i, f_q, \phi_q), \tag{5}$$

其中， $s_q \in \mathbb{R}^L$ 是特征 f_q的落地表示，维度为 $L > 0$, 例如位置、图像和点云； $t_q > 0$ 是检查特征 f_q所需的时长；局部数据D_i^+ 则通过检查结果进行更新。

地面控制站需要收集机器人集群的数据，以了解探索与检查的进展。数据收集过程遵循与（2）相同的通信协议，以更新其局部数据 D_0。

示例 1。 如图 1 所示，仿真中考虑的考古任务部署了 6 架大型无人机作为探索者，搭载高分辨率激光雷达以构建整个遗址的全局地图；部署了 12 架小型无人机作为检查员，搭载相机以拍摄众多潜在特征的特写图像；并部署了 1 个地面控制站，从机器人群中收集、协调并更新特征。■

C. 问题表述

每个机器人 $i \in \mathcal{N}^+$ 的局部规划由一系列导航和各种动作构成，即:

$$\xi_i \triangleq \mathbf{p}_i^1 a_i^1 \cdots \mathbf{p}_i^t a_i^t, \tag{6}$$

其中，$\mathbf{p}_i^t \in \mathcal{W}$ 是航路点序列； a_i^t 是动作序列，即探索者 $i \in \mathcal{N}_{\text{e}}$执行探索或通信，巡检机器人 $i \in \mathcal{N}_{\text{o}}$ 执行检查或通信，地面控制站执行通信。规划目标是为机器人群设计探索、检查和通信策略，使地面控制站收集所有特征所需的总时间最小化，即:

$$\begin{aligned}
&\min_{\{\xi_i\}} \quad T \\
\textbf{s.t.} \quad & \mathcal{W} \subseteq M_0(T); && \text{(7a)} \\
& s_q \subseteq D_0(T),\ \forall q \in Q; && \text{(7b)} \\
& x(t) \in \widehat{\mathcal{X}},\ \forall t \in [0, T]; && \text{(7c)} \\
& (1) - (5),\ \forall i \in \mathcal{N}; && \text{(7d)}
\end{aligned}$$

其中， $T > 0$ 是需要最小化的任务持续时间；约束条件 (7a) 确保地面控制站在时间 $t = T$ 的局部地图 M_0 覆盖整个工作空间；约束条件 (7b) 要求地面控制站的局部数据 $D_0(T)$ 包含整个工作空间内所有特征的检查结果；其他约束条件 (7c)、(7d) 则确保机器人群遵循前文所述的导航、探索、检查和通信模块。

IV. 所提解决方案

所提解决方案由四部分组成：（I）第 IV-A 节所述的所提方法概述；（II）在探索机器人与地面控制站之间通信受限的情况下，第 IV-B 节所述的探索任务与通信事件协调；（III）第 IV-C 节所述的针对一组边界框设计的通信感知三维探索与检查算法；（IV）在地图更新并检测到特征后，第 IV-D 节所述的协同探索、检查与通信的在线适应。

A. 所提方法概述

所提方法通过多层多速率协调框架解决式（7）中的上述优化问题，同时对地面控制站、探索机器人和巡检机器人的协同行为进行联合优化。如图 2 所示，通过利用关于感兴趣区域分布的先验知识，以边界框 $\mathcal{B} \triangleq \{B_i\} \subset \mathcal{W}$ 限定搜索空间，从而提升机器人探索效率；这些边界框包围着内部结构的集群。不过，通过下文所述的无先验探索模块，该框架也可应用于完全未知且没有任何先验信息的工作空间。给定边界框后，地面控制站负责滚动时域分配边界框（边界框） $\mathcal{B} \triangleq \{B_1, B_2, \cdots\} \subset \mathcal{W}$、将子组重新分配至其他未完成的边界框，并收集地图信息 $\mathcal{M}$ 以及特征 $\mathcal{S} \triangleq \{s_1, s_2, \cdots\}$的检查结果。首先，地面控制站根据探索机器人数量、边界框数量和机器人感知能力将所有机器人划分为多个子组。然后，按照下文所述的滚动分配算法，将每个子组分配至最近的边界框。此外，所提方法采用双层通信结构：第一层以较低频率协调地面控制站与各子组，第二层则以较高频率管理组内协作。

1) 地面控制站至子群层：对于 GCS 至子群组层，我们设计了一种间歇通信协议，以促进 GCS 与探索机器人之间的协调，主要关注三个方面：(I) 边界框管理。GCS 将边界框的位置告知子组中的探索者，探索者则向 GCS 报告边界框的完成状态。GCS 采用滚动分配算法，将任何剩余边界框分配给最近的子组，以加快整体任务的进度。(II) 检查结果收集。检查员将检查结果 $\mathcal{S}$ 传送给探索者，随后由探索者将这些结果连同探索地图 $\mathcal{M}$一并转发给 GCS。(III) 会合时间和地点的协调。GCS 与

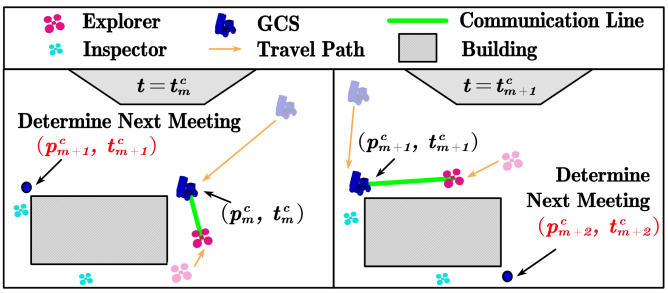

图 3： 地面控制站与探索机器人之间的间歇通信。

探索者利用任务完成时间预测算法，协商下一次会合的时间和地点。为进一步量化任务执行效率，我们引入了用于衡量 GCS 和探索机器人空闲时间的指标。具体而言，τ_0 表示 GCS 等待与探索者会合所花费的时间。对于每个探索者 $i \in \mathcal{N}_e$，其空闲时间定义为 $\tau_i \triangleq \tau_i^- \cup \tau_i^+$，其中 τ_i^- 表示前往与 GCS 或检查员会合地点的移动时间，τ_i^+ 表示会合开始前在会合地点的等待时间。详细说明见第 IV-B 节。

2）子群层：提出一种子组内的主动通信协议，以在通信受限的情况下协调探索者与检查员之间的协作。该协议包含两个主要组成部分：(I) 感兴趣区域分配。探索者将检测到的感兴趣区域分配给检查员。(II) 检查轨迹规划与检查结果中继。检查员将检查结果及其规划轨迹中继给探索者，以协调下一次会合事件。为评估这种组内协调的效率，引入空闲时间 $\tau_j \triangleq \tau_j^- \cup \tau_j^+$ 作为衡量每个检查员 $j \in \mathcal{N}_c$ 效率的指标，其中 τ_j^- 表示没有可供检查的特征时的空闲时段，τ_j^+ 表示前往这些特征所需的移动时间。详细说明见第 IV-C 节。

基于上述多层框架和协调策略，将式 (7) 中的原问题重新表述为在满足相同约束的条件下最小化所有机器人的总空闲时间，即目标函数重新表述如下：

$$\min_{\{\xi_i\}} \{ \sum_{i \in \mathcal{N}^+} \tau_i \}, \tag{8}$$

其中，τ_i 是上文定义的每个机器人 $i \in \mathcal{N}^+$ 的空闲时间。请注意，通过最小化（7）中的 T，可以实现总空闲时间最小化，即最大化探索、检查和通信的任务效率。

注释1. 这里根据各场景中边界框的数量和体积的先验信息，为从业者提供了机器人集群规模的一般指导原则。具体而言，为确保探索、检查和通信并行执行，为每个边界框分配1个地面控制站和1个探索者。检查员的数量由以下公式确定：$|\mathcal{N}_c| = \sum_{i=0}^{|\mathcal{B}|} \binom{2V_{B_i}}{V_{B_{\text{base}}}}$，其中 $V_{B_{\text{base}}}$ 表示单个建筑物的边界框体积，系数2反映了经实证验证的每栋建筑物所需检查员数量。这些数值是完成任务的最低建议配置。降低配置会影响效率，但仍

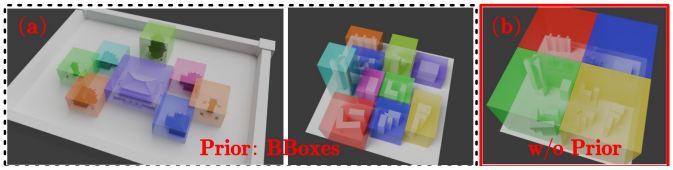

图4： 通过垂直建筑物构建边界框（左）的两个场景；不含任何先验信息的自适应边界框划分（右）。可执行。通常，超过建议数量能够通过提高资源并行性来增强任务效率。

B. 地面控制站至子群层

1) 子问题表述：为确保在线数据采集，在式(2)的通信模块Comm_{0i}，$i \in \mathcal{N}_e$下，要求地面控制站与各子群组通过间歇通信协议频繁会合并通信。用 $\mathcal{C} \triangleq C_1 C_2 \cdots$ 表示通信事件序列，其中 $C_m = (p_m^c,\ t_m^c)$ 表示地面控制站与各子群组内探索机器人的第 m次通信；p_m^c 和 t_m^c 分别表示第 m次通信的位置和时间。间歇通信协议如下：(I) 地面控制站可在满足视距和通信范围条件时与探索机器人通信；(II) 每次通信期间，双方在本地确定下一次会合的时间和地点。随后，双方出发，在下次会合前不再通信。如图3所示，该过程重复进行直至任务终止。在此协议下，探索机器人的空闲时间 $\tau_i \quad i \in \mathcal{N}_e$ 主要源于从探索地点前往预定时间 t_m^c的会面地点 p_m^c 所需的移动时间 $\tau_{i_m}^-$，以及等待地面控制站的等待时间 $\tau_{i_m}^+$。另一方面，地面控制站的空闲时间 τ_{0_m} 是等待探索机器人抵达 p_m^c进行第 m次通信的等待时间。因此，式(8)的目标是在所有通信轮次中最小化探索机器人和地面控制站的总空闲时间。

问题1。 确定最优规划 $\{\xi_i,\ i \in \{0\} \cup \mathcal{N}_e\}$使探索机器人和地面控制站的总空闲时间最小化，即**最小化** $\{\xi_i\} \sum_{m=1}^{M} (\sum_{i \in \mathcal{N}_e} \tau_{i_m} + \tau_{0_m})$，其中 $M > 0$是预定义的规划时域，表示未来的通信轮数。■

为解决问题1，本文提出一种用于构建边界框的参数化方法，如图4(a)所示。对于每栋建筑物，从先验结构数据或低分辨率扫描中提取其主轴对齐占地面 $B_i^{\mathcal{F}} \subset \mathbb{R}^2$，然后将该占地面沿垂直方向延伸，并加入安全裕度$\delta_z \in [\delta_{\min},\ \delta_{\max}]$，以构成立方边界框 $B_i \triangleq B_i^{\mathcal{F}} \times [z_{\text{base}} - \delta_z,\ z_{\text{top}} + \delta_z]$，其中 z_{base} 和 z_{top} 表示结构的标高边界。完整的搜索空间定义为 $\mathcal{B} \triangleq \bigcup_{i=1}^{N} B_i \subset \mathcal{W}$，从而有效涵盖建筑物包络周围 ϵ邻域内的所有表面邻接子空间。通过这种构建方式，机器人集群能够执行表面法向轨迹规划，同时避免计算代价高昂的完整三维重建。对于缺乏先验结构知识的场景，系统将启动自适应边界框生成流程，详见第 IV-E5 节。随后，首先提出一种任务完成时间预测器，用于估算探索者所需的时间

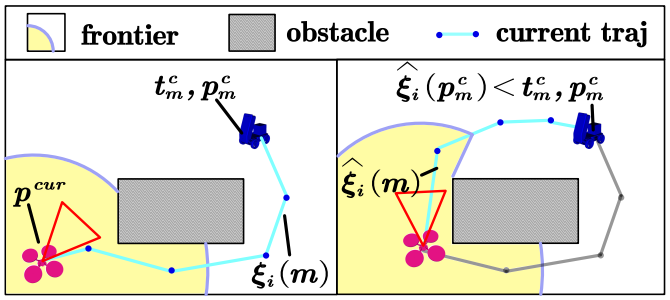

图5： 快速的基于前沿的探索与探索策略的局部自适应 $\{\mathbf{p}_i^t\}$ 由探索者执行。

探索每个边界框 $B_i \in \mathcal{B}$，以便地面控制站安排下一次通信事件。随后，提出一种名为 FF3E(·)的探索算法，使探索者能够在时间 t_m^c抵达会合地点 p_m^c ，从而最小化等待时间 τ_0。最后，提出一种算法，由地面控制站协调与探索者之间的通信事件，以减少等待时间 τ_i^+ 。

2）任务完成时间预测及下一次通信事件的规划：探索者 $i \in \mathcal{N}_e$ 首先探索一段用户定义的时间 $t_{i_m}^e < E_i$，期间根据（4）拟合一组特征 $\mathcal{F}_{i_m}$ ，并接收特征的检查结果 $\mathcal{S}_{i_m}$。然后，探索者 i 估算完成整个边界框 B_i 探索所需的时间，并将其作为下一次通信时间 t_{m+1}^c，即任务完成时间预测器如下：

$$t_{m+1}^c = \left(\frac{V_{B_i}}{|\mathcal{S}_{i_m}|}\right) \cdot \left(\frac{|\mathcal{F}_{i_m}|}{V_{B_i}^m}\right) \cdot t_{i_m}^e, \tag{9}$$

其中， $\mathcal{S}_{i_m} \subset \mathcal{S}$ 是探索者 i 在第 m 次通信期间接收的检查结果集合； $V_{B_i} > 0$ 是边界框 B_i的总体积，而 $V_{B_i}^m$ 表示截至第 m 次通信，探索者 i已探索的体积。若未收到任何特征，即 $\mathcal{S}_{i_m} = \varnothing$，则根据已探索体积的比例近似计算通信时间 t_{m+1}^{ideal}，即 $V_{B_i} t_{i_m}^e / V_{B_i}^m$。此外，从边界框 B_i 的四个角点中选择位置 p_{m+1}^c，使其到探索者 i 当前位置的距离满足以下条件：按照（1）中的导航模块 Navi(·) 运行的 $A^\star$ 算法所估算的到达时间最接近新规划的通信时间 t_{m+1}^c。请注意，所提框架也可集成（9）以外的其他预测算法，例如对现有数据进行回归。

注释 2. 不同于工作 [25] 中地面控制站每轮都与所有探索机器人通信，所提方法根据预测的任务完成时间安排通信，从而无需地面控制站按照预定义顺序与所有探索机器人通信。■

