InforMARL: Scalable Multi-Agent Reinforcement Learning through Intelligent Information Aggregation

Siddharth Nayak

sidnayak@mit.edu

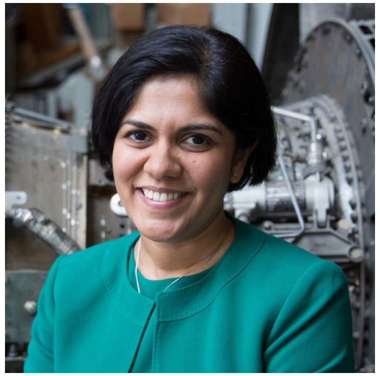
Joint work with:

Kenneth Choi

Wenqi Ding

Sydney Dolan

Karthik Gopalakrishnan



Hamsa Balakrishnan

This research was sponsored in part by the NASA University Leadership initiative (grant #80NSSC20M0163) and the U.S. AFRL/U.S. Air Force Artificial Intelligence Accelerator (Cooperative Agreement Number FA8750-19-2-1000). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of NASA, the U.S. Air Force or the U.S. Government.

DINaMo

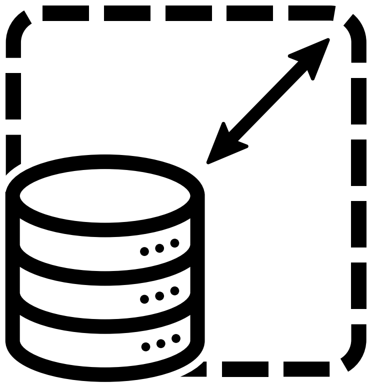
Background and Motivation

Background and Motivation

Key Features Expected from MARL Algorithms

Decentralized Execution	Scalability	Efficiency in training sample complexity

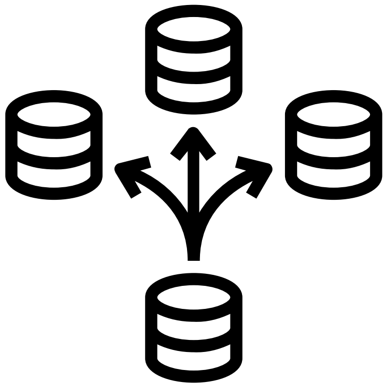
Motivation: Scalability



Scalability

- MARL algorithms should work in scenarios with a large number of agents
- Preferably be agnostic to number of agents in the environment

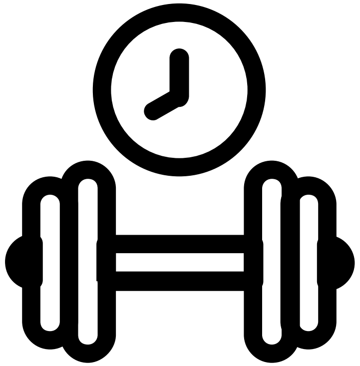
Motivation: Decentralized Execution



Decentralized Execution

- Each agent should be able to take decisions for itself
- Should not depend on a centralized controller

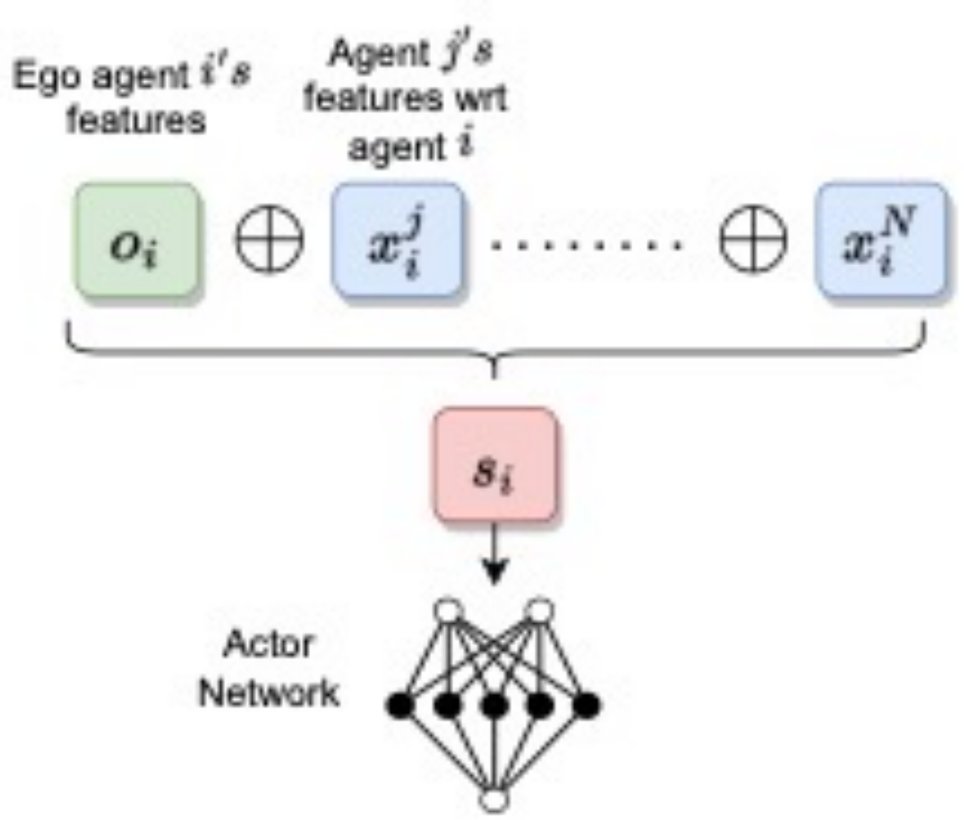
Motivation: Training Time



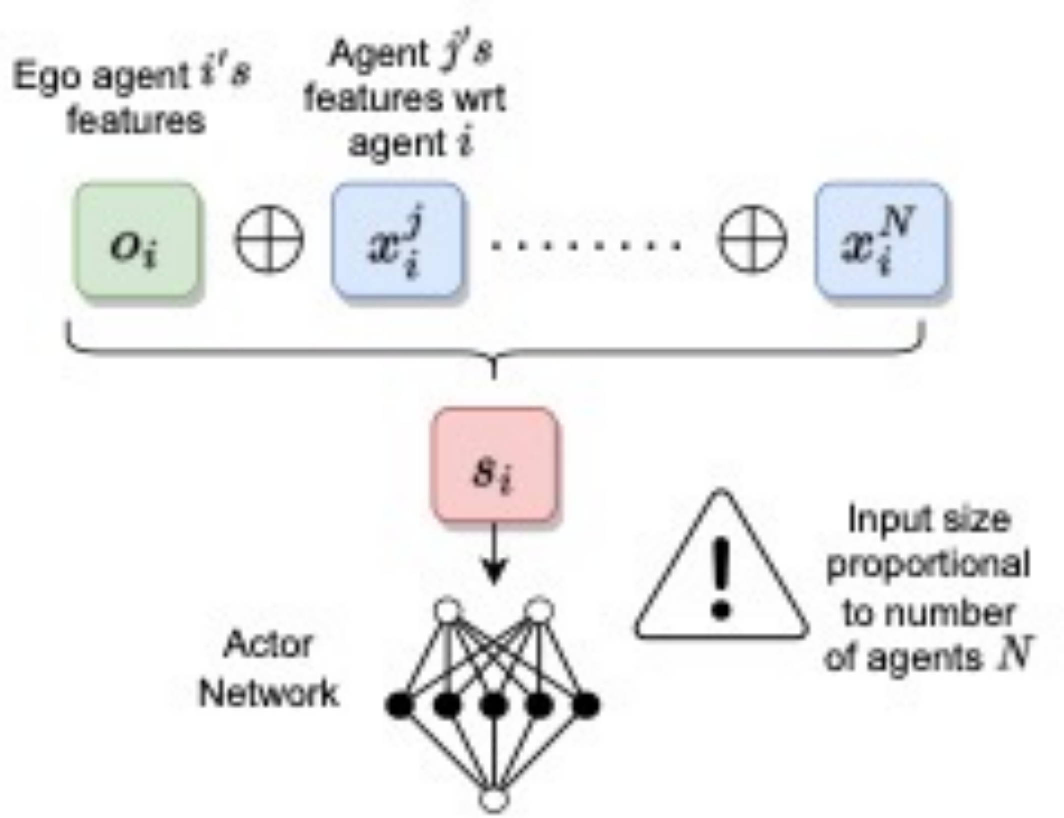
Efficiency in training sample complexity

- MARL algorithms need a lot of training samples because of various issues like non-stationarity, partial observability, etc.
- The amount of training required increases as the environment complexity increases

