Dear Editor and Reviewers:

First and foremost, we sincerely thank the Editor-in-Chief, Associate Editor, and anonymous reviewers for their insightful comments and suggestions. These have been carefully incorporated into the revised manuscript, resulting in a clearer, more informative, and more readable version. We acknowledge the reviewers' concerns regarding grammatical errors, unclear definitions, and poorly formatted figures, which were noted as obstacles to evaluating the paper's technical contributions. In this revision, we have thoroughly addressed these issues by eliminating grammatical errors, refining definitions and notations for clarity, and replotting the figures to enhance readability. All reviewer comments have been carefully addressed, and the corresponding changes have been highlighted in blue for ease of reference. Our detailed responses to the reviewers' comments, provided in a 1:1 correspondence, are outlined below. Yours sincerely,

Ziqi Yang, Roberto Tron

# 1 Response to reviewer 1

**Comment**

This paper proposes a multi-agent mobile robot planning algorithm that is robust agains attackers that would compromise the agents by directing them to unsafe regions. This is achieved by incorporating 1) co-observation constraints so that the agents could watch over each other, 2) reachability constraints so that the region the agents could potentially enter does no overlap with unsafe regions, and 3) sub-team planning algorithm to improve robustness by adding more number of agents.
The proposed method is interesting, but I think the writing must be improved significantly to secure the clarity, and there are a lot of grammar mistakes that need to be fixed. To me, it appears that the draft may have been prepared somewhat hastily, which is not considerate of the reviewers' time.

**Response:** We are grateful for your thoughtful comments and for expressing interest in our proposed method. We sincerely apologize for the grammatical mistakes in the initial submission and acknowledge the importance of presenting our work with clarity and professionalism. In the revised version, we have been extra careful to address all grammar issues and have implemented thorough proofreading to minimize similar mistakes in future submissions. We greatly appreciate your patience and constructive feedback, which have been invaluable in improving the quality of our manuscript.

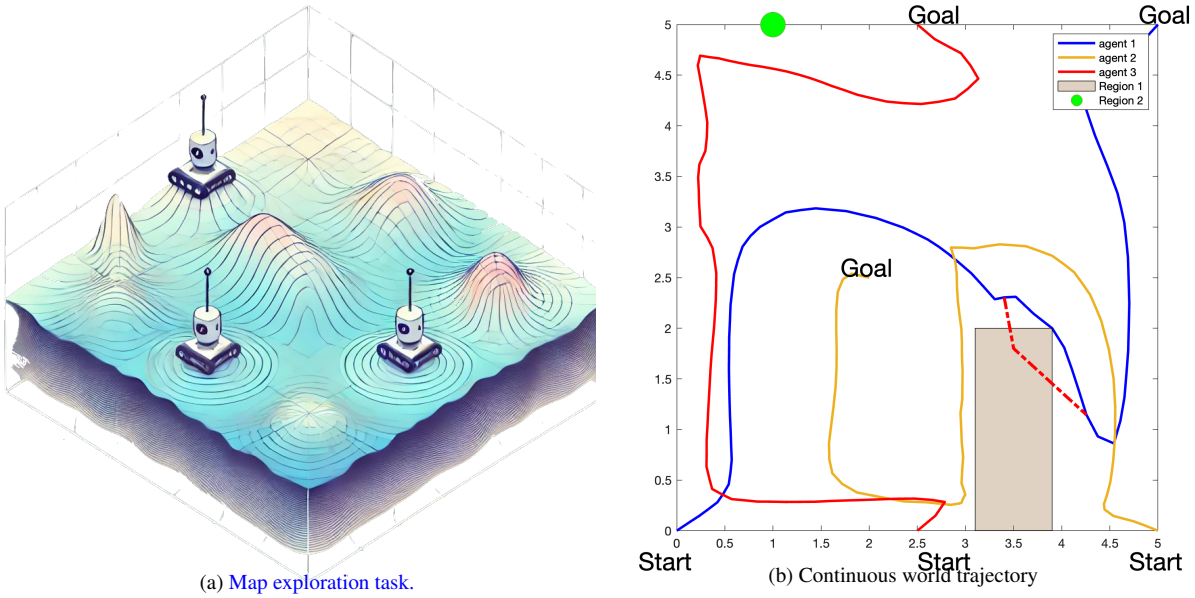
**Comment 1.1**

It seems like Example 1 is introduced as an example to help readers understand the formulation (1) in details. However, it is still vague, and it doesn't easily connect to Figure 3b.

**Response:** Example 1 has been revised to improve clarity, with additional illustrative content and an updated figure added to better showcase the example application and its relevance to the proposed approach.

## Revision:

**Example 1.** *Robots are tasked with navigating an unknown task space to collect sensory data and reconstruct a field (as illustrated in Fig. 2a). The task space is modeled as a grid, where each grid point has an associated value tracked by a Kalman Filter (KF) [28]. The KF estimates uncertainty at each point through its covariance $P_j$ updated based on the measurements taken by robots along the trajectory $\mathbf{q}$. Measurement quality, modeled by a Gaussian radial basis function, decreases with distance from the robot. The optimization objective $\Phi(\mathbf{q}) = \max_j P_j(\mathbf{q})$ is to minimize the maximum uncertainty in reconstructing the field (detailed in [6]).*




(a) Map exploration task.

(b) Continuous world trajectory

Figure 2: (2a) Illustration of a 3 robot map exploration task case. (2b) The unsecured trajectory design optimized for a map exploration task. Potential security breaches, indicated by red dashed lines, highlight paths that could allow unauthorized access to forbidden regions.

Ideally, robots are expected to spread out in a Boustrophedon pattern to efficiently explore the task space, optimize coverage, and minimize reconstruction uncertainty (Fig. 2b). However, we consider the possibility of threats following the *physical masquerade attack* model [2], where a compromised insider (robot) masquerading as a properly functioning agent, attempts to gain access to unauthorized locations without detection. The attacker leverages full knowledge of the motion plan and exploits the compromised robot to provide false self-reports to the central entity (CE). These malicious deviations remain undetected as long as observations from uncompromised robots align with the motion plan. The corresponding security requirement can then be formally defined as:

**Definition 1.** *A multi-robot trajectory plan is secured against plan-deviation attacks if it ensures that any potential deviations to these forbidden regions will cause the corresponding robot to miss their next co-observation with other robots.*

**Comment 1.2**

Personally, I think to much space is dedicated to the details of how reachability constraints are implemented in the ADMM formulation (Sec. D, E, F, G, H). Although they are important details, I think this part can be condensed more, and some of the details can be deferred to the Appendix, so that some space can be dedicated more to improve the clarity of the other sections.

**Response:** As suggested, we have condensed this section and moved part of the "Transformation to Canonical Frame" section into the appendix to streamline the main text and improve readability. We believe that the remaining details on the implementation of the reachability constraints are essential and provide critical context for understanding the proposed method. We appreciate your suggestion, which has helped us enhance the balance and clarity of the manuscript.

**Revision:**

## D Transformation to canonical coordinates

To use the definition of reachability ellipsoid from the section above as computational constraints for ADMM, we first apply a differentiable rigid body transformation to reposition the ellipse $\mathcal{E}$ from the global frame $\mathcal{F}$ to a canonical frame $\mathcal{F}_\mathcal{E}$, where the origin of $\mathcal{F}_\mathcal{E}$ is at the ellipsoid's center $o = \frac{1}{2}(q_1 + q_2)$, and the first axis of $\mathcal{F}_\mathcal{E}$ is aligned with the foci (see Fig.3a for an illustration). From this definition, a unit vector along the first axis of the ellipsoid in the frames $\mathcal{F}$ and $\mathcal{F}_\mathcal{E}$ is given by $\nu_\mathcal{F} = \frac{q_2 - q_1}{\|q_2 - q_1\|}$ and $\nu_\mathcal{E} = [1, 0, 0]^T$, respectively. The full transformation from coordinates $q^\mathcal{E} \in \mathcal{F}$ to $q \in \mathcal{F}$, and its inverse, are then parametrized by a rotation $R^\mathcal{F}_\mathcal{E}$ and a translation $o^\mathcal{F}_\mathcal{E}$ as (we drop the subscript and superscript from $R^\mathcal{F}_\mathcal{E}$ and $o^\mathcal{F}_\mathcal{E}$ to simplify the notation):

$$q = Rq^\mathcal{E} + o, \quad q^\mathcal{E} = R^\mathrm{T}(q - o). \tag{10}$$

where the rotation matrix $R = H(\nu_\mathcal{F}(q_1, q_2), \nu_\mathcal{E})$ is a *Householder rotation* $H$ (a differentiable linear transformation describing the minimal rotation between the two vectors, see Appendix B for details). To simplify the notation, in the following, we will consider $H$ to be a function of $q_1, q_2$ directly, i.e. $H(q_1, q_2)$. More details on (10) are given in Appendix A. Reachability constraints are formulated with respect to different types of forbidden regions a point, a plane, a segment, and a convex polygon.