3) 基于前沿的快速三维探索：稍微滥用一下符号，探索机器人 i当前的局部规划表示为：

$$\xi_i(m) \triangleq \mathbf{p}_i^1 a_i^1 \cdots \mathbf{p}_i^t a_i^t \cdots p_m^c a_i^m, \quad t \leqslant t_m^c, \tag{10}$$

其中， p_m^c, t_m^c 表示地面控制站与探索者 i 之间第 已确认的 m 次通信； a_i^m 表示通信

算法 1: 基于前沿的快速三维探索: FF3E(·)

输入: 当前位置 p_i，前沿集合 Γ，会合位置 p_m^c，会合时间 t_m^c，机器人速度 v_i
输出: 最优路径 $\hat{\xi}_i^\star$

函数 planPath(p_0, $\hat{\xi}'$, T^+):
  如果 $T^+ + T_{p_k^c} > t_k^c$ **then**
    返回 $\hat{\xi}'$;
  $\hat{\xi}^\star \leftarrow \hat{\xi}'$;
  对于每个点 $P \in \Gamma \setminus$ 已执行 **do**
    $D \leftarrow$ 获取距离(p_0, P) ;
    $T_{p_0}^P \leftarrow D/v_i$;
    $\hat{\xi}' \leftarrow \hat{\xi}' + [P]$;
    $T^+ \leftarrow T^+ + T_{p_0}^P$;
    $\xi^+ \leftarrow$规划路径(P, $\hat{\xi}'$, T^+) ;
    如果距离($\hat{\xi}^+$) >距离($\hat{\xi}^\star$) **则**
      $\hat{\xi}^\star \leftarrow \hat{\xi}^+$;
  返回 $\hat{\xi}^\star$;
$D_m^c \leftarrow$获取距离(p_i, p_m^c) ;
$T_{p_m^c} \leftarrow D_m^c / v_i$;
$T' \leftarrow t_m^c - T_{p_m^c}$;
如果 $T' > 0$ **则**
  $\hat{\xi}_i^\star \leftarrow$规划路径($p_i$, [], 0);
  $\hat{\xi}_i^\star \leftarrow \hat{\xi}_i' + [p_m^c]$;
返回 $\hat{\xi}_i^\star$;

动作；而 $\mathbf{p}_i^t, a_i^t$ 表示探索的航路点和动作，这些内容应进行优化，以便在通信时间 t_m^c 内最大限度地探索工作空间。如图5所示，具体而言，将 $\xi_i(m) \triangleq \hat{\mathbf{p}}_i^1 a_i^1 \cdots \hat{\mathbf{p}}_i^t a_i^t \cdots \hat{\mathbf{p}}_m^c a_m^c$ 记作修订后的局部规划，其中 $\{\hat{\mathbf{p}}_i^t\}$表示更新后的航路点， $\hat{\mathbf{p}}_m^c$ 表示新增至会合地点 p_m^c 的航路点。因此，探索者 i 的等待时间为 $\tau_i^+ = \hat{\xi}_i(p_m^c) - t_m^c$，其中 $\hat{\xi}_i(p_m^c)$ 表示预计到达 p_m^c的时间。

问题2。 确定局部规划 $\xi_i(m)$ ，使得：(I) 在通信时间 t_m^c 内最大化已探索区域；(II) 最小化等待时间 τ_i^+ 。■

首先值得注意的是，上述两个目标往往相互冲突。现有基于旅行商问题或带时间窗的旅行商问题的算法 [55] 无法直接应用，因为并非所有前沿都必须访问。因此，式（3）中 Explore(·)模块下的所提解决方案总结于算法1。每一轮 m中，探索者 i 首先检查自己是否有足够的时间 t_m^c到达会合地点 p_m^c ，如第14–16行所示。如果 $T' > 0$不成立，则在第20行返回 $\varnothing$ 。否则，算法通过第18–19行中的 planPath(·) 函数寻找最优航路点 $\xi_i^\star$。具体而言，该函数会在第18行检查已耗费的总时间加上到达 p_m^c所需的时间是否超过约定的会合时间t_m^c 。如果超过，则此函数在第3行返回当前路径 ξ' 。否则，它在第4行将当前路径 ξ' 初始化为最优路径 $\xi^\star$。随后，它在第5行遍历每个前沿 $P \in \Gamma$ ，

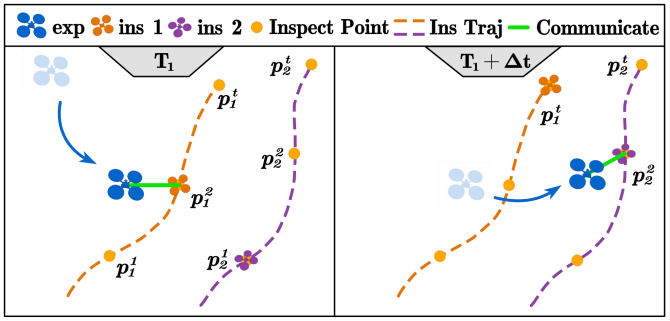

图6：子组内探索机器人（蓝色）与巡检机器人（黄色）之间的主动通信。

在第6–7行计算从其位置 p_0 到 P 的距离 D 和移动时间 $T^P_{p_0}$。之后，在再次调用 planPath$(\cdot)$ 之前，会更新已访问的前沿 ξ' 集合以及已耗费的总时间 T^+，如第10行所示。最后，如果所得路径 ξ^+ 比 $\xi^\star$ 更长，则在第11–12行相应地更新最优路径 $\xi_i^\star$。最后，返回 $\xi_i^\star$ 算法1中的结果，作为问题2的优化解。

算法1的时间复杂度可根据其递归结构逐步分析。在第14至16行的初始化过程中，主要计算到达会合地点 p_m^c 所需的距离和时间，因此复杂度为 $\mathcal{O}(1)$。随后，在递归路径规划过程中，函数planPath$(\cdot)$通过在第10行递归调用自身，探索所有可能的前沿点 $P \in \Gamma$。最坏情况下的时间复杂度为 $\mathcal{O}(|\Gamma|!)$，其中 $|\Gamma|$ 为前沿点总数。最后，第3行更新最优路径的复杂度为 $\mathcal{O}(1)$。因此，算法1的整体复杂度为 $\mathcal{O}(|\Gamma|!)$。

4）地面控制站的通信协调：为了尽量减少地面控制站的空闲时间 τ_0，只需确保在预定时间 t_m^c内到达会合地点 p_m^c。因此，为地面控制站构建了一个带时间窗的旅行商问题（TSP-TW），以优化其在每次通信 $m \in M$中与探索机器人的会合顺序。由于篇幅有限，省略详细描述。总之，本文提出的任务完成时间预测算法、三维探索算法和间歇通信协议，解决了协调地面控制站与探索机器人以减少空闲时间 τ_i 和 τ_0 的子问题。值得注意的是，我们的框架无需借助全球定位系统或实时动态定位等外部定位系统，对地面控制站与探索机器人之间的全局定位进行显式校准。此外，探索区域内任何预先校准的三维坐标都可以作为地面控制站与探索机器人通信的绝对参考，无需额外校准。

引理1. 在所提出的*GCS*至子群组层协调策略下，探索机器人可在有限时间内拟合边界框中的所有特征 $\mathcal{F}$，且地面控制站可在有限时间内收集拟合结果。

证明。 在探索过程中，探索机器人通过算法1探索并拟合特征 $\mathcal{F}_{i_m}$。第1行中的规划模块 planPath$(\cdot)$ 确保访问边界框 B_i 内的所有前沿，并完整探索整个边界框。由于每个边界框 $|\Gamma|$ 中的前沿数量有限，探索任意边界框所需的总时间也有限。此外，探索机器人可以通过滚动分配算法完整探索所有边界框。因此，环境中的所有特征 $\mathcal{F}$ 都能在有限时间内拟合完成。另一方面，通信协议会安排下一次通信事件 $(p_m^c,\ t_m^c)$，探索机器人将遵循第17行的安排。同样，地面控制站通过求解带时间窗的旅行商问题，确保及时抵达会合事件。在每次通信期间，探索机器人都可以将检测到的所有特征传输给地面控制站。由于特征数量和通信轮次均有限，地面控制站可以在有限时间内收集所有结果。■

C. 通信感知的探索与检查

为尽量缩短子组内的空闲时间，我们在有限通信条件下提出了一种同步探索与检查算法。

1) 子问题表述： 不同于间歇通信方案，我们针对子组内的探索者 $i \in \mathcal{N}_e$ 和检查员 $\mathcal{N}_\circ^i \triangleq \{1, \cdots, K\} \subset \mathcal{N}_\circ$，设计了一种基于式（2）中模块 $\text{Comm}_{ij}(\cdot)$ 的新型主动通信协议。该协议分为两个步骤：(I) 探索者 i 在视线约束下与 $\mathcal{N}_\circ^i$ 中的检查员通信；(II) 每次通信期间，探索者不仅要在能量约束下确定与检查员下一次会合的时间和位置，还要将拟合特征分配给检查员，即指定待检查的特征 $\hat{\mathcal{F}}_j \subset \mathcal{F}$，并为每位检查员 $j \in \mathcal{N}_\circ^i$ 更新相应的局部规划 ξ_j^+。如图6所示，该过程会重复进行，直至所有特征均已拟合并检查。

定义1（子组计划）。 子组的整体计划定义为一个四元组：

$$\Xi_i \triangleq (\hat{\mathcal{N}}_i,\ \mathbf{c}_i,\ \boldsymbol{\varphi}_i,\ \boldsymbol{\xi}_i), \tag{11}$$

其中，$\hat{\mathcal{N}}_i \subset \mathcal{N}_\circ^i$是需要通信的检查员子集；$\mathbf{c}_i \triangleq \{(t_j,\ p_j),\ \forall j \in \hat{\mathcal{N}}_i\}$ 是 $\mathcal{N}_i$中每个检查员的会合时间和位置的集合；给定拟合特征集合$\mathcal{F}_i^+$，$\boldsymbol{\varphi}_i \triangleq \{\mathcal{F}_j,\ \forall j \in \mathcal{N}_i\}$ 是分配给每个检查员的新特征集合；而$\boldsymbol{\xi}_i \triangleq \{\xi_j^+,\ \forall j \in \hat{\mathcal{N}}_i\}$ 是所有检查员更新后的局部规划集合。■

问题3。 确定子组的计划 Ξ_i，如式（11）所定义，使子组内探索者和检查员的总空闲时间最小，即：

$$\min_{\Xi_i} \Big\{ \tau_i + \sum_{j \in \mathcal{N}_\circ^i} (\tau_j^+ + \tau_j^-) \Big\}, \tag{12}$$

其中，τ_j^+是检查员$j \in \mathcal{N}^i$。因等待而产生的空闲时间；τ_j^-是除τ_j^+之外的剩余空闲时间。■

2）有限通信条件下的高效三维探索与检查：提出了一种封装为 SOEI$(\cdot)$ 的高效探索与检查算法，以解决上述问题。如图 7 所示并在算法2中概述，该算法由以下三个主要部分组成。

(I) 巡检机器人的选择。最优子组解 $\Xi_i^\star$和空闲时间 $\tau_{\Xi^\star}$ 在第 1 行初始化。然后，算法遍历每个子集 $\hat{\mathcal{N}}_i \subset \mathcal{N}_\circ^i$，以评估相应的空闲时间。请注意，如果没有巡检机器人

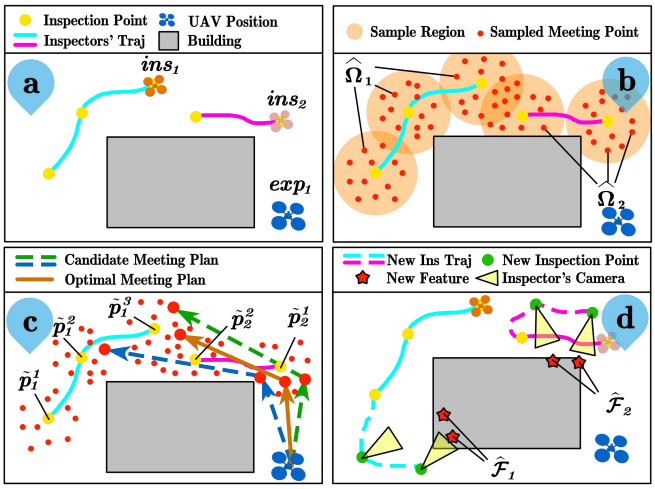

图7： 算法2中所提出的 SOEI 算法的主要组成部分：**(a)** 子组的初始化，包括探索者（蓝色）和检查员（黄色）；**(b)** 候选会合地点 $\{p_j\}$（红色）的采样；**(c)** 会面序列的优化 $\mathbf{c}_i$，包括会合时间和地点（棕色）；以及 **(d)** 将拟合特征 $\mathcal{F}_i^+$（红色星形）分配给检查员 $\widehat{\mathcal{N}}_i$。

被选中，探索者将继续探索。会面序列。本文提出一种(其)于采样的算法，用于确定探索者与 $\widehat{\mathcal{N}}_i$中的巡检机器人之间的最优会合事件序列。在满足视距和通信范围约束的条件下，围绕每个巡检机器人正在执行的局部计划 $\{\xi_j^-\}$ 生成一组采样点 $\widehat{\ }\Omega_j$。用 $\widehat{\ }\mathbf{c}' \triangleq \{\mathbf{c}_i'\}$ 表示潜在的会合事件序列集合，其中每个$\mathbf{c}_i'$ 均通过为每个巡检机器人从 Ω_j 中选取一个采样点，并按照时间顺序遍历这些采样点来构成。随后，通过式（1）中的导航模块 Navi$(\cdot)$，并采用自适应速度以最小化总空闲时间 $\tau_{\Xi^+} = (\tau_i + \sum_{j\in\widehat{\mathcal{N}}_i} \tau_j^+)$，即可获得探索者访问这些采样点的最优轨迹。如第 5 行所示，通过求解以下优化问题，生成会合事件的实际日程$\mathbf{c}_i$ 及相应的空闲时间 τ_{Ξ^+}：

$$\begin{aligned} &\min_{\mathbf{c}_i'} \quad \tau_{\Xi^+} \\ &\text{s.t.} \quad \tau_i^- = \mathbb{A}^\star(\{p_j\}),\ \tau_i^+ = t_j - \xi_j^-(p_j); \qquad \text{(13a)} \\ &\qquad \tau_j^+ = t_j - t_j^{e-},\ \forall j \in \widehat{\mathcal{N}}_i,\ p_j \in \widehat{\Omega}_j, \qquad \text{(13b)} \end{aligned}$$

