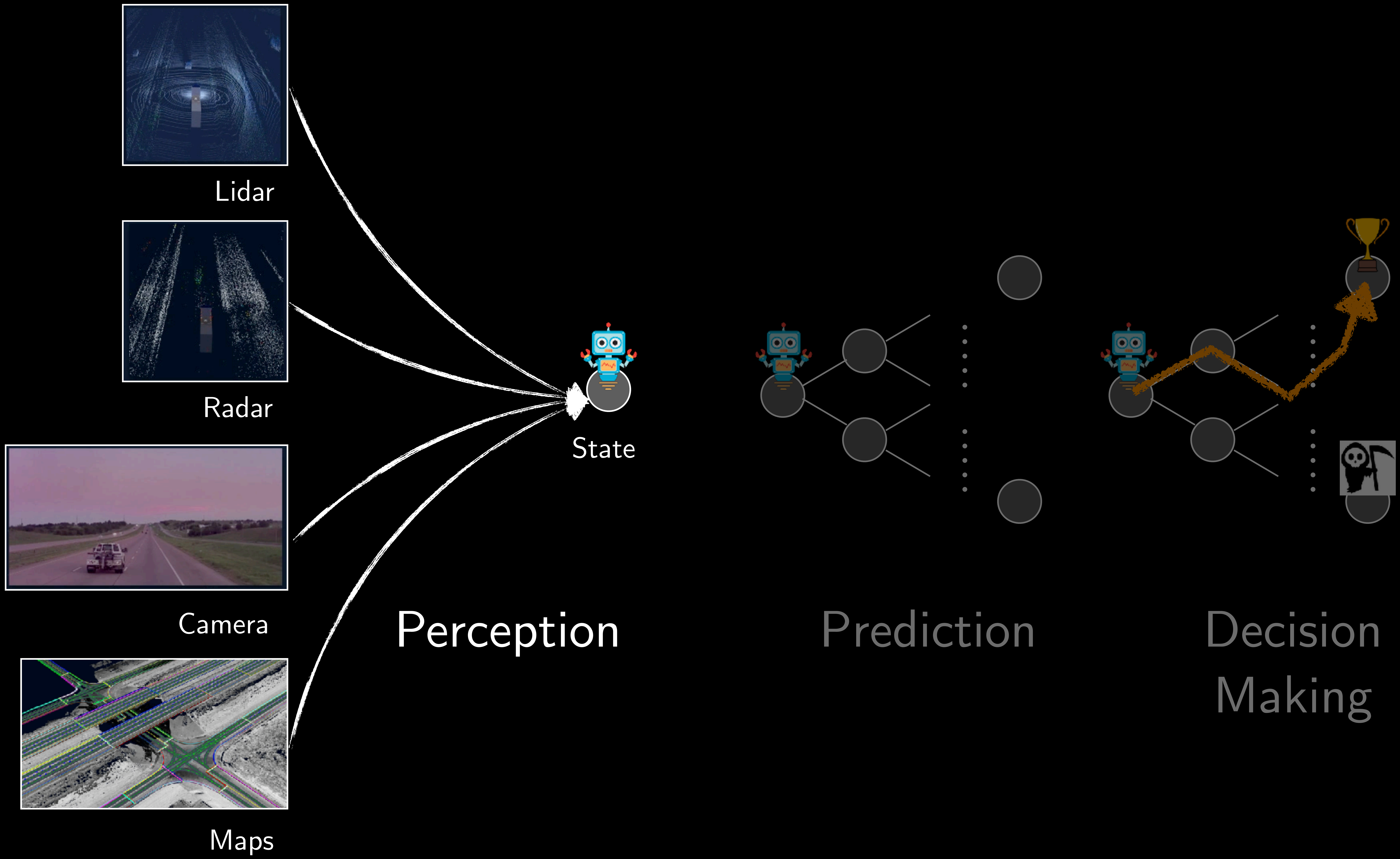
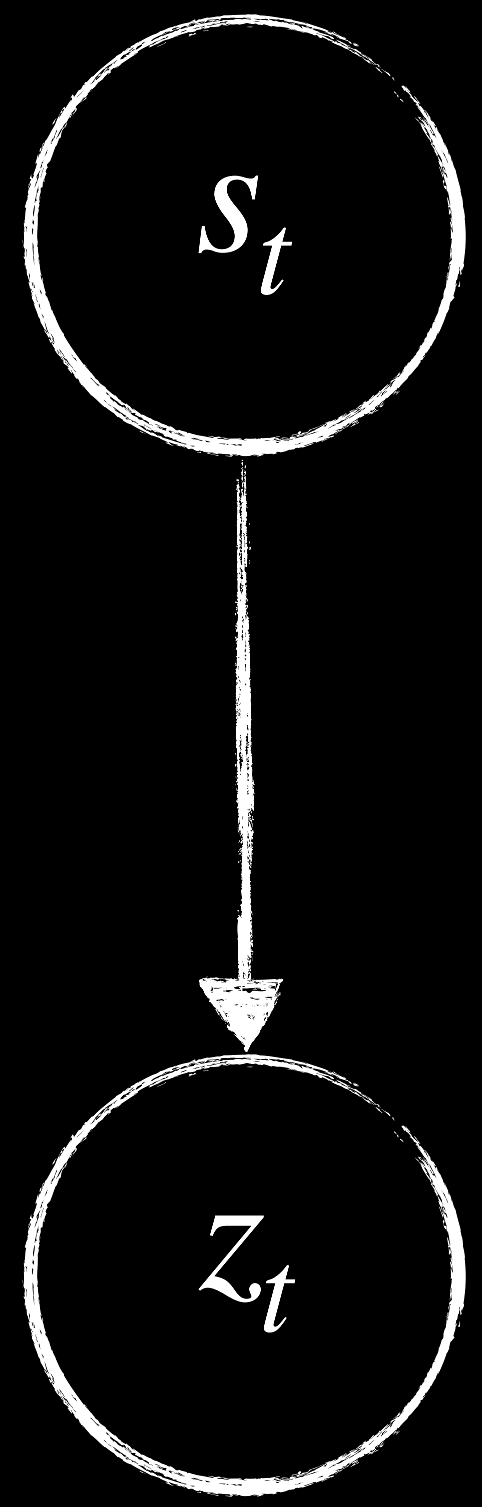
SLAM as Graph Optimization

Sanjiban Choudhury

Cornell Bowers CIS
Computer Science


Lidar
Radar
Camera
Maps
State
Perception
Prediction
Decision Making

Estimate state from observations



Perception so far ...

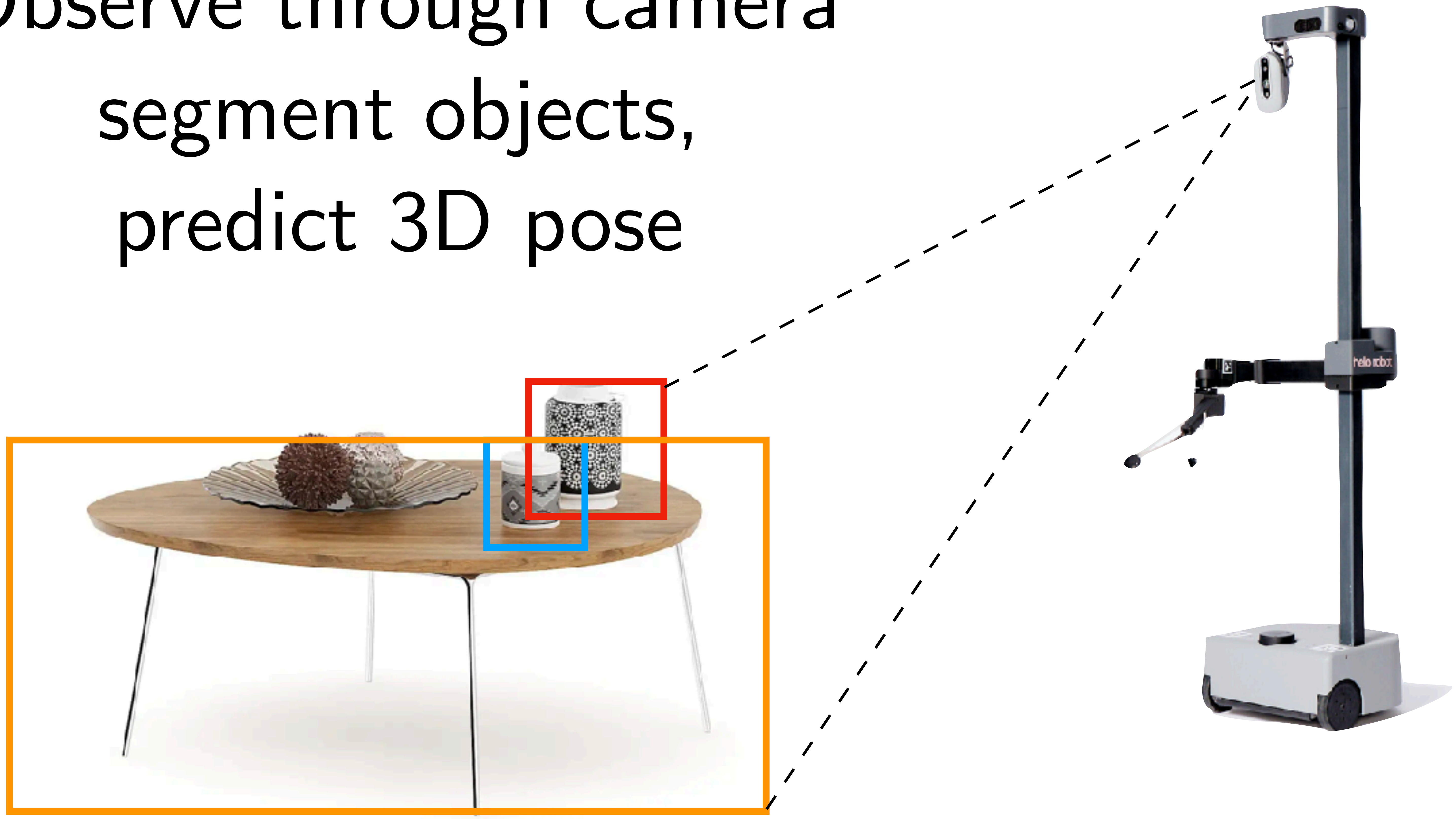
State of objects is unknown

State of the robot is known

Perception so far ...

State of objects is unknown

Observe through camera
segment objects,
predict 3D pose

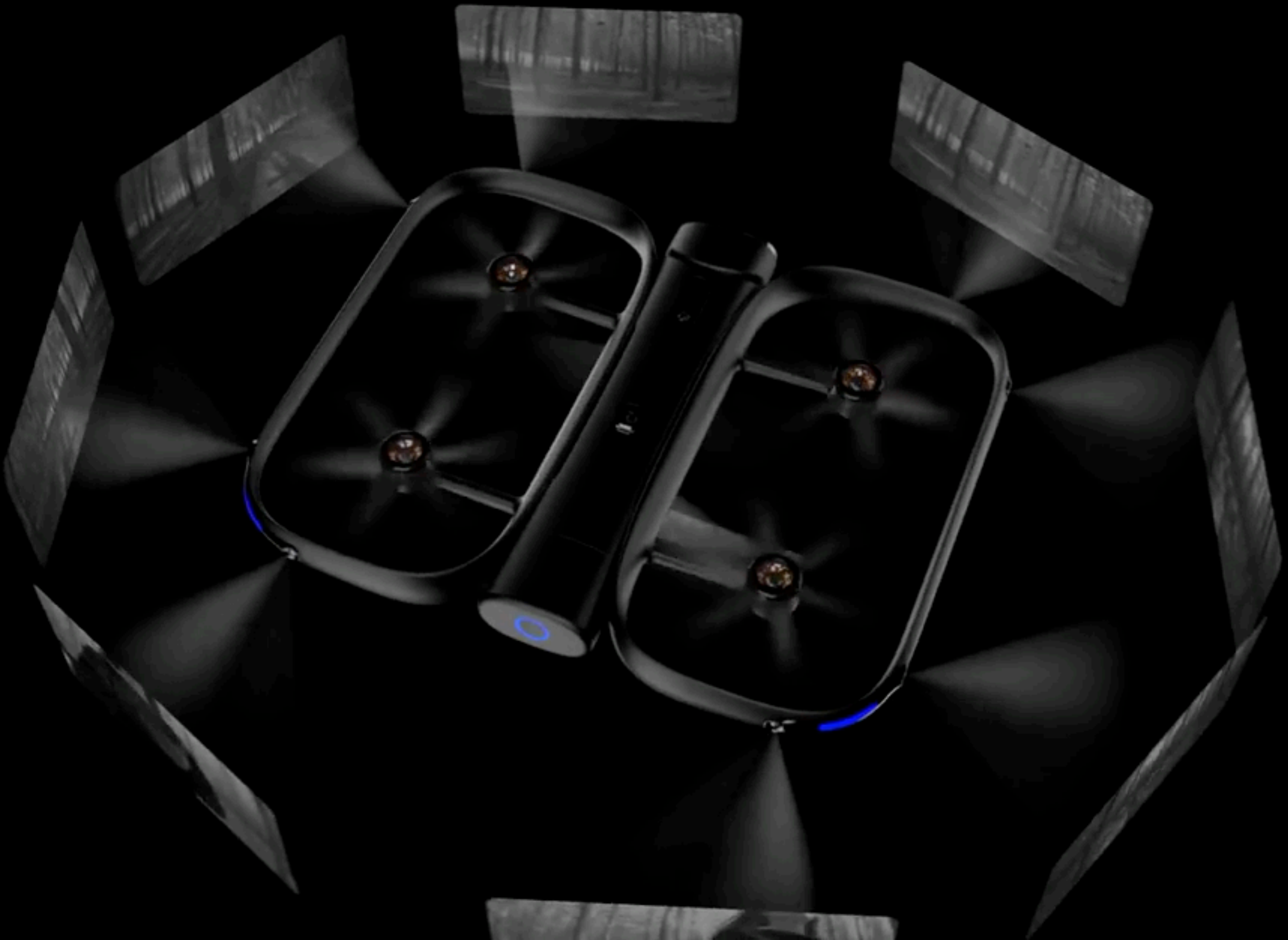


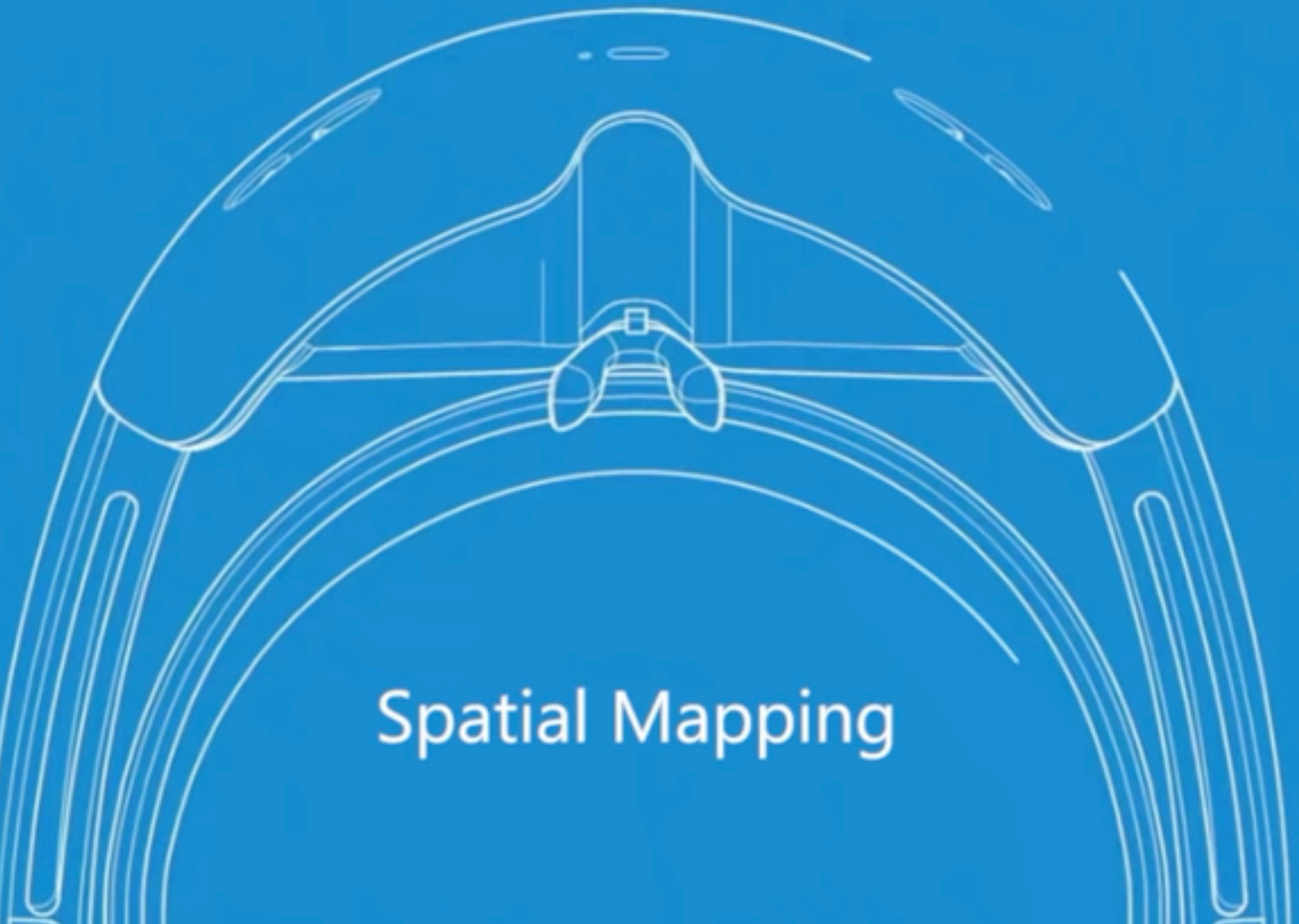
What if we don't know where the robot is?