## F Plane-ellipsoid reachability constraint

For a hyperplane shaped forbidden region $\mathcal{L}(q) = \{q \in \mathbb{R}^m : \mathbf{n}^\mathrm{T} q = d\}$ (Fig.3c), the reachability constraint is $\mathcal{L} \cap \mathcal{E}(q_1, q_2, a) = \emptyset$. When transformed into the canonical frame, the hyperplane can be written as $\mathcal{L}^\mathcal{E}(q^\mathcal{E}) = \{q^\mathcal{E} \in \mathbb{R}^m : \mathbf{n}^\mathrm{T}_\mathcal{E} q^\mathcal{E} = d_\mathcal{E}\}$, with $\mathbf{n}_\mathcal{E} = H(q_1, q_2)\mathbf{n}, \quad d_\mathcal{E} = -\mathbf{n}^\mathrm{T} o + d$.

## G Convex-polygon-ellipse reachability constraint

The reachability constraint for a convex polygon is treated as a union of reachability constraints for each individual segments that define the hyperplane. We define

**ADMM constraint 4** (Convex-polygon-ellipsoid reachability constraint)**.**

$$D(\mathbf{q}) = \begin{bmatrix} D_{seg1}(\mathbf{q}) \\ D_{seg2}(\mathbf{q}) \\ \vdots \end{bmatrix} \tag{30}$$

$$\mathcal{Z} = \{\mathbf{q} \in \mathbb{R}^{nm} : \|D(\mathbf{q})\| = 0\}, \tag{31}$$

$$\Pi_\mathcal{Z}(\mathbf{z}) = 0, \tag{32}$$

where $D_{seg}$ are the constraint functions for all line segments used to define the convex polygon region. For hyperplane $\mathcal{L}^{\mathcal{E}}(q^{\mathcal{E}}) = \{q^{\mathcal{E}} \in \mathbb{R}^m : \mathbf{n}_{\mathcal{E}}^{\mathrm{T}} q^{\mathcal{E}} = d_{\mathcal{E}}\}$ with endpoints $p_1^{\mathcal{E}}$ and $p_2^{\mathcal{E}}$; this segment is defined as:

$$\begin{bmatrix}(p_1^{\mathcal{E}} - p_2^{\mathcal{E}})^{\mathrm{T}} \\ (p_2^{\mathcal{E}} - p_1^{\mathcal{E}})^{\mathrm{T}}\end{bmatrix} p^{\mathcal{E}} \leq \begin{bmatrix} p_2^{\mathcal{E}\mathrm{T}} \\ -p_1^{\mathcal{E}\mathrm{T}}\end{bmatrix} (p_1^{\mathcal{E}} - p_2^{\mathcal{E}}), \quad \mathbf{n}_{\mathcal{E}}^{\mathrm{T}} q^{\mathcal{E}} = d_{\mathcal{E}}. \tag{33}$$

When the *tangent interpolation point* $p_{\mathcal{L}}^{\mathcal{E}}$ stays within the segment (i.e. red region in Fig. 3d), the constraint is a plane-ellipse constraint in Section II-F. Otherwise (i.e. brown region in Fig. 3d), the constraint is a point-ellipse constraint in Section II-F.

**ADMM constraint 5** (Line-segment-ellipsoid reachability constraint)**.**

$$D_{seg}(\mathbf{q}) = \begin{cases} D_{p_1}(\mathbf{q}) & (p_1^{\mathcal{E}} - p_2^{\mathcal{E}})^{\mathrm{T}}(p_{\mathcal{L}}^{\mathcal{E}} - p_2^{\mathcal{E}}) < 0 \\ D_{p_2}(\mathbf{q}) & (p_2^{\mathcal{E}} - p_1^{\mathcal{E}})^{\mathrm{T}}(p_{\mathcal{L}}^{\mathcal{E}} - p_1^{\mathcal{E}}) < 0 \\ D_{p_{\mathcal{L}}^{\mathcal{E}}}(\mathbf{q}) & \textit{otherwise} \end{cases} \tag{34}$$

$$\mathcal{Z} = \{\mathbf{q} \in \mathbb{R}^{nm} : \|D_{seg}(\mathbf{q})\| = 0\}, \tag{35}$$

$$\Pi_{\mathcal{Z}}(\mathbf{z}) = 0, \tag{36}$$

where $D_{p_1}$ and $D_{p_2}$ are the point-ellipsoid constraint projection function (11) respect to $p_1^{\mathcal{E}}$ and $p_2^{\mathcal{E}}$, and $D_{p_{\mathcal{L}}^{\mathcal{E}}}$ is the plane-ellipsoid constraint (22). with respect to frame $\mathcal{L}^{\mathcal{E}}$. This constraint needs to be supplemented with a convex obstacle constraint for the polygon (i.e. keep foci waypoints outside the region, introduced in [6]) to prevent cases where the ellipse is a subset of the region.

**Comment 1.3**

The word "checkpoint" appears abruptly in page 6 without concrete definition, although it seems like it is an important component of the formulation, and an important design choice for the results. For example, in the way Figure 3 is presented, I don't get a good idea why the checkpoint location is there in (c).

**Response:** For the feedback regarding the term "checkpoint", the original intent of "checkpoint" in Section 2 was to illustrate the limitations of the method proposed, specifically to show that even if APMAPF yields a valid solution, transforming this solution to a continuous domain may still need additional security measure to guarantee security. To address the concern, we have replaced the term "checkpoints" with "key locations" in Section 2 to avoid confusion. Additionally, we have added a formal definition of "checkpoints" and the "checkpoint graph" later in Section 3 to provide a clear and precise explanation of these concepts. These updates ensure that the term is used consistently and that its role in the methodology is explicitly defined.

**Revision:**

## H Secured planning results and limitation

We employ the Attack-Proof MAPF(APMAPF) solver [2] on an 8 by 8 grid world with a similar setup to generate a MAPF plan with a co-observation schedule in a grid-world application (Fig.4a). This result is transformed to a continuous configuration space and serves as the initial trajectory input for the ADMM solver with an additional task function targeting the map exploration task. Co-observation schedules are set up using the APMAPF algorithm for two forbidden regions. Reachability constraints are added to ensure an empty intersection between all robots' reachability regions during co-observations and the forbidden regions. It is important to emphasize that an APMAPF solution does not guarantee existence, nor does it ensure a successful transition to the continuous configuration space. In this case, while an APMAPF solution may be found, the secured, attack-proof solution becomes infeasible in the continuous setting because the robots' mobility is no longer restricted to adjacent grids. Additional security measures, such as stationary security cameras or surveillance robots, need to be incorporated. For instance, in this scenario, we deploy a security camera as additional security measure to observe agent 3 at time 8 to ensure security.

The simulation result, shown in Fig.4b, displays reachability regions as black ellipsoids, demonstrating empty intersections with Zones 2 and 3. Explicit constraints between reachability regions and obstacles are not activated, assuming basic obstacle avoidance capabilities in robots. The intersections between obstacles and ellipsoids, as observed between agent 3 and Zone 1, are deemed tolerable. All constraints are satisfied, and agents have effectively spread across the map for optimal exploration tasks.

1) Limitations: Our solution demonstrates the potential of planning with reachability and co-observation to enhance the security of multi-agent systems. However, two primary challenges need to be addressed. Firstly, achieving a co-observation and reachability-secured plan is not always feasible, particularly when obstacles or restricted regions separate the robots, preventing

them from establishing co-observation schedules or finding reachability-secured paths. For example, agent 3 in Fig.4b requires additional security measures to create secured reachability areas. Secondly, security requirements can impact overall system performance, as illustrated by the comparison between Fig.2b and Fig.4b. The introduction of security constraints resulted in the top left corner remaining unexplored by any robots. This trade-off between security and system performance is particularly significant as system performance is a key factor in the decision of multi-agent system deployments. These challenges are further addressed in Section III, ensuring the effective integration of reachability and co-observation in securing multi-agent systems.

**Comment 1.4**

Similarly, how the trajectories are broken down to multiple ellipsoid components are not clear, is it a design choice to make the constraints less conservative than setting a single ellipsoid with the start and goal point?

**Response:** The reachability ellipsoids are designed to work in conjunction with the co-observation technique. Instead of using a single ellipsoid defined by the start and goal points, the ellipsoid are defined by smaller ellipsoids between two consecutive co-observation times. This approach ensures that the constraints are aligned with the co-observation schedule, reducing conservativeness while maintaining security guarantees. We also make modification to Definition 4 (Remark 1 in the original manuscript) to make this more clear.

## Revision:

**Definition 4.** *A multi-robot trajectory is secured against plan-deviation attacks if there exists a co-observation plan such that the reachability region between each consecutive co-observation does not intersect with any forbidden regions.*

**Comment 1.5**

Similarly, in page 7, "key locations" and "checkpoint graph" are mentioned without clear explanations what they are.