其中，t_j^{e-} 是巡检机器人 j正在执行的局部规划 ξ_j^- 的结束时间戳；$\xi_j^-(t_j)$ 返回巡检机器人 j 在时间 t_j的位置，而$\xi_j^-(p_j)$ 输出巡检机器人 j位于位置 p_j时的时间戳。此外，约束（13a）使用 $A^\star$ 算法计算探索者 i 的行程空闲时间以及巡检机器人 j的等待时间；（13b）量化探索者 i的等待空闲时间。由于非解析约束（13a）无法直接应用于传统优化工具，因此采用遗传算法，并遵循标准步骤：将潜在序列$\mathbf{c}_i'$ 联合编码为染色体；将适应度函数定义为总空闲时间的倒数，即 Fitness $= 1/\tau_{\Xi^+}$；通过遗传算法施加约束（13）；并遵循标准遗传算法流程 [56]，以获得实际会面序列$\mathbf{c}_i$。最后，OptMeet算法返回总空闲时间 τ_{Ξ^+}，如第5行所示。

算法 2:同步优化探索与检查 SOEI$(\cdot)$

输入： $\mathcal{F}_i^+$，$\mathcal{N}_\circ^i$，$\{\xi_j^-\}$。

输出： $\Xi_i^\star$。

1 $\Xi_i^\star \leftarrow$无，$\tau_{\Xi^\star} \leftarrow \infty$；
2 **对每个** $\widehat{\mathcal{N}}_i \subset \mathcal{N}_\circ^i$ **执行**
3 　$\widehat{\ }\mathbf{c}' =$视距采样$(\{\xi_j^-\})$;
4 　**对每个** $\mathbf{c}'_i \in \widehat{\ }\mathbf{c}$ **执行**
5 　　$(\mathbf{c}_i, \tau_{\Xi^+}) \leftarrow$OptMeet$(\mathbf{c}_i', \{\xi_j^-\})$ 由（13）；
6 　　$(\varphi_i, \boldsymbol{\xi}_i, \tau_{\Xi^-}) \leftarrow$ MVRP$(\mathbf{c}_i, \mathcal{F}^+, \{\xi_j^-\})$；
7 　　$\tau_\Xi \leftarrow \tau_{\Xi^-} + \tau_{\Xi^+}$；
8 　　**如果** $\tau_\Xi < \tau_{\Xi^\star}$，**那么**
9 　　　$\tau_{\Xi^\star} \leftarrow \tau_\Xi$；
10 　　　$\Xi_i^\star \leftarrow (\widehat{\mathcal{N}}_i, \mathbf{c}_i, \varphi_i, \boldsymbol{\xi}_i)$；
11 **返回** $\Xi_i^\star$；

（III）特征分配。根据已确认的会面事件，随后将存储的新特征 $\mathcal{F}^+$ 分配给巡检机器人，并追加到其局部规划末尾。为确定最优分配并最小化空闲时间 τ_{Ξ^-}，本文构建并求解一个 MVRP，通过将局部规划的终点设为初始位置，并将 $\mathcal{F}^+$ 中的所有特征设为待访问位置，从而获得 φ_i。获得 φ_i 后，通过在已分配的特征之间插入最短路径，更新后续局部规划$\{\xi_j^+\}$，并计算总空闲时间 τ_j^-。最后，将 τ_{Ξ^-} 与 τ_{Ξ^+} 相加，得到总空闲时间 τ_Ξ（第7行）。如果该时间小于当前最优空闲时间 $\tau_{\Xi^\star}$，则使用此解更新最优子组解 $\Xi_i^\star$（第9–10行）。需要说明的是，由于同一组内采用相对定位，我们的框架不要求组内共享全局坐标。具体而言，进入边界框时，所有组员均以入口点作为局部原点。探索机器人以该原点为参照，对检测到的特征（结构元素、检查目标）进行地图绘制。最后，巡检机器人直接根据这些相对坐标导航，无需在不同组之间进行外部对齐。

3) 算法摘要：值得注意的是，子集 $\widehat{\mathcal{N}}_i$的数量与 $\mathcal{N}^i$。的大小呈组合关系，即$\mathcal{O}(2^{|\mathcal{N}^i_\circ|})$。生成所有潜在会面序列 $\widehat{\ }\mathbf{c}$的复杂度为阶乘级，即 $\mathcal{O}(|\widehat{\mathcal{N}}_i|!)$。接着，为确定最优会合事件序列$\mathbf{c}_i$，遗传算法的复杂度为 $\mathcal{O}(|\widehat{\mathcal{N}}_i|^3)$。最后，用于分配特征的 MVRP 算法的复杂度为 $O(|\widehat{\mathcal{N}}_i|^{|\mathcal{F}^+|})$。

引理2。 采用针对子群层的算法*2*后，所有特征 $\mathcal{F}$ 均可被检查，且结果 $\mathcal{S}$可由探索机器人在有限时间内收集。

证明。 在每次通信事件中，探索者通过第 6 行的 MVRP 算法，将$\widehat{\ }$新发现的特征 $\mathcal{F}^+$ 分配给选定的检查员 $\mathcal{N}_i$，从而确保每位$\widehat{\ }$检查员都被分配到一组特征 $\mathcal{F}_j$。由于特征 $\mathcal{F}$ 的数量有限，递归特征分配可保证最终将所有特征都分配给检查员。

另一方面，主动通信协议可确保探索者与每位检查员主动通信多次，以尽量减少因等待造成的空闲时间。此外，检查员会将检查结果 $\mathcal{S}$ 传输给探索者。由于检查员 $\mathcal{N}_\circ$和通信事件的数量有限，收集所有检查结果 $\mathcal{S}$ 所需的总时间也有限。■

D. 在线执行与适应

1) 探索任务滚动分配：由于地面控制站完全掌握所有边界框 $\mathcal{B}$ 的位置，并在地面控制站-子群组层直接与探索机器人协调，因此由地面控制站将这些边界框分配给机器人集群是合理的。此外，地面控制站并非一次性分配所有边界框，而是采用滚动时域下的滚动分配策略。这种方式尤其有用，因为 $\mathcal{B}$ 中边界框的实际结构未知，难以预测每个边界框的探索时间。在初始阶段，地面控制站将每个探索者 $i \in \mathcal{N}_\text{e}$ 分配到最近的边界框 $B_i \in \mathcal{B}$。然后，根据边界框 B_i 的体积 V_{B_i}，为每个探索者分配特定数量的巡检机器人，组成一个子组，记为 $\mathcal{N}_\circ^i \subset \mathcal{N}_\circ$，即

$$|\mathcal{N}_\circ^i| = \left\lfloor |\mathcal{N}_\circ| \frac{V_{B_i}}{\sum_i V_{B_i}} \right\rfloor, \tag{14}$$

其中，$|\mathcal{N}_\circ|$ 是巡检机器人的总数；$\lfloor\cdot\rfloor$ 表示向下取整函数。在线执行期间，当地面控制站与探索者在指定位置 p_m^c 会合时，每个探索者都会向地面控制站报告当前的 B_i 探索状态。一旦 B_i 完成探索并反馈所有已检查特征，地面控制站便会将下一个最近的边界框分配给整个子组执行，直至所有边界框均已分配，如图 8 所示。

2）处理会合时间不匹配：由于运动不确定性和跟踪误差，实际到达指定会合地点的时间可能与计划的会合时间不同，尤其是在未知或复杂环境中。这称为会合时间不匹配。为应对这一挑战，提出了一系列机制：（I）如果地面控制站等待探索者的时间超过给定容差 $\delta > 0$，即 $\tau_0 \geqslant \delta$，则地面控制站会假定该探索者发生故障，并转而与下一位探索者通信。反之，如果探索者等待地面控制站的时间超过 δ，即 $\tau_i^+ \geqslant \delta$，探索者必须等待地面控制站到达；（II）在一个子组内，如果探索者等待巡检机器人的时间超过 δ，探索者将继续执行其局部规划。由于探索者完全掌握所有巡检机器人的局部规划，因此会将下一次预定会合事件追加到该巡检机器人的局部规划末尾。如果在下一次会合时再次超过这一容差，探索者便会假定该巡检机器人发生故障，随后将讨论故障恢复机制。最后，当可能出现调度冲突，导致探索者需要在相近时间分别与地面控制站和巡检机器人会合时，地面控制站具有更高优先级。这样的优先级排序是合理的，因为 GCS-explorer 通信发生得不频繁，而探索者与巡检机器人的通信则更为频繁因此，剩余数据可以延后至后续会合时传输。

3）探索过程中的局部自适应：在三维在线探索过程中，尽管通过算法1可根据现有前沿综合得出每个探索者 $i \in \mathcal{N}_\text{e}$ 的最优路径 $\xi_i^\star$，但探索过程中可能会产生新的前沿，因此需要进行局部自适应。为解决这一问题，可采用滚动时域方式触发算法1，将新的前沿加入现有集合并重新规划路径。这个迭代重新规划过程会持续进行，直到探索者在 t_m^c内到达已确认的会合地点 p_m^c。

定理1。所提出的*SLEI3D*框架可确保所有已检查特征 $\mathcal{S}$ 在有限时间内由地面控制站收集，同时使(8)中的总空闲时间最小化。

证明。引理1保证探索机器人在有限时间内拟合所有特征 $\mathcal{F}$，而引理2则确保检查员在有限时间内将所有特征 $\mathcal{F}$ 分配、检查并传输给探索机器人。在间歇通信协议下，地面控制站求解带时间窗的旅行商问题，在预定时间 t_m^c到达会合地点 p_m^c，届时探索机器人会传输已检查特征 $\mathcal{S}$。由于特征数量 $\mathcal{F}$ 和通信轮次均有限，地面控制站可在有限时间内收集所有检查结果 $\mathcal{S}$。此外，在线适应策略能够有效缓解不确定性造成的延误和故障。因此，SLEI3D 框架保证所有特征 $\mathcal{F}$均可在有限时间内收集并送回地面控制站。另一方面，如问题1所重新表述，地面控制站到子群组层的算法会协调地面控制站与探索机器人的会合事件，以尽量缩短等待时间。如问题3所述，SOEI算法$(\cdot)$会协调每个子组中的探索机器人和检查员，以尽量缩短空闲时间。因此，给定当前已知的特征集合，机器人集群与地面控制站的整体规划可使（8）中的总空闲时间最小化，并最大限度地提高收集已探索和已检查特征的效率。证毕。■

E. 泛化

1) 机器人故障与恢复：考虑以下情况：(I) 探索者故障：假设探索者 i 在 $t_\text{f} > 0$发生故障，地面控制站会在预定的时间 t_m^c 和位置 p_m^c 等待探索者 i，最长等待时间为 δ。随后，地面控制站判定探索者 i已发生故障，并在其局部规划中直接转向下一位探索者。之后各轮的会合事件将不再考虑发生故障的探索者。此外，一旦另一个子组完成对其分配边界框的探索，就会将其重新分配到发生故障的探索者所负责的边界框，以便与 $\mathcal{N}_\circ^i$中的其余巡检机器人一起重新开始探索与检查任务。(II) 巡检机器人故障：假设巡检机器人 $j \in \mathcal{N}_\circ^i$ 在 $t_\text{f} > 0$发生故障，同一子组中的探索者 i 在等待最长时长 δ后会检测到其故障。随后，探索者 i 将通过更新 $\mathcal{N}_\circ^i \leftarrow \mathcal{N}^i{}_\circ \backslash \{j\}$，将巡检机器人 j 从子组中移除，然后应用算法2重新分配其余巡检机器人。

除了上述措施外，机器人之间（GCS-explorer 或 explorer-inspector）的通信不稳定性还可以通过冗余来

解决。基线机制会在超过 δ 的长时间断连后识别故障，但这可能会对瞬态中断反应过度。因此，我们提出一种硬件层的恢复机制：每台机器人都配备一个专用网状网络，可在通信中断期间启用。该网络能够立即转发关键状态更新，同时维持本地协调。这种集成式方法能够确保机器人发生永久性故障以及通信网络暂时退化时的韧性。

2）多地面控制站：在大规模场景中，可能会有多个地面控制站。在这种情况下，假设各地面控制站具备相似的能力，且彼此始终保持连接，例如能够了解探索进度和已检查特征。对于与探索机器人的通信，由于部分探索机器人已预先指定与某个特定地面控制站通信，因此应修改第IV-B节中的间歇通信协议，将单个带时间窗的旅行商问题替换为多个并行的带时间窗的旅行商问题，以便为每个地面控制站生成导航与通信局部规划，从而提高整体效率。

3）高优先级特征：如果特征具有不同的优先级，例如某些特征可能需要立即响应，则可通过为与高优先级特征相关的巡检任务添加优先级约束，来修改算法2中的特征分配模块。也就是说，应采用类似带优先级或优先约束的多车辆路径问题（MVRP-PC）的算法。这样便可优先检查高优先级特征，同时将总空闲时间降至最低。

4）自发会面：在执行过程中，地面控制站、探索机器人或巡检机器人可能会在已确认的会合地点之外的其他位置相遇，这种情况称为自发会面。在这种情况下，它们会交换各自的局部数据，但不会协调下一次会合事件，也不会修改各自的局部规划，从而确保所有已确认的会合仍然有效。

5）完全未知环境：当缺少有关边界框的先验信息时，本文提出一种边界框自适应分区方法，根据协同探索的实时进展动态分配未探索区域 $\mathcal{W}$ 。需要指出的是，该方法依赖于有关环境形状与大小的先验信息；对于形状不规则的工作空间，会先用规则边界框将其框定，再进行划分。如图4（b）所示，整个空间 $\mathcal{W}$ 最初被划分为 $\{B_j\}$个体积相等的立方子空间 $|\mathcal{N}_{\mathrm{e}}|$。随后，每个子组通过上述框架自主探索分配给它的 B_j ，并在完成后通知地面控制站。地面控制站通过 $\Psi \triangleq \mathcal{W}\backslash\bigcup_{j\in\mathcal{I}_{\mathrm{comp}}} B_j$跟踪未探索前沿的集合，其中 $\mathcal{I}_{\mathrm{comp}}$ 为已探索区域的集合。该集合 Ψ 通过迭代八叉树分区动态划分为新的边界框集合，同时保持所有坐标轴对齐。当 $\Psi=\varnothing$ 或最小值(维度$(B_j)) \leqslant 10.0$米时，迭代终止。后续巡检按照算法2所述流程进行，通过层次分解确保完全覆盖。

6）能量约束下的舰队管理：当机器人能量容量有限且需要充电时，我们可以通过在各层进行修改来采用我们的框架。首先，机器人模型应纳入这些约束。具体而言，每个机器人 $i \in \mathcal{N}$ 的能量容量有限，为 $E_i(t) \leqslant \underline{E} \in \mathbb{R}^+$，在运行过程中会以 $E_i(t)=-\alpha_i$ 的速率变化，而在指定充电站 $P_{\mathrm{chg}} \subset \mathcal{W}$充电时，则以 $\beta_i > 0$ 的速率$E_i(t)=\beta_i$ 变化 $\alpha_i > 0$ 。当$E_i(t) \leqslant E \in \mathbb{R}^+$ 且机器人位于充电站时，将触发充电过程，充满电所需时长为 $t_{\mathrm{b}}=(E-\underline{E})/\beta_i$ 。该约束形式化表述如下： $0<\underline{E}_i(t) \leqslant \overline{E}, \quad \forall t \in[0, \quad T], \quad \forall i \in \mathcal{N}$。