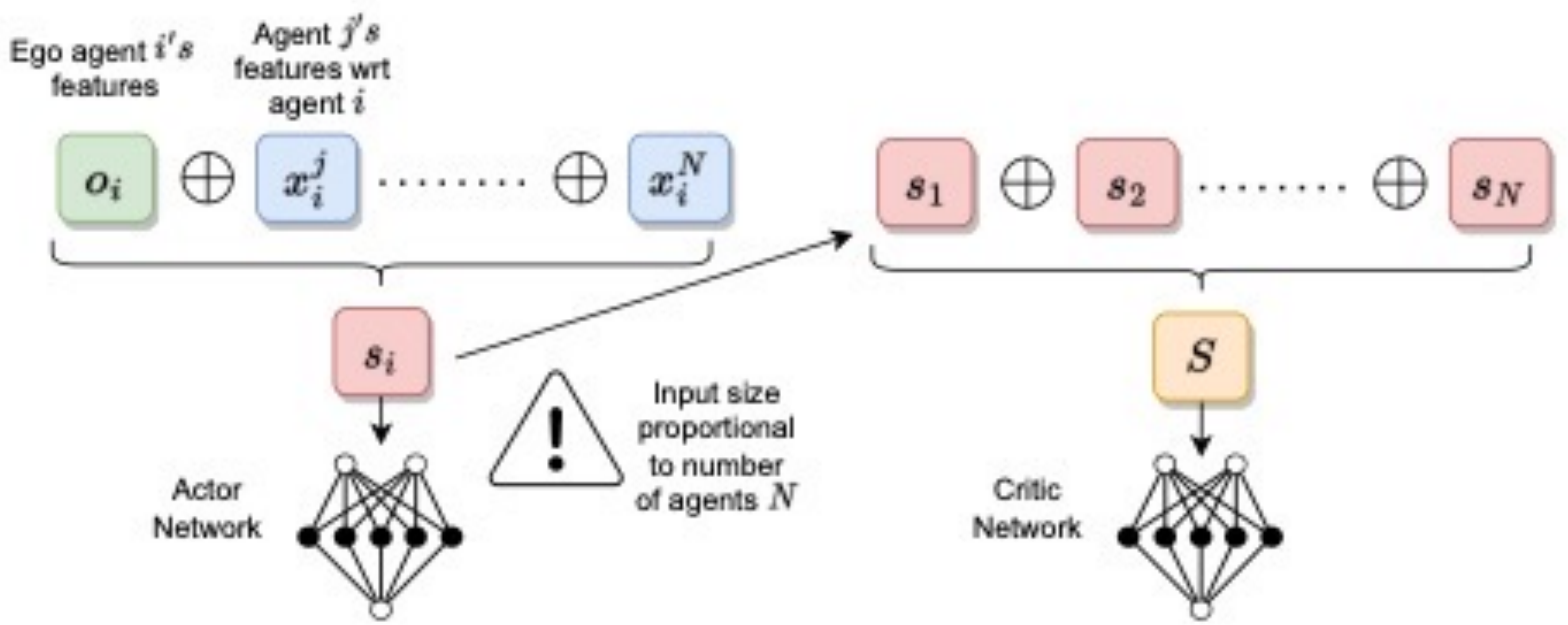
Motivation: Prior Approaches



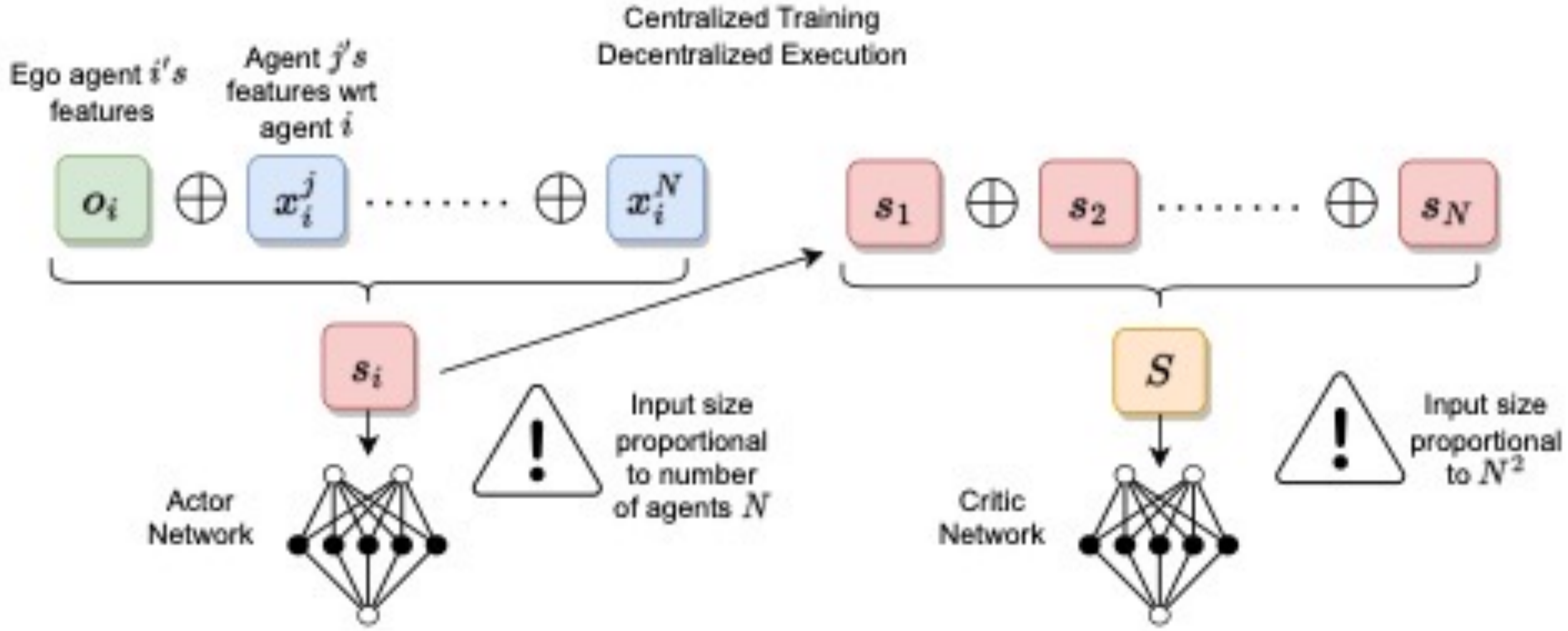
Motivation: Prior Approaches



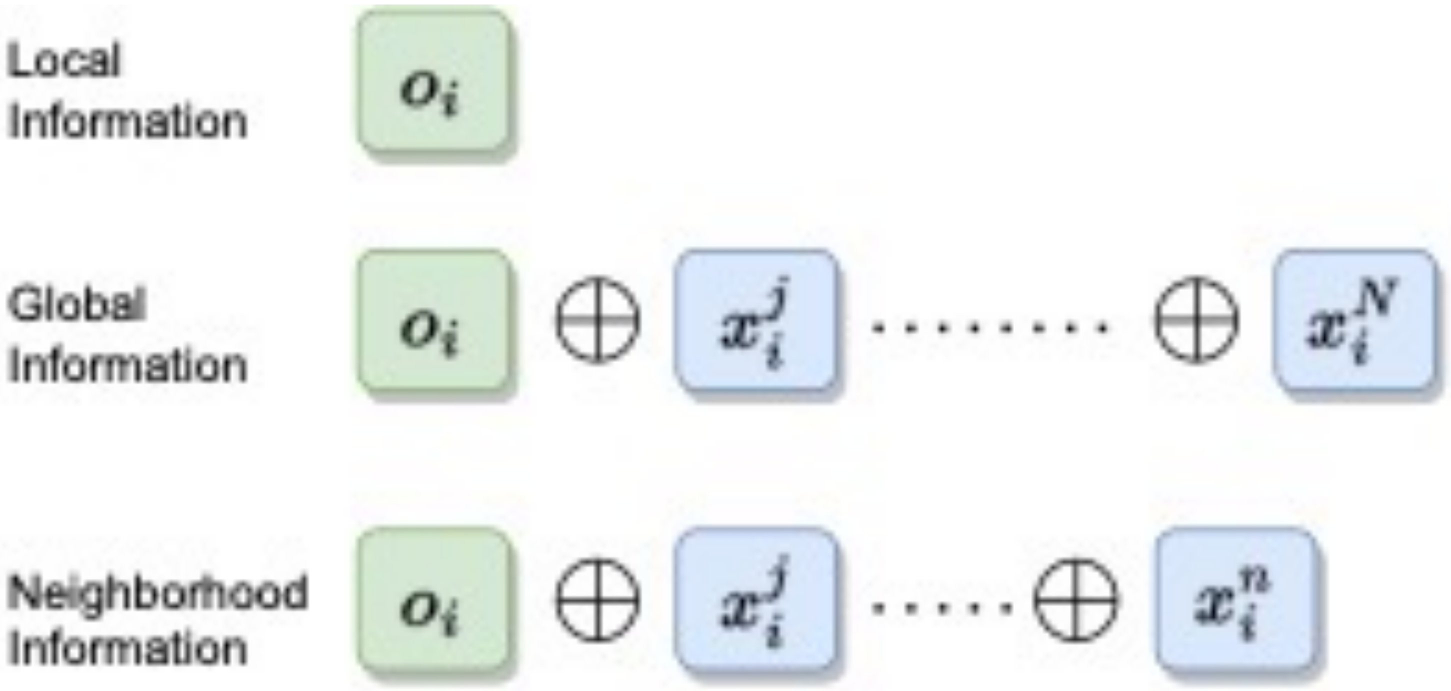
Motivation: Prior Approaches



Motivation: Prior Approaches

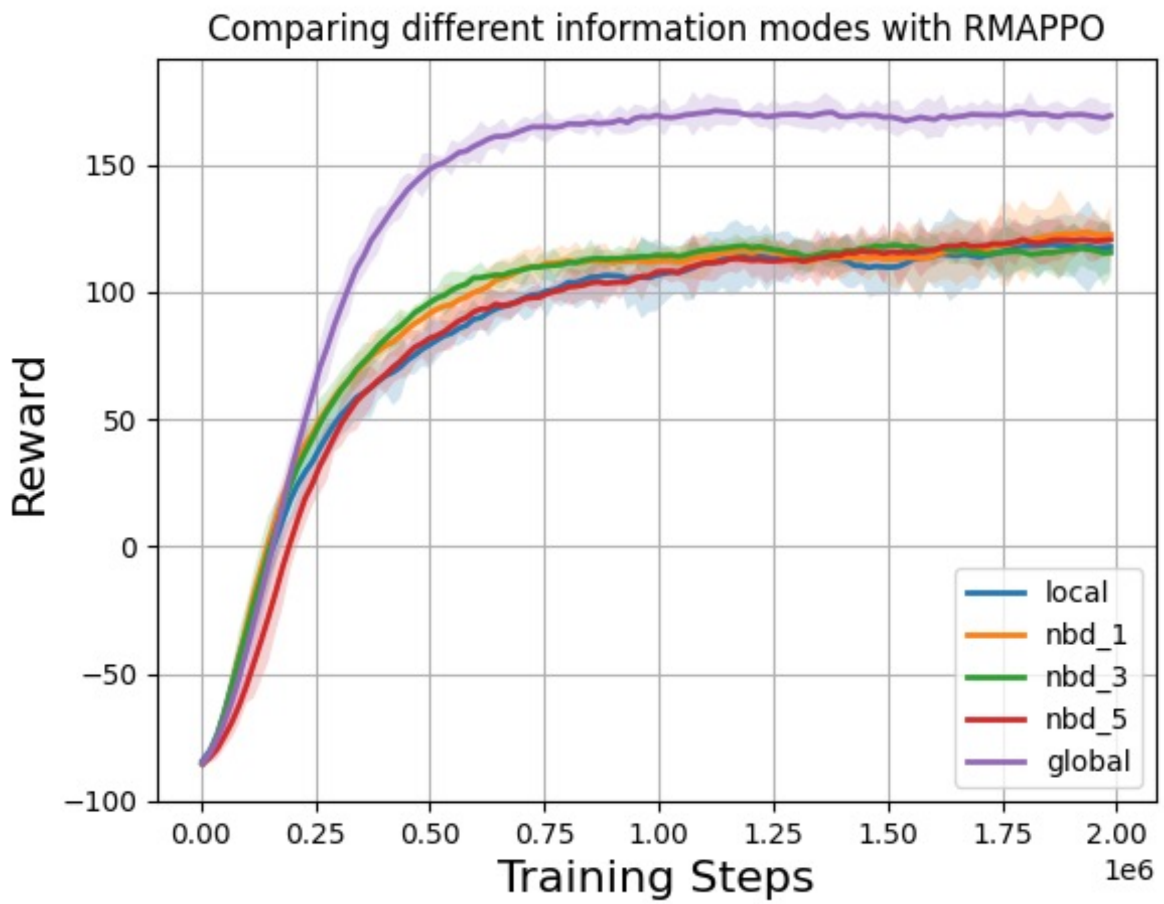