**Response:** In Section III, we add additional definition to 'checkpoint' and 'checkpoint graph'

## Revision:

Our strategy is based on the following concept:

**Definition 6.** *The $i$-th* checkpoint $v_{pi} = (q_p, t_i)$ *for sub-team $p$ is defined as a location $q_{pi} = q_p(t_i)$ and time $t_i$. It represents either the first or last instance where a robot is co-observed by a teammate, and serves as a point where the robot can leave or join a new team.*

For simplicity, let $\mathcal{I}_{v_i}$ denote the sub-team to which $v_i$ belongs. To ensure the security of the reference trajectory, the reachability region between consecutive checkpoints must avoid intersections with forbidden regions. This requirement can be formally stated as:

**Remark 2.** *A set of checkpoints $V_p = \{v_{p0}, \ldots, v_{pT}\}$ (arranged in ascending order of $t_{v_{pi}}$) can secure the reference trajectory for sub-team $p$, if $\mathcal{E}(q_{v_{pi}}, q_{v_{p(i+1)}}, t_{v_{pi}}, t_{v_{p(i+1)}}) \cap F = \emptyset$ for every $i$, where $F$ is the union of all forbidden regions.*

**Definition 7.** *The planned trajectory $\{q_p\}_{p=1}^{N_p}$ is represented as a directed* checkpoint graph *$G = (V, E)$, where $V = (\cup_p V_p) \cup V_c$ is the set of verticies representing waypoints on the planned trajectory and $E = E_t \cup E_c$ is the set of edges representing feasible paths connecting $V$. The components of the checkpoint graph are defined as follows:*

**Checkpoints** $V_p$ *where additional robots are required for security through co-observation.*

**Trajectory edges** $E_t$ *represent the original planned path of the primary robot.*

**Connection vertices** $V_c$ *represent non-checkpoint locations linked via $E_c$.*

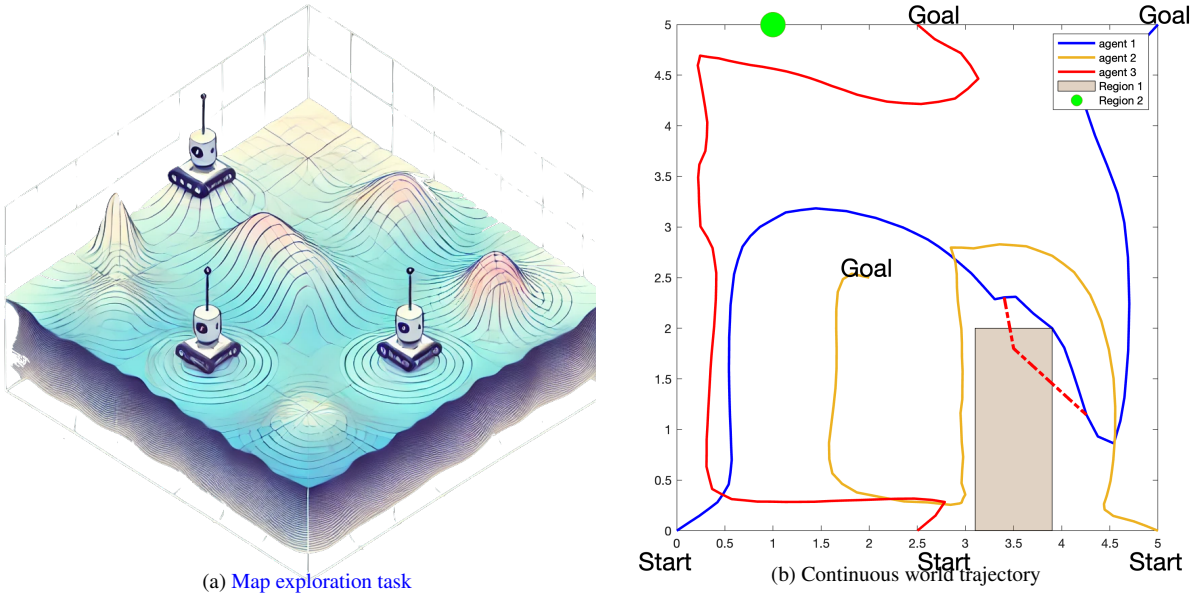
**Cross-trajectory edges** $E_c$ *represent feasible paths where robots deviate from the original trajectory to join other teams or provide co-observation support, at least one endpoint be a checkpoint.*

**Comment 1.6**

The plots are not reader-friendly at all. In all figures, trajectory lines are too thin. The legend text is too small. In figure 2, there is no number scale for (b) and I am even not sure if this corresponds to the same scenario as (a) and (c). In Figure 5, the color of the sub-team robots seem like not matching with the main robots.

**Response:** We have re-plotted the figures to make them more reader-friendly by thickening trajectory lines, enlarging legend text, and ensuring consistent coloring between sub-team robots and main robots in Figure 7 and Figure 8 (Figure 5 and 6 in original manuscript).
For Figure 2, we appreciate the comment regarding the lack of a number scale especially on (b,c,d). The intention of these figures are to illustrate the geometric relationships between ellipsoids and different shapes, as well as the transformation of coordinates. To maintain focus on these aspects, we chose to omit the number scale and believe that it is not essential for conveying the intended concept.

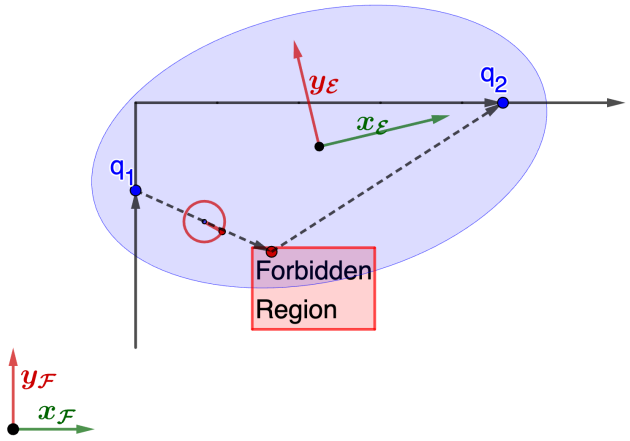
## Revision:



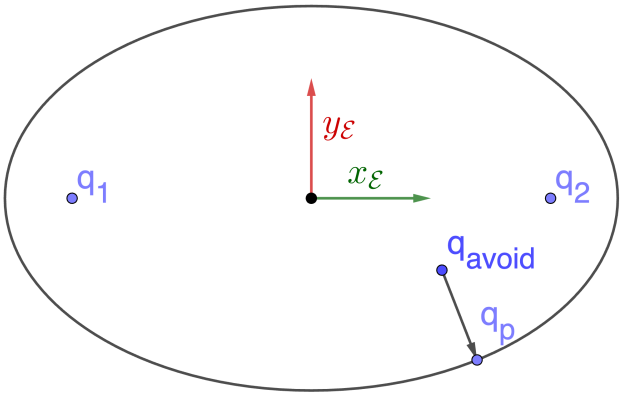

(a) Map exploration task

(b) Continuous world trajectory

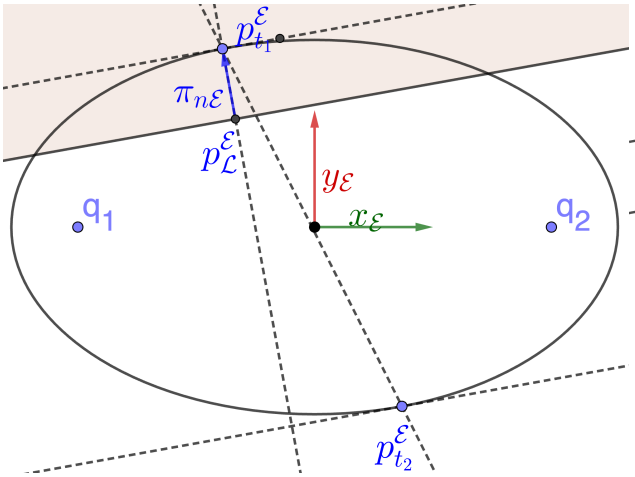
Figure 2: (2a) Illustration of a 3 robot map exploration task case. (2b) The unsecured trajectory design optimized for a map exploration task. Potential security breaches, indicated by red dashed lines, highlight paths that could allow unauthorized access to forbidden regions.