为解决上述约束，所提框架进行了以下三项关键修改：（I）如第 IV-B2 节所述，对已分配边界框的完成时间进行预测时，必须将充电时长考虑在内。因此，地面控制站与探索机器人之间规划的通信时间 t^{c}_{m+1}修改如下：

$$\tilde{t}^{\mathrm{c}}_{m+1}=t^{\mathrm{c}}_{m+1}+\left\lceil\frac{t^{\mathrm{c}}_{m+1}-t^{\mathrm{c}}_{m}}{\overline{E}-\underline{E}}\right\rceil\cdot(t_b+T_{\mathrm{chg}}), \qquad (15)$$

其中， $\lceil\cdot\rceil$ 用于计算最小充电周期数， T_{chg} 表示从四个角点之一到充电站的往返行程时间， t_b 为最大充电时长；（II）在按照第 IV-B4 节所述进行地面控制站至子群组的协调时，地面控制站与探索机器人之间规划的会合时间 t^{c}_m 也应在以下情况下作相应修改：（i）当剩余能量 $(E_0-E)/\alpha \geqslant \Delta t_m$, 时，其中 $\Delta t_m = t^{\mathrm{c}}_m - t^{\mathrm{c}}_{m-1}$ 表示两次连续会合之间的时间间隔，规划的 t^{c}_m 仍满足能量约束。（ii）如果$(E_0-E)/\alpha < \Delta t_m$,地面控制站需要中途充电, 则会合时间调整如下：

$$\tilde{t}^{\mathrm{c}}_m=t^{\mathrm{c}}_m+\left\lceil\frac{\Delta t_m-E_0}{\overline{E}-\underline{E}}\right\rceil\cdot(T^0_{\mathrm{chg}}+t_b),$$

其中， T^0_{chg} 表示会合点 p^{c}_m 与充电站之间的往返行程时间；（III）在每个子组内部进行协调时，可对通过求解 MVRP 得到的初始分配 ϕ_i（如第 IV-C2 节所述）进行重新规划，为每个巡检机器人生成不受能量约束的路径序列 $\mathcal{F}^+$ ，即预测每个巡检机器人的局部路径上的能量消耗，并按需插入充电事件。随后巡检事件的计划时间戳也会相应调整。

V. 数值实验

为进一步验证所提方法的有效性，我们针对大规模系统开展了大量数值实验，并将其性能与若干最先进的方法进行比较。所提方法基于 ROS 框架，使用 Python3 实现，并在配备 Intel(R) i9-13900KF 24核CPU @3.0GHZ 和 RTX-4090 GPU 的工作站上进行测试。仿真视频可见补充材料。

A. 系统描述

仿真机器人集群由作为探索者和巡检机器人的无人机，以及作为地面控制站的无人地面车辆组成，并部署在轻量级模拟器 [57], [58]中。如图 10 所示，

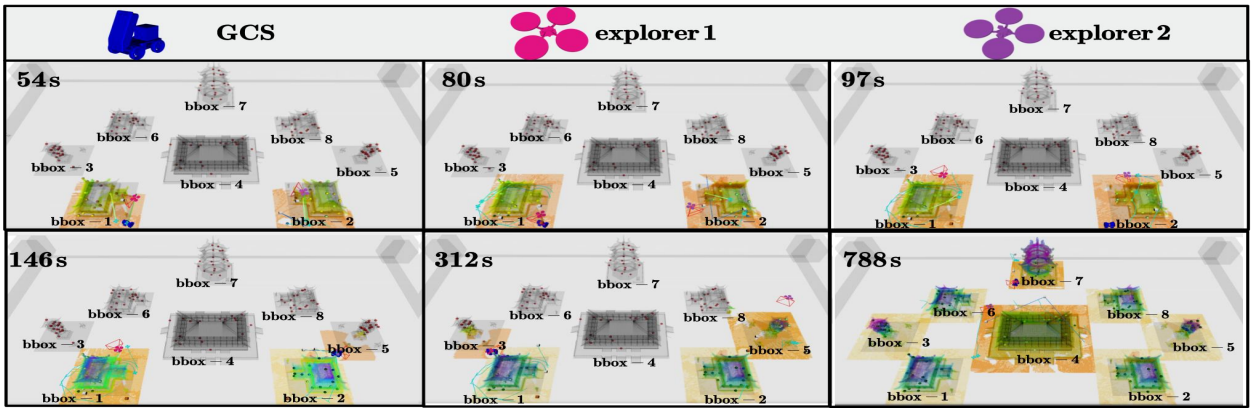

图 8： 场景A中包含 1 个地面控制站、2 个探索机器人和 4 个巡检机器人的 GCS到子组层快照。在 $t = 146s$ 时，BBox-5 被重新分配给探索机器人 2；在 $t = 312s$ 时，BBox-3 被重新分配给探索机器人 1。

考虑以下四 种场景。场景A：在规模为 $60 \times 50 \times 9m^3$ 的古代建筑集群中部署 1 个地面控制站、2 个探索者和 4 个巡检机器人，以验证第 V-B1 节所述 GCS至子群层的性能；场景-B：在规模为 $18 \times 20 \times 15m^3$ 的大型单体建筑结构中部署 1 个探索者和 3 个巡检机器人，以验证第 V-B2 节所述子组层的性能；场景-C：在规模为 $70 \times 60 \times 30m^3$的中型城市中部署 1 个地面控制站、4 个探索者和 8 个巡检机器人，以验证第 V-B3 节所述整体方案的性能；场景-D：在规模为$120 \times 160 \times 20m^3$ 的大型古代遗址中部署 1 个地面控制站、8 个探索者和 40 个巡检机器人，以验证其在大规模场景中的整体性能。

为弥合高保真仿真中理论验证与实际部署之间的差距，我们的实验框架由三个经过精心设计的核心模块构成：感知、规划和通信。系统采用一种新颖的轻量级架构，其实现方式不同于传统的 Gazebo 工作流，具体包括：(I) 通过传感器仿真实现感知真实性：仿真激光测距仪和 RGB-D 相机，并配置噪声模型（测距误差为 ±2cm，深度失真为 5%），在复现 Gazebo 插件功能的同时消除计算开销。(II) 环境抽象：通过集成 RVIZ，将作业空间离散为点云表示（体素分辨率为 0.1m），在保留几何保真度的同时实现实时碰撞检测。(III) ROS 原生互操作性：标准化的消息接口确保仿真平台与物理平台之间能够无缝切换。系统实现细节如下：通过 [59] 生成三维欧几里得有符号距离场（ESDF）地图，作为局部地图；使用 5 m范围深度相机和里程计传感器构建环境的体积地图。随后，感兴趣区域（AoI）识别流程以 0.5 Hz的频率并行处理同步数据流：利用 MiDas 网络和 DeepLabV3+ 预测深度与语义信息，分析 RGB 图像以重建几何特征，并通过 YOLOv7 进行目标识别。每个探索者和巡检机器人都使用动力学运动规划器进行导航，以生成无碰撞路径；路径包含多个在拓扑上相异的航路点，以刻画三维环境的结构复杂性，最大速度为 $2m/s$ ，加速度为 $2m/s^2$。配备视场（FOV）相机的巡检机器人可执行特征拟合、检测和检查；该相机左右视角为 $90°$，前视角为60°，检测距离为 $5\ m$。特征进入视场后，探索机器人或巡检机器人可自动通过式（4）中的 Fit(·)进行特征拟合，或通过式（5）中的 Inspect(·)进行检查。此外，机器人集群中的通信范围限制为 $5m$，且地面控制站、探索机器人和巡检机器人之间保持视距（LOS）。

B. 结果

以下总结了与四种场景相关的结果，包括对机器人故障、多地面控制站和高优先级特征的泛化。最后，针对机器人群规模、各种不确定性以及巡检任务的不同持续时间，进行了可扩展性和鲁棒性分析。

1）地面控制站至子群层的评估：如图 8 所示，场景A包含 1 个地面控制站、2 个探索机器人和 4 个巡检机器人，需要探索包含 8 个边界框的区域。为启动探索过程，地面控制站根据机器人群规模和边界框数量，采用滚动分配策略，将两名巡检机器人分配给每个探索机器人，组成子群。随后，将这些子群分配至最近的边界框，平均分配时间为 $1ms$。接着，每个探索机器人由算法1引导，平均规划时间为 $26ms$；该算法根据当前位置、速度和已确认的会面事件 $\mathcal{C}$，动态调整探索路径。同时，通信协调算法的平均计算时间为 $3.5ms$，用于确定地面控制站与探索机器人之间的最优访问顺序。例如，如图 8 所示，在第一次会面期间，地面控制站先前往 BBox- 1 与探索机器人 2 会面，然后前往 BBox-2 与探索机器人 1 会面，接着返回 BBox-1 再次与探索机器人 2 交互，最后再次前往 BBox-5 与探索机器人 1 会面。式（9）中的任务完成时间预测算法平均耗时 $0.47ms$，平均预测误差约为 $13.2s$。以下时间线可进一步说明执行过程。在 $t = 54s$，地面控制站按照与探索机器人 2 已确认的会面事件行动，并提前抵达

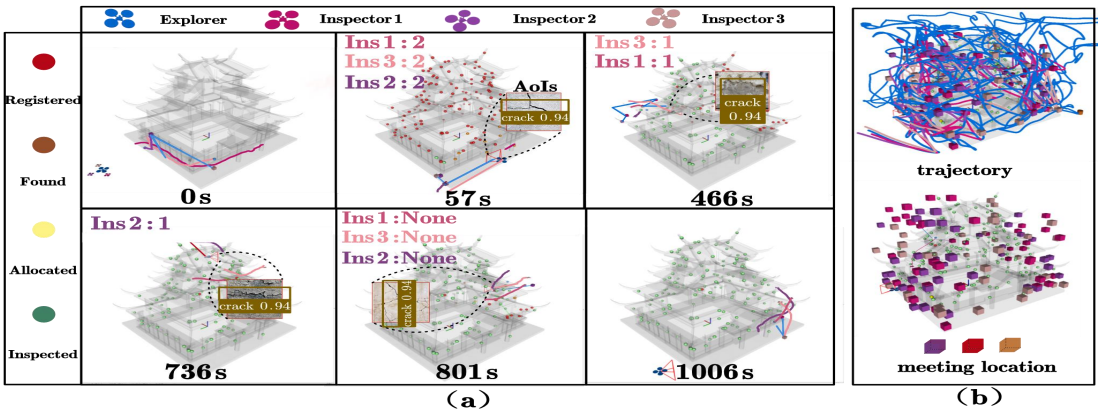

图9： 场景-B中由一台探索机器人和三台巡检机器人组成的子群层演化过程：（**a**）探索机器人和巡检机器人在不同时间点的规划轨迹快照；（**b**）所有机器人的完整轨迹及完整会合事件。

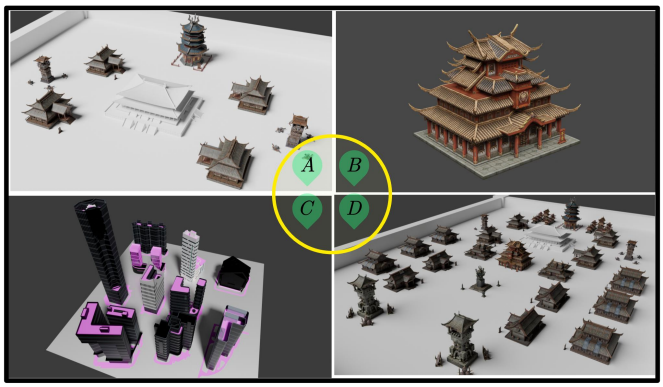

图10： 数值实验中测试的场景：包含一组小型古代建筑的场景A；包含一座单独古代建筑的场景-B；包含中等规模城市建筑的场景-C；以及包含大型古代遗址的场景-D。

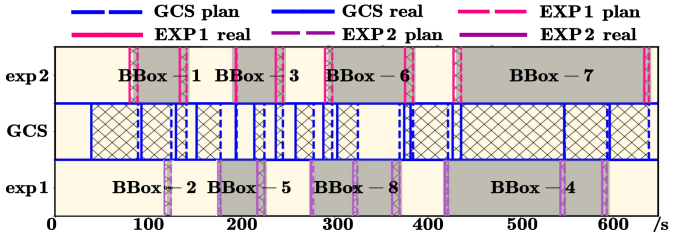

图11： 场景A中地面控制站与两台探索机器人之间的规划会合事件和实际会合事件。地面控制站的规划会合时间（蓝色虚线）与探索机器人（粉色和紫色虚线）的时间相一致，表明会合事件和目标边界框已得到确认。实线表示实际到达时间，该时间通过网格区域与同一会合事件的规划时间相绑定。

BBox-1 的会面地点，如图 11 所示。在 $t = 80s$，探索机器人 2 抵达指定地点与地面控制站会面。同样，地面控制站于 $t = 97s$在预定时间之前抵达 BBox-2 的会面地点，等待探索机器人 1。在 $t = 146s$，探索机器人 1 与地面控制站会面，并接到探索 BBox-5 的新任务。于 $t = 312s$ 与地面控制站会面后，探索机器人 1 发送已检查特征，地面控制站将最近的 BBox-3分配为探索机器人 1 的新任务。在 $t = 788s$ 之后，地面控制站收集齐全部 150 个特征。

此外，如图11所示，每个边界框的会合次数少于3次，即所提出的算法能够准确预测每项探索任务的完成时间。此外，值得注意的是，所有探索机器人在会合事件中的实际到达时间始终早于计划会合时间，这表明算法1能够确保它们在指定时间窗口内及时到达，从而减少地面控制站和探索机器人的空闲时间。

