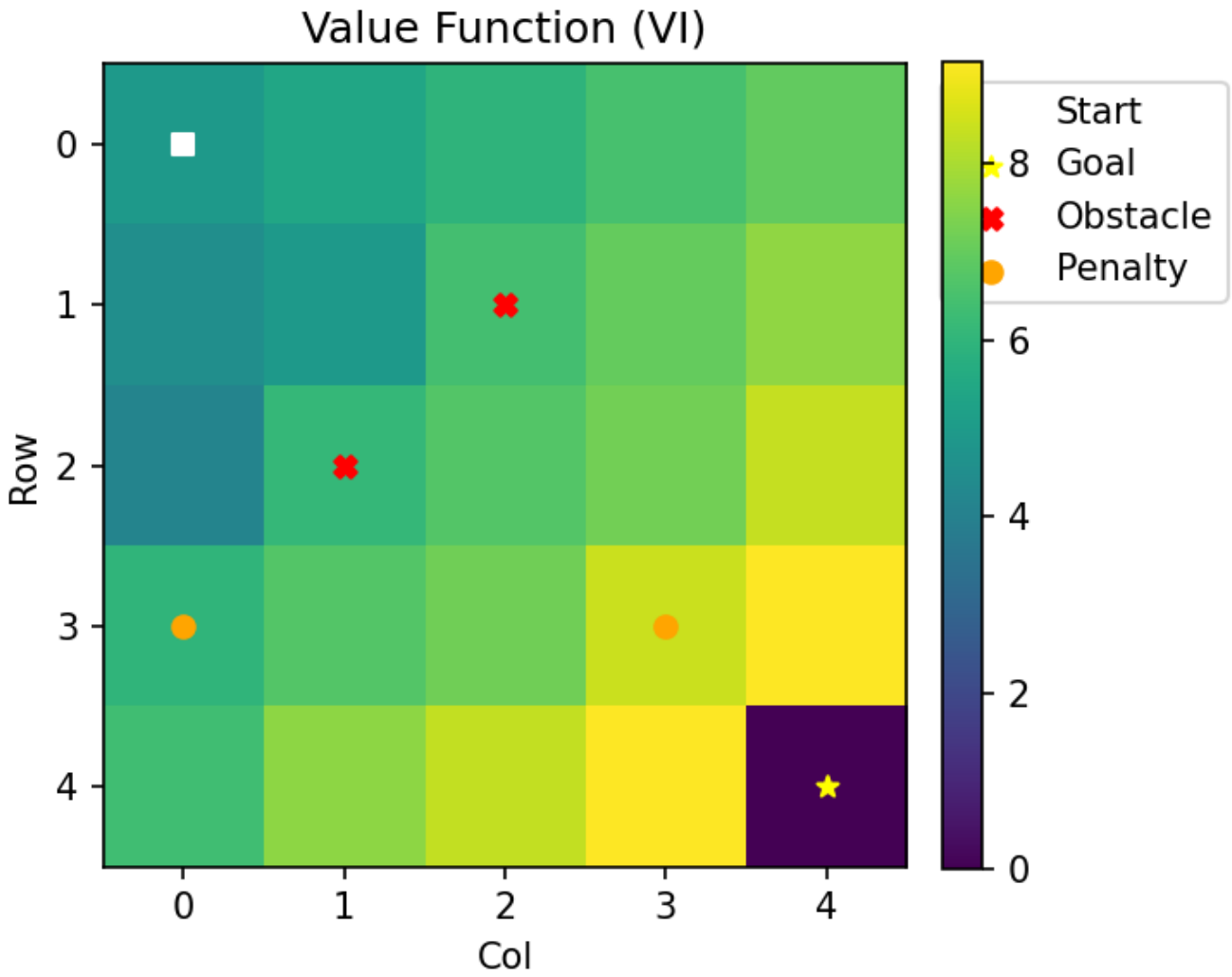
Problem 1

Iterations to Convergence

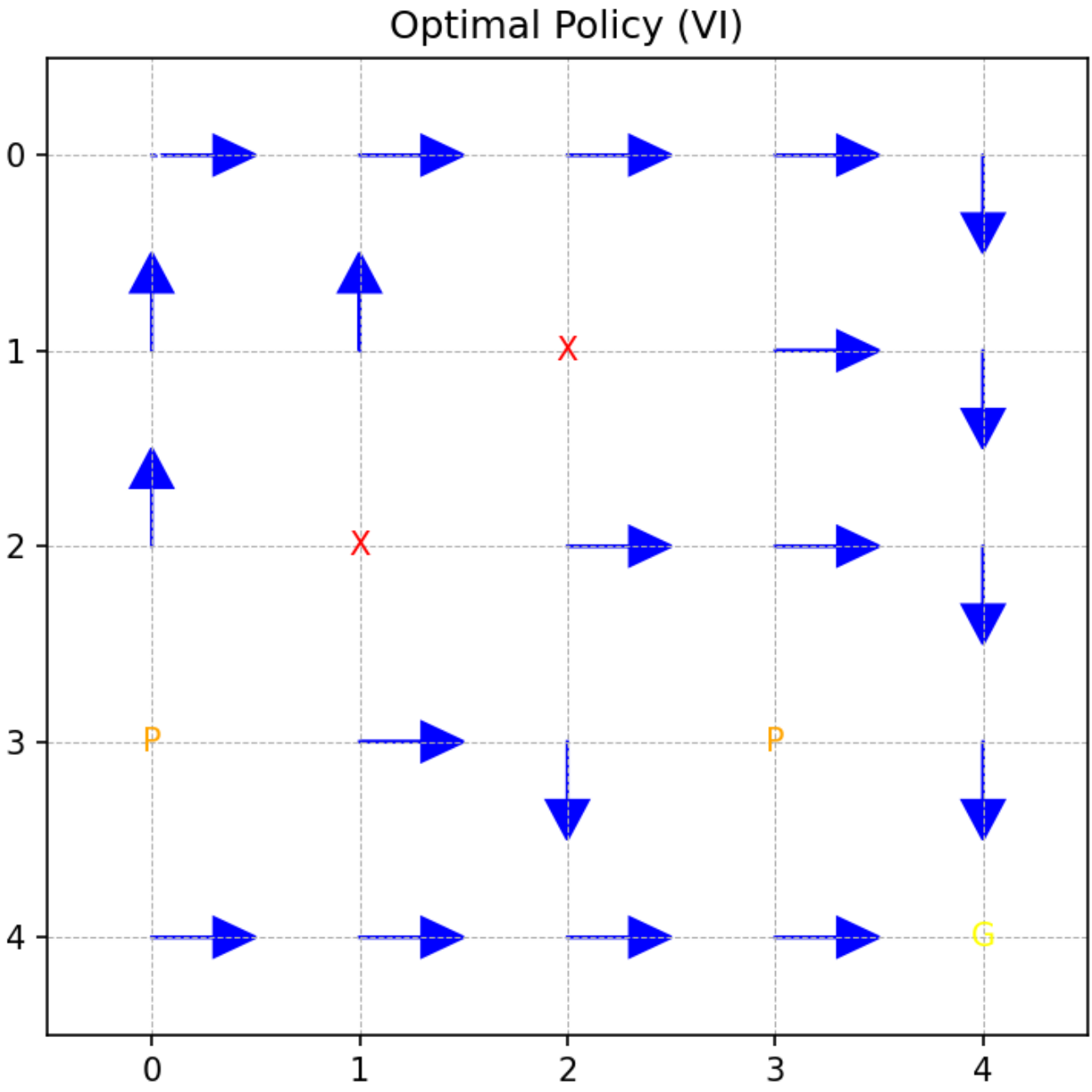
- Value Iteration: **31** sweeps to hit the convergence threshold.
- Q-Iteration: **31** sweeps as well. The matching counts confirm the tabular solvers reach the same fixed point under the provided tolerance.