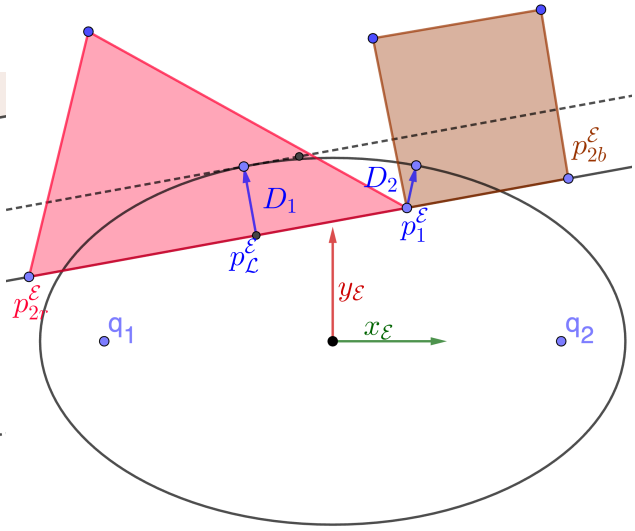

(a) Showcase of a reachability region




(b) Point-ellipsoid constraint

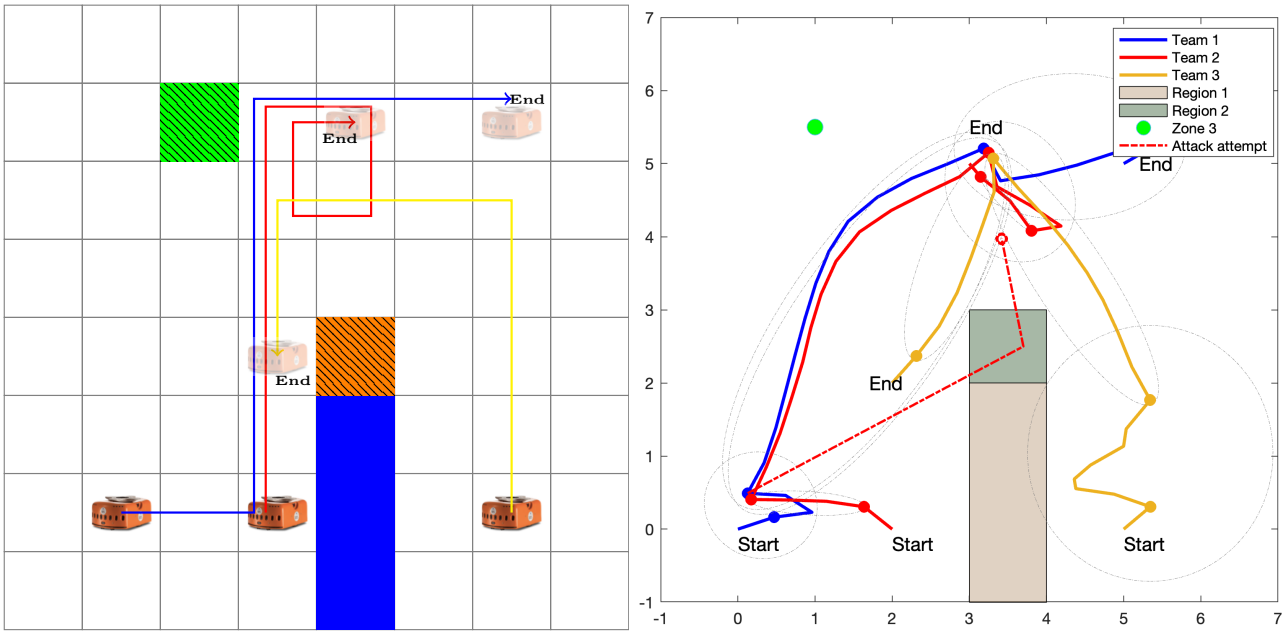



(c) Plane-ellipse constraint



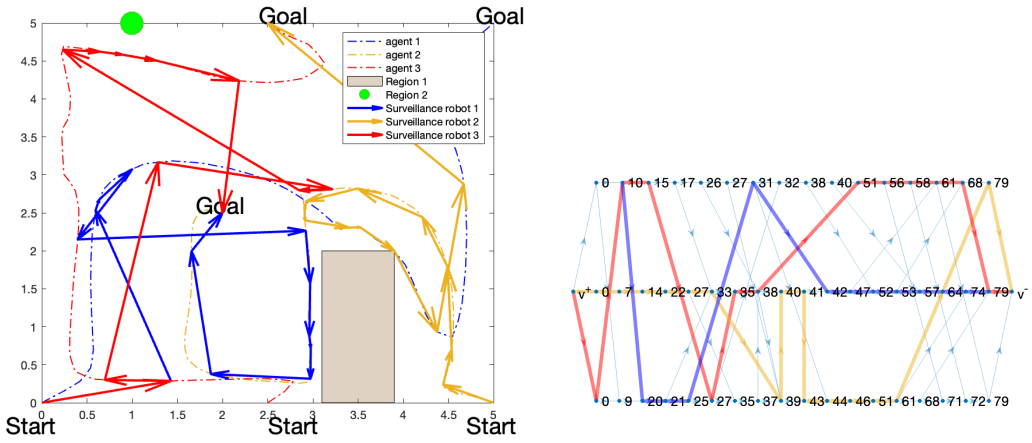

(d) Polygon-ellipse constraint

Figure 3: (3a) The black line is the planned trajectory, $q_1$ and $q_2$ are two co-observed locations where the robots are expected at given times $t_1$ and $t_2$, dashed lines show a possible trajectory of a compromised robot during the unobserved period. The axis of global frame $\mathcal{F}$ and canonical frame $\mathcal{F}_\mathcal{E}$ are shown as $x_\mathcal{F}, y_\mathcal{F}$ and $x_\mathcal{E}, y_\mathcal{E}$ respectively. (3b) For point-ellipsoid constraint, $q_{avoid}$ is projected to the areas outside the ellipsoid to $q_p$. (3c) For plane-ellipsoid constraint, the projection is simplified to the point-ellipse constraint that projects point $p_L$ outside the ellipse to $p_t$. (3d) For convex-polygon-ellipsoid constraint, the projection is either a plane-ellipse constraint $D_1$ (for the red region) or a point-ellipse constraint $D_2$ (for the brown region).



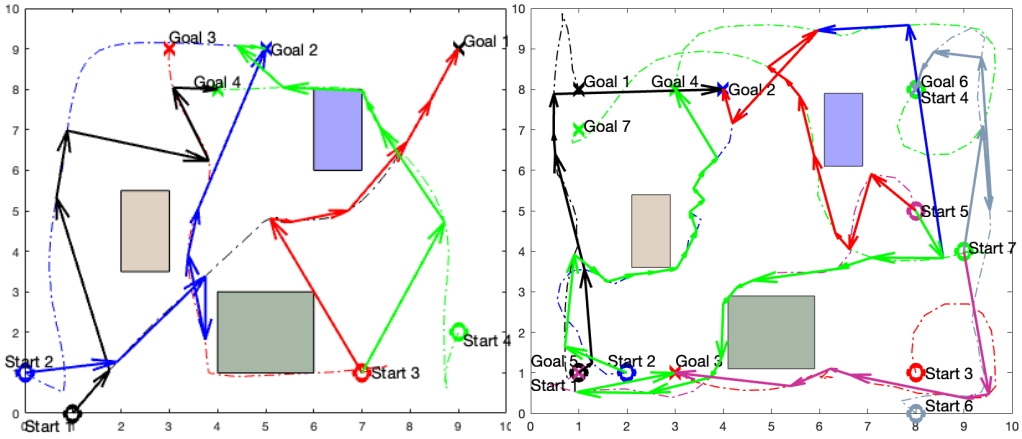

(a) Grid world result (b) Secured result.

Figure 4: Trajectory design of a three-robot system based on the grid world result. Zone 1 is an obstacle, Zone 2 and Zone 3 are forbidden regions. (4a) Result of the APMAPF algorithm in $8 \times 8$ grid-world without the map exploring objective. (4b) The secured trajectory with the incorporation of co-observation generated by APMAPF and additional reachability constraints. Attack attempts into the forbidden region (red dashed line) will cause team 1 miss the next scheduled co-observation.




(a) Cross-trajectory co-observation plan (b) Checkpoint graph and result flows

Figure 7: (7a) The cross-trajectory co-observation plan is shown as arrows on top of the original plan in Fig. 7a. (7b) The checkpoint graph and resulting flows highlighted.




(a) 4-agent case (b) 7-agent case

Figure 8: (8a) Cross-trajectory co-observation result of a 4 sub-teams task. (8b) Cross-trajectory co-observation result of a 7 sub-teams task.

**Comment 1.7**

The author assumes "Zone 1" is tolerable and Zone 2 and 3 are hard constraints, but what if the environment is the opposite? Do you get infeasible solution? It is very skeptical that the author is making deliberate assumptions to make the example work in a special case.

**Response:** We acknowledge the limitation of the method proposed in Section II, as it relies on the specific setup, and a safe solution is not always guaranteed to exist. This limitation is explicitly discussed in the manuscript. The chosen result is deliberate, not to make the example work in a special case, but to illustrate that even when APMAPF finds a solution in the grid world, such a plan may fail to guarantee security when transformed into the continuous domain, requiring further manual adjustments. In the revised version, we have further emphasized these limitations and the challenges they aim to address in the simulation results.

**Revision:**

## H Secured planning results and limitation

We employ the Attack-Proof MAPF(APMAPF) solver [2] on an 8 by 8 grid world with a similar setup to generate a MAPF plan with a co-observation schedule in a grid-world application (Fig.4a). This result is transformed to a continuous configuration space and serves as the initial trajectory input for the ADMM solver with an additional task function targeting the map exploration task. Co-observation schedules are set up using the APMAPF algorithm for two forbidden regions. Reachability constraints are added to ensure an empty intersection between all robots' reachability regions during co-observations and the forbidden regions. It is important to emphasize that an APMAPF solution does not guarantee existence, nor does it ensure a successful transition to the continuous configuration space. In this case, while an APMAPF solution may be found, the secured, attack-proof solution becomes infeasible in the continuous setting because the robots' mobility is no longer restricted to adjacent grids. Additional security measures, such as stationary security cameras or surveillance robots, need to be incorporated. For instance, in this scenario, we deploy a security camera as additional security measure to observe agent 3 at time 8 to ensure security.

The simulation result, shown in Fig.4b, displays reachability regions as black ellipsoids, demonstrating empty intersections with Zones 2 and 3. Explicit constraints between reachability regions and obstacles are not activated, assuming basic obstacle avoidance capabilities in robots. The intersections between obstacles and ellipsoids, as observed between agent 3 and Zone 1, are deemed tolerable. All constraints are satisfied, and agents have effectively spread across the map for optimal exploration tasks.