Motivating Experiment



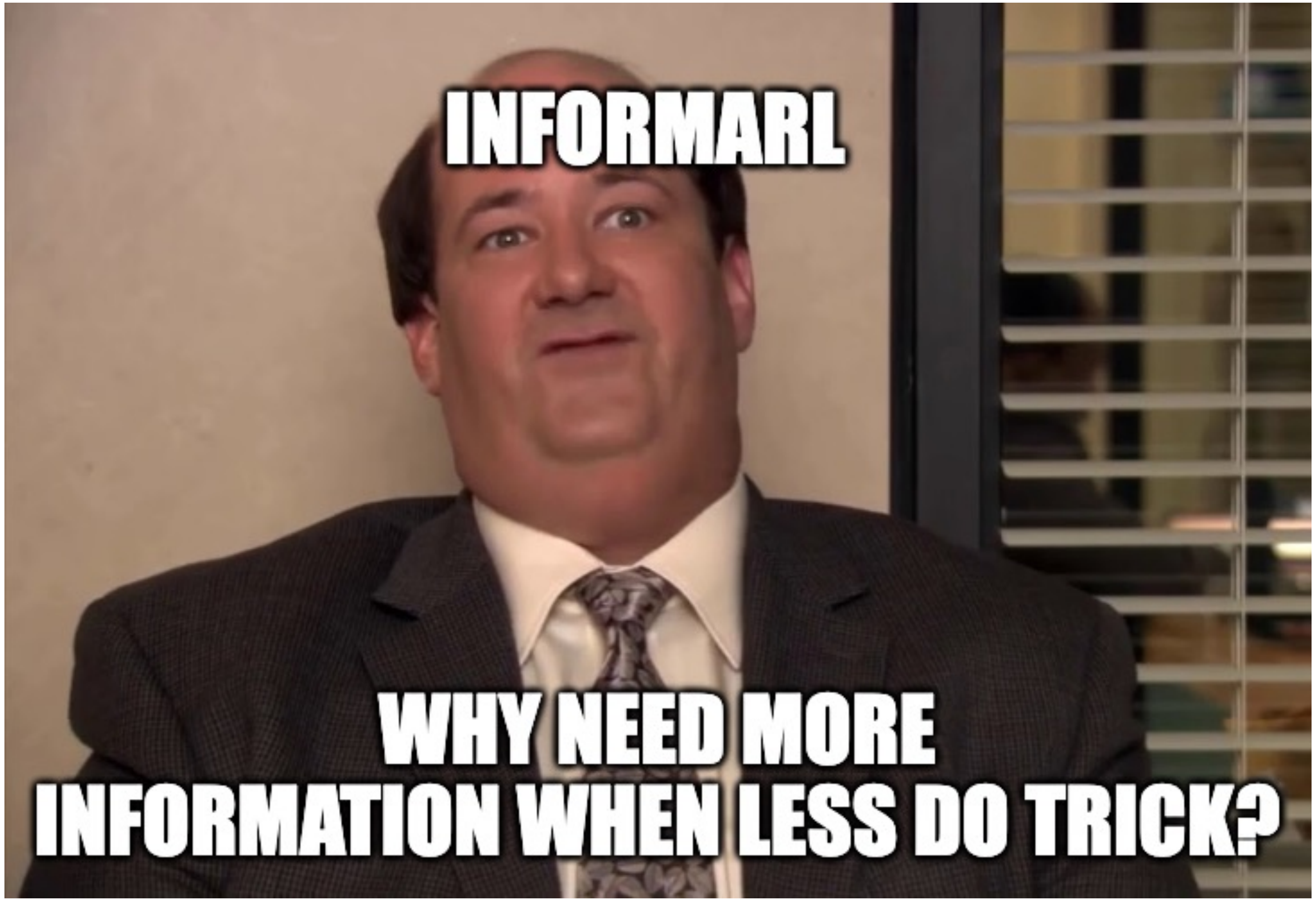
$n < N$

Motivating Experiment



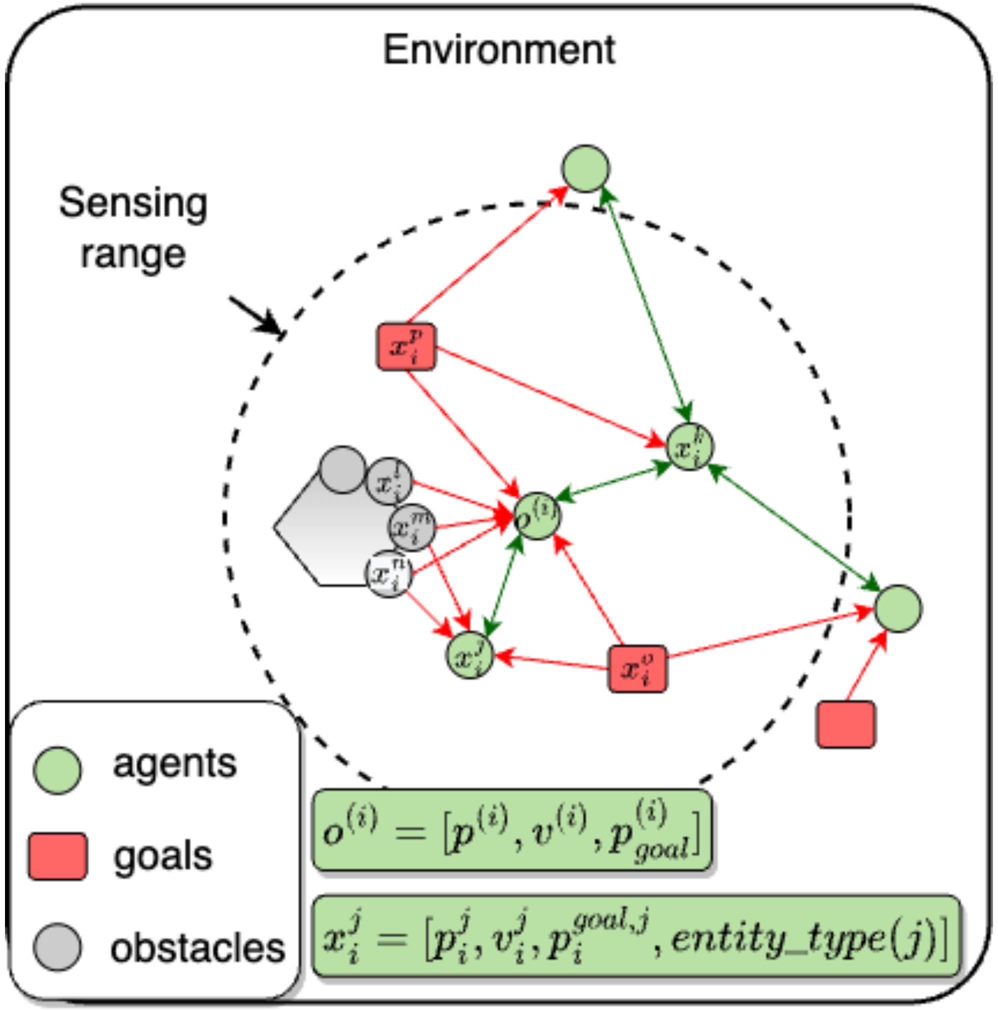
- In practice, we just have local information about the neighborhood
- And naïve concatenation of neighborhood information doesn't work