Final Value Function



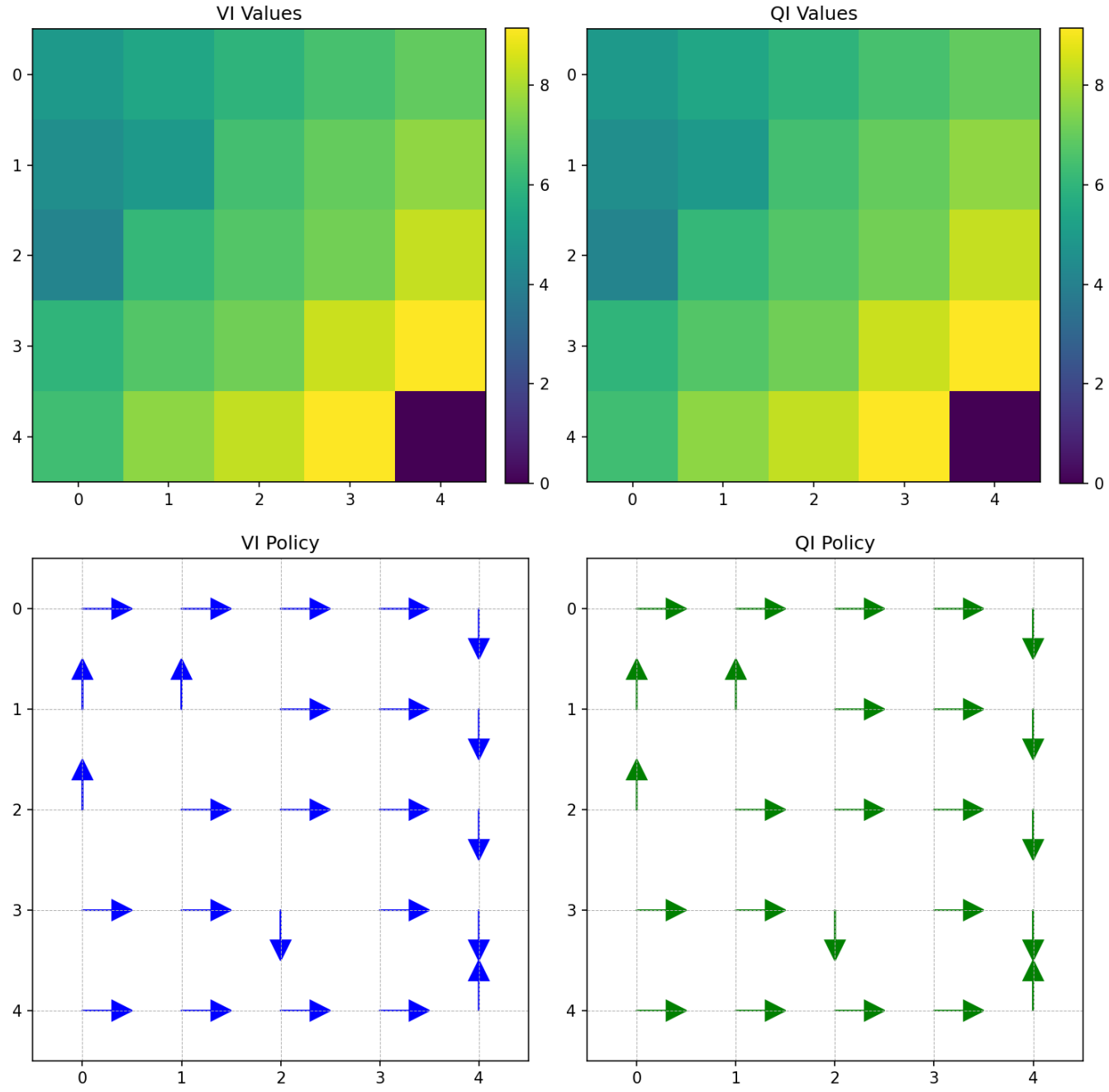
The heatmap highlights the state value that rises near the goal states and drops near terminal penalty and obstacles.