Position of robot is unknown

Real World Applications

Skydio

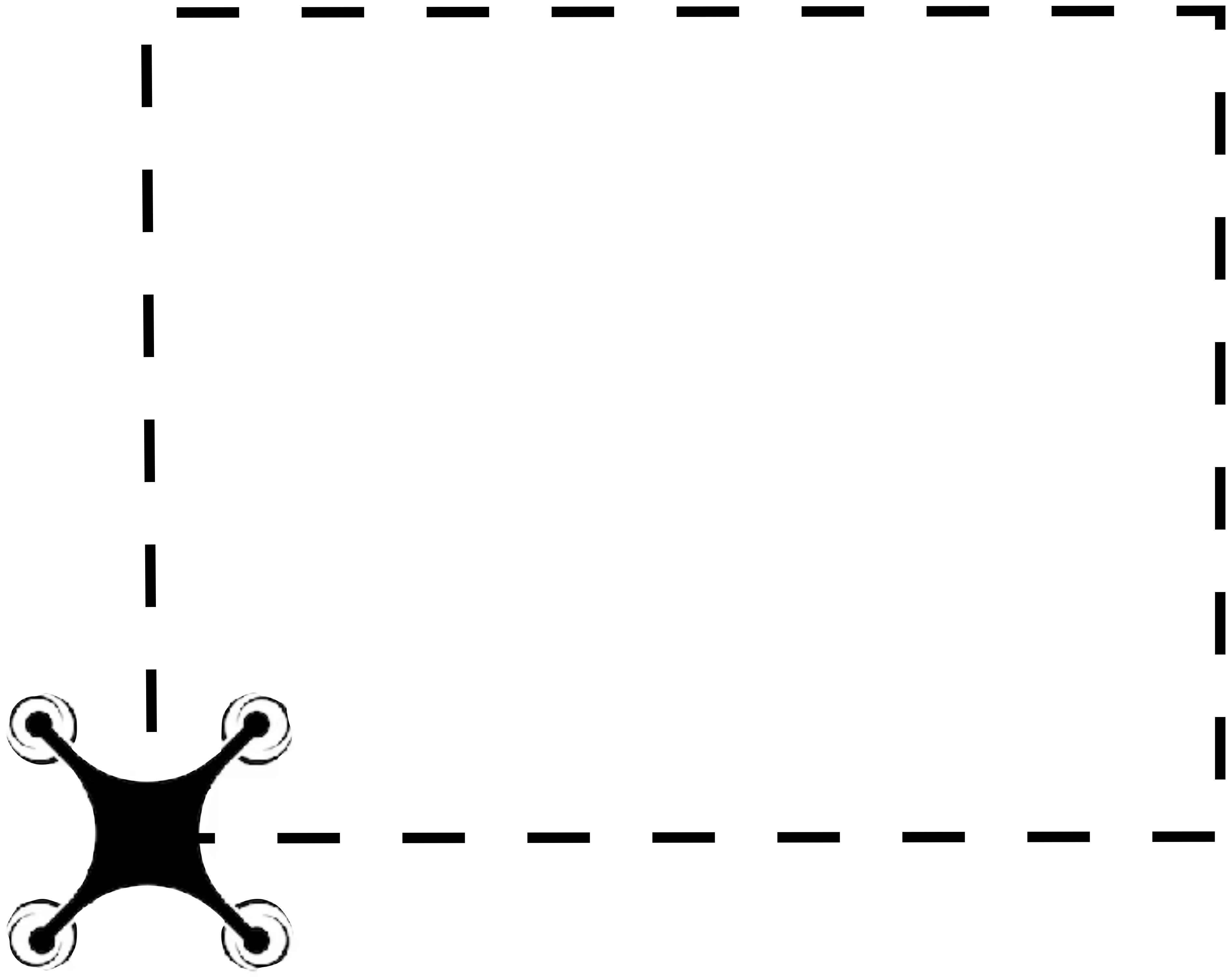




Spatial Mapping

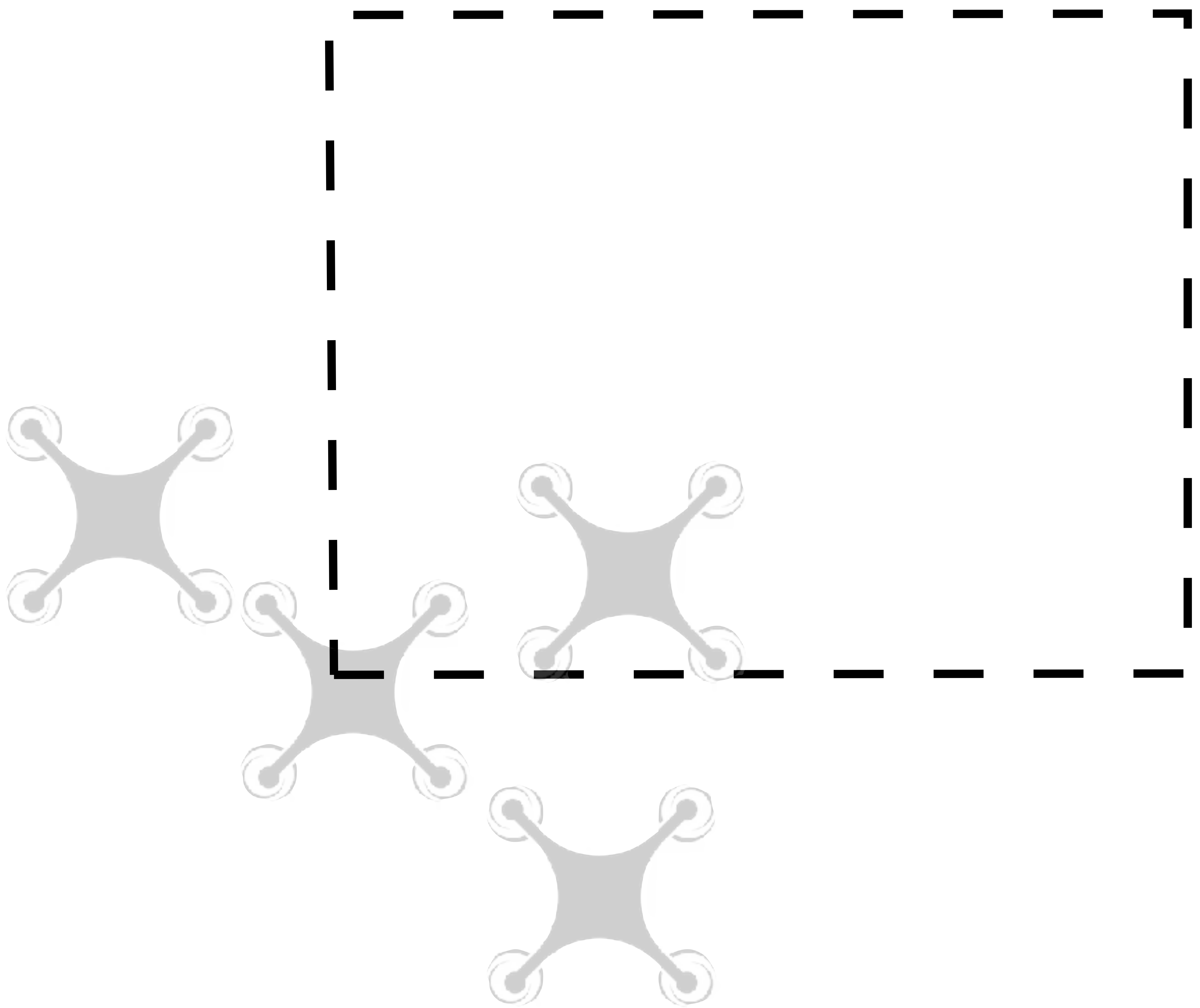
Microsoft HoloLens

A Toy Example



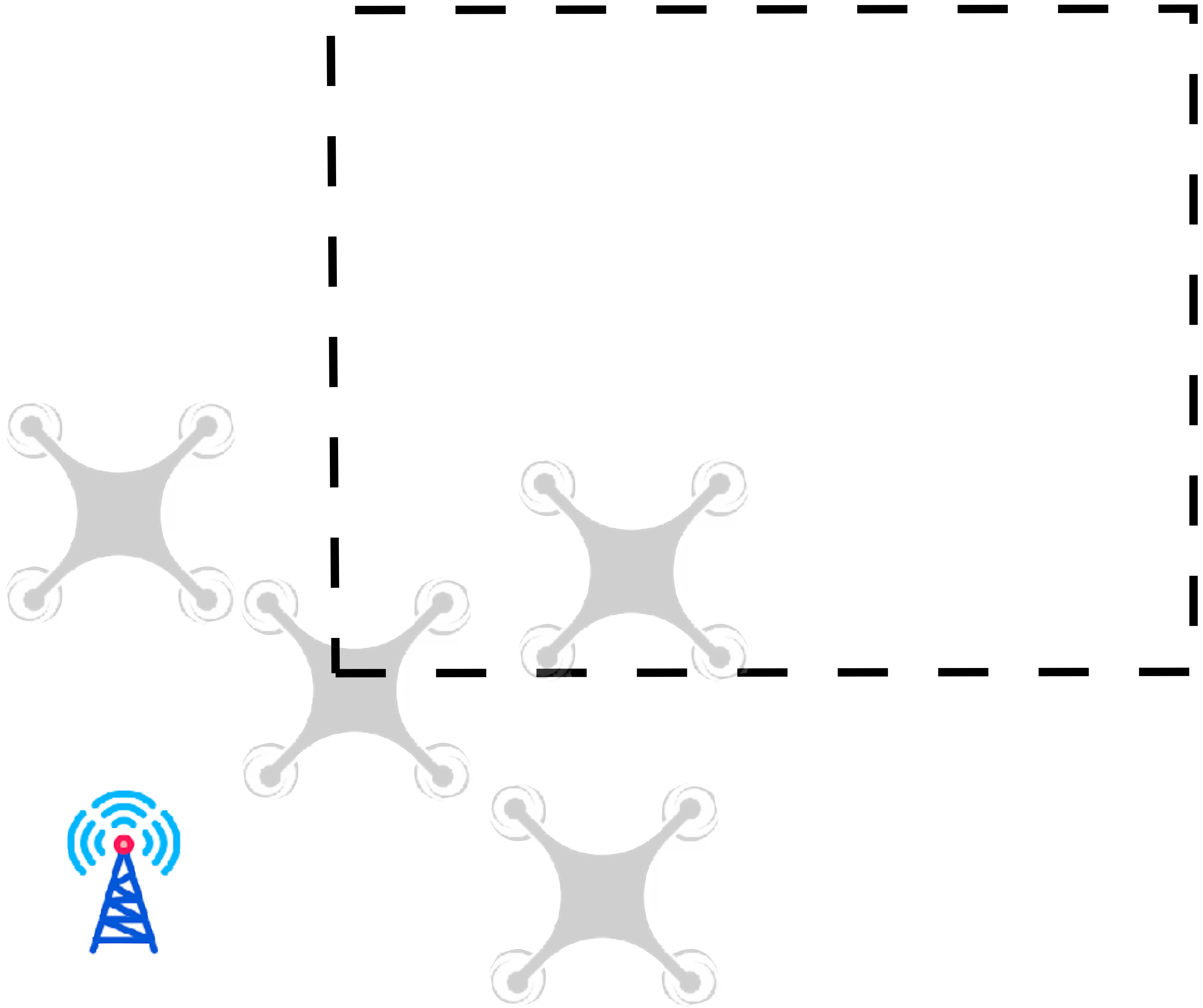
We have a drone that we are flying around in a circuit