Motivating Experiment

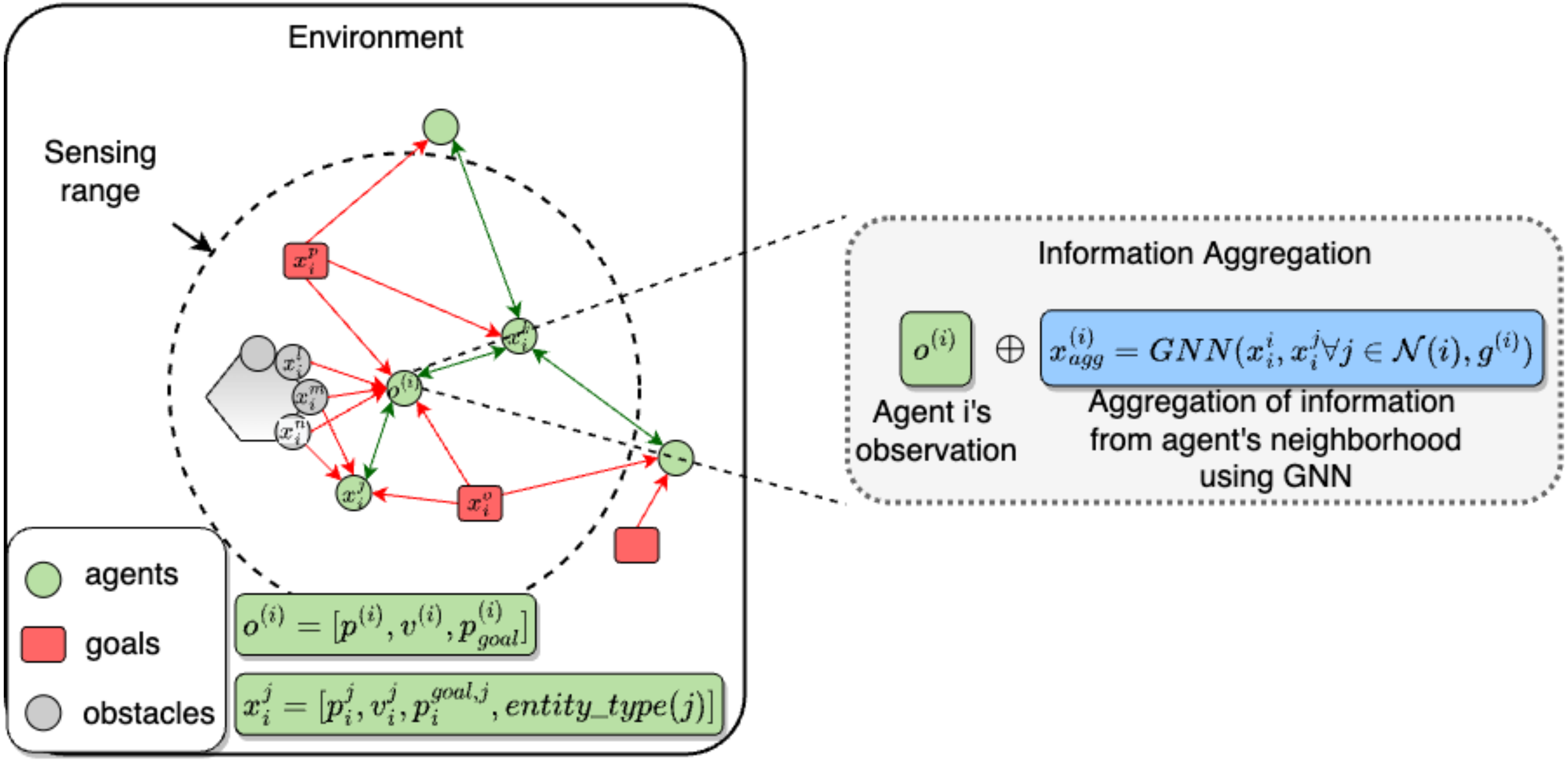


DINaMo

Method

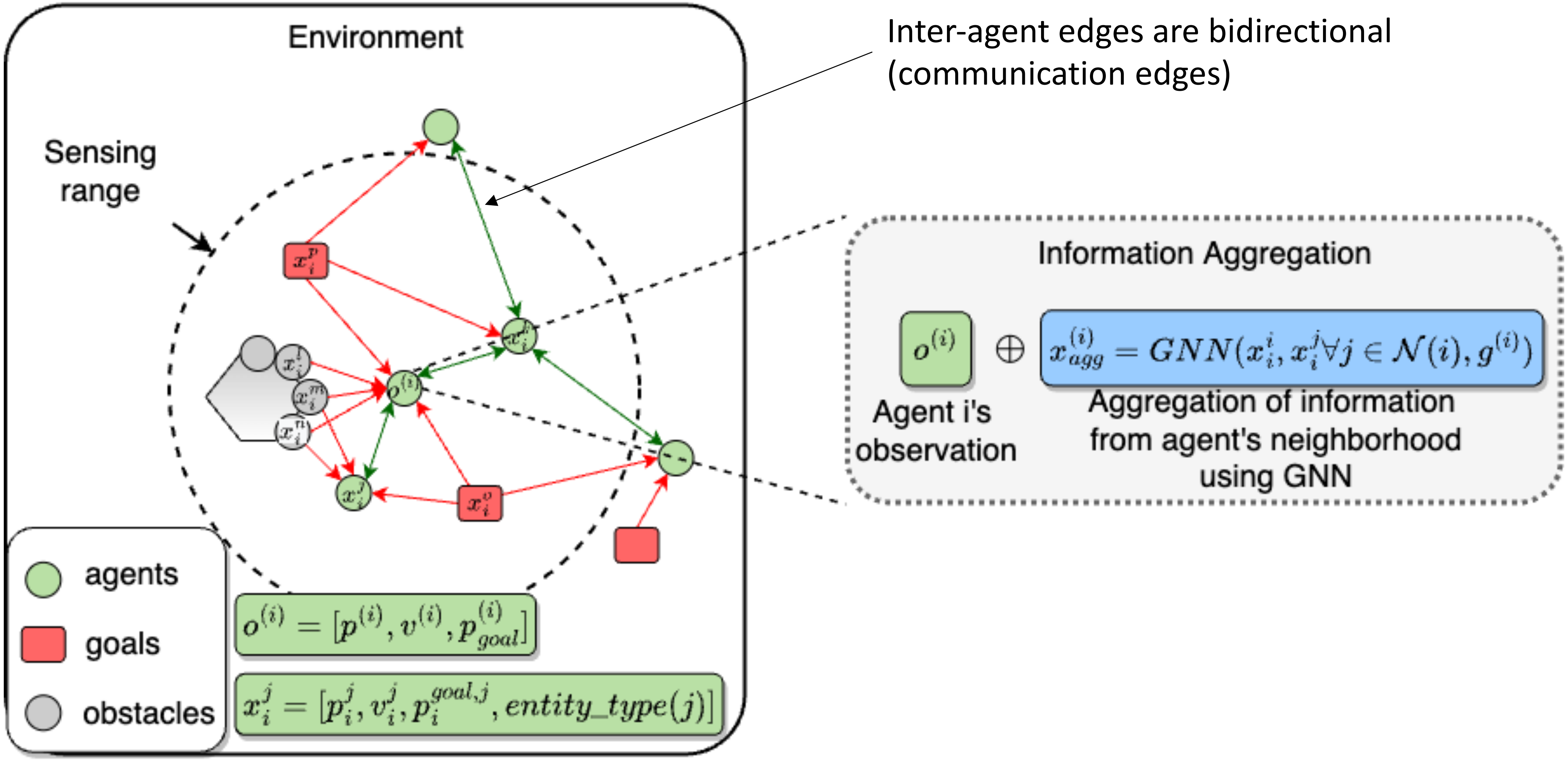

Environment
Sensing range
x_i^p
x_i^k
x_i^l
x_i^m
x_i^n
$o^{(i)}$
x_i^j
x_i^o
agents
goals
obstacles
$o^{(i)} = [p^{(i)}, v^{(i)}, p^{(i)}_{goal}]$
$x_i^j = [p_i^j, v_i^j, p_i^{goal,j}, entity\_type(j)]$