1) Limitations: Our solution demonstrates the potential of planning with reachability and co-observation to enhance the security of multi-agent systems. However, two primary challenges need to be addressed. Firstly, achieving a co-observation and reachability-secured plan is not always feasible, particularly when obstacles or restricted regions separate the robots, preventing them from establishing co-observation schedules or finding reachability-secured paths. For example, agent 3 in Fig.4b requires additional security measures to create secured reachability areas. Secondly, security requirements can impact overall system

performance, as illustrated by the comparison between Fig.2b and Fig.4b. The introduction of security constraints resulted in the top left corner remaining unexplored by any robots. This trade-off between security and system performance is particularly significant as system performance is a key factor in the decision of multi-agent system deployments. These challenges are further addressed in Section III, ensuring the effective integration of reachability and co-observation in securing multi-agent systems.

**Comment 1.8**

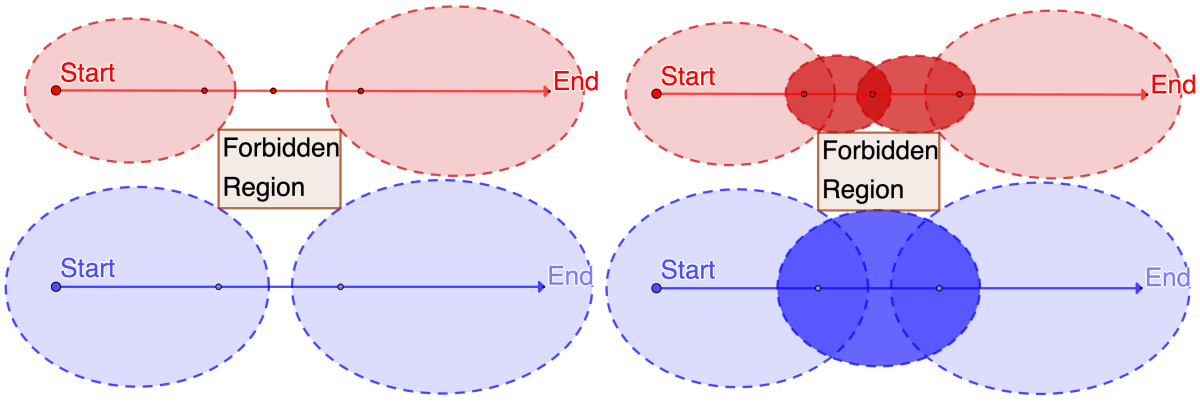
p 8 - A heuristic approach is provided (Algorithm 1); Is it almost impossible for the reader to decipher this algorithm without any explanations in words.

**Response:** We have re-written the algorithm with more detailed and add illustration figure to improve readability and make the approach easier to follow.

**Revision:**

**Algorithm 1** Checkpoint Graph Initialization for Team $\mathcal{I}_p$

1: **Initialization:** Set $t_0 \leftarrow 0$ and $t_2 \leftarrow T$, and add $(q_p(t_0), t_0)$ and $(q_p(t_2), t_2)$ to $V_p$ as *start* and *end vertices*.
2: **while** $\mathcal{E}(q_p(t_0), q_p(t_2), t_0, t_2) \cap F \neq \emptyset$ **and** $t_0 \leq t_2$ **do**
3: **Forward Search:** Find the largest $t_1 \in \{t_0 + 1, \ldots, t_2\}$ such that $\mathcal{E}(q_p(t_0), q_p(t_1), t_0, t_1) \cap F = \emptyset$. Once found, add $(q_p(t_1), t_1)$ to $V_p$.
4: **Backward Search:** Find the smallest $t_3 \in \{t_1, \ldots, t_2 - 1\}$ such that $\mathcal{E}(q_p(t_2), q_p(t_3), t_2, t_3) \cap F = \emptyset$. Once found, add $(q_p(t_3), t_3)$ to $V_p$.
5: **Update Indices:** Set $t_0 \leftarrow t_1$ and $t_2 \leftarrow t_3$.
6: **end while**




(a) (b)

Figure 5: Checkpoint generation example

**Comment 1.9**

Similar to the comment above, it will be good to introduce what the “network multi-flow problem” is.

**Response:** We have added additional content to introduce what the "network multi-flow problem" in our setup.

**Revision:** **In Section III.A, we make the following modification**

We assume that the planned trajectory for each sub-team has a robot operating on it. The co-observation planning problem is then transformed into a network multi-flow problem by modeling the additional robots as flows traveling through $G$. These flows either follow the planned trajectory via trajectory edges $E_t$ or switch teams via cross-trajectory edges $E_c$ to ensure valid and secured paths. Additionally, they are required to visit every checkpoint $V_p$ to meet the security requirements introduced in 2. This formulation reduces the co-observation problem to finding valid flow patterns in $G$. The problem is then formulated into linear objectives and constraints, enabling the use of Mixed-Integer Linear Programming (MILP) techniques to efficiently compute feasible solutions.

**Comment 1.10**

p3 - “rotation derived from a modifie version of Householder transformations [25].” ; At least a brief introduction of Householder transformations (in words) must be included for readers who are not familiar with it.

**Response:** A brief introduction of Householder transformation is added.

**Revision:** **In Section II.D we modified:**

the rotation matrix $R = H(\nu_{\mathcal{F}}(q_1, q_2), \nu_{\mathcal{E}})$ is a *Householder rotation* $H$ (a differentiable linear transformation describing the minimal rotation between the two vectors, see Appendix B for details).

**Comment 1.11**

This reachability ellipsoid is an over approximation of the exact reachability region in Definition 3.”; A proof or at least a reference to the proof should be provided.

**Response:** While we believe that a formal proof is not necessary for this context, we have added additional content to make the approximation more intuitive. This includes a clearer explanation of how the reachability ellipsoid relates to the exact reachability region. These additions aim to enhance the reader’s understanding without requiring formal proof.

**Revision:**

**Definition 2.** *Consider a robot $i$ starting from $q_1$ at time $t_1$ and reaching $q_2$ at time $t_2$. The* reachability region *between $t_1$ and $t_2$ is defined as the set of points $q'$ in the free configuration space for which there exists a kinematically feasible trajectory containing $q'$.*

For simplicity, in this paper we consider only a maximum velocity constraint $v_{max}$; the reachability region for $q'$ can then be over-approximated by the following (the over-approximation is obtained by neglecting physical obstacles):

**Definition 3.** *The* reachability ellipsoid $\mathcal{E}$ *is defined as the region* $\mathcal{E}(q_1, q_2, t_1, t_2) = \{\tilde{q} \in \mathbb{R}^n : d(q_1, \tilde{q}) + d(\tilde{q}, q_2) < 2a\}$, *where* $a = \frac{v_{max}}{2}(t_2 - t_1)$.

This region is an ellipsoid because it represents the set of points $q'$ whose sum of distances two *foci* $q_1$ and $q_2$ is less than $2a$, where $a$ is the major radius of the ellipsoid.

**Comment 1.12**

Other grammar related comments.

**Response:** We have carefully reviewed and corrected all listed grammar mistakes, as well as others that were not explicitly mentioned.

## 2 Response to reviewer 2

**Comment**

This paper proposed a novel multi-robot optimal planning algorithm that integrated mutual observations and introduced reachability constraints for enhanced security, which ensures that, even with adversarial movements, compromised robots cannot breach forbidden regions without missing scheduled co-observations. The contribution of this paper is hard evaluate due to the organization.

**Response:** We sincerely thank the reviewer for their valuable comments and feedback. We acknowledge the concern regarding the organization of the manuscript, which may have made it challenging to fully evaluate the contributions. In the revised version, we have carefully restructured the manuscript to improve its organization and readability, ensuring that the key contributions are presented more clearly and effectively. We greatly appreciate your valuable feedback, which has been instrumental in enhancing the quality of our work.

**Comment 2.1**

The dimension of p is erroneously defined;

**Response:** After carefully reviewing the manuscript, we believe the reviewer might have meant the dimension of $q$ rather than $p$. In the revised version, we have corrected the definition of $q$ to ensure clarity and accuracy. Thank you for bringing this to our attention.

**Revision:** **In Section II, first paragraph, we modified**

We denote the trajectory as $\mathbf{q}_i = [q_{i0} \dots q_{iT}] \in \mathbb{R}^{m\times T}$, where $q_{ij} \in \mathbb{R}^m$ is the waypoint of agent $i$ in a $m$ dimensional state space, $T$ is the time horizon. For a total of $n_p$ robots, trajectories can be represented as an aggregated vector $\mathbf{q} = \text{stack}(\mathbf{q}_1, \dots, \mathbf{q}_{n_p}) \in \mathbb{R}^{mn_p\times T}$, where $\text{stack}(\cdot)$ denotes the vertical stacking operation.

**Comment 2.2**

There are many grammar errors, please check it carefully

**Response:** We have carefully reviewed and corrected all identified grammar issues, as well as others not explicitly mentioned. The revised manuscript has undergone thorough proofreading to ensure clarity and grammatical accuracy.