2）子群组层评估：本节重点介绍场景A中的组内动态（如图10所示），以及不同检查时间 t_q 对整体性能的影响。

(I) 整体执行。如图9所示，在场景-B中执行任务期间，系统会监控子组内机器人的运动和动作，同时进行探索和特征拟合。如果发现了新特征，算法2便会每隔 $5s$ 触发一次，以优化子组解 Ξ_i，平均计算时间为 $1.384s$。具体而言，给定所有检查员正在执行的局部规划 $\{\xi_j^-\}$ 后，探索者会使用 SmapleLos$(\cdot)$ 函数，其平均运行时间为 $1.3s$；同时，OptMeet$(\cdot)$ 的平均运行时间为 $0.04ms$ ，用于优化检查员 $\mathcal{N}_i$ 和会合事件$\mathbf{c}_{i\circ}$ 随后，探索者调用经过修改的 MVRP$(\cdot)$算法，将新拟合的特征分配给选定的检查员，平均运行时间为 $0.08s$。最后，探索者调用 Navi$(\cdot)$ ，根据检查员正在执行的局部规划 $\{\xi_j^-\}$ 和分配的特征 φ_i更新其局部规划 $\{\xi_j^+\}$ ，平均计算时间为 $0.05s$。在通信阶段，探索者将分配的特征 φ_i 和更新后的局部规划$\{\xi_j^+\}$ 发送给相应的检查员，确保子组内任务的一致性。以下描述了几个有代表性的时刻：在 $t = 57s$，探索者决定先与巡检机器人 1 会合，并为其分配2个特征；然后与巡检机器人 3 会合，并为其分配2个特征；最后与巡检机器人 2 会合，并为其分配2个特征。在 $t = 466s$，探索者先与巡检机器人 3 会合，为其分配1个特征，随后与巡检机器人 1 会合并为其分配1个特征。由于行驶距离较长，探索者避免与巡检机器人 2 会合，从而减少了总空闲时间。在 $t = 736s$，探索者选择只与巡检机器人 2 会合，并将新发现的特征分配给它。这些特征都在巡检机器人 2 附近，而且没有剩余特征需要巡检。在 $t = 801s$，由于所有检查员都在执行

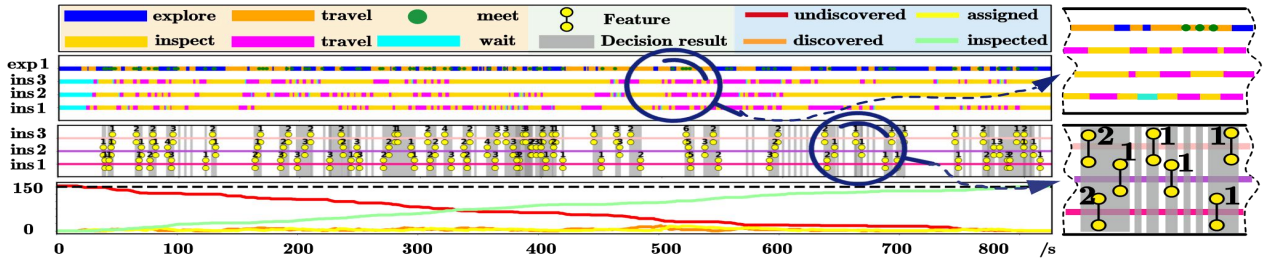

图 12：顶部：场景-B 中子组内所有机器人的状态变化：探索机器人处于探索（深蓝色）、行进（深橙色）和会合（绿色）状态；巡检机器人处于巡检（浅橙色）、行进（粉色）和等待（青色）状态。**中部**：探索机器人在与巡检机器人会合期间的规划结果，以灰色方框表示。时间线上的锤形符号及不同巡检机器人表示计划会合事件的顺序，以及分配给每个巡检机器人的特征数量。**底部**：不同特征相关状态随时间的变化：未发现（红色）、由探索机器人发现（橙色）、分配给巡检机器人（黄色）以及由巡检机器人检查（绿色）。

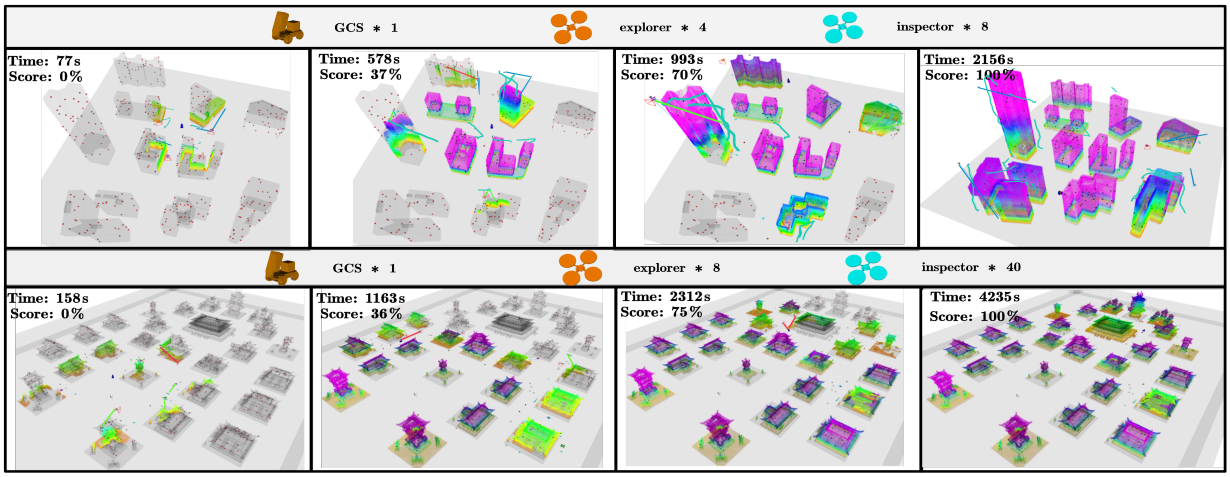

图 13： 场景-C 在 1 个地面控制站、4 个探索机器人和 8 个巡检机器人条件下的整体执行快照（**顶部**）；以及场景-D 在 1 个地面控制站、8 个探索机器人和 40 个巡检机器人条件下的整体执行快照（底部</style）。

巡检任务且距离较远，探索者没有与任何检查员会合。此外，还展示了子组内所有机器人的完整轨迹以及子组的会合地点。可以看出，由于需要在线适应与所有检查员的会合事件，探索者的轨迹相当复杂；而随着特征逐步添加到检查员的局部规划中，检查员的轨迹则更加平滑。

最后但同样重要的是，图 12 展示了每个机器人的状态在线变化，例如“探索”“行进”“检查”和“等待”；“已登记”“已发现”“已分配”和“已检查”特征数量的变化；以及探索机器人与巡检机器人之间的通信事件。可以看出，探索机器人将 60.35% 的时间用于探索，并覆盖整个场景中的感兴趣区域。巡检机器人主要前往特征所在位置进行检查，平均等待时间为 $33s$。此外，中部图显示，探索机器人在不同时间会与全部、部分或没有巡检机器人会合，平均通信周期为 $6.8s$。底部子图显示，150 个特征在 $1006s$ 内全部完成检查，覆盖率达到 100%。

(II)不同检查时间。为评估不同检查时间 t_q 对我们所提方法性能的影响，我们使用不同的 t_q进行了一系列实验，范围从 $1s$到 $10s$。用于比较的指标包括每次会合分配的特征平均数量。结果表明，分配的特征数量随着 t_q 从 $1s$ 增加到 $8s$，单调地从 3.11 降至 1.65。因此，如果检查这些特征所需的时间更短，就需要更频繁地会合。

3）全流程仿真：为验证图 13 所示的整体性能，我们针对不同规模的工作空间和机器人集群开展了大规模仿真：场景-C 配置了 1 个地面控制站、4 个探索机器人和 8 个巡检机器人；场景-D 配置了 1 个地面控制站、8 个探索机器人和 40 个巡检机器人。如前所述，这两个层级以不同的频率集成。在场景-C 中，地面控制站与探索机器人之间的平均会合周期为 $40.6s$，探索机器人与巡检机器人之间的平均会合周期为 $6.8s$。相比之下，场景-D 中这两个频率分别为 $63.2s$ 和 $5.5s$。滚动分配策略通过动态重新分配边界框，进一步提升了不同子组的效率。在场景-C 中，探索机器人于 $578s$和 $993s$完成任务，随后将其重新分配至附近的边界框，以最大限度地减少空闲时间。同样，在场景-D 中，探索机器人于 $1163s$和 $2321s$重新分配。平均而言，场景-C 中每个探索机器人分配到 2.5 个边界框，场景-D 中每个探索机器人分配到 3.3 个边界框，平均任务时间分别为 $529s$ 和 $513s$。这表明，所提方法能够在不同场景下有效平衡各子组的任务分配。值得注意的是，在两种场景中，探索机器人均能以 100% 的成功率按时到达会合地点；而在场景-D 中，地面控制站偶尔会因杂乱环境中的任务重叠而出现轻微延迟

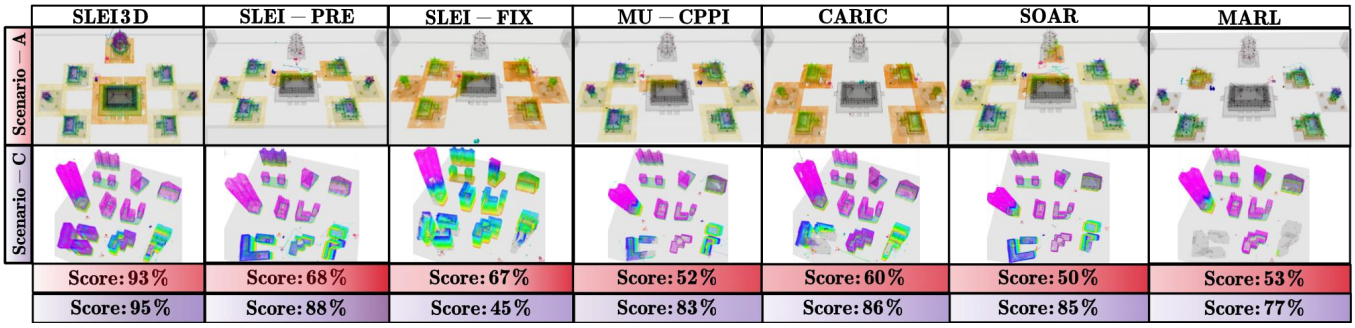

图14： 所提方法（SLEI3D）与场景A和场景-C中六种基线方法的最终得分比较。

。此外，在两种场景中，地面控制站与各子组之间的平均通信次数均少于 3 次，表明所提算法能够有效预测完成时间。子群层在复杂环境中也展现出良好的适应性，例如场景-C 中杂乱的建筑物以及场景-D 中错综复杂的结构。在这两种情况下，所提方法在场景-C 中对 $1006s$、在场景-D 中对 $2006s$ 均实现了 100% 的特征覆盖。

整体效率受到多种因素的影响，其中探索机器人的探索效率对于提升特征发现和检查过程尤为关键。量化探索效率的一个重要指标是重叠率，它反映了机器人之间的冗余覆盖。在我们的框架中，通过为机器人集群中的专用探索者预先分配边界框（边界框），完全消除了组间探索重叠。然而，在四个场景中，单个探索者执行组内操作而非多探索者探索，其平均重叠率为15%，即在场景A–D中分别为20%，10%，13%，和17%。这一结果与 [6]中单机器人探索的最先进水平重叠率一致。通过严格保持组间重叠率为零，并将组内重叠抑制在最低水平，所提方法通过系统性减少冗余探索，显著提升了探索效率。

4) 通信约束下的最大机器人群规模：首先，我们的实验测量结果表明，在场景 B–D 中，地面控制站与各子组之间每次交互的平均通信带宽约为 1.9 Mbps（分别为 2.018 Mbps、1.755 Mbps 和 1.940 Mbps）。带宽消耗与每次会面传输特征的平均数量（11、9.5 和 10.5）成正比。值得注意的是，即使在我们的问题设置所固有的特征密集条件下，系统仍能保持这种带宽效率（1.9 Mbps），这表明该算法能够在保持运行保真度的同时实现紧凑的信息编码。该框架采用**单对单链路** 范式来应对通信异步性，其中成对交互严格避免通过共享带宽，以瞬时且集中式的方式进行多机器人同时通信。因此，除了底层通信硬件和任务环境所施加的限制外，该框架不会对可行的机器人群规模施加额外约束。

C. 比较

为进一步验证所提方法（即 **SLEI3D**）的有效性，我们将其与**六种** 有力的基线方法进行了定量比较：

(i)**CARIC**，该方法基于 [47] 中提出的同步探索与检查分层框架。机器人被划分为多个团队，每个团队独立分配至特定子区域，然后通过视距通信将检查结果传回静态地面控制站。为确保比较公平，我们进行了修改，使地面控制站能够移动并与探索机器人通信。

(ii)**SOAR**，该方法基于 [46] 中的工作。它仅由一台探索机器人识别未知区域并生成视点，随后通过全局全对全通信将这些视点分配给巡检机器人。然而，该方法无法处理有限通信条件下涉及多台探索机器人和巡检机器人的场景，也未考虑将信息传回地面控制站的需求。为确保与我们的方法公平比较，SOAR 仅用于将特征从探索机器人分配给巡检机器人，其余组件则遵循所提框架，以适应问题设定。

(iii)**MU-CPPI**，该方法基于 [45], 中提出的先探索后检查框架。机器人首先探索整个环境，然后根据探索结果生成局部检查规划。然而，该方法未考虑可移动地面控制站、有限通信约束以及同步探索与检查。因此，该方法只能用于子组层，以引导子组中的探索机器人与巡检机器人会合。其余组件遵循所提框架，以匹配问题设定。

(iv) **多智能体强化学习**（MARL），一种通过多智能体强化学习（MARL）实现协作探索的多机器人策略，如 [32]–[34] 所示。这些工作开发了端到端协调策略，对探索策略和无碰撞运动规划进行联合优化。然而，它们仅关注问题表述中的探索阶段，未解决群间与群内的通信需求。因此，如第 IV-B3 节所述，MARL 模块仅用于替代探索机器人的探索策略，同时