Method



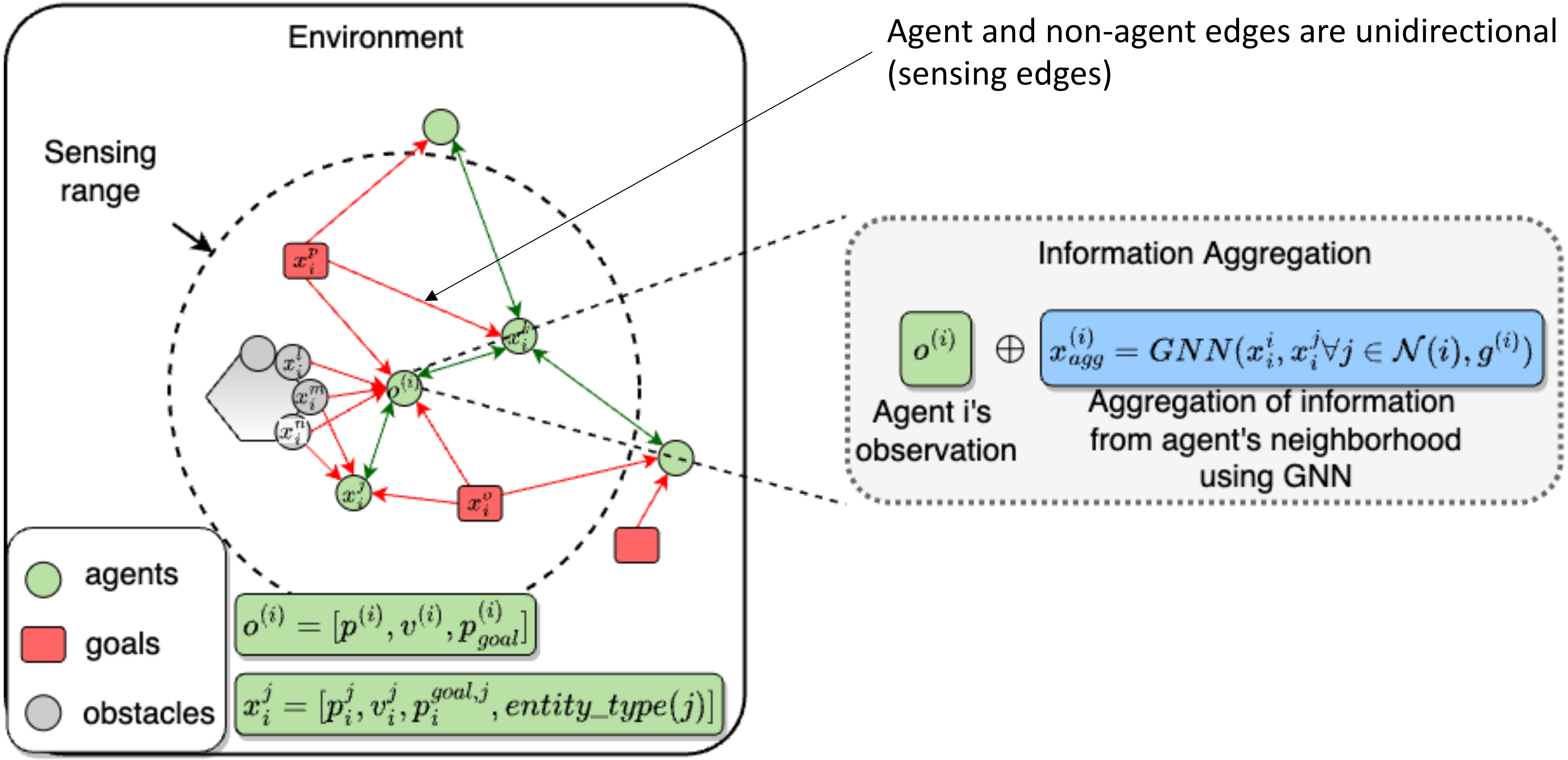
Environment
Sensing range
agents
goals
obstacles
$o^{(i)} = [p^{(i)}, v^{(i)}, p^{(i)}_{goal}]$
$x^j_i = [p^j_i, v^j_i, p^{goal,j}_i, entity\_type(j)]$
Information Aggregation
$o^{(i)}$
$\oplus$
$x^{(i)}_{agg} = GNN(x^i_i, x^j_i \forall j \in \mathcal{N}(i), g^{(i)})$
Agent i's observation
Aggregation of information from agent's neighborhood using GNN

Method



Inter-agent edges are bidirectional (communication edges)

Method

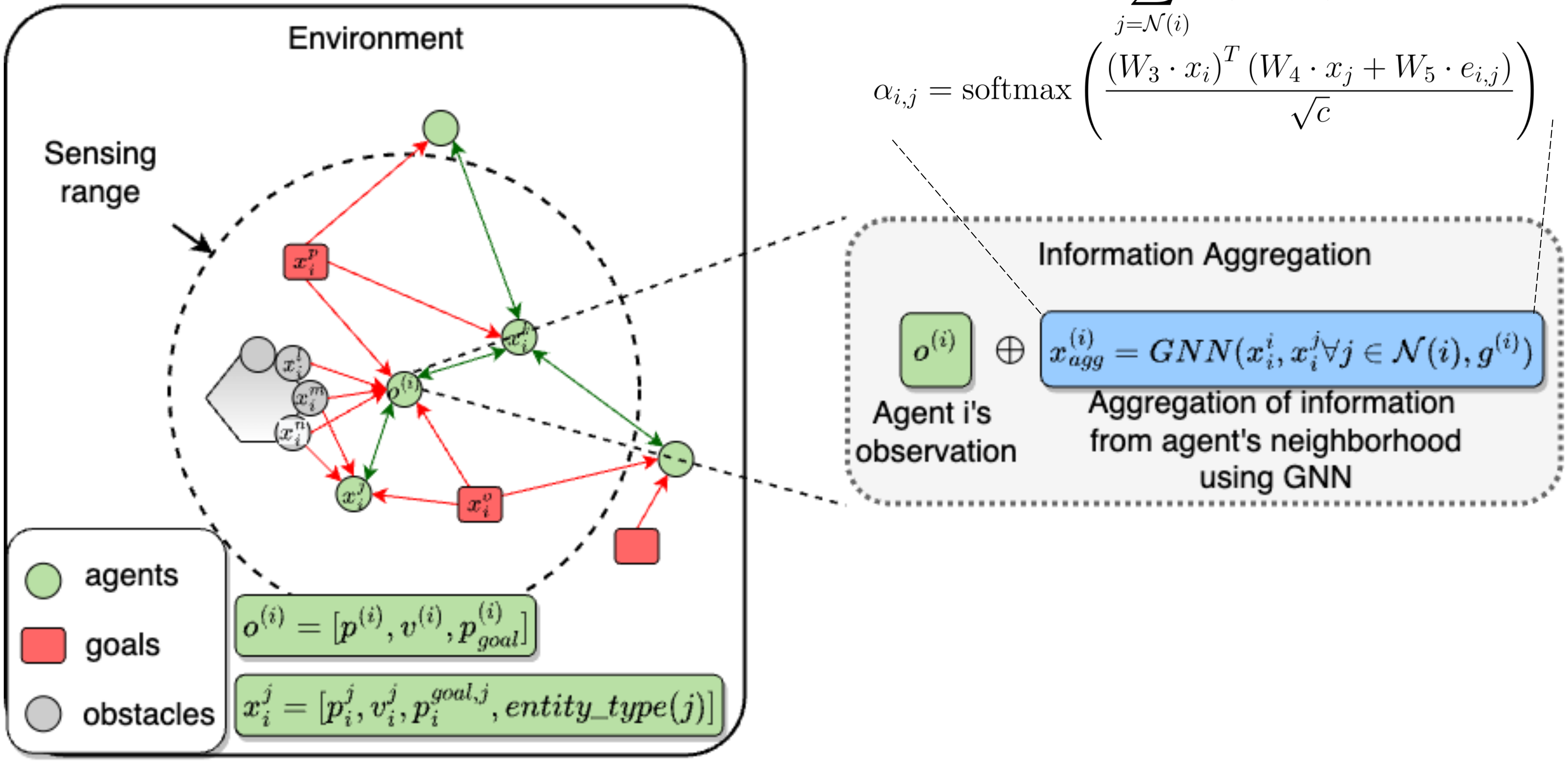


Agent and non-agent edges are unidirectional (sensing edges)