**Comment 2.3**

In Definition 1, is reasonable to say “any potential deviations”?

**Response:** We believe that this phrase is reasonable and essential to the definition and intention of security against plan-deviation attacks. The definition is designed to ensure that the trajectory plan accounts for all possible paths a compromised robot might take toward forbidden regions. By explicitly requiring that any such deviation results in the robot missing its next co-observation, the definition provides a rigorous and comprehensive security guarantee. As such, the phrase "any potential deviations" accurately reflects the intended scope of the definition.

**Comment 2.4**

In the definition of $U$, what does “,” mean?

**Response:** We are not entirely sure what the specific concern or issue is in this context. However, if the reviewer is referring to $u'$ , it was intended to denote the vector $u$ before normalization. To avoid any potential confusion, we have revised the notation in the updated manuscript. We appreciate your feedback and have taken steps to ensure clarity in this definition.

**Revision:**

**Definition 9.** *Let $\nu_1$ and $\nu_2$ be two unitary vectors ($\|\nu_1\| = \|\nu_2\| = 1$). Define the normalized vector $u = \frac{\nu_1+\nu_2}{\|\nu_1+\nu_2\|}$, the* Householder rotation *$H(\nu_1, \nu_2)$ is defined as*

$$H(\nu_1, \nu_2) = 2uu^{\mathrm{T}} - I. \tag{42}$$

**Comment 2.5**

How to solve (9a)?

**Response:** In practical applications, Eq.4a (originally 9a) is solved iteratively using a nonlinear optimization solver, such as *fmincon*. Additionally, the derivatives of the constraint projection functions, as derived later in the manuscript, are specifically utilized to solve the optimization problem more efficiently and quickly. These derivatives enhance the solver's performance by providing necessary gradient information, improving convergence speed. We have clarified these points in the revised manuscript to provide further insight into the solution process.

## Revision:

**Remark 1.** *In practical application, (4a) is solved iteratively via a nonlinear optimization solver such as* fmincon *[30]. The use of explicit gradients of the constraint functions (derived below) greatly enhances the solver's efficiency and accuracy (with respect to the use of numerical differentiation). At the same time, (4b) and (4c) have closed form solutions.*

[30] The MathWorks, Inc., MATLAB Optimization Toolbox, The Math- Works, Inc., Natick, Massachusetts, 2024, version R2024b, https://www. mathworks.com/products/optimization.html.

**Comment 2.6**

How to define $d_{\max}$ in (14)?

**Response:** $d_{max}$ is the maximum distance between a pair of robots that the secure can be ensured through co-observation.

## Revision:

**ADMM constraint 1** (Co-observation constraint)**.**

$$D(\mathbf{q}) = \overrightarrow{q_{aj}q_{bj}}, \tag{7}$$

$$\mathcal{Z} = \{\mathbf{z} \mid \|\mathbf{z}\| \leq d_{max}\}, \tag{8}$$

$$\Pi_{\mathcal{Z}}(z) = \begin{cases} d_{max}\frac{\mathbf{z}}{\|\mathbf{z}\|} & if\ \|\mathbf{z}\| > d_{max}, \\ \mathbf{z} & otherwise, \end{cases} \tag{9}$$

*where $a, b$ are the indices of the pair of agents required for a mutual inspection, $d_{max}$ is the maximum distance between a pair of robots that the secure can be ensured through co-observation.*

**Comment 2.7**

In Definition 4, it is unclear that $\mathcal{E}$ is an ellipsoid. There is no definition of the function $d(\cdot, \cdot)$;

**Response:** In the revised manuscript, we have added additional explanations to clarify that $\mathcal{E}$ is an ellipsoid and to explicitly define the function $d(\cdot, \cdot)$ to avoid confusion and ensure clarity.

## Revision:

**Definition 2.** *Consider a robot $i$ starting from $q_1$ at time $t_1$ and reaching $q_2$ at time $t_2$. The* reachability region *between $t_1$ and $t_2$ is defined as the set of points $q'$ in the free configuration space for which there exists a kinematically feasible trajectory containing $q'$.*

For simplicity, in this paper we consider only a maximum velocity constraint $v_{max}$; the reachability region for $q'$ can then be over-approximated by the following (the over-approximation is obtained by neglecting physical obstacles):

**Definition 3.** *The* reachability ellipsoid *$\mathcal{E}$ is defined as the region $\mathcal{E}(q_1, q_2, t_1, t_2) = \{\tilde{q} \in \mathbb{R}^n : d(q_1, \tilde{q}) + d(\tilde{q}, q_2) < 2a\}$, where $a = \frac{v_{max}}{2}(t_2 - t_1)$, and $d(\cdot, \cdot)$ denotes the Euclidean distance between two points.*

This region is an ellipsoid because it represents the set of points $q'$ whose sum of distances two *foci* $q_1$ and $q_2$ is less than $2a$, where $a$ is the major radius of the ellipsoid.

**Comment 2.8**

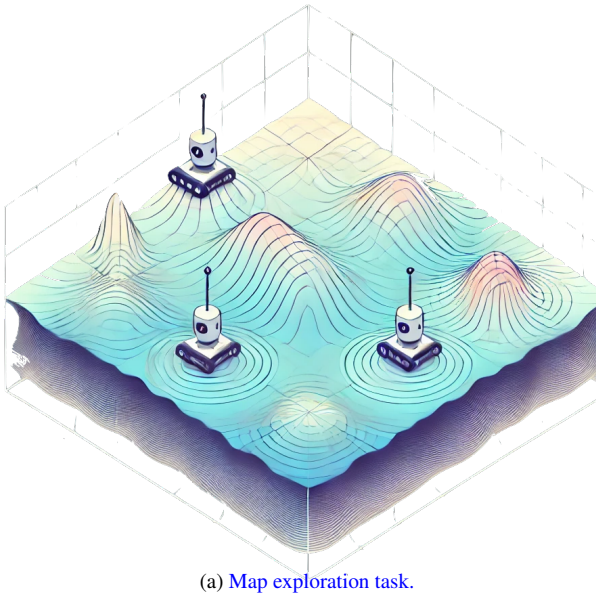
It is unclear about how to ensure the security. Where is the attack?

**Response:** We have added a detailed description of the attacker model in the revised manuscript, outlining how a compromised robot attempts to gain unauthorized access while masquerading as legitimate. The mechanisms for ensuring security are formally described in Definition 1 and further extended in Definition 4 and Remark 2, which outline the requirements and constraints necessary to detect and mitigate such attacks effectively. We also add potential attacks to the unsecured and secured result figures.

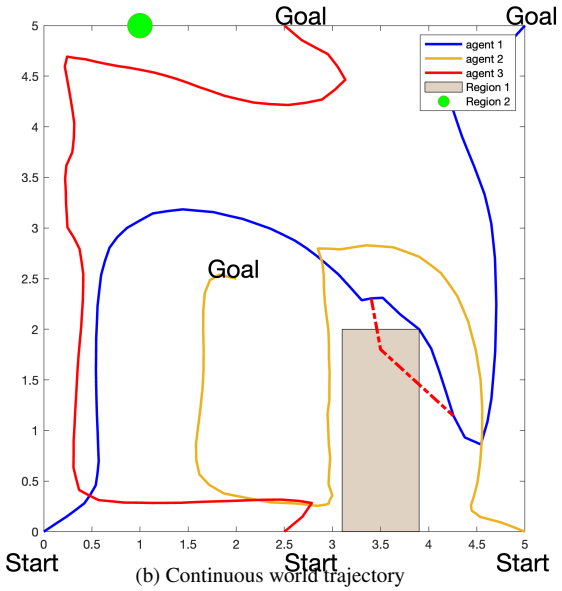
**Revision:** **In Section I, Introduction, Paragraph 1, we revised**

A countermeasure to these *plan-deviation attacks* [2]–[4] is to use the onboard sensing capabilities of the robots themselves to perform inter-robot *co-observations* and detect unusual behavior. These mutual observations establish a *co-observation schedule* alongside the planned path, ensuring that any attempts by a compromised robot to violate safety constraints (such as transgressing into forbidden regions) would break the observation plan and be promptly detected.

**In Section II, after Example 1 we revised:**



(a) Map exploration task.




(b) Continuous world trajectory

Figure 2: (2a) Illustration of a 3 robot map exploration task case. (2b) The unsecured trajectory design optimized for a map exploration task. Potential security breaches, indicated by red dashed lines, highlight paths that could allow unauthorized access to forbidden regions.