T=0



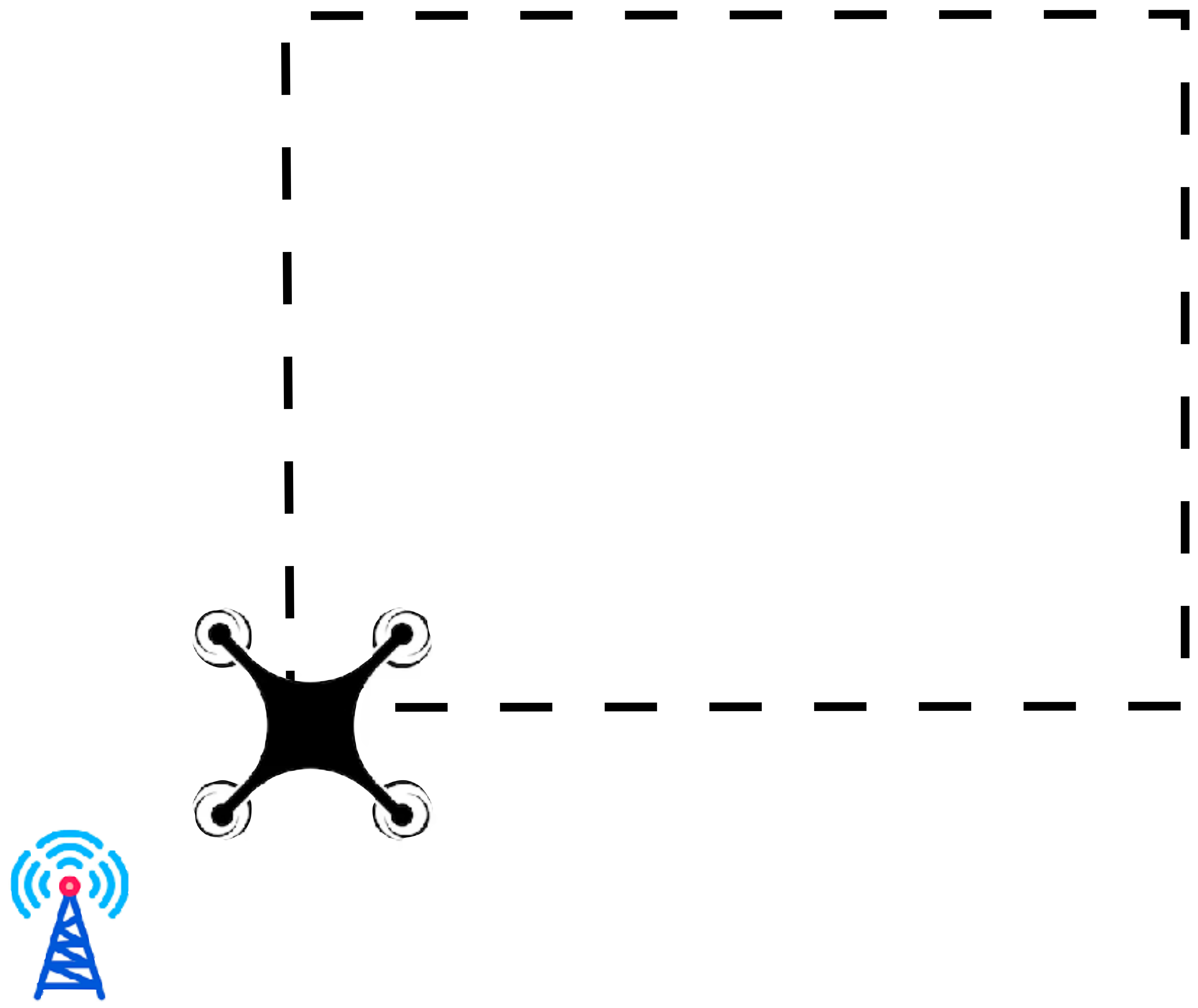
The 2D position is unknown

T=0



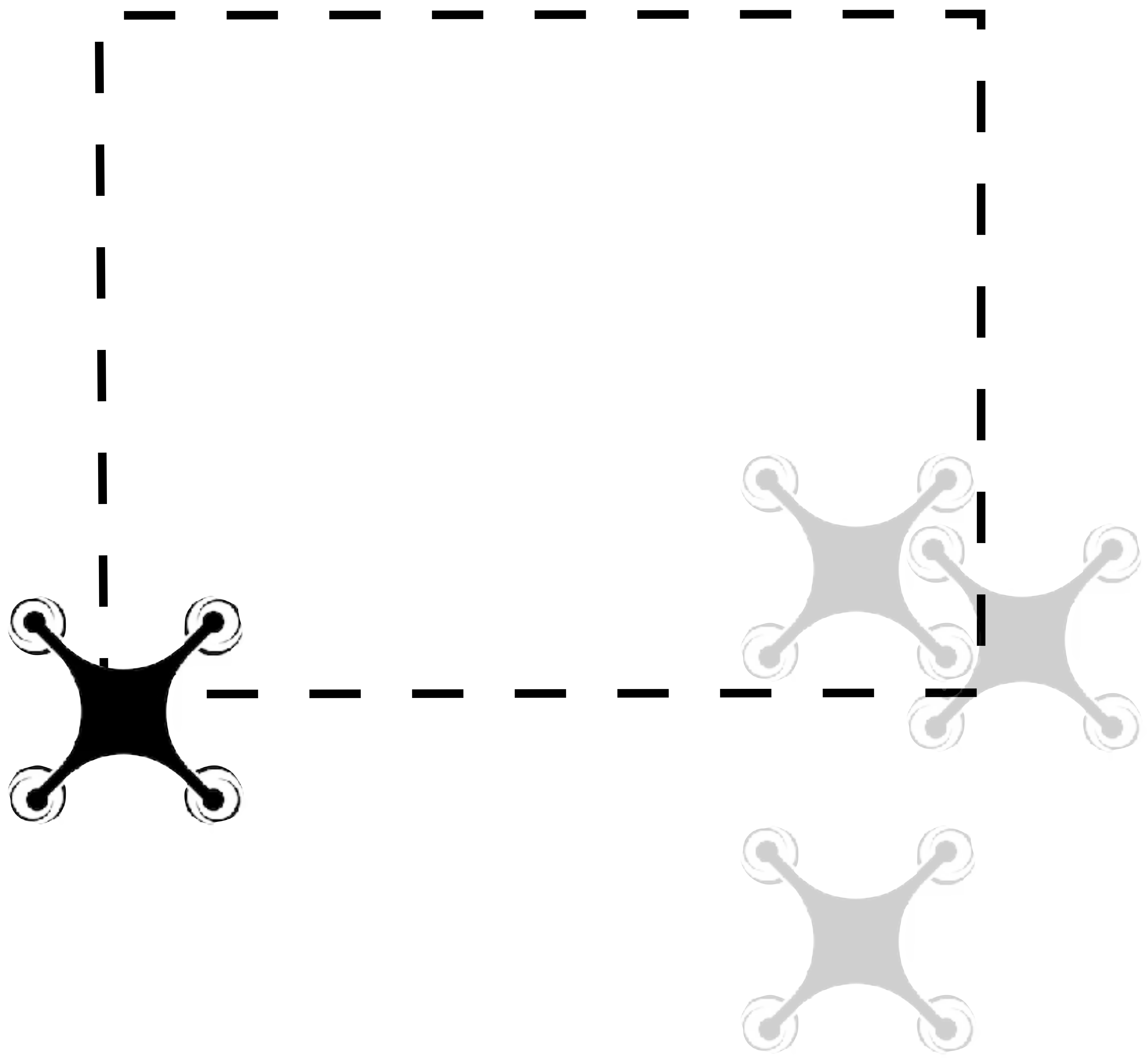
It observes a **landmark** whose position is known

T=0



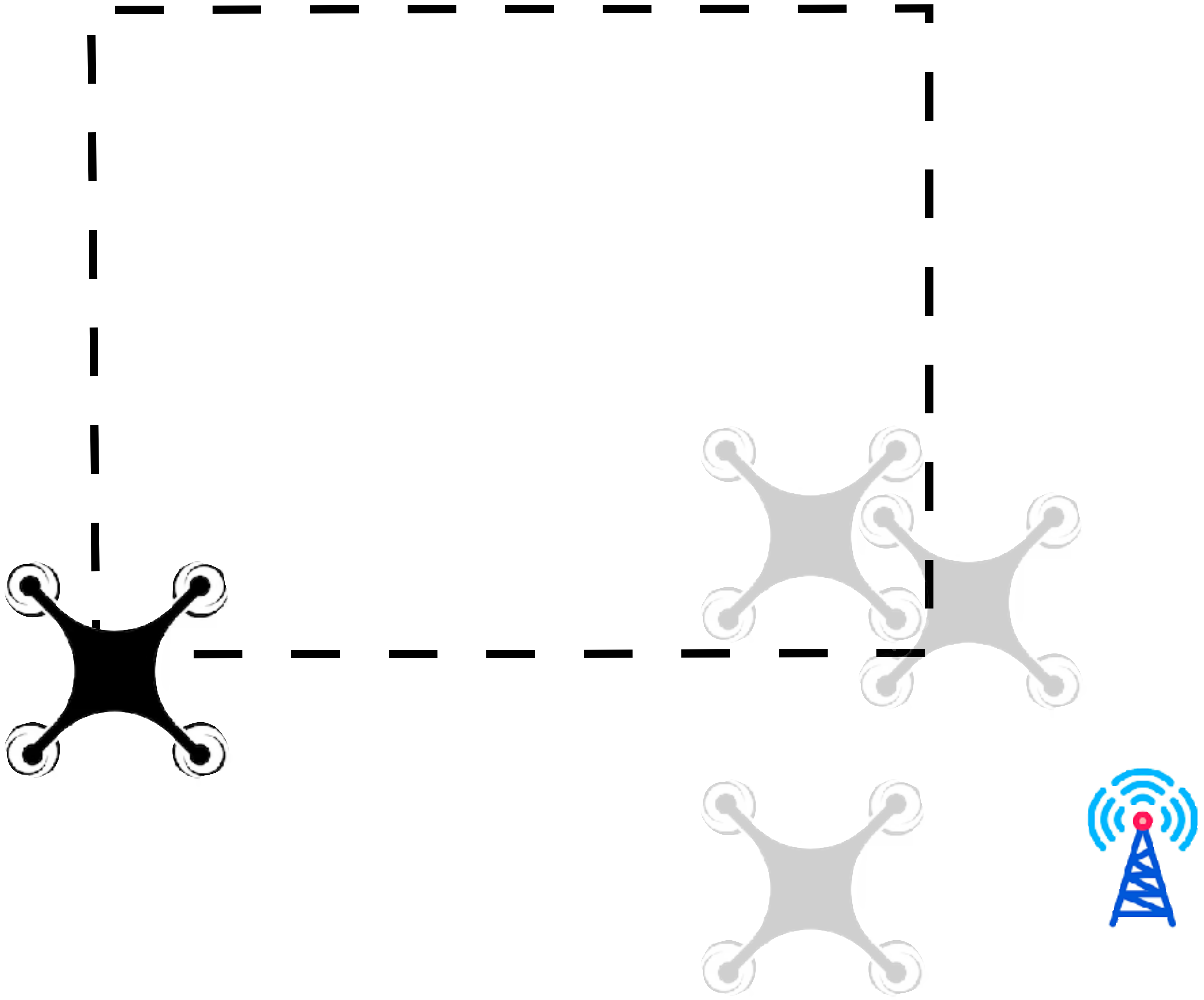
Using this observation, the robot updates it's position