表I 两个场景下的基线比较。

场景	方法	完成率（%）				完成时间（s）				GCS-EXP 会面（#）				EXP-INS 会面（#）				总浪费时间（s）			
		平均值	标准差	最大值	最小值	平均值	标准差	最大值	最小值	平均值	标准差	最大值	最小值	平均值	标准差	最大值	最小值	平均值	标准差	最大值	最小值
场景A	CARIC	97.6	2.1	100	96	1009	29.6	1037	978	7.3	0.6	8	7	57.3	1.15	58	56	1967	72.4	2018.3	1884.2
	SLEI-PRE	92	2	94	90	988.7	116.3	1100	868	10.3	0.6	11	10	90	6.1	97	86	2259.4	292.9	2586.2	2020.5
	SOAR	73	1	74	72	1110	141.1	1254	972	10.3	1.5	12	9	174	14.1	189	161	2716.1	603.9	3304.6	2097.8
	MU-CPPI	83.7	2.5	86	81	1239	72.4	1322	1189	9.7	1.5	11	8	83	4.4	86	78	3104.1	467.9	3616.8	2700.1
	SLEI-FIX	96	2.7	98	93	939.7	66.30	1002	870	10	0	10	10	52.7	1.2	54	52	1551.9	85.9	1636.6	1464.8
	MARL	92.4	4.2	97.1	87.5	932.5	120.3	1060	800.3	13.5	4.2	18	9	93.5	13.2	108	88	2218.8	512.9	3000.8	1737.2
	SLEI3D	**94.0**	1	94	93	**743.7**	208.5	976	573	**13.7**	3.5	17	10	**84.7**	11.9	98	75	**1434.5**	17.0	1454.1	1423.6
场景C	CARIC	98.7	0.6	99	98	2659	317.5	3007	2385	8.7	0.6	9	8	182.7	11.1	194	172	10032	2178.2	12288	7940.4
	SLEI-PRE	95.3	1.2	96	94	1767	102.9	1882	1684	38.3	1.2	39	37	199.36.8		207	194	8085.5	411.1	8459.7	7645.4
	SOAR	94	1	95	93	1908	54.2	1954	1848	43	3	46	40	415.3	11.2	425	403	8542.1	616.3	9229.8	8039.9
	MU-CPPI	94.7	0.6	95	94	2117	51.5	2172	2070	44	10.2	55	35	165.329.1		198	142	9725.3	1048.1	10742	8648.6
	SLEI-FIX	96.7	1.2	98	96	2462	365.5	2870	2165	14	0	14	14	188.725.5		218	172	8809.4	490.1	9252.3	8282.9
	MARL	77.8	5.5	85	70	1648	210.8	2000	1314	37	0	37	37	162.728.6		200	130	8479.1	504.2	9000.1	7800.5
	SLEI3D	**96.7**	1.5	98	95	**1699**66.7		1776	1658	**42**	3.5	44	38	**279.7**80.8		353	193	**7920.3**733.6		8767.4	7496.8

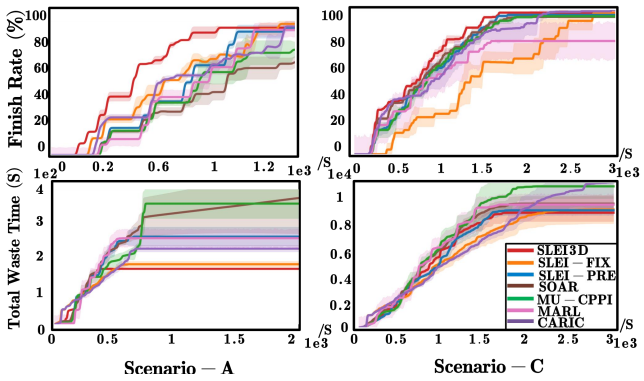

图15： 所提方法与六种基线方法在三次运行中随时间采集到的特征数量对比。

保留所提框架的其他组件，以确保系统兼容性。(v) **SLEI-PRE**，其中所有边界框最初都分配给探索机器人，即每个探索机器人按照固定顺序探索指定的边界框，而不是依赖地面控制站进行动态分配。(vi) **SLEI-FIX**，其中地面控制站在整个过程中始终停留在其初始位置。因此，探索机器人需要按固定间隔返回地面控制站进行通信。

上述方法在场景A和场景-C中均进行了评估，每个场景各进行3次测试。实验结果汇总于图14和表I。图15展示了地面控制站随时间采集到的特征，其中场景A的执行时间为 600s ，场景-C为 1660s。在两个场景中，与CARIC（1009s 和 2659s）、SOAR（1110s 和 1908s）、MU- CPPI（1239s 和 2117s）、多智能体强化学习（932.5s 和 1648s）、SLEI- PRE（988.7s和 1767s）以及SLEI-FIX（939.7s 和 2462s）相比，SLEI3D在场景A和场景-C中收集所有特征所用时间最短（因此效率最高），分别为 743.7s 和 1699s 。此外，显然，与SLEI-FIX（939.7s和 2462s）相比，采用可移动地面控制站有助于更快地完成任务并减少空闲时间。此外，尽管SLEI3D和SLEI-PRE的完成率相近（（94.0%和96.7%）与（92%和95.3%）），但边界框的动态分配与SLEI- PRE相比，可大幅减少空闲时间。SLEI3D还促进了探索机器人与巡检机器人之间更有效的协作。这一点从探索机器人与巡检机器人之间更多的会合事件中可见一斑，尤其是在场景-C中，SLEI3D的平均会合次数为279.7次，而SLEI-PRE为199.3次，SLEI-FIX为188.7次。与CARIC相比，这种差异更为显著。CARIC采用较简单的组内通信策略，但缺乏适应复杂任务的能力。SLEI3D能够更快地完成任务（（743.7s和 1699s）与（1009s 和 2659s）相比），且空闲时间更少（（1434.5s和 7920.3s）与（1967s和 10032s）相比）。此外，专门针对场景A训练的多智能体强化学习在迁移至场景-C后，性能显著下降。如表I所示，两个场景间的成功率从92.4%大幅降至77.8%，足以说明这一点。相比之下，我们的方法展现出更强的泛化能力，在两个场景中均保持稳定的成功率（94.0%与96.7%）。这种性能差距凸显了多智能体强化学习等纯学习方法的固有局限性：它们往往会对训练环境过拟合。我们的混合架构通过自适应任务分配机制和几何感知协同有效缓解了这一问题，使其在熟悉和全新的场景中都能保持鲁棒的性能。此外，多智能体强化学习表现出较高的不稳定性，成功率偏差较大（4.2%与5.5%）；而SLEI3D在多次运行中则展现出稳定的鲁棒性（1.0%与1.5%）。最后，SLEI3D在两个场景中的空闲时间均最短，场景A和场景-C中分别为 1434.5s 和 7920.3s；相比之下，CARIC为（1967s 和 10032s）、SOAR为（2716.1s 和 8542.1s）、MU- CPPI为（3104.1s 和 9725.3s）、多智能体强化学习为（2218.8s 和 8479.1s）、SLEI-PRE为（2259.4s 和 8085.5s），SLEI-FIX为（1551.9s和 8809.4s）。总之，SLEI3D有效应对了CARIC、SOAR和多智能体强化学习等最先进的框架，以及SLEI-FIX和SLEI-PRE等SLEI变体所带来的挑战。

此外，如图16所示，与基线方法（CARIC、SOAR 和 MU-CPPI）的计算时间对比分析表明，SLEI3D 的平均计算时

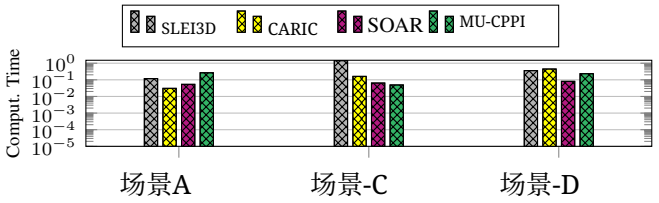

图16： SLEI3D 与其他方法在不同场景下计算时间的比较。

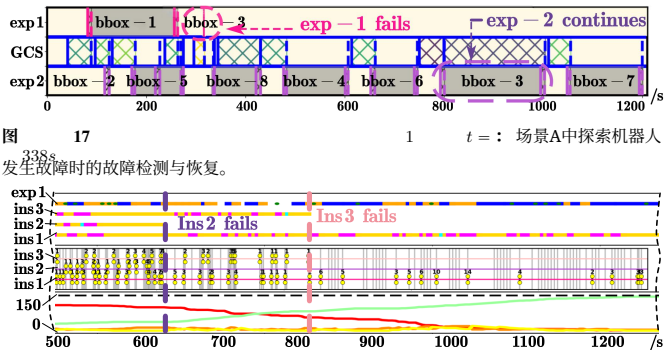

图 17 1 $t =$ ： 场景A中探索机器人发生故障时的故障检测与恢复。338s

图18： 场景B中巡检机器人 2 首先在 $t = 683s$发生故障，随后巡检机器人 3 在 $t = 825s$ 发生故障时的故障检测与恢复。

间较长（0.6325 s vs. 0.2128 s vs. 0.0662 s vs. 0.1827 s），这主要是由于 SOEI(·) 优化所致。MU-CPPI 的处理时间最短，而 SLEI3D 的计算时间稳定（0.118–1.4225s）。尽管计算成本较高，SLEI3D 的方法仍带来了显著的性能提升，正如表I中的分析所示。

D. 泛化

1) 机器人故障：如第 IV-E 节所述，所提方法能够在执行过程中检测并恢复探索机器人和巡检机器人可能发生的机器人故障。具体而言，参数δ 被设置为 $10s$。

(I)探索器故障。考虑与第 V-B1 节相同的设置：在场景A中配置 1 个地面控制站、2 个探索机器人和 4 个巡检机器人，其中一个探索机器人在线发生故障。地面控制站与探索机器人之间的最终会合结果如图 17 所示。探索机器人 1 在 $t = 338s$ 之前于 BBox-3 与地面控制站会合时发生故障。地面控制站等待 $10s$ 后，在预定义的时间和位置检测到该故障。随后，地面控制站直接前往 BBox-8 的下一会合事件，与探索机器人 2 进行通信。如图 17 所示，与计划时间相比，地面控制站的等待时间导致其到达 BBox-8 时出现轻微延迟。根据所提故障恢复机制，隶属于探索机器人 2 的整个子组被重新分配至 BBox-3 以重新开始任务，因为发生故障的探索机器人 1 无法将 BBox-3 的检查结果中继给地面控制站。整体任务于 $1215s$完成，时间略长于图 8 中无故障情况下的 $788s$ 。

（II）**检查器故障**。考虑与第 V-B2 节相同的设置，但两台巡检机器人在执行过程中连续发生故障。机器人状态、决策结果和特征完成状态的总体变化总结于图 18 中。巡检机器人 2 在 $t = 683s$ 与探索者会面之前发生故障，探索者在其局部规划结束时等待 $10s$ 后检测到该故障。随后，巡检机器人 2 被

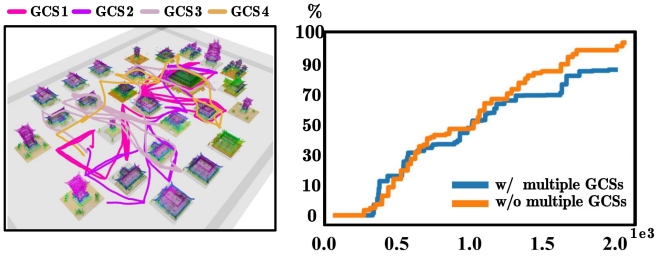

图 19： 场景-D，包含 4 个地面控制站、8 个探索机器人和 40 个巡检机器人：所有地面控制站的最终会合轨迹（左）；随时间变化的采集到的特征对比（右）。

排除在探索者已知的活动检查器集合之外。此外，很明显，一旦探索者检测到巡检机器人 2 发生故障，就不会再向巡检机器人 2 分配任何特征。巡检机器人 3 在 $t = 825s$发生故障后，探索者采取类似的方式检测到该故障。最后，整体任务在 $t = 1380s$完成，这一时间略高于无故障标称场景中的 $1006s$ 。值得注意的是，在巡检机器人 2 和 3 发生故障后，探索者倾向于减少会面次数，但向巡检机器人 1 分配更多特征。

2) 多地面控制站：在场景-D中，针对两种配置对所提出的 SLEI3D 算法进行了评估：一种采用包含 4 个地面控制站、8 个探索机器人和 40 个巡检机器人的多地面控制站设置，另一种采用单地面控制站配置。具体而言，每个地面控制站都预先分配了特定的边界框子集和一组探索机器人，每个探索机器人均独立运行。图 19 展示了所有地面控制站的最终轨迹，以及采集到的特征随时间的变化。显然，每个地面控制站都成功与分配给自己的探索机器人会合，从而收集到所有特征。不过值得注意的是，与单地面控制站配置相比，采用多个地面控制站的 SLEI3D 在初期能够更快地收集特征。然而，随着进程推进，SLEI3D 逐渐追上并最终超越了多地面控制站配置的表现。这是因为所有探索机器人都从同一位置出发，并被分配了大小相近的边界框。换言之，针对多个地面控制站提出的路由方案，使这些边界框的完成时间几乎同步，从而各地面控制站能够同时从各自分配的边界框中收集特征。相比之下，SLEI3D 只能依次从多个边界框中收集特征，因此特征收集速度慢得多。随着任务推进，待探索的边界框子集在体积上差异显著，导致完成时间的分布更加多样。因此，随着特征收集速率越来越不依赖于分布式协调，多个地面控制站的优势也随之减弱。这表明，多地面控制站配置尤其适用于需要在分布广泛的位置同时收集特征的场景。

3) 高优先级特征：为进一步验证 SLEI3D 对高优先级特征的处理效果，我们从场景-B 的 150 个特征中选取 50 个作为高优先级特征。图20 展示了巡检机器人轨迹的快照以及这些特征的优先级，表明高优先级特征会优先得到巡检。也就是说，巡检机器人

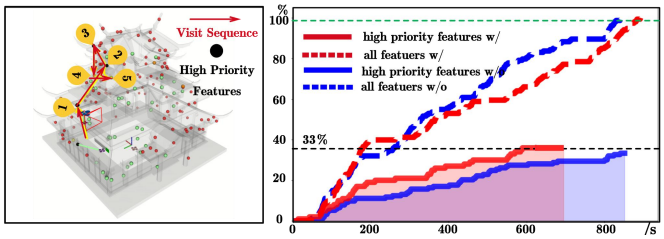

图 20： 场景-B中包含150个特征和50个高优先级特征时的高优先级特征：高优先级特征的检查序列（左）；已收集的高优先级特征随时间的变化对比（右）。

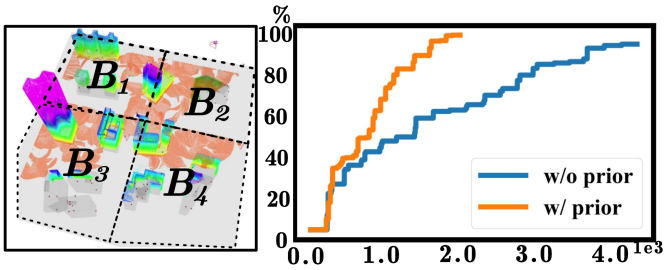

图 21： 场景-C下系统在没有任何先验信息时的整体演化（左）；已收集特征随时间的变化对比（右）。

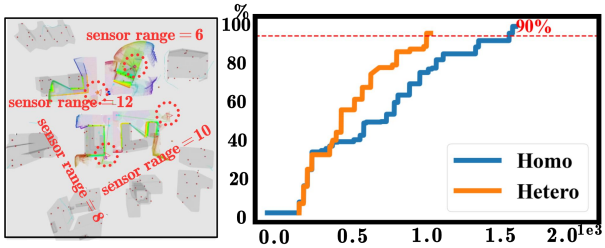

图 22： 场景-C下，配备1个地面控制站、4个探索机器人的异构传感器范围（左）；已收集特征随时间的变化对比（右）。

会优先处理 5 个特征中的 3 个高优先级特征。此外，图中还展示了高优先级特征或所有特征随时间的收集情况。可以看出，与 SLEI3D 相比，考虑高优先级特征的 SLEI3D 收集高优先级特征的速度快得多，但总体完成时间略长（$842s$与 $893s$ 相比）。这是因为优先处理高优先级特征会妨碍对所有特征的分配进行同步优化，从而导致总体完成时间增加。