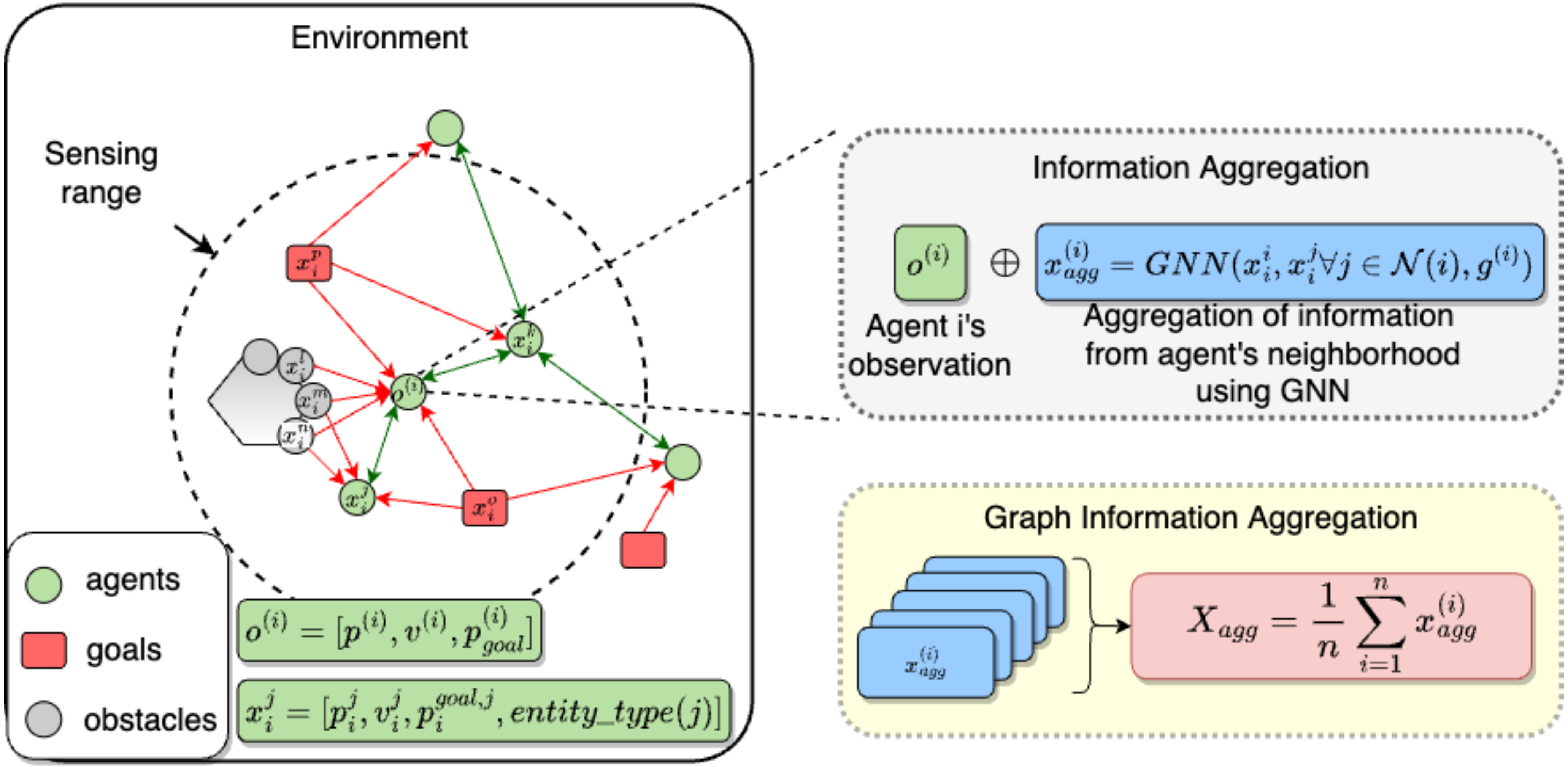
Method

$$x_i' = W_1 \cdot x_i + \sum_{j=\mathcal{N}(i)} \alpha_{i,j} W_2 \cdot x_j$$

$$\alpha_{i,j} = \text{softmax}\left(\frac{(W_3 \cdot x_i)^T (W_4 \cdot x_j + W_5 \cdot e_{i,j})}{\sqrt{c}}\right)$$

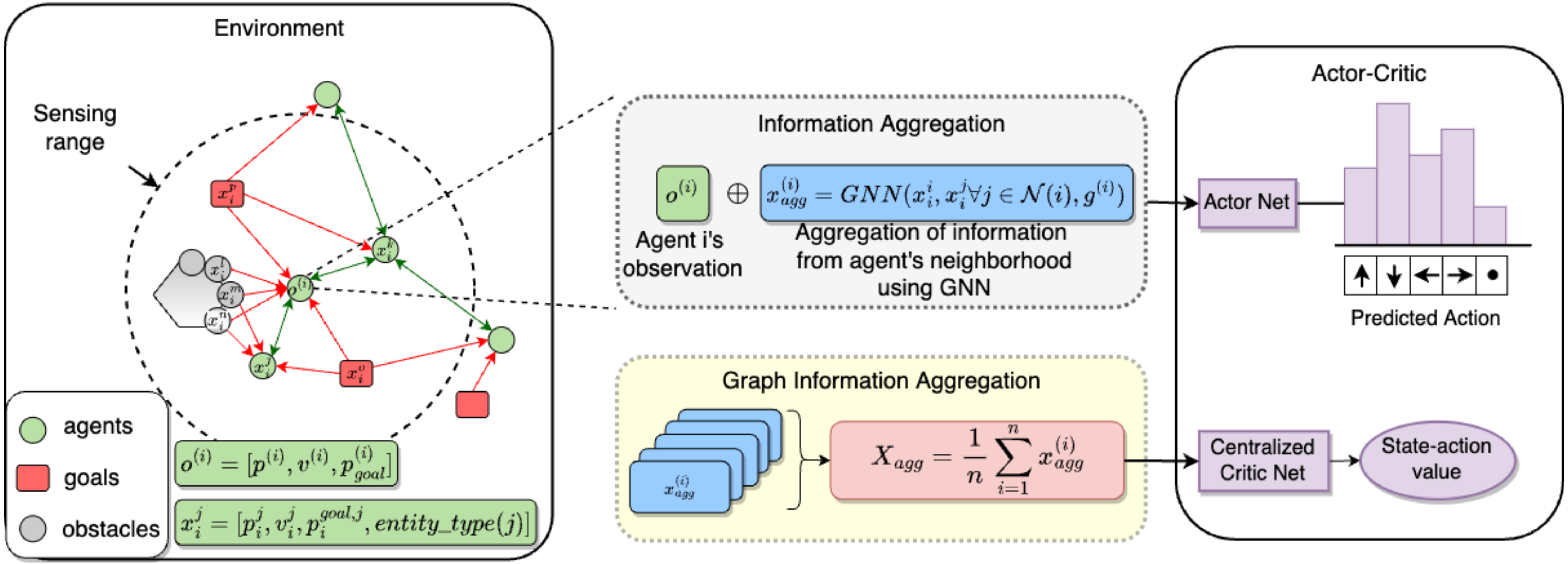


Method



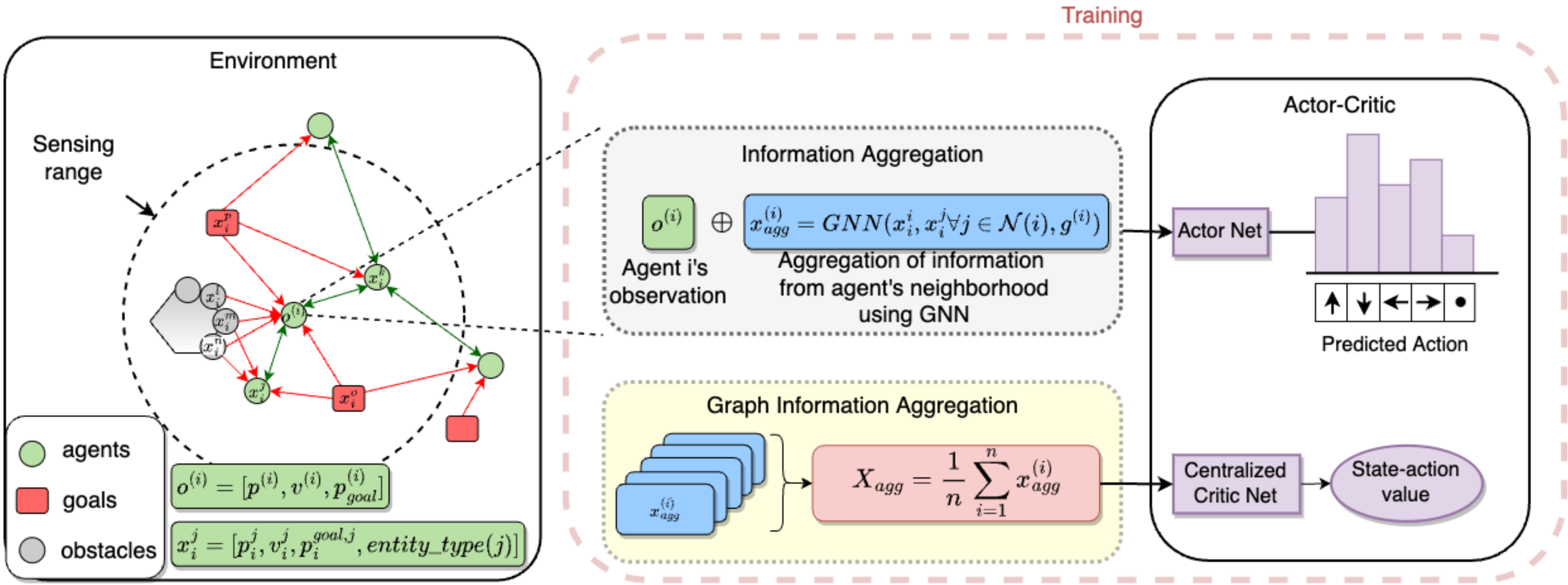
Environment
Sensing range
x_i^p
x_i^k
x_i^l
x_i^m
x_i^n
$o^{(i)}$
x_i^j
x_i^o
agents
goals
obstacles
$o^{(i)} = [p^{(i)}, v^{(i)}, p^{(i)}_{goal}]$
$x_i^j = [p_i^j, v_i^j, p_i^{goal,j}, entity\_type(j)]$
Information Aggregation
$o^{(i)}$
$\oplus$
$x^{(i)}_{agg} = GNN(x_i^i, x_i^j \forall j \in \mathcal{N}(i), g^{(i)})$
Agent i's observation
Aggregation of information from agent's neighborhood using GNN
Graph Information Aggregation
$x^{(i)}_{agg}$
$X_{agg} = \frac{1}{n}\sum_{i=1}^{n} x^{(i)}_{agg}$