T=1



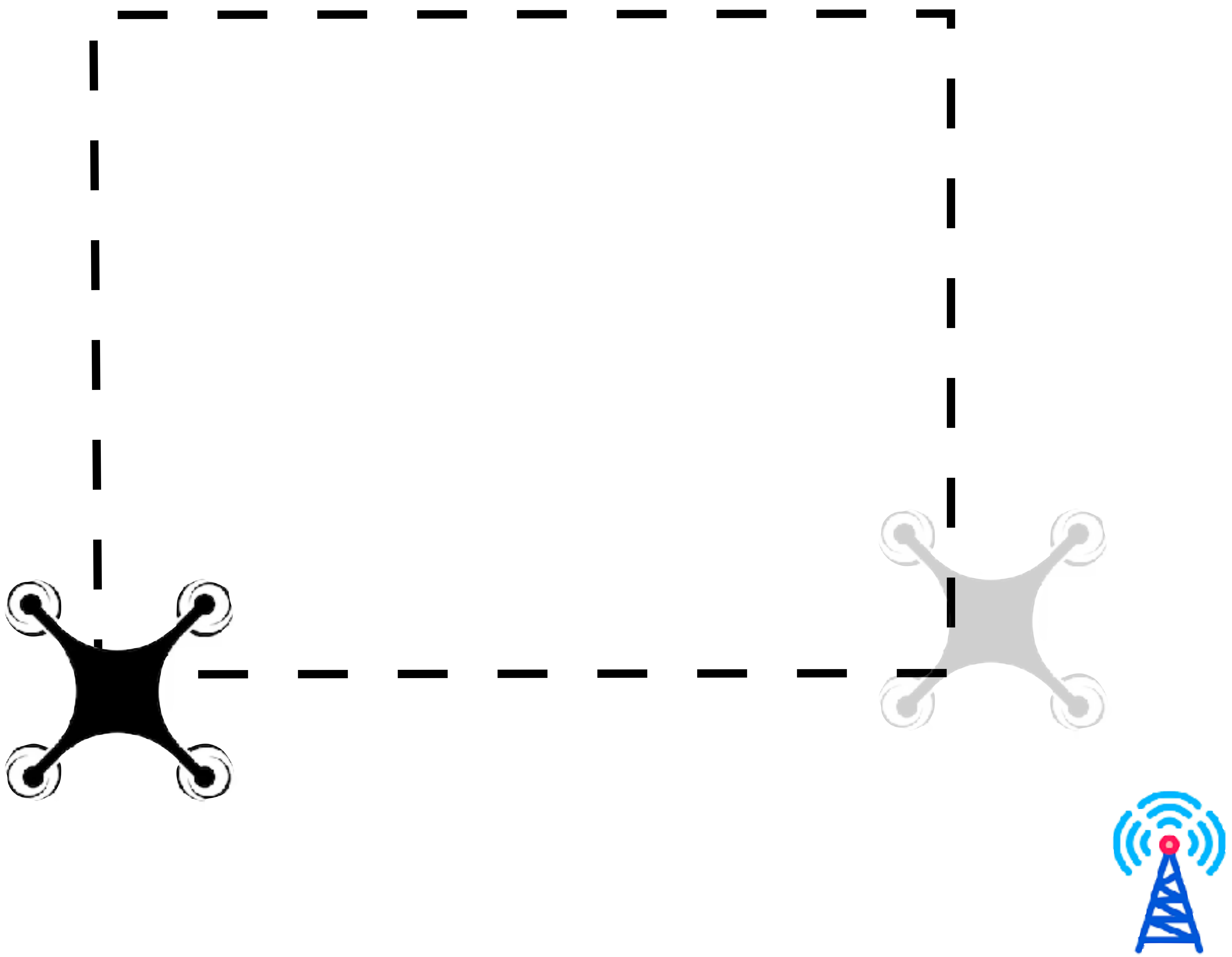
Predict the next pose based on dynamics

T=1



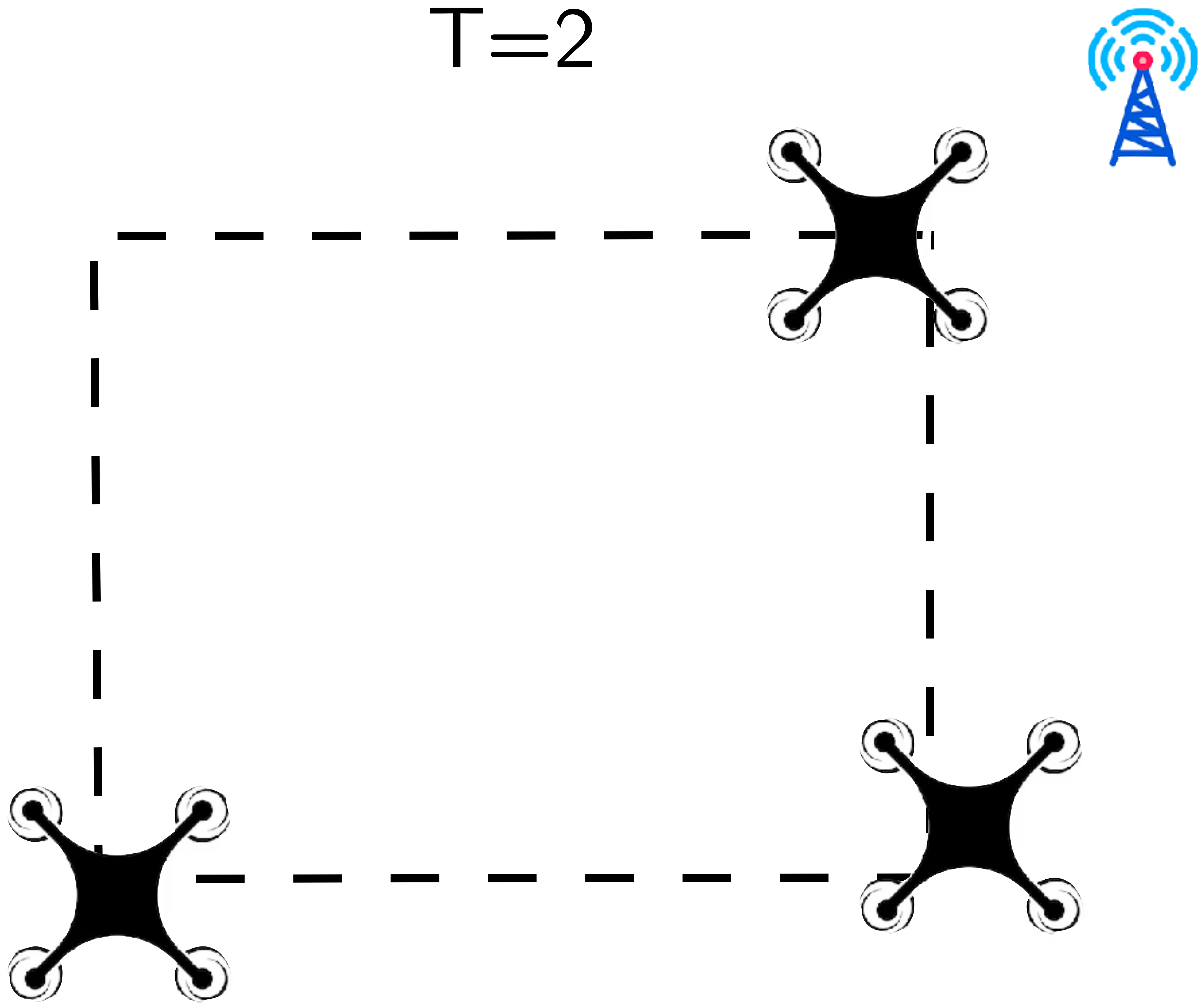
Observe a landmark

T=1

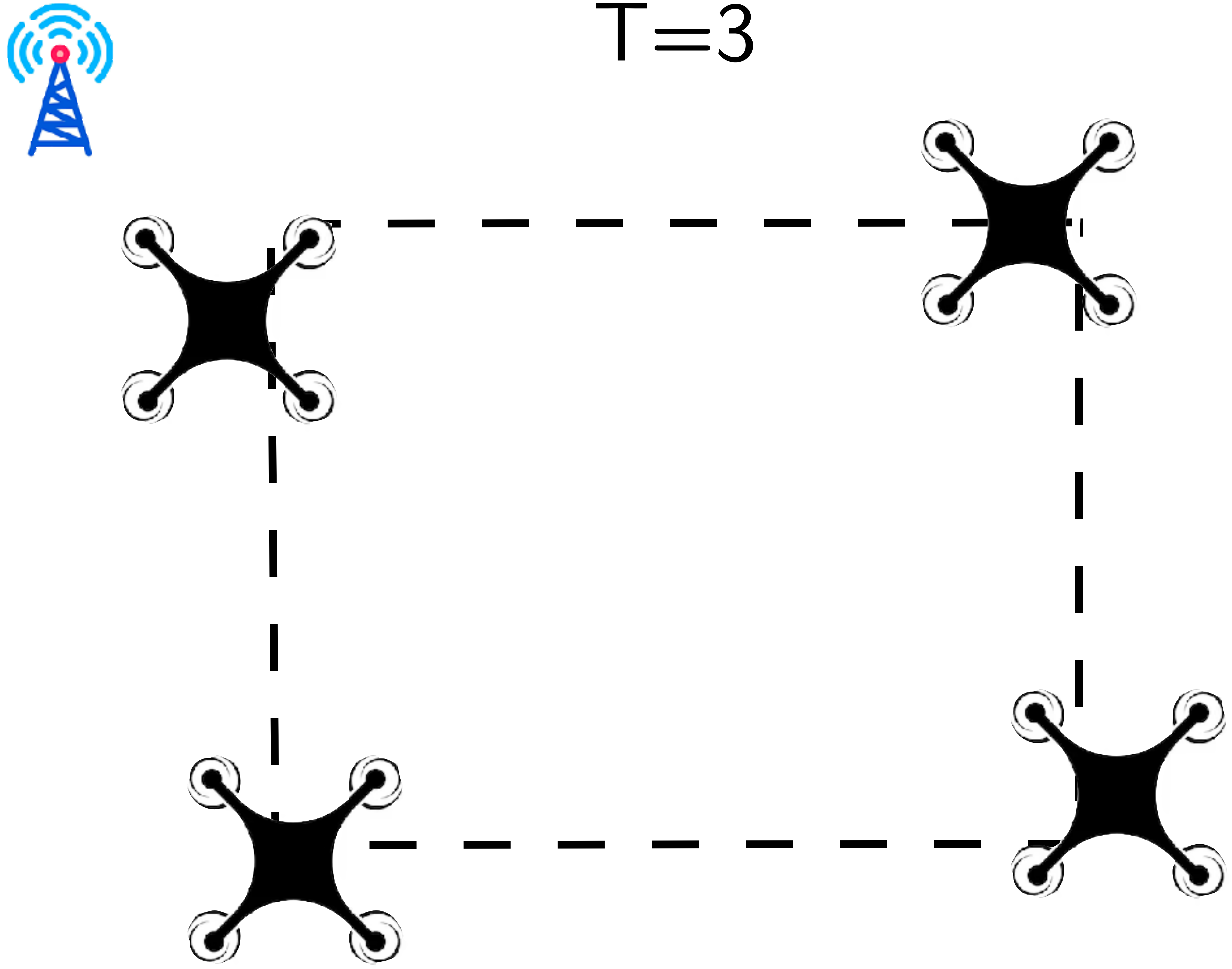


Update pose

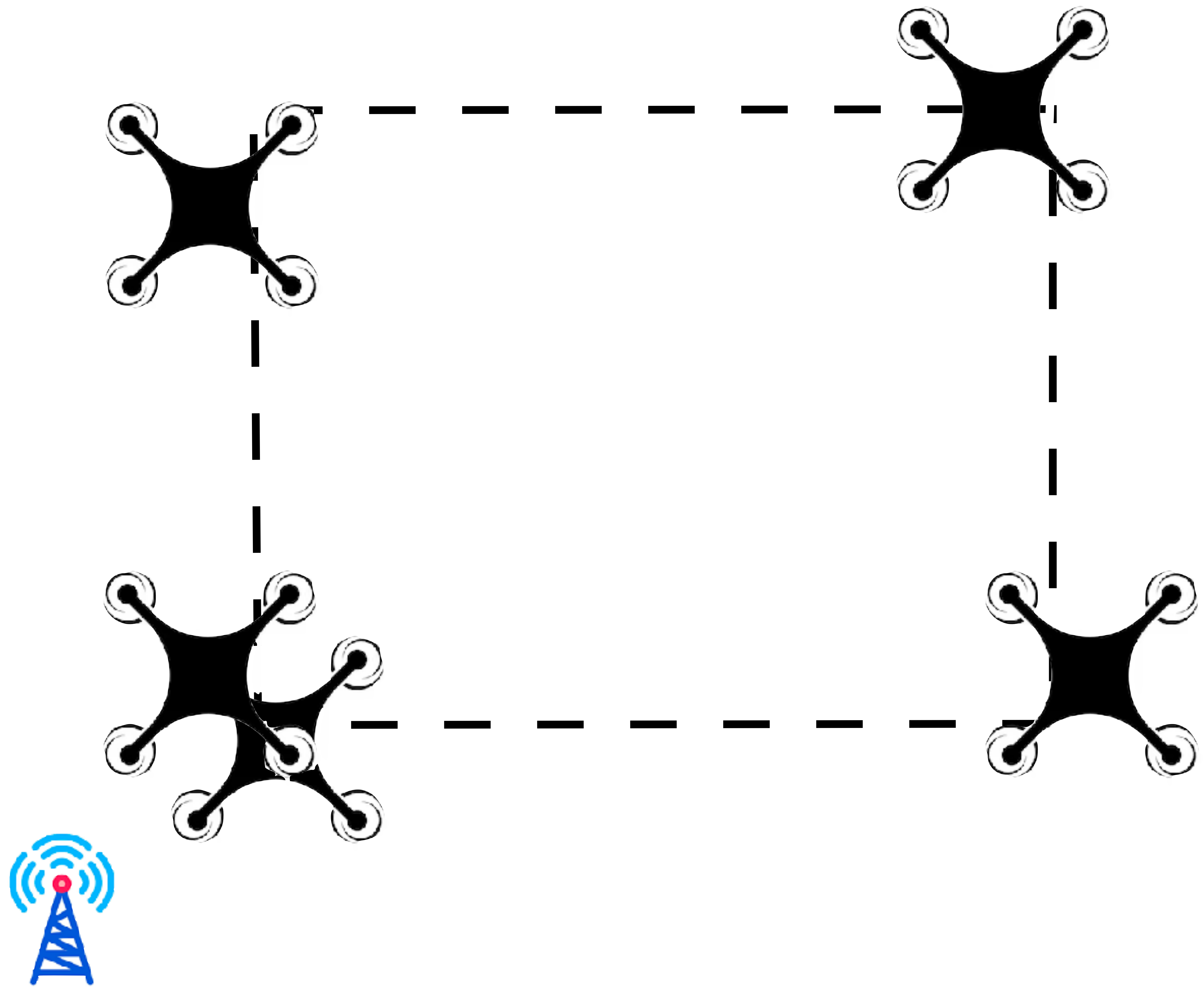
T=2



T=3



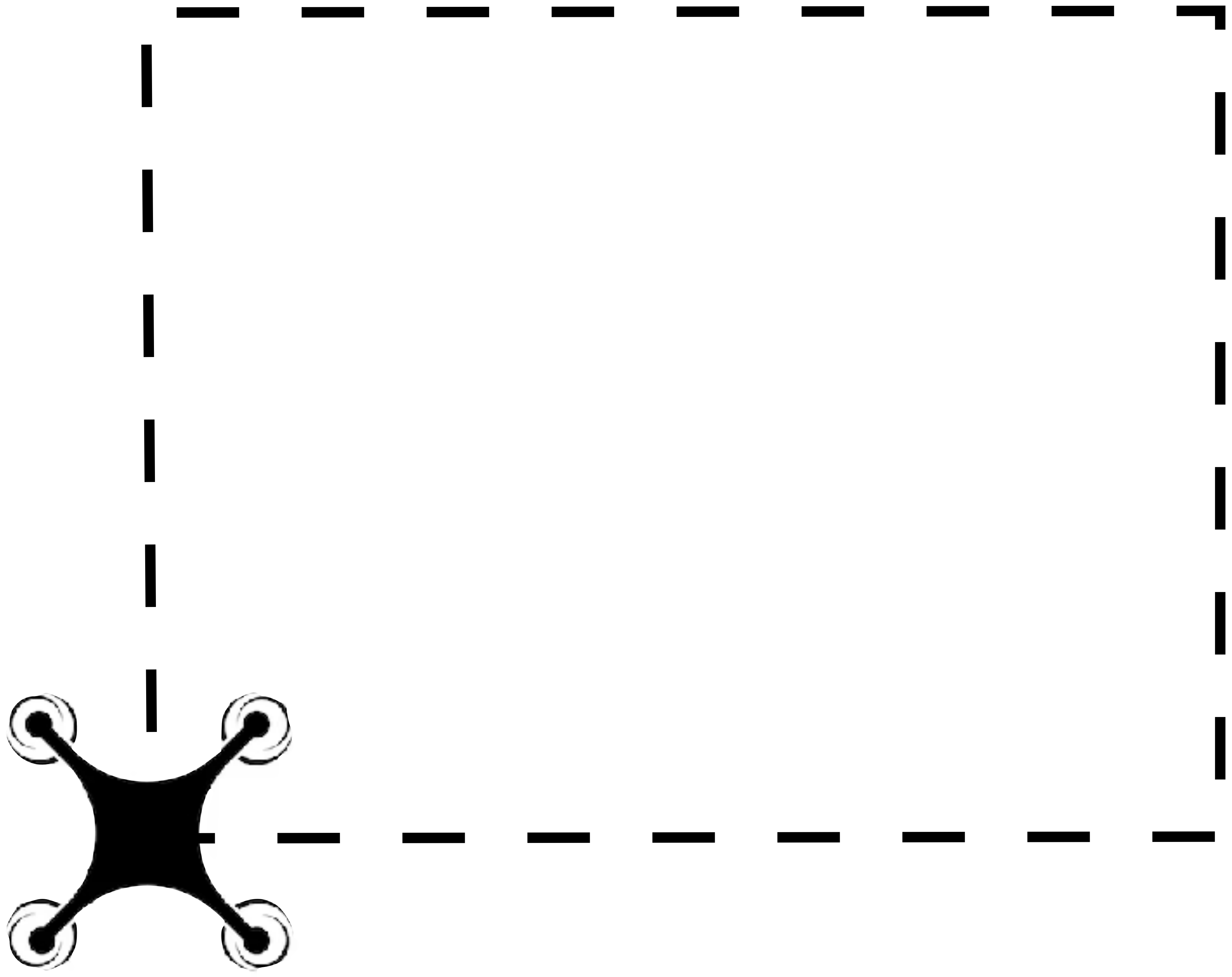
T=4



How do we mathematically solve for the poses at t=0,1,2,3,4 ?

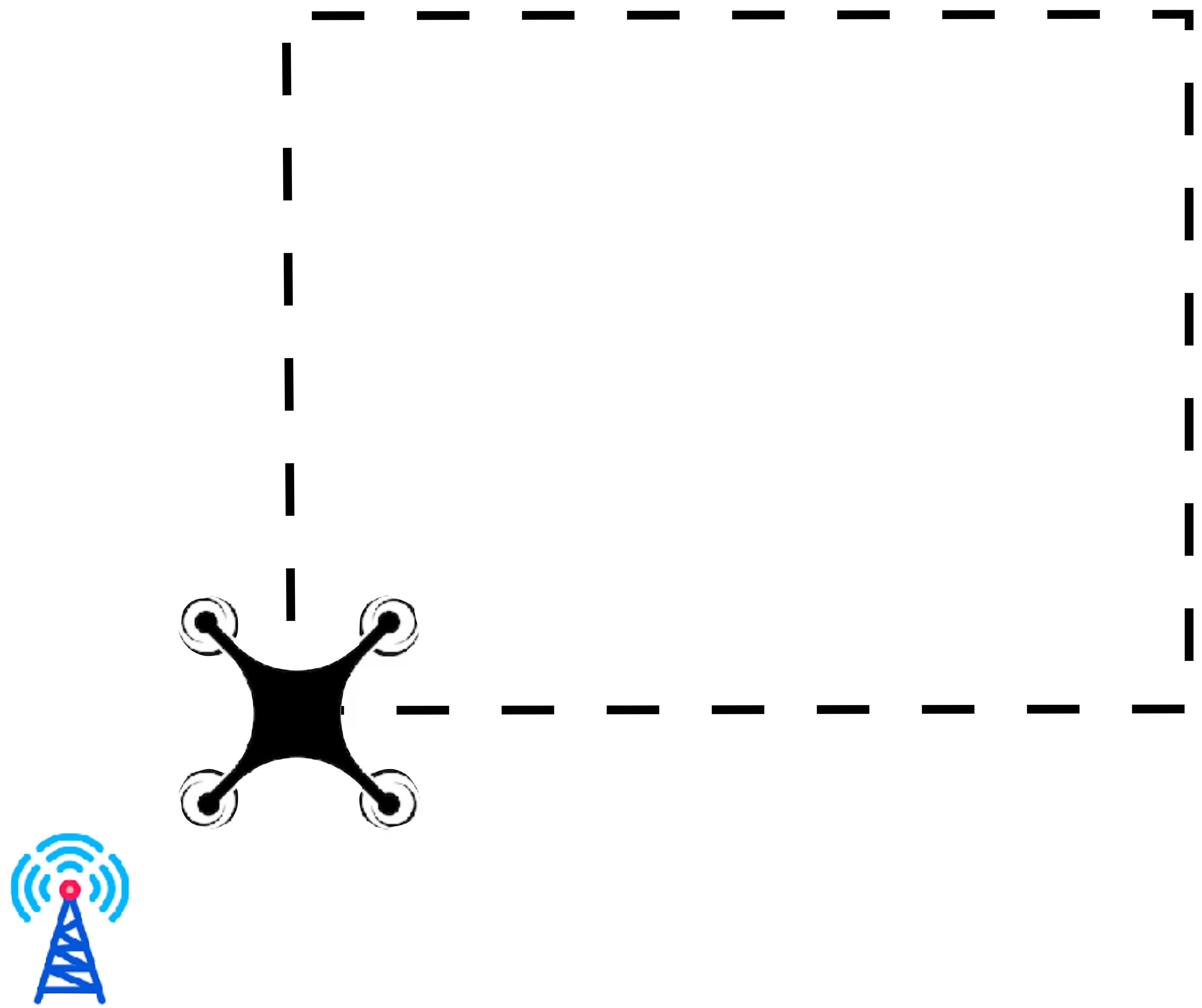
Let's do math!

Now … what if we don’t know all the landmarks?