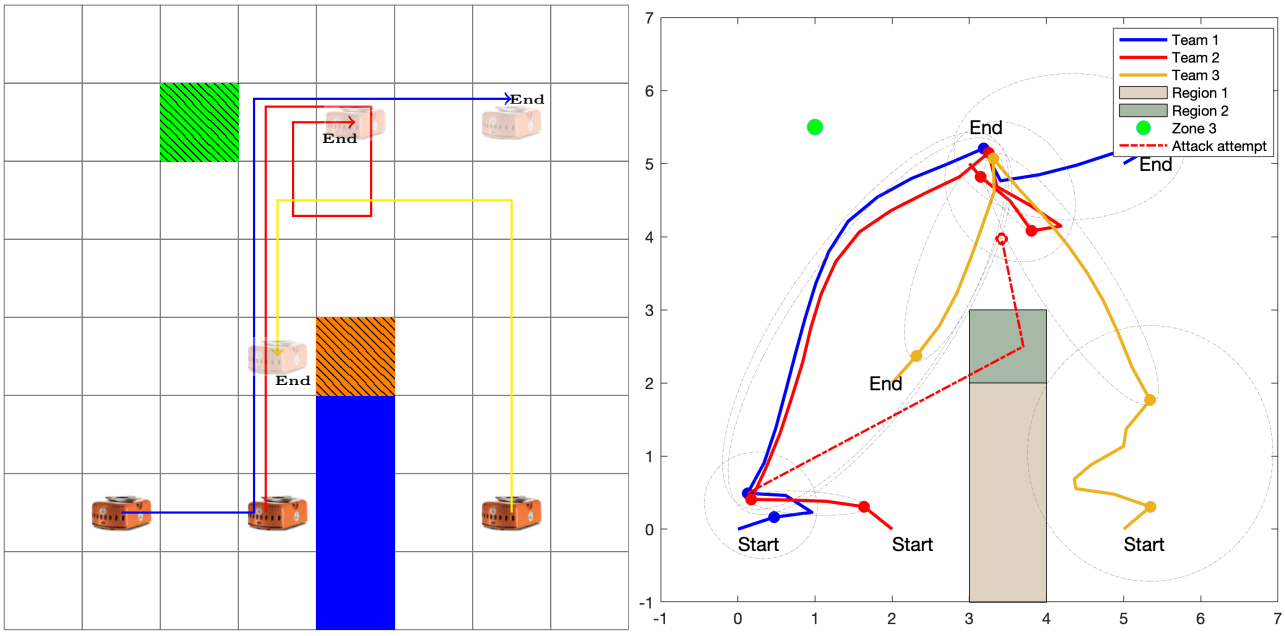
Ideally, robots are expected to spread out in a Boustrophedon pattern to efficiently explore the task space, optimize coverage, and minimize reconstruction uncertainty (Fig. 2b). However, we consider the possibility of threats following the *physical masquerade attack* model [2], where a compromised insider (robot) masquerading as a properly functioning agent, attempts to gain access to unauthorized locations without detection. The attacker leverages full knowledge of the motion plan and exploits the compromised robot to provide false self-reports to the central entity (CE). These malicious deviations remain undetected as long as observations from uncompromised robots align with the motion plan. The corresponding security requirement can then be formally defined as:

**Definition 1.** *A multi-robot trajectory plan is secured against plan-deviation attacks if it ensures that any potential deviations to these forbidden regions will cause the corresponding robot to miss their next co-observation with other robots.*

**In section II.C:**

The condition that a reachability ellipsoid $\mathcal{E}(q_1, q_2)$ does not intersect with any forbidden region is sufficient to secure the trajectory of the robot between $q_1$ and $q_2$ according to Definition 1: any deviation to a point $p_o$ outside $\mathcal{E}(q_1, q_2)$ will cause the robot to miss a potential observation at $p_2$, $t_2$. Definition 1 can then be restated as follows:

**Definition 4.** *A multi-robot trajectory is secured against plan-deviation attacks if there exists a co-observation plan such that the reachability region between each consecutive co-observation does not intersect with any forbidden regions.*




(a) Grid world result

(b) Secured result.

Figure 4: Trajectory design of a three-robot system based on the grid world result. Zone 1 is an obstacle, Zone 2 and Zone 3 are forbidden regions. (4a) Result of the APMAPF algorithm in $8 \times 8$ grid-world without the map exploring objective. (4b) The secured trajectory with the incorporation of co-observation generated by APMAPF and additional reachability constraints. Attack attempts into the forbidden region (red dashed line) will cause team 1 miss the next scheduled co-observation.

**In Section III.A:**

To ensure the security of the reference trajectory, the reachability region between consecutive checkpoints must avoid intersections with forbidden regions. This requirement can be formally stated as:

**Remark 2.** *A set of checkpoints $V_p = \{v_{p0}, \ldots, v_{pT}\}$ (arranged in ascending order of $t_{v_{pi}}$) can secure the reference trajectory for sub-team p, if $\mathcal{E}(q_{v_{pi}}, q_{v_{p(i+1)}}, t_{v_{pi}}, t_{v_{p(i+1)}}) \cap F = \emptyset$ for every $i$, where $F$ is the union of all forbidden regions.*

**Comment 2.9**

Please check the reference format carefully.

**Response:** Thank you for pointing this out. We have carefully reviewed and revised the reference format in the manuscript to ensure consistency and compliance with the journal's guidelines.

# 3 Response to reviewer 3

**Comment 3.1**

lease check the grammar and typos, here are some within the first two pages: . . . I stop marking these typos after page 2, but please do a careful proof-read of your own paper.

**Response:** We sincerely apologize for the grammatical mistakes in the initial submission and acknowledge the importance of presenting our work with clarity and professionalism. In the revised version, we have been extra careful to address all grammar issues and have implemented thorough proofreading to minimize similar mistakes in future submissions. We greatly appreciate your patience and constructive feedback, which have been invaluable in improving the quality of our manuscript.

**Comment 3.2**

Some definitions and symbols are not clearly explained: 1) First paragraph of Sec 2: the dimension of $q$ should be $n_p * m * T$ right? Further, how is it aggregated? In what order?

**Response:** In the revised manuscript, we have clarified that the dimension of $q$ have been clearified and have further explained the aggregation process.

**Revision:** **In Section II, we modified:**

We denote the trajectory as $\mathbf{q}_i = [q_{i0} \dots q_{iT}] \in \mathbb{R}^{m \times T}$, where $q_{ij} \in \mathbb{R}^m$ is the waypoint of agent $i$ in a $m$ dimensional state space, $T$ is the time horizon. For a total of $n_p$ robots, trajectories can be represented as an aggregated vector $\mathbf{q} = \text{stack}(\mathbf{q}_1, \dots, \mathbf{q}_{n_p}) \in \mathbb{R}^{mn_p \times T}$, where $\text{stack}(\cdot)$ denotes the vertical stacking operation.

**Comment 3.3**

Def 2: what is $\hat{v}_F$ and $\hat{v}_\epsilon$? Also $v_F$ and $v_e$ are defined later, so not proper to use here.

**Response:** We acknowledge that in Definition 2 (Definition 9 in the revised manuscript) $\hat{v}_F$ and $\hat{v}_\epsilon$ were typographical errors and have been corrected in the revised manuscript. Additionally, the terms $v_F$ and $v_e$ have been modified to ensure proper introduction and to avoid confusion.

**Revision:**

**Definition 9.** *Let $\nu_1$ and $\nu_2$ be two unitary vectors ($\|\nu_1\| = \|\nu_2\| = 1$). Define the normalized vector $u = \frac{\nu_1+\nu_2}{\|\nu_1+\nu_2\|}$, the* Householder rotation *$H(\nu_1, \nu_2)$ is defined as*

$$H(\nu_1, \nu_2) = 2uu^{\mathrm{T}} - I. \tag{42}$$

**Comment 3.4**

3) What is the difference between lowercase letters and bold-type lowercase letters? If bold-type denotes vectors, then why $x(t)$ in eq(2), u, v's in eq (3) not in bold-type?

**Response:** We apologize for the confusion regarding the notation. To address this, we have added additional clarification in the revised manuscript. Specifically, we use non-bold symbols to denote single-agent states (e.g. $q_{ij}$) and scalars, while bold symbols represent aggregated states of a single robot or multiple robots (e.g., $\mathbf{q}$). This distinction has been made explicit to avoid further ambiguity.

**Revision:** ***Notation***. In this paper, we use non-bold symbols to denote single-agent states (e.g. $q_{ij}$) and scalars, and bold symbols to represent aggregated states of single robot and multiple robots (e.g. $\mathbf{q}$).

**Comment 3.5**

3) Def 3: the 'workspace' is not specified before. The function d(q1,q2) is not defined. Further, this def is confusing, do you want to say that the reachability region is the union of all trajectory states, such that starting from q(t1) = q1, you are able to reach q(t2) = q2 while satisfying the speed constraints?
4) Def 4: what is $v_{maq}$? Why is the ellipsoid an over-approximation of reachability region?

**Response:** Thank you for pointing out the issues on Definition 3 (Definition 2 in the new manuscript). In the revised manuscript, we have modified the term 'workspace' for clarity, refined the definition to avoid confusion, and added a detailed description of the function $d(q_1, q_2)$, which represents the Euclidean distance between $q_1$ and $q_2$ . These changes aim to ensure the definition accurately conveys that the reachability region is the union of all trajectory states satisfying the specified conditions.
As for Defnintion 4 (Definition 3 in the new manuscript), we acknowledge that the term $v_{maq}$ as a typographical error, and it has been corrected to $v_{max}$ in the revised manuscript. Additionally, we have provided a more detailed explanation to clarify why the ellipsoid is an over-approximation of the reachability region. Specifically, it represents the set of points $q'$ whose sum of distances two foci $q_1$ and $q_2$ is less than $2a$, where $a$ is the major radius of the ellipsoid.