4) 完全未知环境：为验证 SLEI3D 在完全未知环境中的能力，在 1 个地面控制站、4 个探索机器人和 8 个检查机器人条件下，针对场景-C 开展了有无边界框先验信息的对比实验。如图 21 所示，无先验信息的探索中，探索机器人的特征发现率明显较低。这一延迟源于缺少边界框，迫使探索机器人穷举探索均匀子空间，而无法优先探索目标区域。因此，地面控制站收集特征的时间显著延长（1832 秒对比 4000 秒），验证了所提出自适应划分机制的有效性。这也凸显了边界框先验信息能够加速探索与检查整体任务的完成。

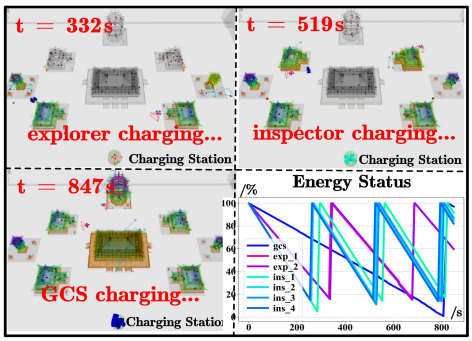

图 23： 场景A下，不同角色的机器人返回固定充电站进行充电的代表性快照，以及整个机器人群能量状态的整体演化。

5) 探索机器人的异构能力：为验证对异构机器人集群的支持，将场景-C 中的探索机器人设置为具有不同的感知范围（$6m$、 $8m$、 $10m$、 $12m$），并与采用统一 $5m$传感器的同构单元进行对比。如图 22 所示，借助更大的感知范围，异构系统能够更快地发现特征并调度检查机器人。与同构系统相比，收集约 90% 特征所需的时间减少了 40%（$1005s$对比 $1656s$）。值得注意的是，地面控制站通过滚动时域分配，优先将具有更大感知范围的探索机器人分配至更大的边界框，这与式（7）所示的时间最小化目标一致。结果证实，所提出框架能够适配异构机器人集群，且整体性能与传感器范围的增加呈正相关。

6) 能量约束下的舰队管理：为验证所提框架能够处理第 IV-E6 节所述的能量约束，在场景A中部署了一支队伍，包括1个能量为$E_0 = 800\ mAh$ 的地面控制站、2个探索机器人（$E_i = 400\ mAh,\ \forall i \in \mathcal{N}_e$）和4个巡检机器人（$E_j = 300\ mAh,\ \forall j \in \mathcal{N}_o$）。当任何机器人的能量水平低于$E = 20\%E$时，便会启动统一的充电协议，所有充电站的充电速率均为每秒 $\beta = 10\%E$。图 23 中记录的关键事件展示了系统在能量约束下的整体效率：在$t = 332$ 秒时，探索机器人 1达到临界阈值后启动充电协议（如能量状态图所示）；随后，巡检机器人 3也在 $t = 540$ 秒时发生了类似事件。地面控制站在$t = 780$ 秒时按计划充电，同时维持与探索机器人的预定通信事件。在878秒的任务时长内，所有感兴趣区域均完成检查，队伍分别为地面控制站、探索机器人和巡检机器人进行了1次、2次和3次充电。在任何时刻，所有机器人的能量水平均高于5%。

E. 计算复杂度分析

为便于从业者在更广泛的场景中应用所提框架，本文分析了该框架在不同规模、不同检查时间和机群规模下的计算复杂度。具体而言，计算时间主要由三部分构成：（I）地面控制站的局部规划，包括通信事件的协调和边界框的滚动分配；

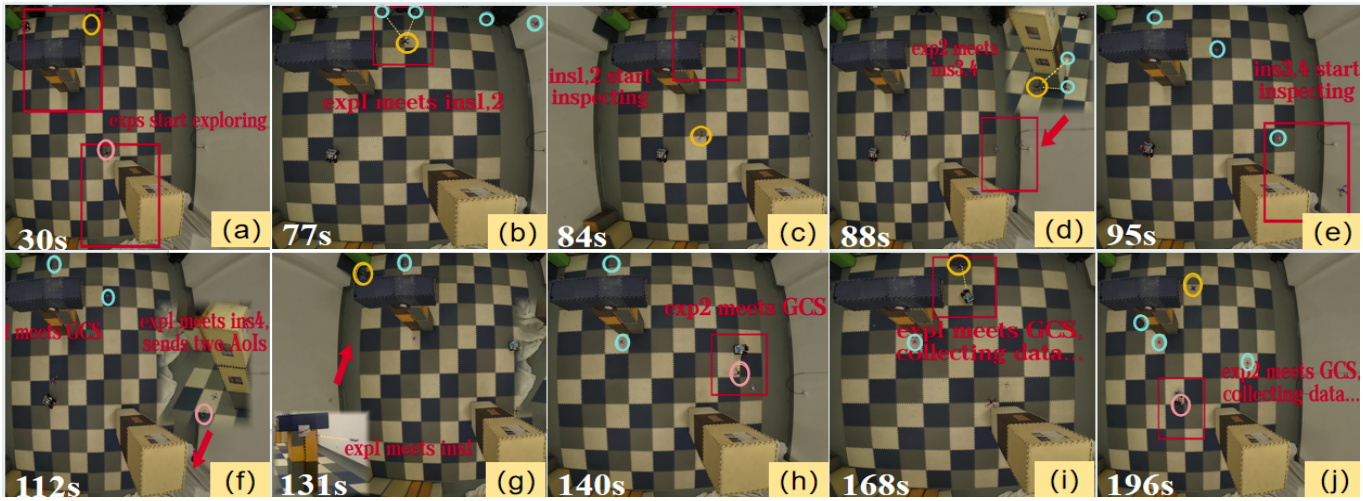

图 24： 涉及 2 台探索机器人（以黄色和粉色圆圈标记）、4 台检查员（以靛蓝色圆圈标记）和 1 个地面控制站的硬件实验快照，展示了对 2 个结构和 9 个感兴趣区域进行协同检查的过程。**(a)**：探索机器人开始探索；**(b)**：探索机器人 1 识别出 2 个感兴趣区域，并与检查员 1 和 2 协调；**(c)**：将感兴趣区域分配给检查员 1 和 2；**(d)**：探索机器人 2 发现另外 2 个感兴趣区域，并与检查员 3 和 4 协调；**(e)**：将感兴趣区域分配给检查员 3 和 4，开始检查；**(f)**：探索机器人 1 首次向地面控制站传输数据；**(g)**：探索机器人 1 发现另一个感兴趣区域，并与检查员 1 协调；**(h)**：探索机器人 2 首次与地面控制站建立通信；**(i)**：探索机器人 1 将已检查特征及后续会合情况传输至地面控制站；**(j)**：探索机器人 2 将已检查特征及后续会合情况传输至地面控制站。

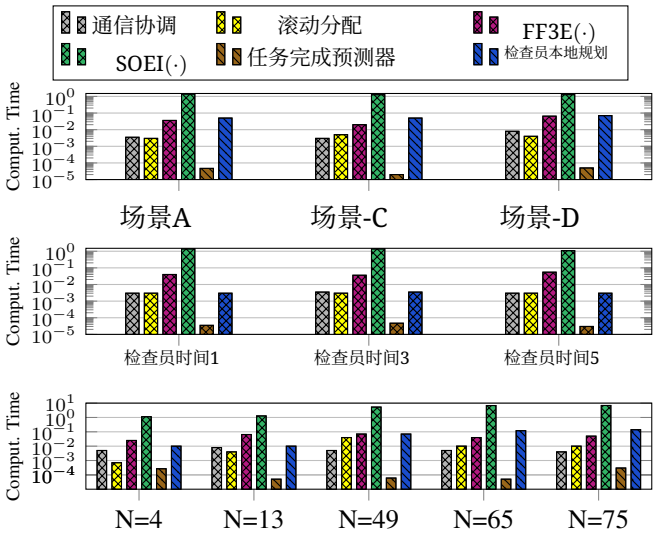

图25： 所提方法主要组件在三个场景中的计算时间（上）；场景-C中不同检查时间下计算时间的变化（中）；场景A中不同机群规模下的计算时间（下）。

（II）探索机器人的局部规划，包括用于探索的 FF3E(·)、通过 SOEI(·) 进行的规划以及完成时间预测器；（III）巡检机器人的局部规划，包括局部规划的更新$\{\xi_j^+\}$。

首先，分析了上述各步骤在不同场景下的计算时间：在图 10 的三个场景中，部署 1 个地面控制站、4 个探索机器人和 8 个巡检机器人，并采用检查时间 $t_q = 3s$。如图 25 所示，在所有场景中，除探索机器人的 SOEI(·) 外，所有模块的计算时间均少于 $0.1s$。SOEI(·) 算法依赖迭代优化过程，因此占据了地面控制站局部规划的大部分计算时间。尽管如此，每个步骤的总计算时间平均均少于 $1.3s$，在所有场景中的平均值。其次，在场景-C 中，针对不同的检查时间 t_q，即 $1s$、$3s$ 和 $5s$，对 1 个地面控制站、4 个探索机器人和 8 个巡检机器人的情况进行了相同的分析。结果表明，检查时间对 SOEI(·) 模块的计算时间影响较小，但每个步骤的总体计算时间平均均少于 $0.25s$。此外，还在场景A中针对不同机群规模进行了相同分析，机群规模分别为 4（1 个地面控制站、1 个探索机器人、2 个巡检机器人）、13（1 个地面控制站、4 个探索机器人、8 个巡检机器人）、49（1 个地面控制站、8 个探索机器人、40 个巡检机器人）、65（5 个地面控制站、10 个探索机器人、50 个巡检机器人）和 75（5 个地面控制站、10 个探索机器人、60 个巡检机器人）。结果表明，随着机群规模增大，计算时间稳步增长，平均值从 $0.13s$ 增至 $6.91s$；这是因为探索机器人和巡检机器人的数量增加时，规划需要更多交互才能收敛。尽管如此，$N = 75$ 的总体计算时间仍低于 $8.82s$，展现了其应用于大规模机器人集群的可扩展性。

F. 硬件实验

1) 实验设置：我们还进行了硬件实验，以验证所提框架的实际可行性。如图 24 所示，实验场地 $5m \times 5m \times 3m$内有两个微型自然三维结构：一个尺寸为 $1.8m \times 0.3m \times 1.9m$ 的 T形结构和一个尺寸为 $0.5m \times 0.5m \times 2.0m$的矩形结构。其表面上共印有 9 个作为感兴趣特征的裂缝，位置未知。此外，两架配备单目摄像机的“Tello”无人机 [60] 作为探索者，通过三维重建执行探索任务。这些探索者以 $0.7m/s$的速度飞行，使用运动学-动力学规划器避障，通信范围为 $0.4m$。此外，四架“Crazyflie”无人机 [61]在 $0.3m/s$,处作为巡检机器人运行，采用改进的 A* 算法，并使用基于优先级的避碰方法，通信范围为 $0.2m$。表面上的感兴趣区域由 YOLOv7 软件包识别，语义分割模块引导探索者的视场朝向目标建筑物；通过融合 RGB 图像与估计的深度信息重建三维环境，其中深度信息由 MiDas 网络根据“Tello”的单目图像预测。

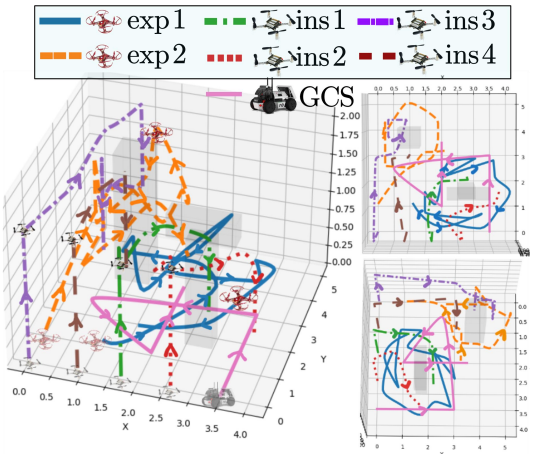

图 26： 所有 8 台机器人的完整轨迹，覆盖 $200s$ 的任务时长；箭头表示不同视图中的时间线：前视图（**左侧**）、顶视图（**右上**）和侧视图（**右下**）。

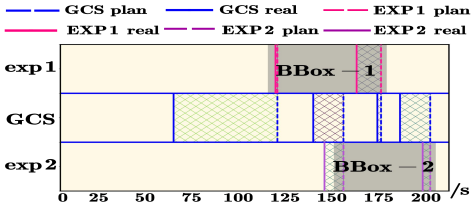

图 27： 硬件实验中地面控制站与两台探索机器人的计划会合事件和实际会合事件。地面控制站（蓝色虚线）计划的会合时间与探索机器人（粉色和紫色虚线）的会合时间一致，表明会合事件已确认，并确定了目标边界框。

此外，语义分割模块引导探索者的视场朝向目标建筑物，并通过融合 RGB 图像与估计的深度信息来重建三维环境；深度信息由 MiDas 网络根据“Tello”的单目图像预测 [62]。最后，一台“LIMO” [63] 四轮差速地面机器人充当可移动地面控制站，使用 EAI XL2激光雷达进行实时建图，并利用 move_base 软件包实现自主导航。所有机器人均由配备 20 台红外摄像机的 OptiTrack动作捕捉系统跟踪，以获取定位数据；同时，所有机器人间通信都严格受视距（LOS）约束限制。需要指出的是，机器人之间的通信约束由中心节点集中实施，包括相对距离、障碍物遮挡和带宽。

2）结果：硬件实验验证了所提框架在 $200s$任务中的性能。如图 24 所示，2 个探索机器人于 $t = 30s$开始探索，如图 24(a) 中的红框所示。探索机器人 1 在 $t = 77s$ 发现了 2 个可能包含裂缝的感兴趣区域，并通过后续的通信会面与巡检机器人 1 和 2 协调，如图 24(b) 所示。感兴趣区域按照 SOEI算法进行分配，引导巡检机器人前往各自的区域，如图 24(c) 所示。与此同时，探索机器人 2 于图 24(d) 中的 $t = 88s$ 发现了另外 2 个感兴趣区域，并在图 24(e-f) 中与巡检机器人 3 和 4 协调。同时，探索机器人 1 于图 24(f) 中的 $t = 112s$ 首次向可移动地面控制站传输数据，其中包括