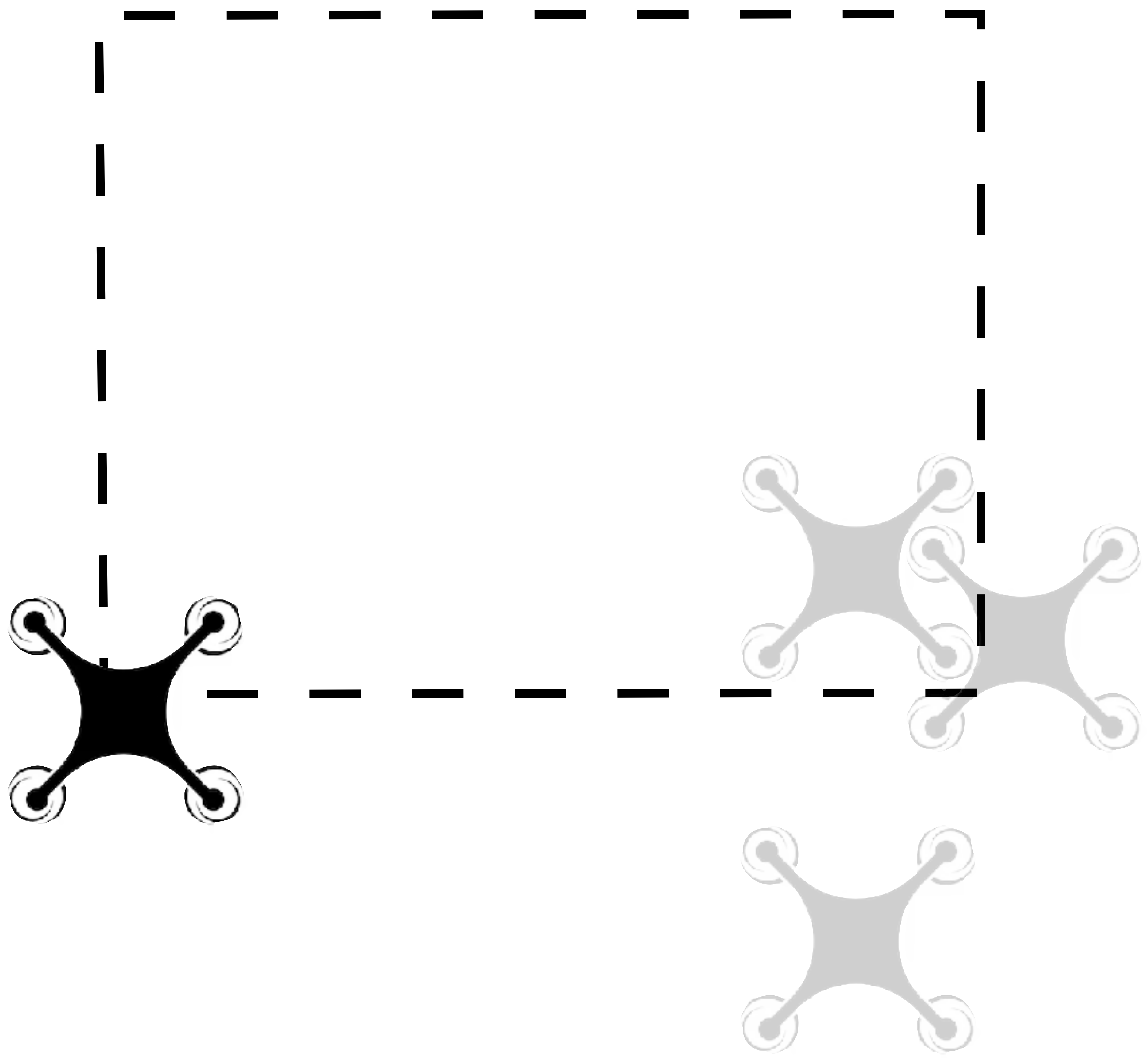
We have a drone that we are flying around in a circuit

T=0



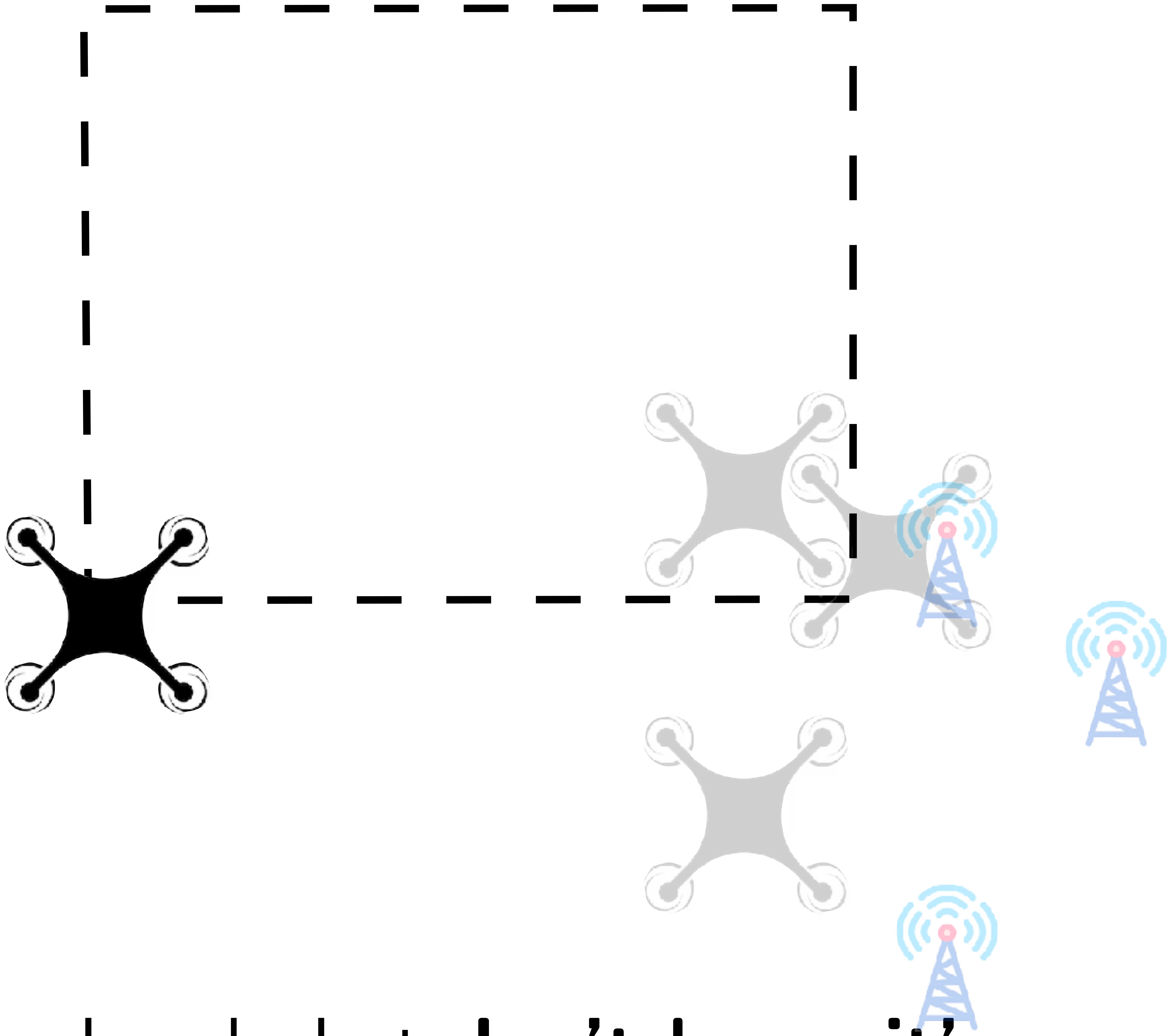
Let's say we know the pose at t=0, landmark at t=0

T=1



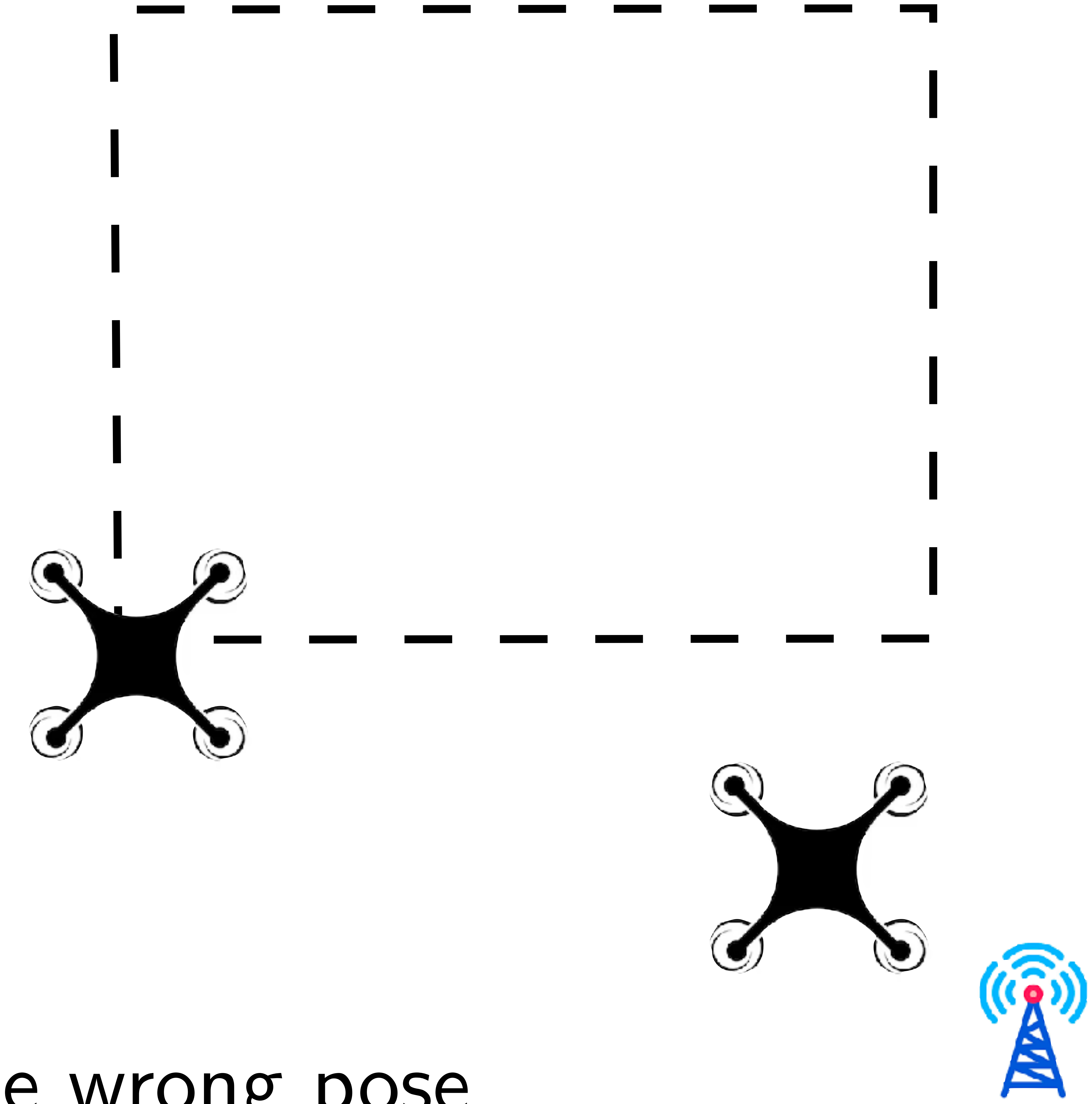
The pose at t=1 is **unknown.**

T=1



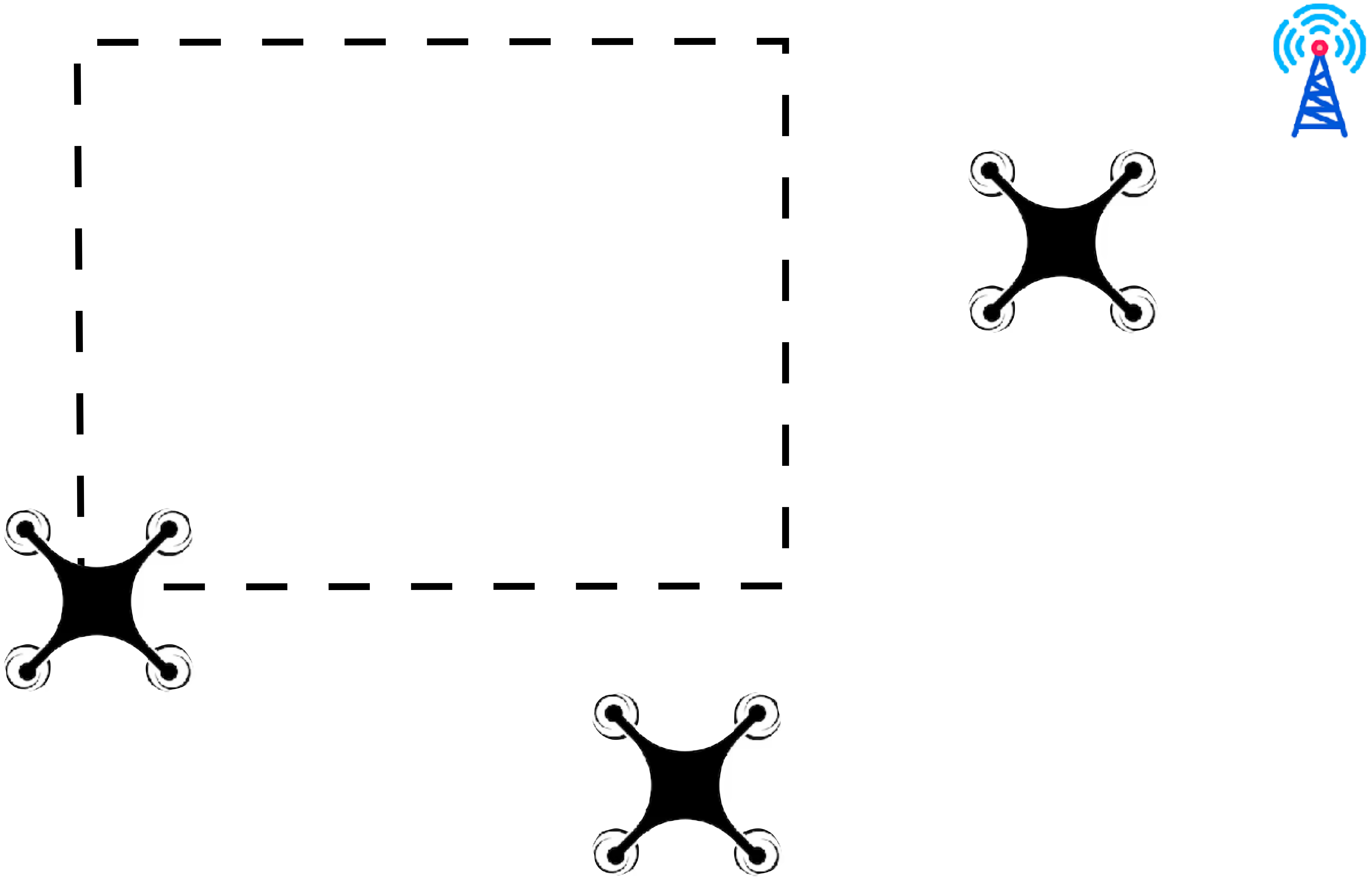
We observe a landmark. but **don't know it's pose either.**