Method



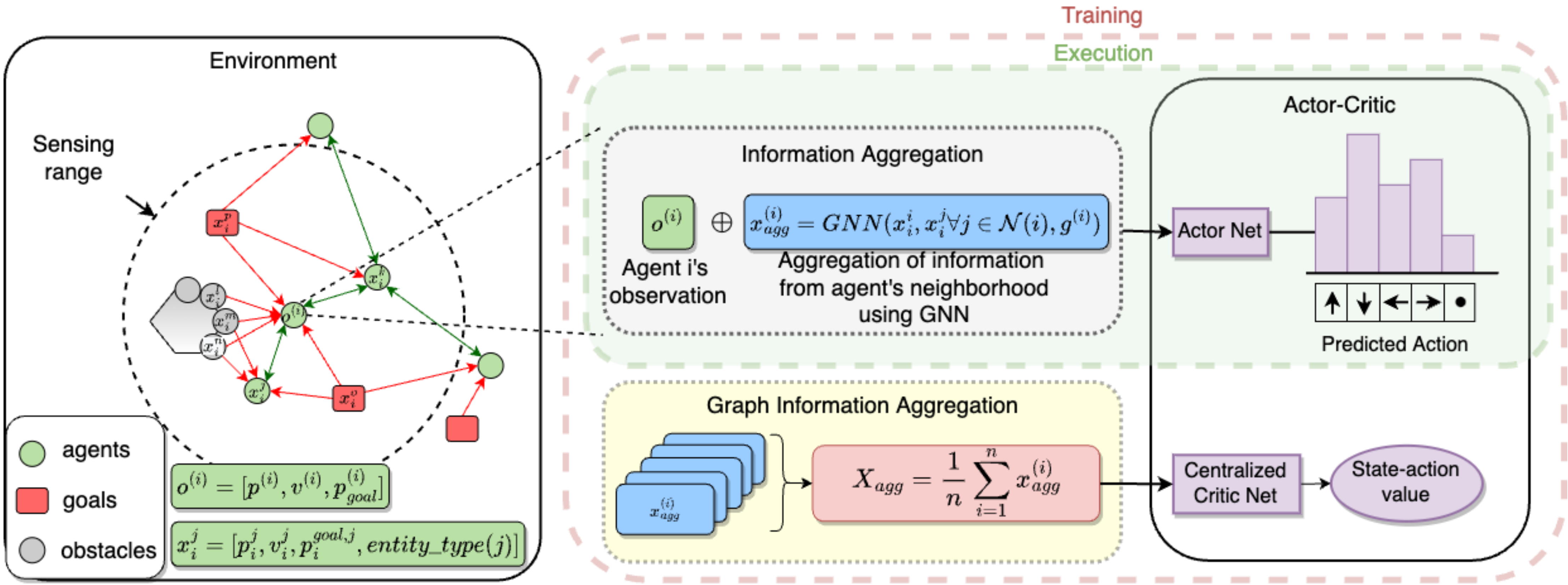
Environment
Sensing range
agents
goals
obstacles
$o^{(i)} = [p^{(i)}, v^{(i)}, p^{(i)}_{goal}]$
$x^j_i = [p^j_i, v^j_i, p^{goal,j}_i, entity\_type(j)]$
Information Aggregation
$o^{(i)}$
$x^{(i)}_{agg} = GNN(x^i_i, x^j_i \forall j \in \mathcal{N}(i), g^{(i)})$
Agent i's observation
Aggregation of information from agent's neighborhood using GNN
Graph Information Aggregation
$X_{agg} = \frac{1}{n} \sum_{i=1}^{n} x^{(i)}_{agg}$
Actor-Critic
Actor Net
Predicted Action
Centralized Critic Net
State-action value

Method



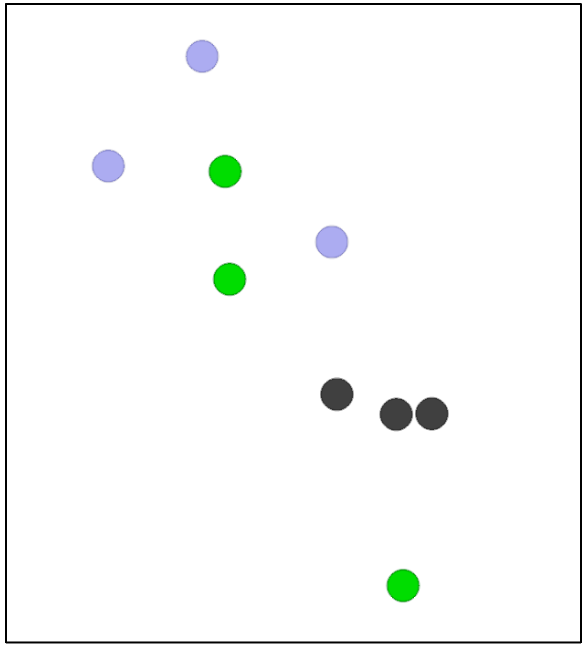
Training
Environment
Sensing range
agents
goals
obstacles
$o^{(i)} = [p^{(i)}, v^{(i)}, p^{(i)}_{goal}]$
$x_i^j = [p_i^j, v_i^j, p_i^{goal,j}, entity\_type(j)]$
Information Aggregation
$o^{(i)}$
Agent i's observation
$x^{(i)}_{agg} = GNN(x_i^i, x_i^j \forall j \in \mathcal{N}(i), g^{(i)})$
Aggregation of information from agent's neighborhood using GNN
Graph Information Aggregation
$x^{(i)}_{agg}$
$X_{agg} = \frac{1}{n}\sum_{i=1}^{n} x^{(i)}_{agg}$
Actor-Critic
Actor Net
Predicted Action
Centralized Critic Net
State-action value

Method

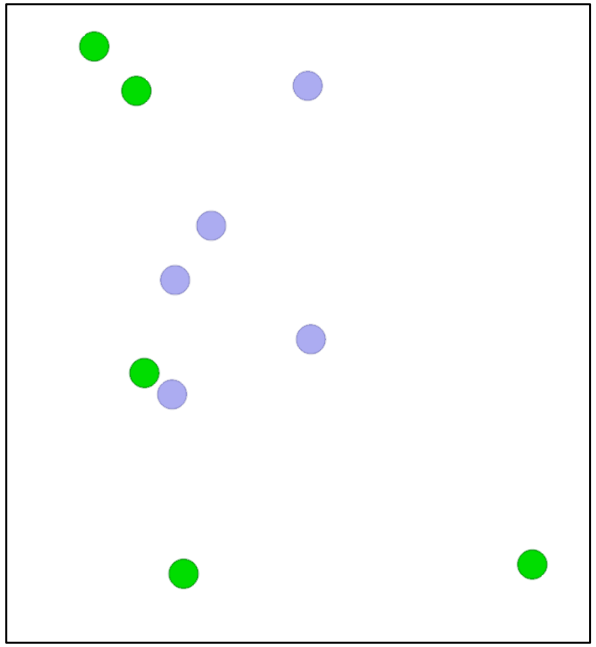


Environment
Sensing range
agents
goals
obstacles
$o^{(i)} = [p^{(i)}, v^{(i)}, p^{(i)}_{goal}]$
$x_i^j = [p_i^j, v_i^j, p_i^{goal,j}, entity\_type(j)]$
Training
Execution
Information Aggregation
$o^{(i)}$
Agent i's observation
$x^{(i)}_{agg} = GNN(x_i^i, x_i^j \forall j \in \mathcal{N}(i), g^{(i)})$
Aggregation of information from agent's neighborhood using GNN
Graph Information Aggregation
$X_{agg} = \frac{1}{n}\sum_{i=1}^{n} x^{(i)}_{agg}$
Actor-Critic
Actor Net
Predicted Action
Centralized Critic Net
State-action value