## Revision:

**Definition 2.** *Consider a robot $i$ starting from $q_1$ at time $t_1$ and reaching $q_2$ at time $t_2$. The* reachability region *between $t_1$ and $t_2$ is defined as the set of points $q'$ in the free configuration space for which there exists a kinematically feasible trajectory containing $q'$.*

For simplicity, in this paper we consider only a maximum velocity constraint $v_{max}$; the reachability region for $q'$ can then be over-approximated by the following (the over-approximation is obtained by neglecting physical obstacles):

**Definition 3.** *The* reachability ellipsoid *$\mathcal{E}$ is defined as the region $\mathcal{E}(q_1, q_2, t_1, t_2) = \{\tilde{q} \in \mathbb{R}^n : d(q_1, \tilde{q}) + d(\tilde{q}, q_2) < 2a\}$, where $a = \frac{v_{max}}{2}(t_2 - t_1)$, and $d(\cdot, \cdot)$ denotes the Euclidean distance between two points.*

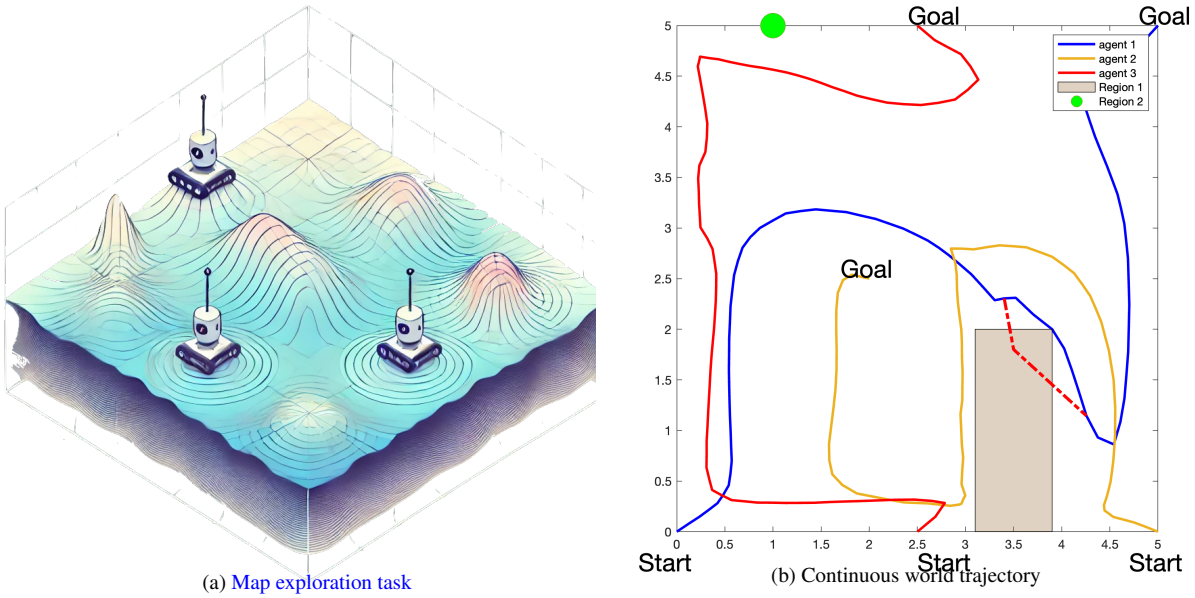
This region is an ellipsoid because it represents the set of points $q'$ whose sum of distances two *foci* $q_1$ and $q_2$ is less than $2a$, where $a$ is the major radius of the ellipsoid.

**Comment 3.6**

Fig 3: legends are too small.

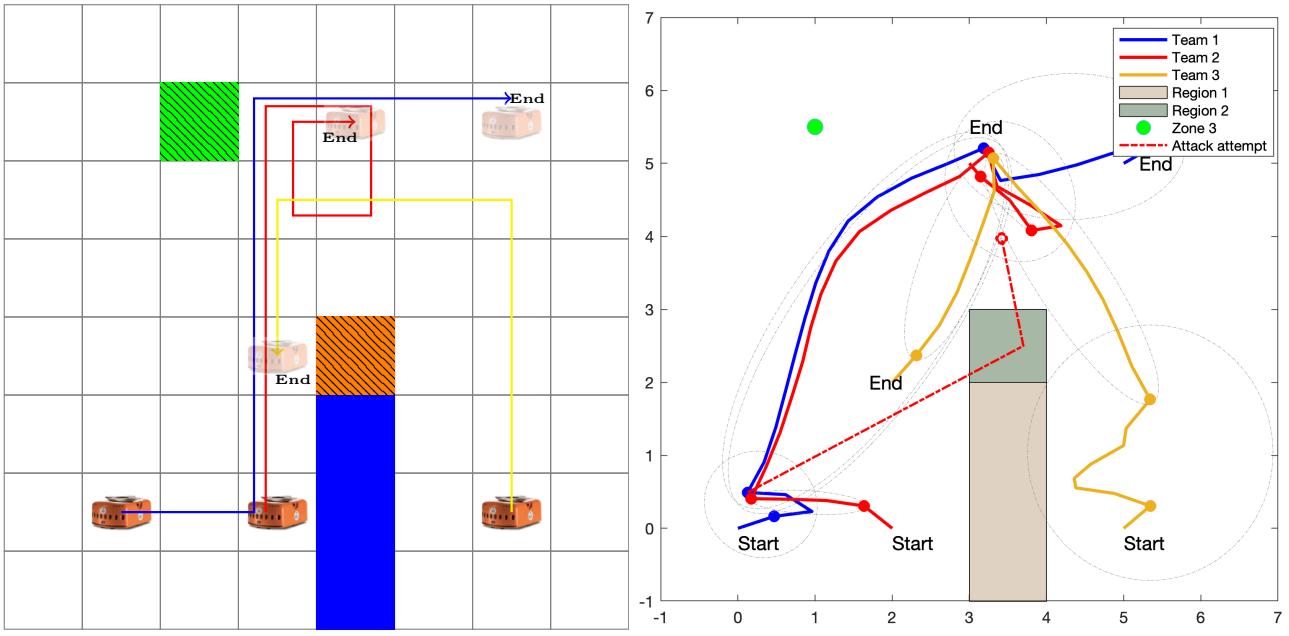
**Response:** We acknowledge that the legends and other aspects of the figures in the original manuscript were not clear. In the revised manuscript, we have addressed these issues by replotting the figures, including Figure 3, to ensure readability and clarity.

## Revision:




(a) Map exploration task (b) Continuous world trajectory

Figure 2: (2a) Illustration of a 3 robot map exploration task case. (2b) The unsecured trajectory design optimized for a map exploration task. Potential security breaches, indicated by red dashed lines, highlight paths that could allow unauthorized access to forbidden regions.




(a) Grid world result (b) Secured result.

Figure 4: Trajectory design of a three-robot system based on the grid world result. Zone 1 is an obstacle, Zone 2 and Zone 3 are forbidden regions. (4a) Result of the APMAPF algorithm in $8 \times 8$ grid-world without the map exploring objective. (4b) The secured trajectory with the incorporation of co-observation generated by APMAPF and additional reachability constraints. Attack attempts into the forbidden region (red dashed line) will cause team 1 miss the next scheduled co-observation.

**Comment 3.7**

4. The idea of the reachability ellipse seems very similar to another idea called 'safe flight corridor' from "Liu, Sikang, et al. "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments." IEEE Robotics and Automation Letters 2.3 (2017): 1688-1695." What is this essential difference between these two methods?

**Response:** Thank you for highlighting the potential similarity between the reachability ellipse and the Safe Flight Corridor (SFC) method. Our approach differs fundamentally in its purpose and application. Unlike the ellipsoids in IRIS and SFC, which are designed to approximate collision-free spaces, the reachability region in our method explicitly maps all reachable states given several known states (and regardless of collision) and is tailored to address security concerns.
In the revised manuscript, we have included more detailed content to clarify this distinction and provide a comprehensive explanation of the differences between our approach and SFC.

**Revision:** Ellipsoids are also used in other path-planning methods like Iterative Regional Inflation by Semidefinite programming (IRIS) [24],[25] and the Safe Flight Corridor (SFC) [26],[27]. Differing from the reachability region that requires mapping all reachable states given several known states, the ellipsoids of IRIS and SFC focus on approximating the safe collision-free space rather than addressing security applications.

[24] R. Deits and R. Tedrake, "Computing large convex regions of obstacle- free space through semidefinite programming," in Algorithmic Founda- tions of Robotics XI: Selected Contributions of the Eleventh International Workshop on the Algorithmic Foundations of Robotics. Springer, 2015, pp. 109–124.
[25] A. Ray, A. Pierson, and D. Rus, "Free-space ellipsoid graphs for multi- agent target monitoring," in 2022 International Conference on Robotics and Automation (ICRA). IEEE, 2022, pp. 6860–6866.
[26] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments," IEEE Robotics and Automation Letters, vol. 2, no. 3, pp. 1688–1695, 2017.
[27] D. Fan, Q. Liu, C. Zhao, K. Guo, Z. Yang, X. Yu, and L. Guo, "Flying in narrow spaces: Prioritizing safety with disturbance-aware control," IEEE Robotics and Automation Letters, 2024.