T=1

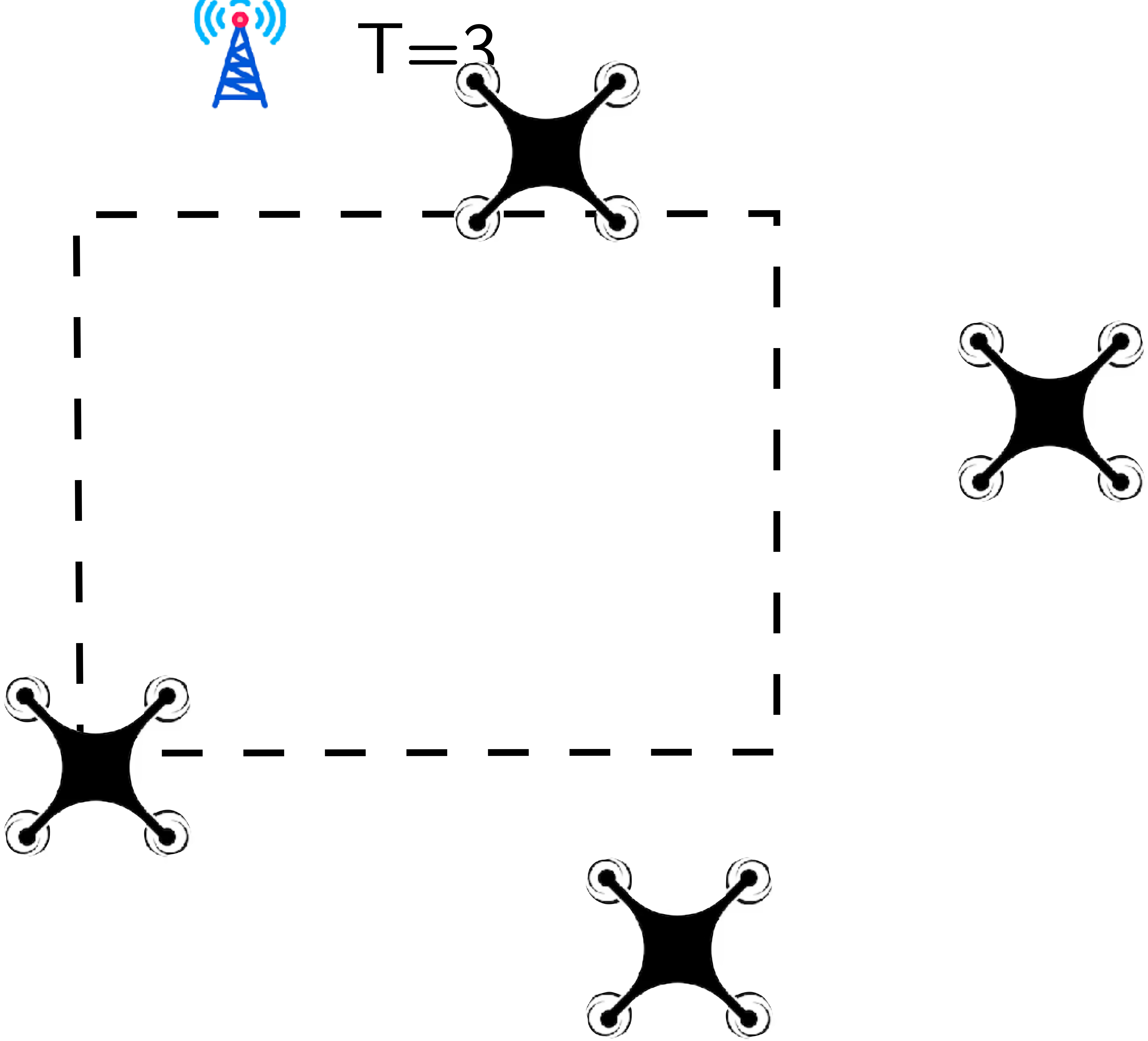


We latch on to the wrong pose

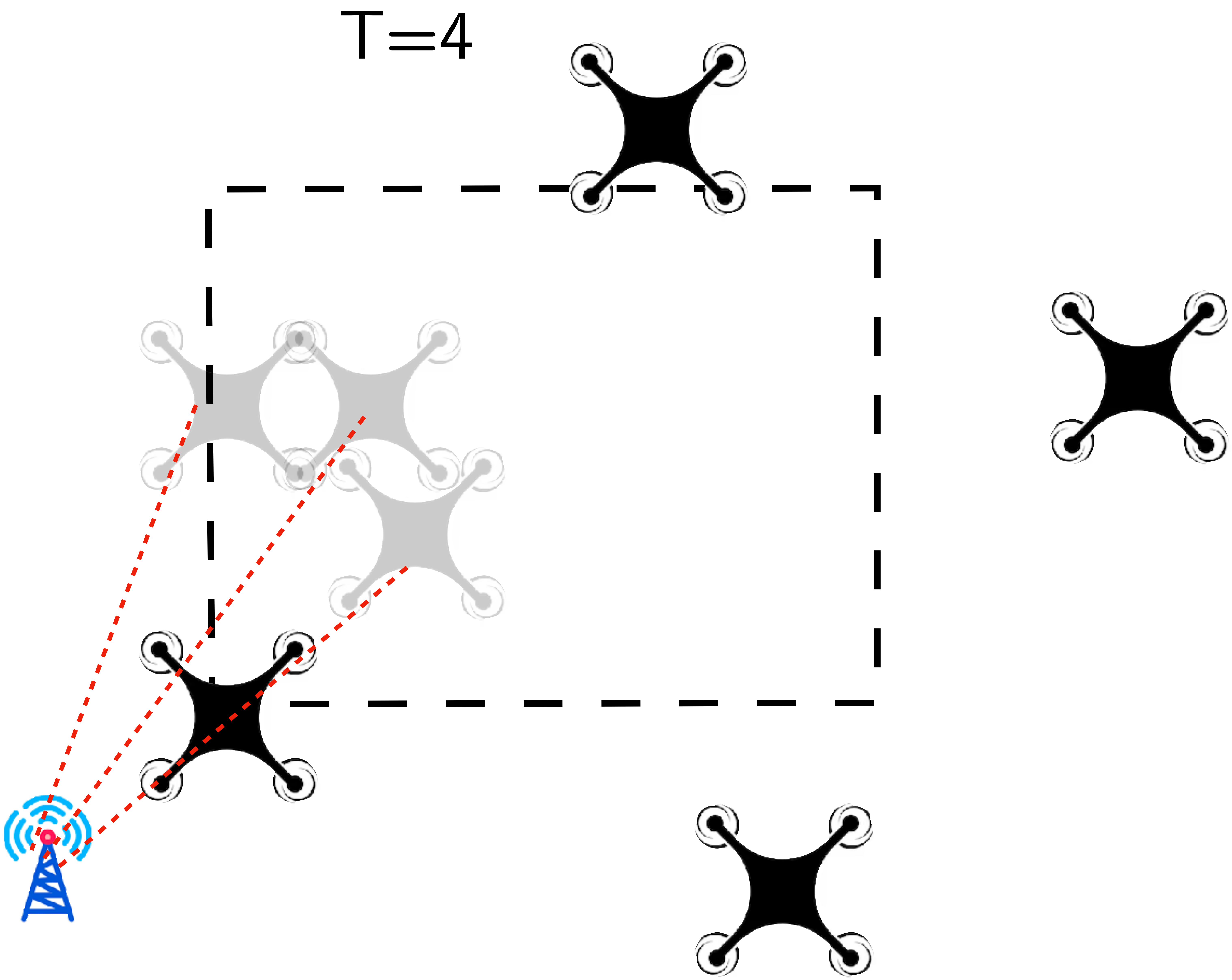
T=2



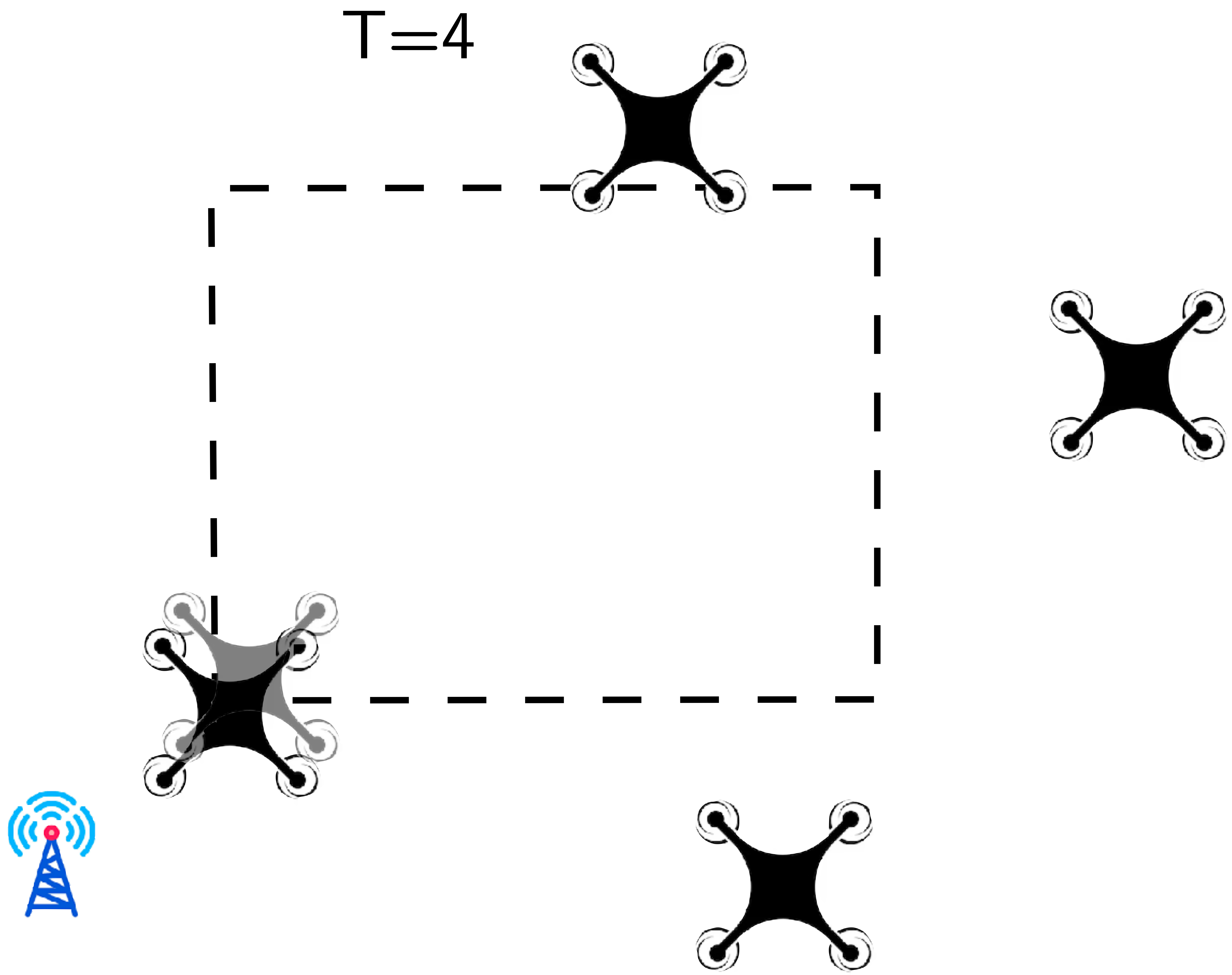
Continue deviating further ..



Continue deviating further ..

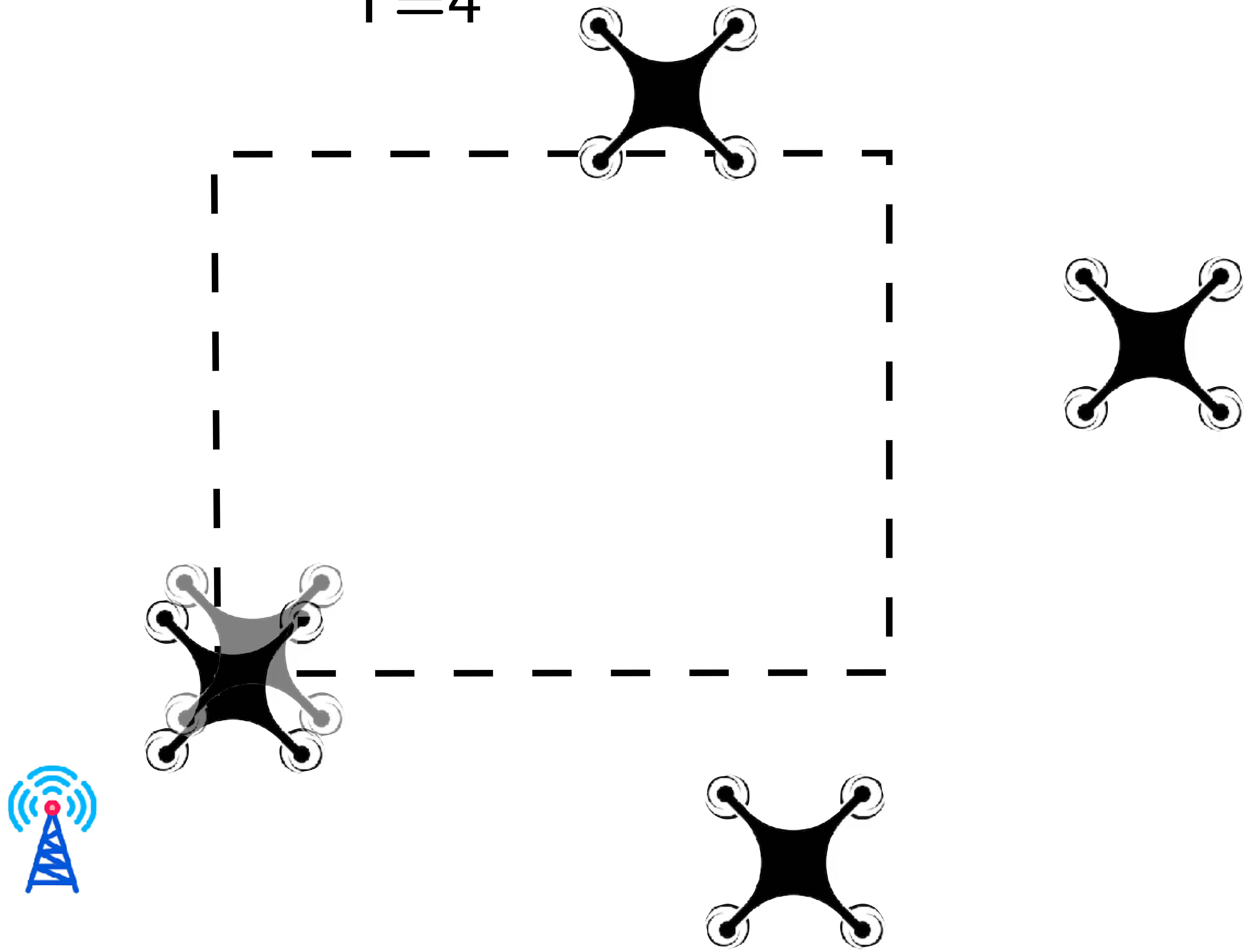


Now at t=4, we see the **same landmark** as t=0



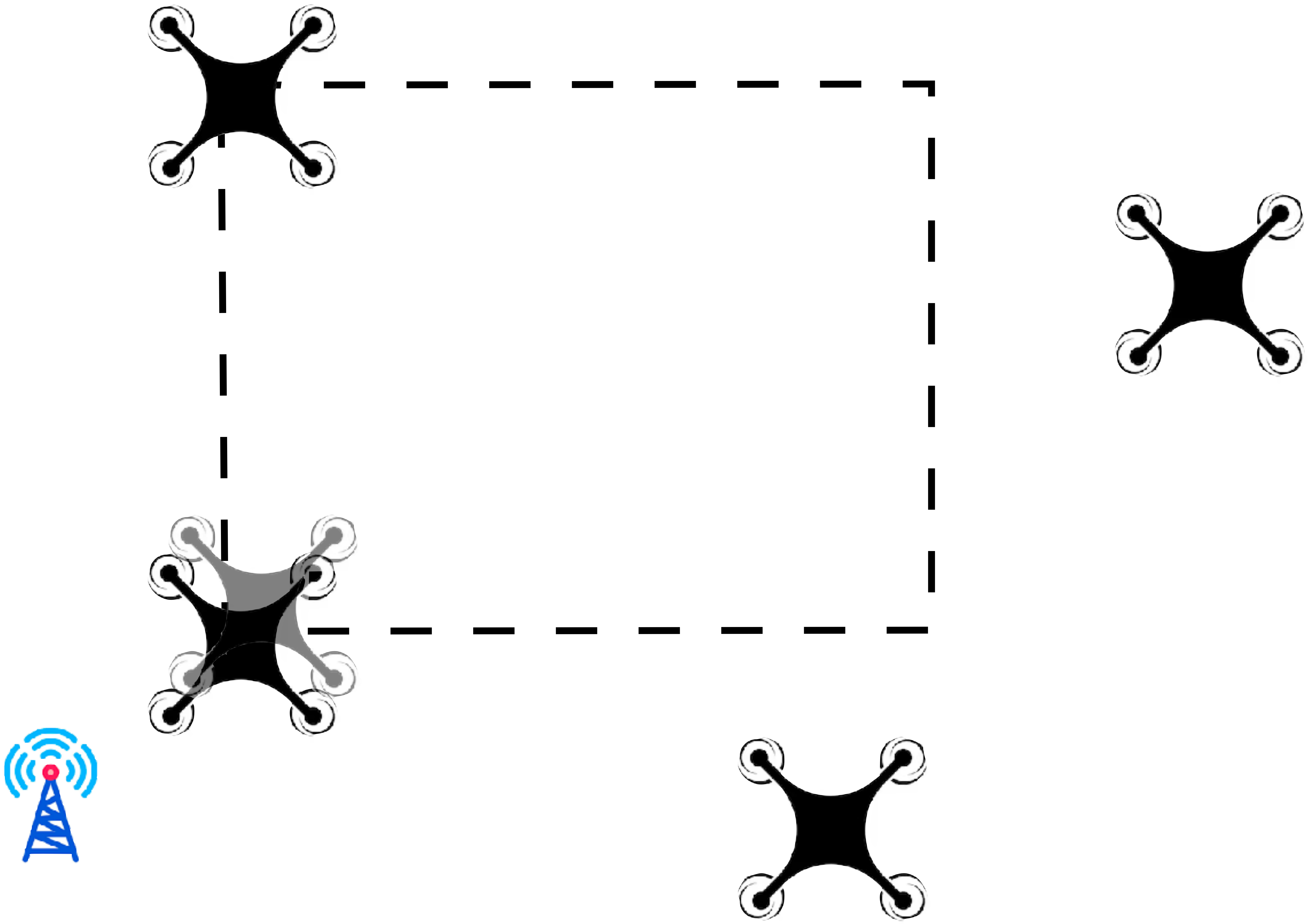
This should "snap" us to the correct position!