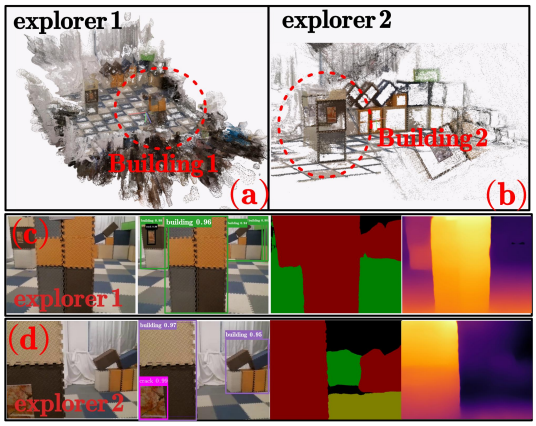

图 28： 两台搭载单目摄像机的探索机器人在三维环境中的感知结果，包括：探索机器人 1（(**a**)）和探索机器人 2（(**b**)）重建的三维环境；探索机器人 1（(**c**)）和探索机器人 2（(**d**)）基于原始相机图像的感兴趣区域和建筑结构识别结果、感兴趣区域检测、语义分割和深度估计。

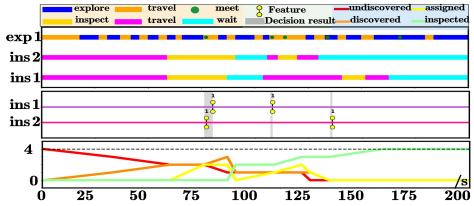

图 29： **顶部**：一个子组内机器人状态的变化；**中部**：探索机器人 1 与巡检机器人 1 和 2 在会合事件期间的规划结果；**底部**：随时间变化的已发现、已分配和已检查特征数量。

已检查特征及后续会面。在 $t = 131s$，探索机器人 1 发现了另一个感兴趣区域，并与巡检机器人 1 协调；与此同时，探索机器人 2 于图 24(h) 中的 $t = 140s$ 首次与地面控制站建立通信。两个探索机器人均严格遵循已确认的会面事件，最后一次与地面控制站通信分别如图 24(i-j) 中的 $t = 168s$ 和 $t = 196s$ 所示。此时，地面控制站已获得所有三维结构和已检查的感兴趣区域。此外，图 26 记录了所有机器人的轨迹，表明：（i）探索机器人实现了完全覆盖，且所有感兴趣区域均已成功检查，如图 28 所示；（ii）地面控制站在四个位置与两个探索机器人协调，以实现最优会面事件，如图 27 所示；（iii）所有机器人均未与其他机器人或三维结构发生碰撞。最后，图 29 总结了第一个子组中机器人状态的详细演变，验证了所提方案在运动不确定性和避碰导致导航时间波动时的鲁棒性。硬件实验的详细视频见补充材料。

VI. 结论

本研究针对有限通信条件下异构机器人集群在大规模未知三维环境中同步探索、检查与通信的挑战。所提出的 SLEI3D 框架采用多层多速率策略，根据感知能力将机器人集群划分为多个子组：配备远程传感器的探索者负责识别感兴趣区域（AoI），而

配备近距离传感器的检查员则负责检查这些特征。地面控制站（GCS）分配任务，并通过地面控制站与探索者之间的间歇式协议，以及探索者与检查员之间的主动式协议来促进通信，从而简化任务分配和数据收集。大量仿真验证了该框架的可扩展性，以及其在各种场景和大规模机器人集群中的适用性。

未来工作包括：（I）通过实现多机器人探索来提高子组的探索效率，这需要更加复杂的通信协议；（II）开发不依赖先验信息（如边界框）的探索策略；（III）引入多样化的特征类型和巡检机器人角色，以应对更加复杂的任务需求；（IV）集成更加逼真的通信模型，包括非视距（NLOS）、信号衰减和多径效应，从而增强系统在现实复杂环境中的鲁棒性。

参考文献

[1] M. Hinostroza and Lekkas，“采用多机器人执行自主巡检与维护作业：初步结果”，载于《IEEE/ASME机电一体化嵌入式系统应用国际会议》（MESA）论文集，2024年，第1–8页。[2] K. Zhang and Pakrashi，“使用多旋翼无人机对海上浮式风力涡轮机进行巡检：文献综述与趋势”，《传感器》，第24卷，第3期，第911页，2024年。[3] J. Jiang et al.，“桥梁缆索维护机器人的关键结构与技术——综述”，《机器人智能与自动化》，第45卷，第1期，第121–143页，2025年。[4] M. Afrazi and K. Lee，“机器人技术在隧道检测与维护中的应用：综合综述”，《地下基础设施进展》，第1–24页，2025年。[5] G. Best et al.，“具备完整任务空中自主能力的多机器人多传感器多样化环境探索”，《国际机器人研究杂志》（IJRR），第43卷，第4期，第485–512页，2024年。[6] C. Wang et al.，“未知环境中的协同探索多机器人系统：综述”，arXiv预印本，arXiv:2503 07278，2025年。。[7] B. Yamauchi，“一种用于自主探索的基于边界的方案”，载于《IEEE机器人与自动化国际会议》（ICRA）论文集，1997年，第146–151页。[8] R. G. Colares and L. Chaimowicz，“下一个边界：结合信息增益与距离成本的分布式多机器人探索”，载于《ACM应用计算专题讨论会》论文集，2016年，第268–274页。[9] B. Zhou et al.，“Racer：采用分布式多无人机系统的快速协同探索”，《IEEE机器人学汇刊》（T-RO），第39卷，第3期，第1816–1835页，2023年。[10] I. Patil et al.，“用于快速多机器人搜索的基于图的同步覆盖与探索规划”，arXiv预印本，arXiv:2303 02259，2023年。。[11] M. Naazare, F. G. Rosas, and D. Schulz，“面向移动操作机器人的三维探索与巡检在线下一最佳视点规划器”，《IEEE机器人与自动化快报》，第7卷，第2期，第3779–3786页，2022年。[12] C. Gao et al.，“基于无人机的先探索后利用系统，用于自主室内设施检测与场景重建”，《自动化施工》，第148卷，第104753页，2023年。[13] M. Allan et al.，“面向月球探索任务的行星探测车仿真”，载于《IEEE航空航天会议》论文集，2019年，第1–19页。[14] M. S. Couceiro，“群体机器人学在搜救应用中的概述”，《人工智能：概念、方法、工具与应用》，第1522–1561页，2017年。[15] M. L. Brutto, A. Borruso, and A. D’argenio，“用于考古遗址摄影测量数据采集的无人机系统”，《数字时代国际遗产杂志》，第1卷，第1期，第7–13页，2012年。[16] P. P. Ray，“机器人事物互联网：概念、技术与挑战”，《IEEE访问》，第4卷，第9489–9500页，2016年。[17] G. Hu, W. P. Tay, and Y. Wen，“云机器人：架构、挑战与应用”，《IEEE网络》，第26卷，第3期，第21–28页，2012年。[18] T. Murayama and A. Iwasaki，“考虑视距通信的多机器人网络双连通性控制”，《机器人与机电一体化杂志》，第35卷，第4期，第1028–1037页，2023年。

[19] A. R. Mosteo，《服务机器人中的多机器人任务分配：从无限通信范围到有限通信范围》，博士学位论文，萨拉戈萨大学，2010年。[20] M. Guo 和 M. M. Zavlanos，《缓冲区约束和间歇通信条件下的多机器人数据采集》，IEEE Robotics Transactions (TR-O)，第34卷，第4期，第1082–1097页，2018年。[21] M. Saboia 等，《Achord：面向间歇性连接的通信感知型多机器人协同》，IEEE Robotics and Automation Letters，第7卷，第4期，第10 184–10 191页，2022年。[22] Y. Kantaros 等，《移动机器人网络的时序逻辑任务规划与间歇性连接控制》，IEEE Transactions on Automatic Control，第64卷，第10期，第4105–4120页，2019年。[23] A. R. da Silva 等，《间歇性会合条件下通信受限的多机器人探索》，arXiv 预印本 arXiv:2309 13494，2023年。[24] K. Milidonis 等，《无人机在聚光太阳能热系统规划、运行与维护中的应用：综述》，Solar Energy，第254卷，第182–194页，2023年。[25] Z. Tian 等，《 ihero：通信资源稀缺条件下的交互式以人为本探索与监督》，arXiv 预印本 arXiv:2405 12571，2024年。[26] S. S harma 和 R. Tiwari，《多机器人区域探索技术与算法综述》，载于《国际计算技术与信息通信技术会议论文集》(ICCTIC)，IEEE，2016年，第151–158页。[27] X. Hu 和 H. Pei，《多机器人协同探索综述》，载于《Journal of Physics: Conference Series》，第2798卷，第1期，2024年，页码012037。[28] A. Hussein 等，《搜救任务中的多机器人任务分配》，载于《Journal of Physics: Conference Series》，第570卷，第5期，2014年，页码052006。[29] Q. Dong 等，《基于动态拓扑图上的 Voronoi 划分实现快速且通信高效的多无人机探索》，arXiv 预印本 arXiv:2408.05808，2024年。[30] F. Cabrera-Mora 和 J. Xiao，《一种用于多机器人探索的泛洪算法》，IEEE Transactions on Systems, Man, and Cybernetics，第42卷，第3期，第850–863页，2012年。[31] D. P. Doolittle，《群体遗传学：基本原理》。Springer Science & Business Media，2012年，第16卷。[32] G. Battocletti 等，《未知环境中自主飞行器的基于强化学习的路径规划》，载于 AIAA AVIATION 2021 论坛，2021年，页码3016。[33] S. Zhu 等，《 Maexp：基于强化学习的多智能体探索通用平台》，载于 IEEE ICRA 会议论文集，2024年，第5155–5161页。[34] J. Chen 等，《利用多机器人系统对三维不确定结构进行基于元强化学习的协同表面检测》，载于 IEEE ICRA 会议论文集，2024年，第7201–7207页。[35] F. Amigoni 等，《通信受限环境中的多机器人探索：综述》，IEEE Intelligent Systems，第32卷，第6期，第48–57页，2017年。[36] T. Vaquero 等，《自主多漫游车协作开展火星洞穴探索的方法：初步结果》，载于国际人工智能、机器人与太空自动化研讨会 (SAIRAS) 论文集，2018年。[37] K. Cesare 等，《有限通信和电池条件下的多无人机探索》，载于 IEEE ICRA 会议论文集，2015年，第2230–2235页。[38] Y. Gao 等，《会合—合并—任务：用于大规模通信受限探索的多机器人协调框架》，载于 IEEE IROS 会议论文集，2022年，第13 700–13 707页。[39] E. Stump 等，《基于可见性的机器人编队部署以维持通信》，载于 IEEE ICRA 会议论文集，2011年，第4498–4505页。[40] Y. Pei 等，《移动机器人网络中兼顾连接性与带宽的实时探索》，Wireless Communications and Mobile Computing，第13卷，第9期，第847–863页，2013年。[41] M. N. Rooker 和 A. Birk，《无线网络约束下的多机器人探索》，Control Engineering Practice，第15卷，第4期，第435–445页，2007年。[42] C. Feng 等，《FC-Planner：一种骨架引导规划框架，用于复杂三维场景的快速空中覆盖》，载于 IEEE ICRA 会议论文集，2024年，第8686–8692页。[43] W. Jing 等，《利用路径基元采样和基元覆盖图进行视觉检测的覆盖路径规划》，载于 IEEE IROS 会议论文集，2019年，第1472–1479页。[44] ——，《面向无人机三维视觉覆盖任务的基于采样的视点规划。收录于2016年IEEE》，载于 IEEE IROS 会议论文集，第1815卷，1808，页码2016。[45] ——，《用于大型复杂结构检测的多无人机覆盖路径规划》，载于 IEEE IROS 会议论文集，2020年，第1480–1486页。

[46] M. Zhang 等，《SOAR：异构无人机协同探索与拍摄，实现快速自主重建》，arXiv 预印本 arXiv:2409.02738，2024年。[47] X. Xu 等，《异构空中系统的一种高性价比协同探索与检查策略》，arXiv 预印本 arXiv:2403.01225，2024年。[48] W. Sheng 等，《有限范围通信条件下的多机器人区域探索》，载于 IEEE IROS 会议论文集，2004年，第1414–1419页。[49] A. R. Mosteo 等，《有限通信范围下的多机器人路径规划》，载于 IEEE ICRA 会议论文集，2008年，第1531–1536页。[50] M. Santos 和 M. Egerstedt，《利用有限通信实现具有异构感知能力的多机器人团队覆盖控制》，载于 IEEE IROS 会议论文集，2018年，第5313–5319页。[51] H. Khalil，《非线性系统》，第3版。新泽西州上萨德尔河：Prentice Hall，2002年。[52] A. Hornung 等，《OctoMap：一种基于八叉树的高效概率三维建图框架》，Autonomous Robots，第34卷，第189–206页，2013年。[53] H. Chen 等，《一种用于遥感图像语义分割的改进 Deeplabv3+ 轻量级网络》，Complex & Intelligent Systems，第10卷，第2期，第2839–2849页，2024年。[54] C.-Y. Wang 等，《YOLOv7：可训练的免费技巧集合为实时目标检测器树立新标杆》，载于 IEEE/CVF 计算机视觉与模式识别会议 (CVPR) 论文集，2023年，第7464–7475页。[55] B. Zhou 等，《FUEL：利用增量式前沿结构和分层规划实现快速无人机探索》，IEEE Robotics and Automation Letters，第6卷，第2期，第779–786页，2021年。[56] B. Alhijawi 和 A. Awajan，《遗传算法：理论、遗传算子、求解方法与应用》，Evolutionary Intelligence，第17卷，第3期，第1245–1256页，2024年。[57]《Marsim》。 [在线]。地址：https://github.com/hku-mars/MARSIM。[58]《FUEL》。 [在线]。地址：https://github.com/HKUST-Aerial-Robotics/ FUEL。[59] L. Han 等，《FIESTA：用于空中机器人的在线运动规划的快速增量式欧几里得距离场》，载于 IEEE IROS 会议论文集，2019年，第4423–4430页。[60]《Tello》。 [在线]。地址：https://www.ryzerobotics.com/cn/tello。[61]《Crazyflie》。 [在线]。地址：https://www.bitcraze.io/products/old-products/crazyflie-2-1/。[62] R. Birkl 等，《MiDaS v3.1——用于稳健单目相对深度估计的模型库》，arXiv 预印本 arXiv:2307.14460，2023年。[63]《LIMO》。 [在线]。地址：https://www.agilex.ai/education/4。