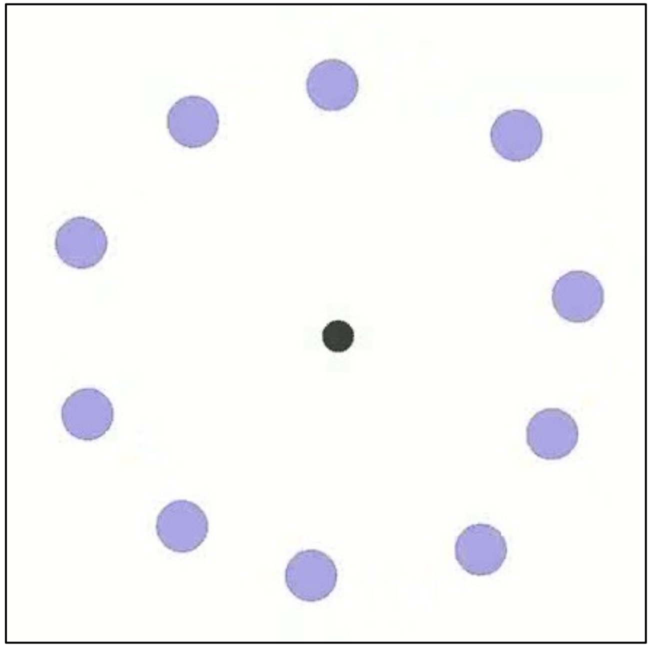
Experiments: Environments



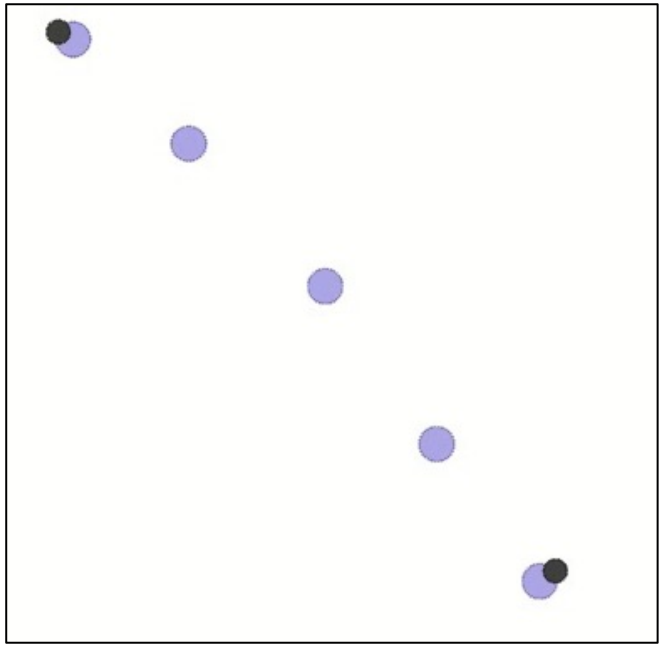
Target



Coverage

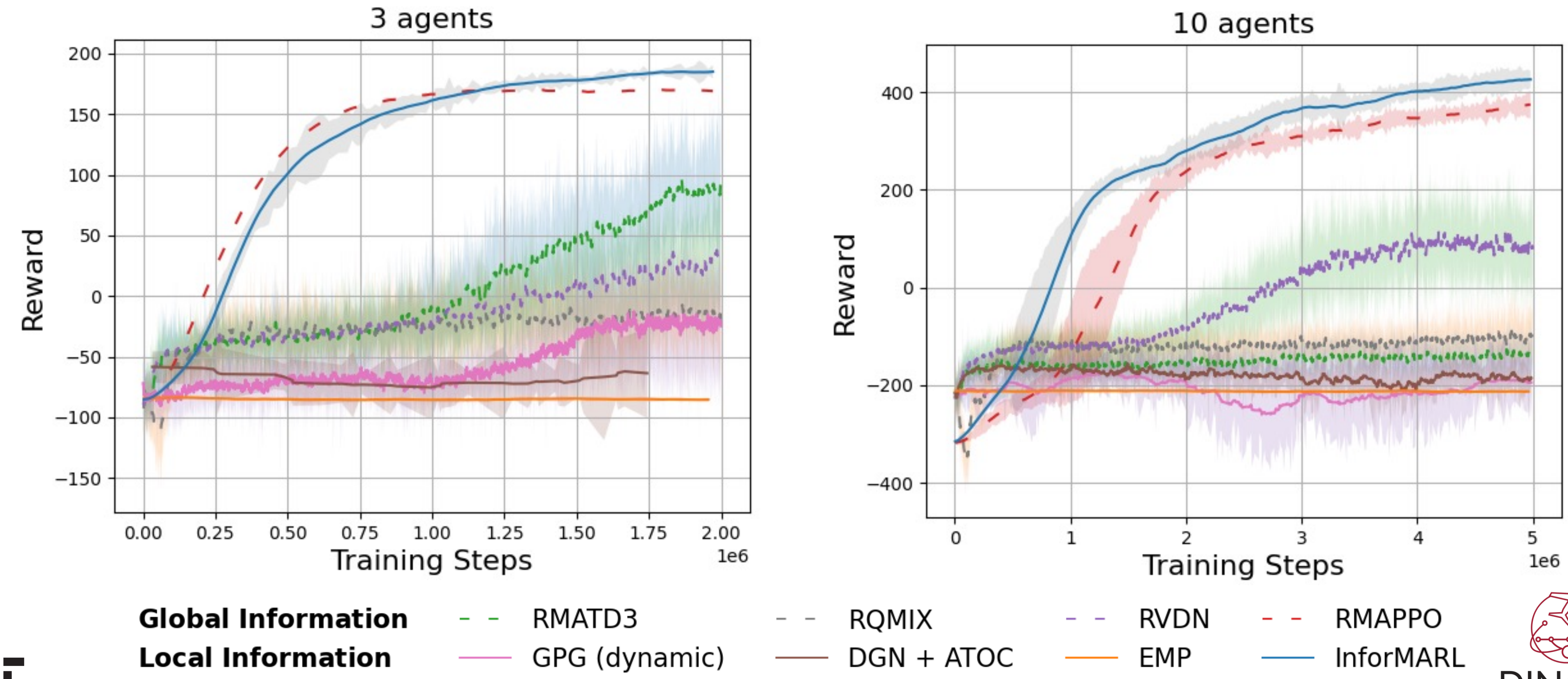


Formation



Line Formation

Experiments: Sample complexity



Experiments: Scalability

↑ - higher better
↓ - lower better

Testing \ Training		n=3	n=7	n=10
m=3	Reward/agent ↑	63.21	63.25	62.87
	Avg. completion time ↓	0.39	0.40	0.40
	Avg. #collisions/agent ↓	0.40	0.46	0.49
	Completion rate ↑	100%	100%	99%
m=7	Reward/agent ↑	61.16	62.23	61.32
	Avg. completion time ↓	0.38	0.40	0.40
	Avg. #collisions/agent ↓	0.74	0.66	0.70
	Completion rate ↑	100%	100%	100%
m=10	Reward/agent ↑	58.59	58.23	58.67
	Avg. completion time ↓	0.38	0.40	0.39
	Avg. #collisions/agent ↓	0.95	0.88	0.87
	Completion rate ↑	100%	99%	100%
m=15	Reward/agent ↑	53.19	53.46	54.21
	Avg. completion time ↓	0.39	0.40	0.40
	Avg. #collisions/agent ↓	1.28	1.21	1.20
	Completion rate ↑	100%	99%	99%

DINaMo

Experiments: Other environments

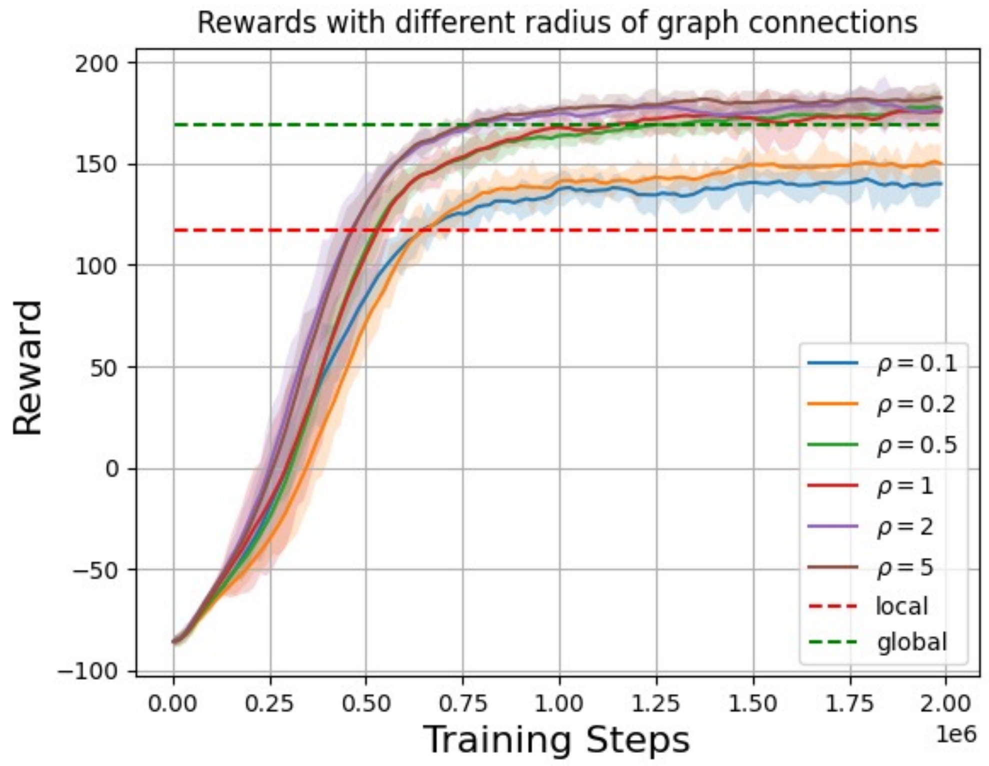
↑ - higher better
↓ - lower better

Task environment	m	Metric	RMAPPO (global info)	InforMARL (local info)
Coverage	m=3	Avg. completion time ↓	0.34	0.36
		Completion rate ↑	100%	100%
	m=7	Avg. completion time ↓	0.42	0.43
		Completion rate ↑	100%	99%
Formation	m=3	Avg. completion time ↓	0.31	0.30
		Completion rate ↑	100%	100%
	m=7	Avg. completion time ↓	0.47	0.43
		Completion rate ↑	100%	100%
Line	m=3	Avg. completion time ↓	0.24	0.21
		Completion rate ↑	100%	100%
	m=7	Avg. completion time ↓	0.38	0.36
		Completion rate ↑	100%	100%

DINaMo

Effect of Sensing Radius

- Vary the sensing radius for InforMARL
- Diminishing returns in performance from increasing sensing radius



Conclusions

- InforMARL uses a graph neural network (GNN)-based architecture for **scalable** multi-agent RL in a **decentralized** fashion.
- InforMARL is **transferable** to scenarios with a different number of entities in the environment than what it was trained on.
- InforMARL has **better sample complexity** than most other standard MARL algorithms with global observations
- Add strict safety constraints for guaranteeing no collisions

Thank You

Questions we couldn’t get to?

Ideas to collaborate?

Drop me an email at sidnayak@mit.edu