T=4



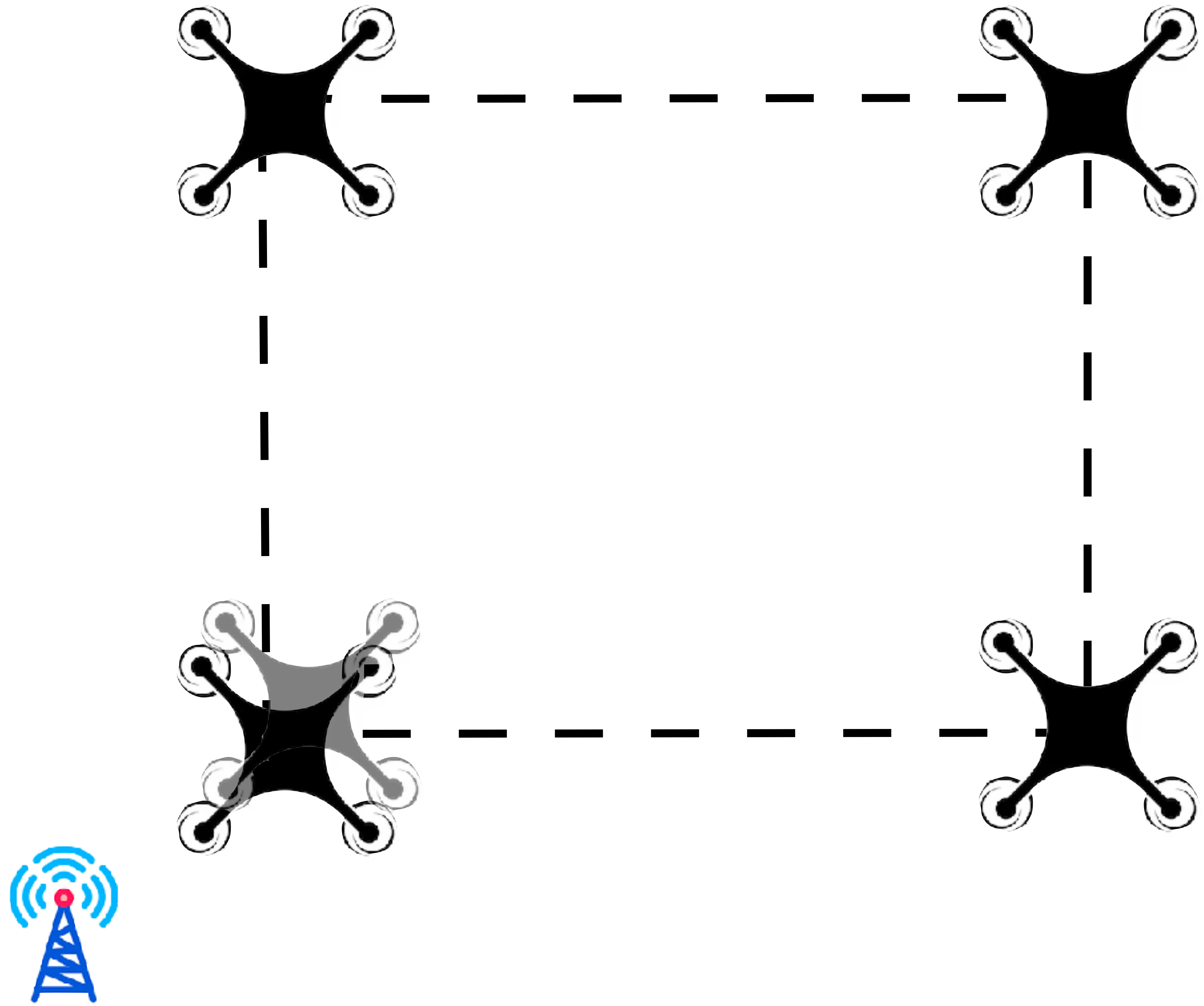
Now the estimate at T=3 is inconsistent

T=4



We correct that one as well

T=4



Correct t=2, t=1!

What is the key insight?

At every timestep, we have to solve for the entire sequence of poses and landmarks

How do we do this mathematically?

SLAM (Simultaneous Localization and Mapping)

The SLAM Problem (t=0)

Robot

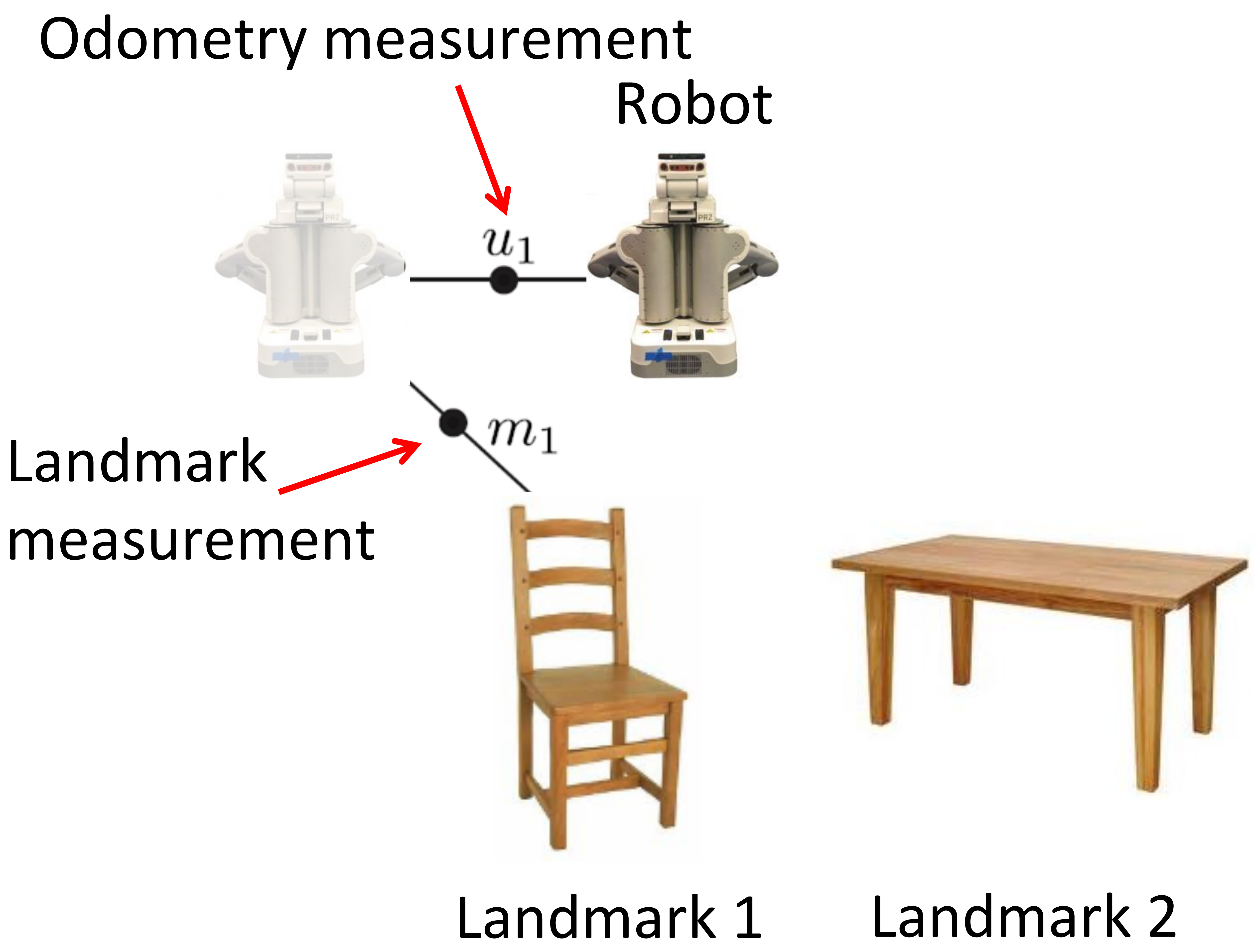
Landmark measurement

Landmark

Onboard sensors:

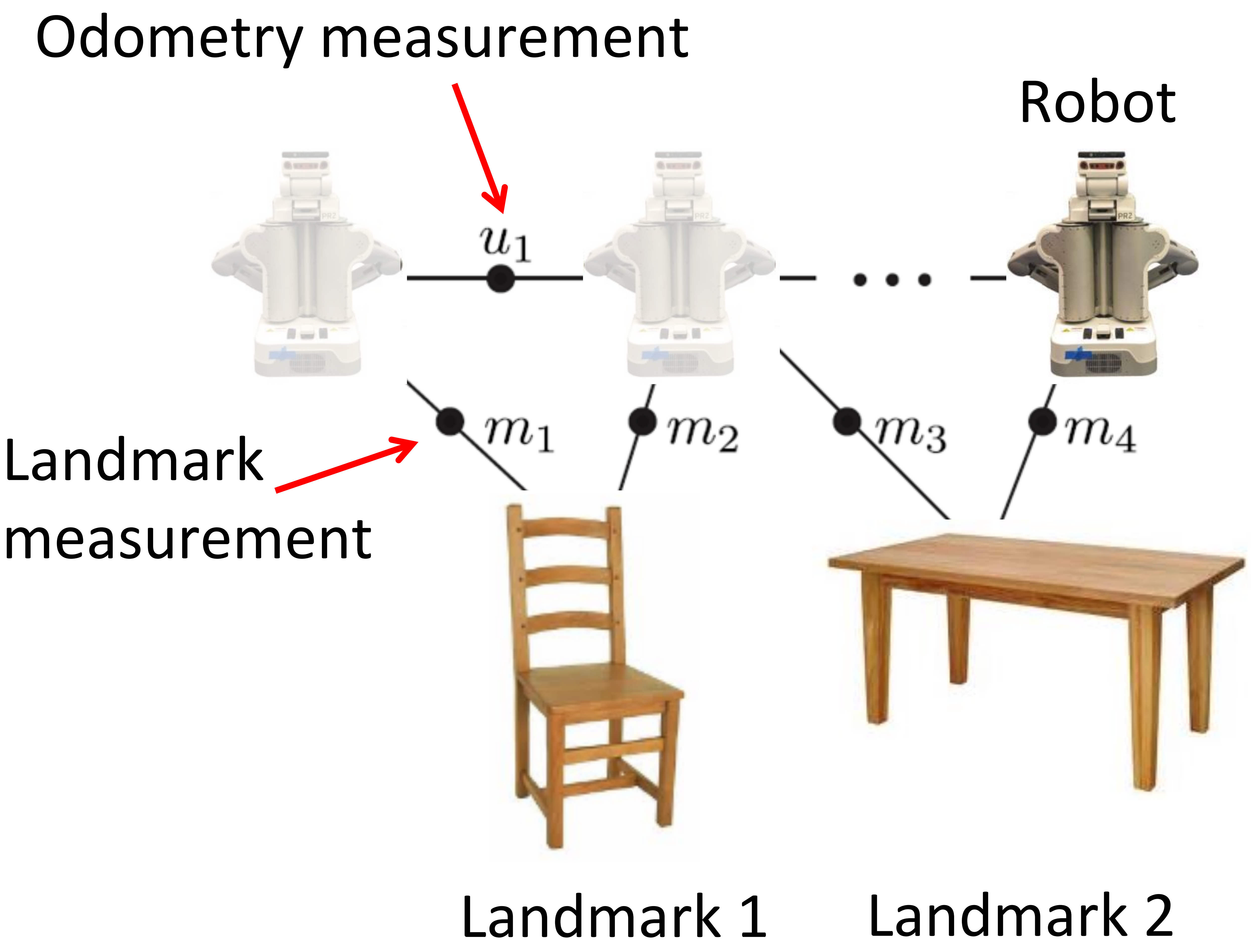
- Wheel odometry
- Inertial measurement unit (gyro, accelerometer)
- Sonar
- Laser range finder
- Camera
- RGB-D sensors

The SLAM Problem (t=1)



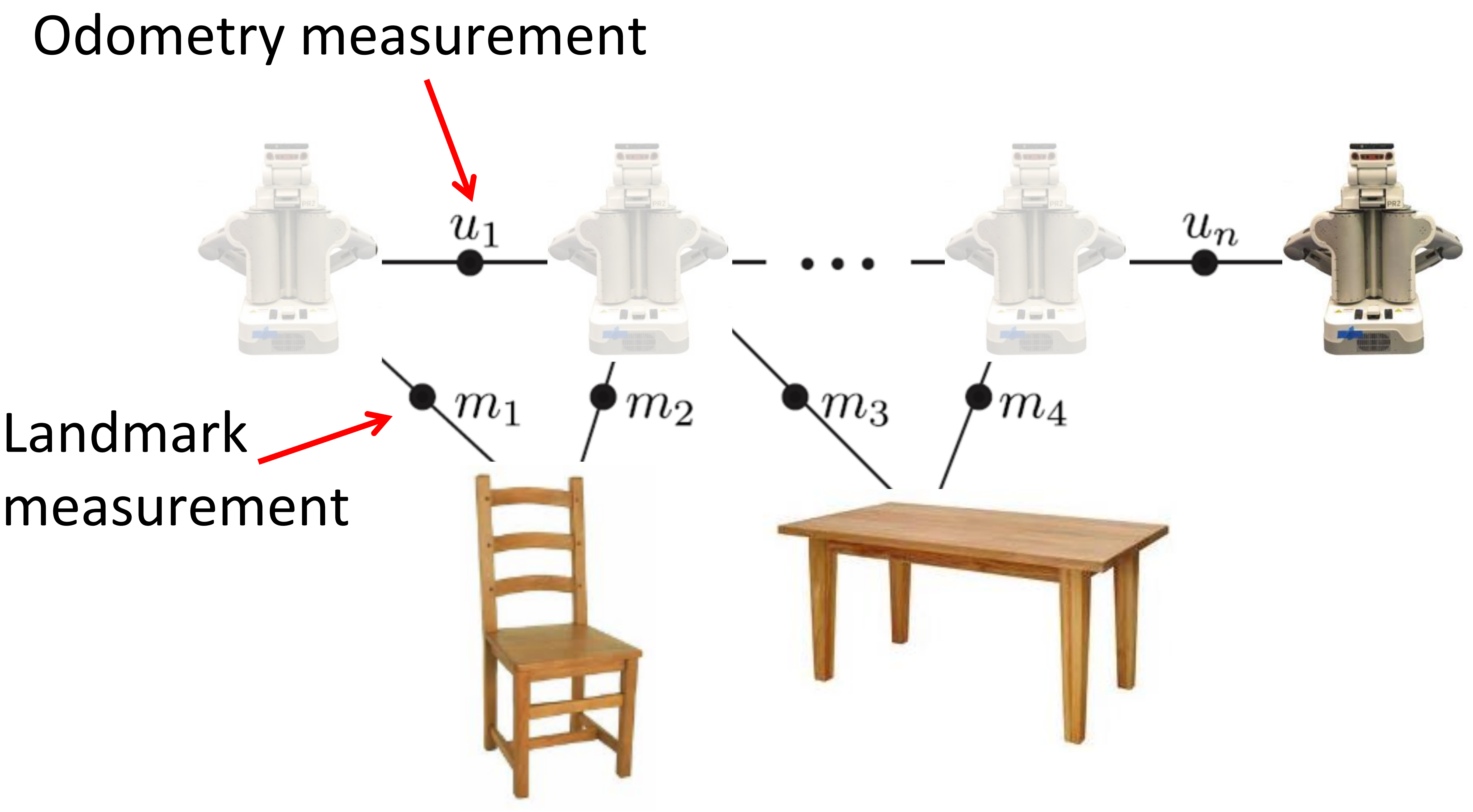
Credit CMU: Robot Localization and Mapping (16-833).

The SLAM Problem (t=n-1)



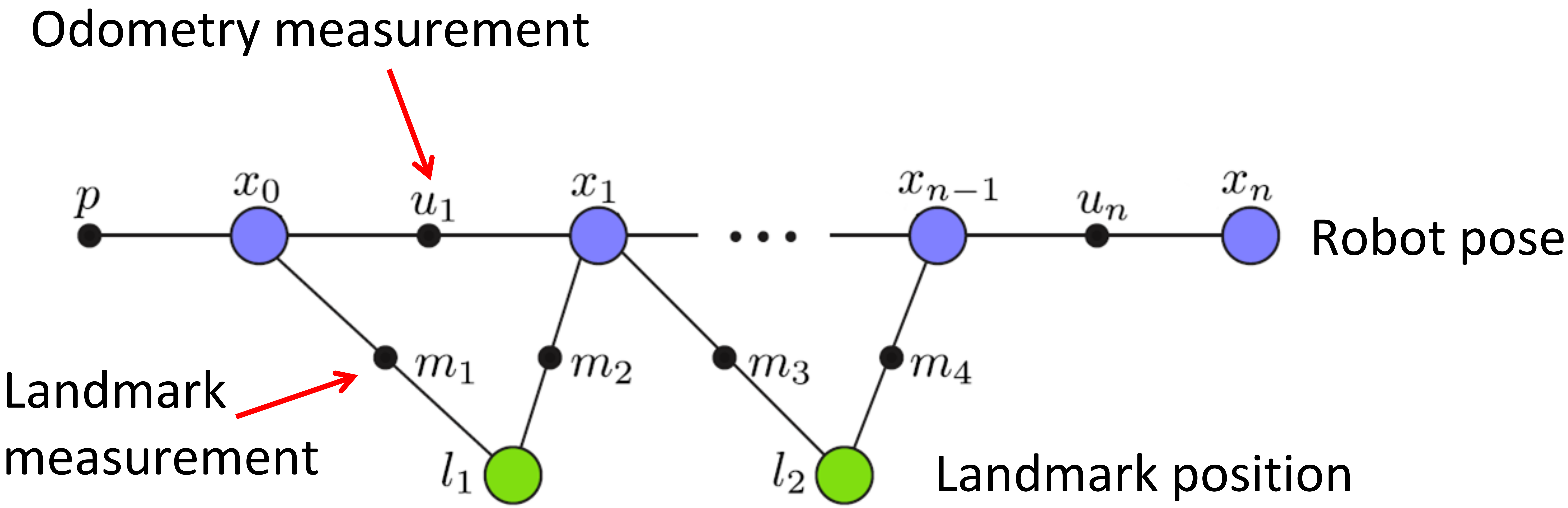
Credit CMU: Robot Localization and Mapping (16-833).

The SLAM Problem (t=n)



Credit CMU: Robot Localization and Mapping (16-833).

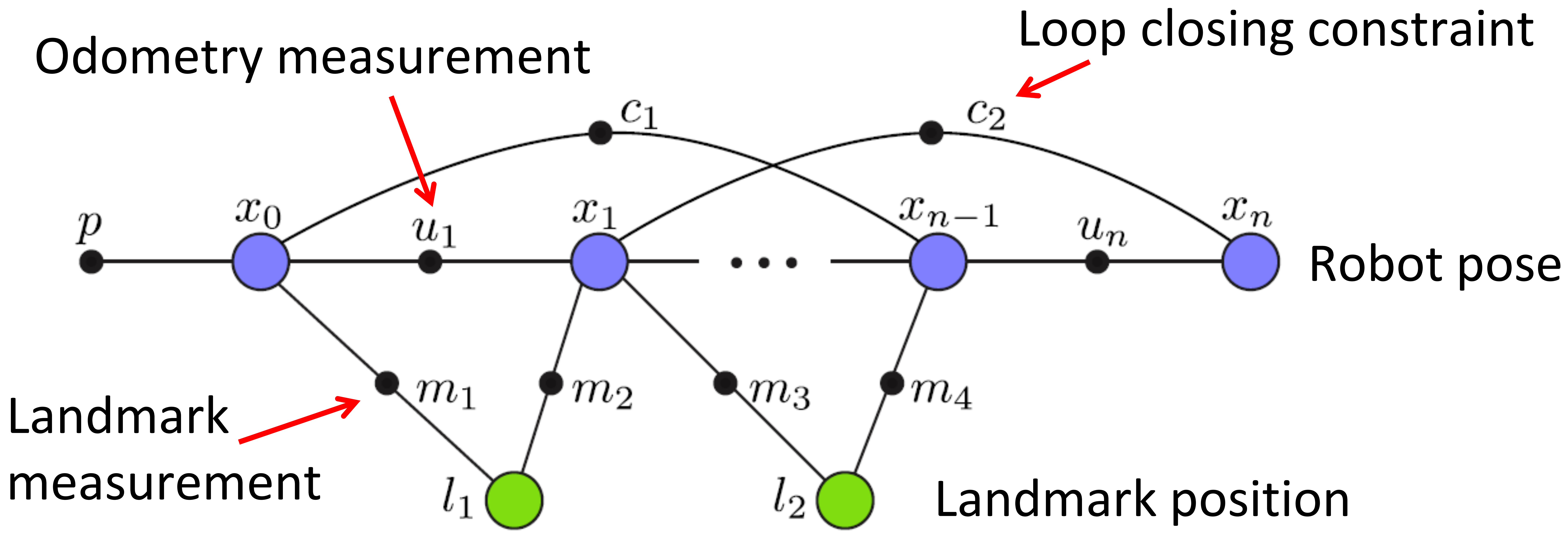
Factor Graph Representation of SLAM



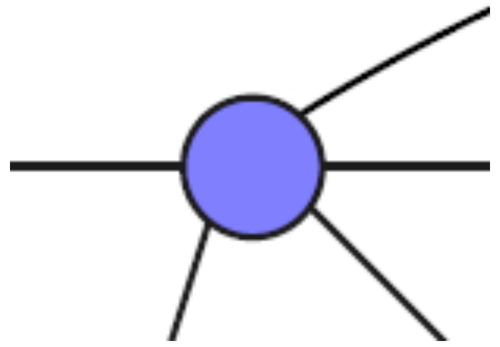
Bipartite graph with ***variable nodes*** and ***factor nodes***

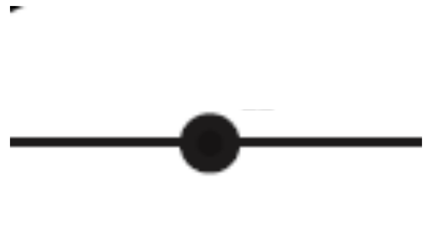
Credit CMU: Robot Localization and Mapping (16-833).

Factor Graph Representation of SLAM



Bipartite graph with ***variable nodes*** and ***factor nodes***





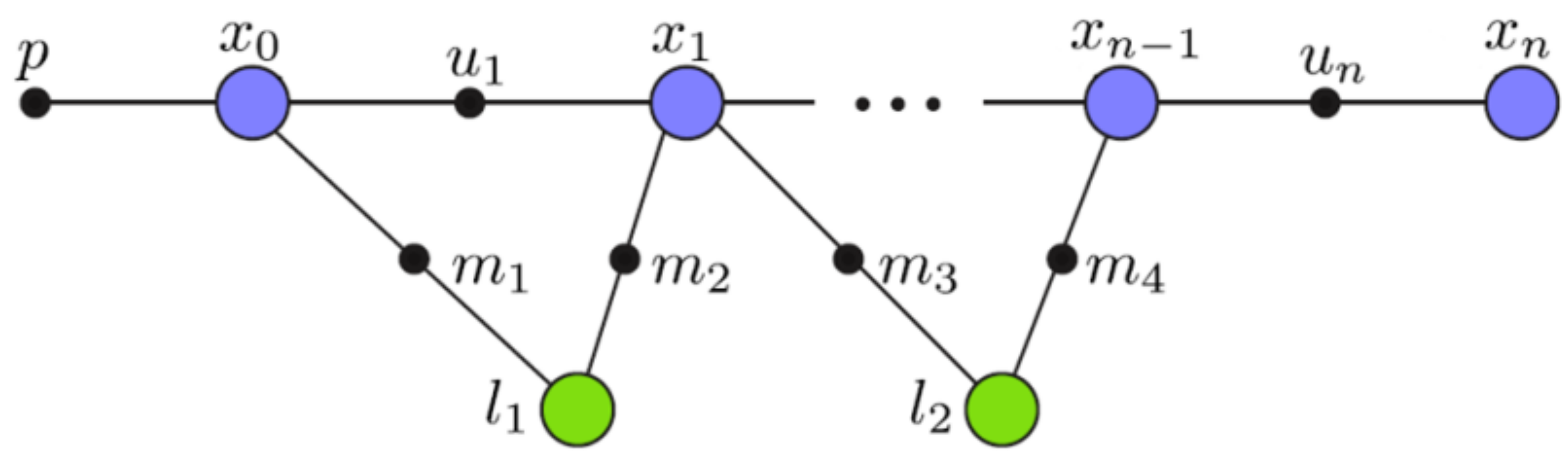
Credit CMU: Robot Localization and Mapping (16-833).

Variables and Measurements

- Variables:

$$\Theta = \{x_0, x_1 \cdots x_n, l_1, l_2\}$$

Might include other quantities such as lines, planes and calibration parameters



- Measurements:

$$Z = \{p, u_1 \cdots u_n, m_1 \cdots m_4\}$$

p is a prior to fix the gauge freedom (all other measurements are relative!)

Credit CMU: Robot Localization and Mapping (16-833).

Finding the Best Solution

Our goal is to find the Θ that maximizes $p(\Theta|Z)$

Bayes Rule

Our goal is to find the Θ that maximizes $p(\Theta|Z)$

Likelihood Prior

Posterior $$p(\Theta|Z) = \frac{p(Z|\Theta)\, p(\Theta)}{p(Z)}$$

Evidence

Note:

- While the measurements Z are given, the generative sensor models provide us with likelihood functions $\mathrm{L}(\Theta; z_i) \propto p(z_i|\Theta)$
- Evidence is independent of Θ

Credit CMU: Robot Localization and Mapping (16-833).

Maximum Likelihood and Maximum A Posteriori

- Maximum A Posteriori (MAP)

$$\Theta_{MAP} = \text{argmax}_{\Theta}\ p(Z|\Theta)\, p(\Theta)$$

- Maximum Likelihood Estimator (MLE)

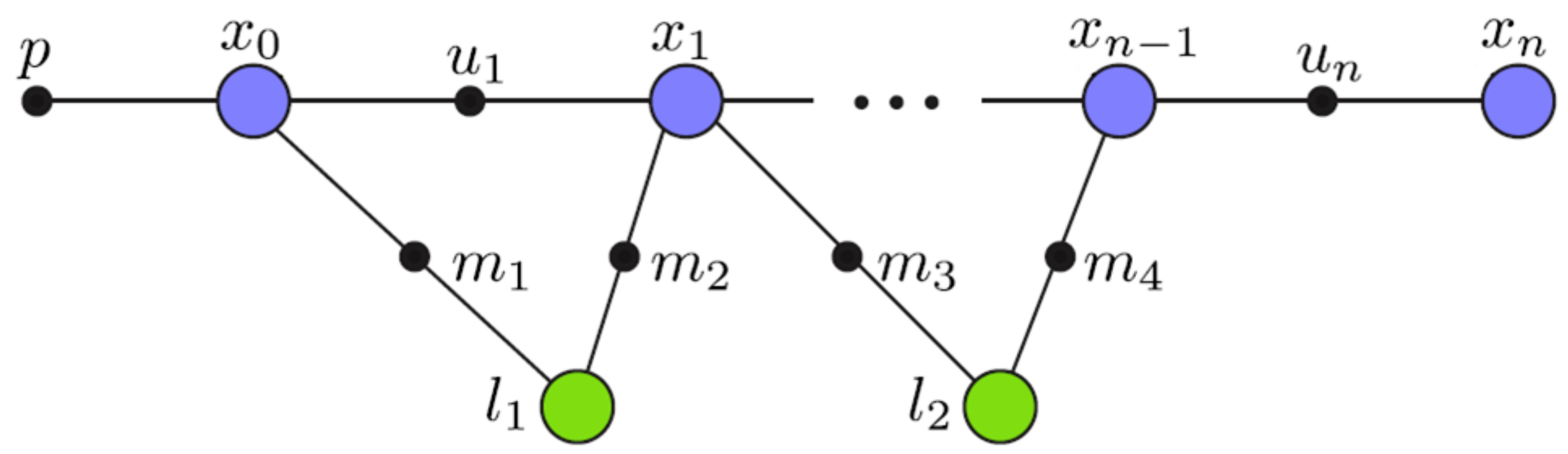
$$\Theta_{MLE} = \text{argmax}_{\Theta}\ L(\Theta; Z)$$

Credit CMU: Robot Localization and Mapping (16-833).

Factorization of Probability Density

- Conditional independence:

$$p(z_1 z_2|\Theta) = p\,(z_1|\Theta)\,p\,(z_2|\Theta)$$

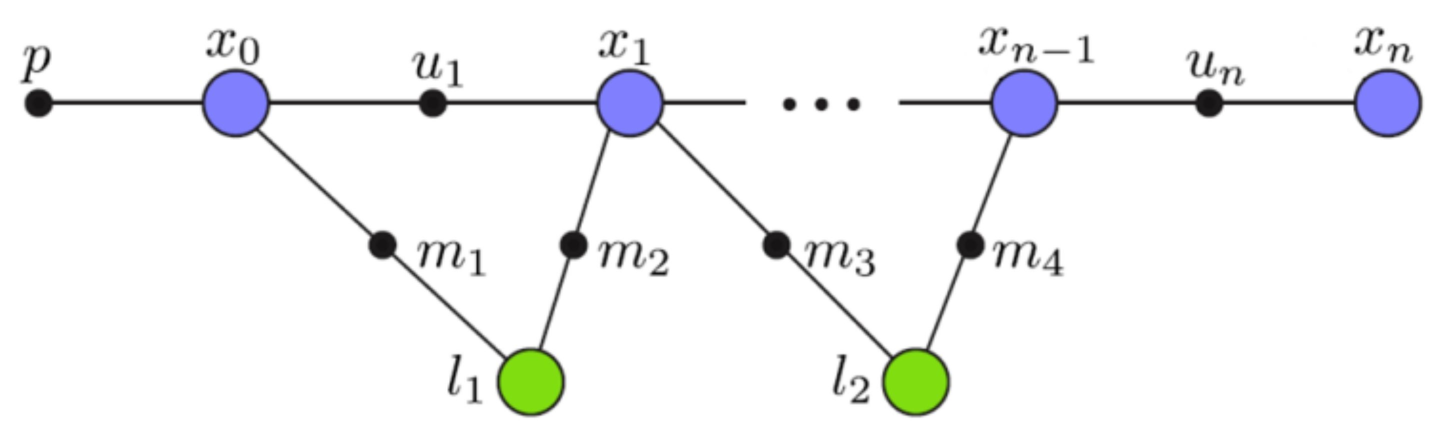


$$\Leftrightarrow \quad \text{argmax}_\Theta \prod_{z \in Z} p\,(z|\Theta)$$

$$\text{argmax}_\Theta\ p(p|\Theta)\,p\,(u_1|\Theta) \cdots p(u_n|\Theta)\,p\,(m_1|\Theta) \cdots p(m_4|\Theta)$$

Credit CMU: Robot Localization and Mapping (16-833).

SLAM as a Least-Squares Problem



$$\text{argmax}_{\Theta} \prod_{z \in Z} p\left(z|\Theta\right)$$

Gaussian noise

$$\text{argmin}_{\Theta} \sum_{i} \|h_i(\Theta) - z_i\|^2$$

$$\text{argmin}_{\theta} \|A\theta - b\|^2$$

Normal equations:

$$A^T A\theta = A^T b$$

Credit CMU: Robot Localization and Mapping (16-833).