Optimal Policy Visualization



Arrows indicate the greedy action per cell derived from the final value function. Policy arrows point toward the goal basin while detouring around obstacle and penalty cells.

Value Iteration vs Q-Iteration



- Both algorithms converged in 31 iterations and got same values and optimal policies. I think it may be because it is a relatively simple problem (few states or few degree of freedom).

Effect of Stochastic Transitions

Stochastic transitions smear probability mass toward unintended neighboring cells, making the optimal policy favor safer routes that maintain high expected value even when slips occur. This also lowers peak values near the goal compared to a deterministic setting because agents must budget for chance failures on each step.

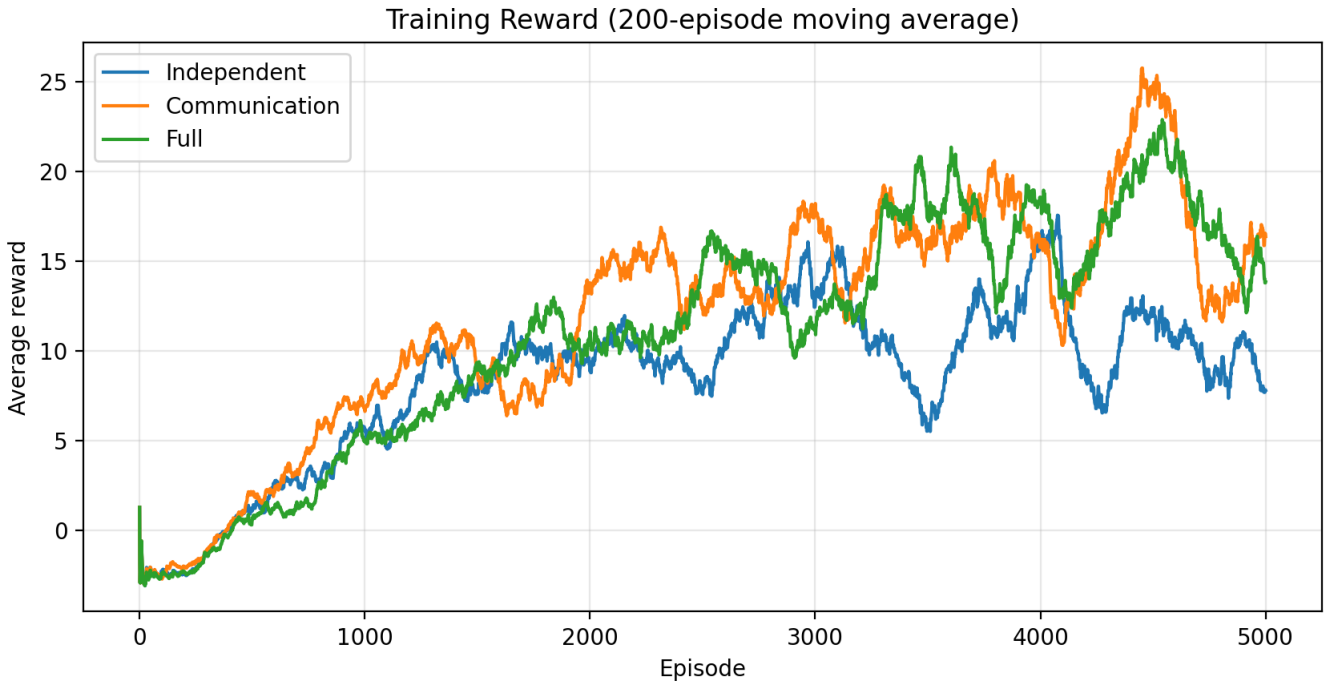
Problem 2

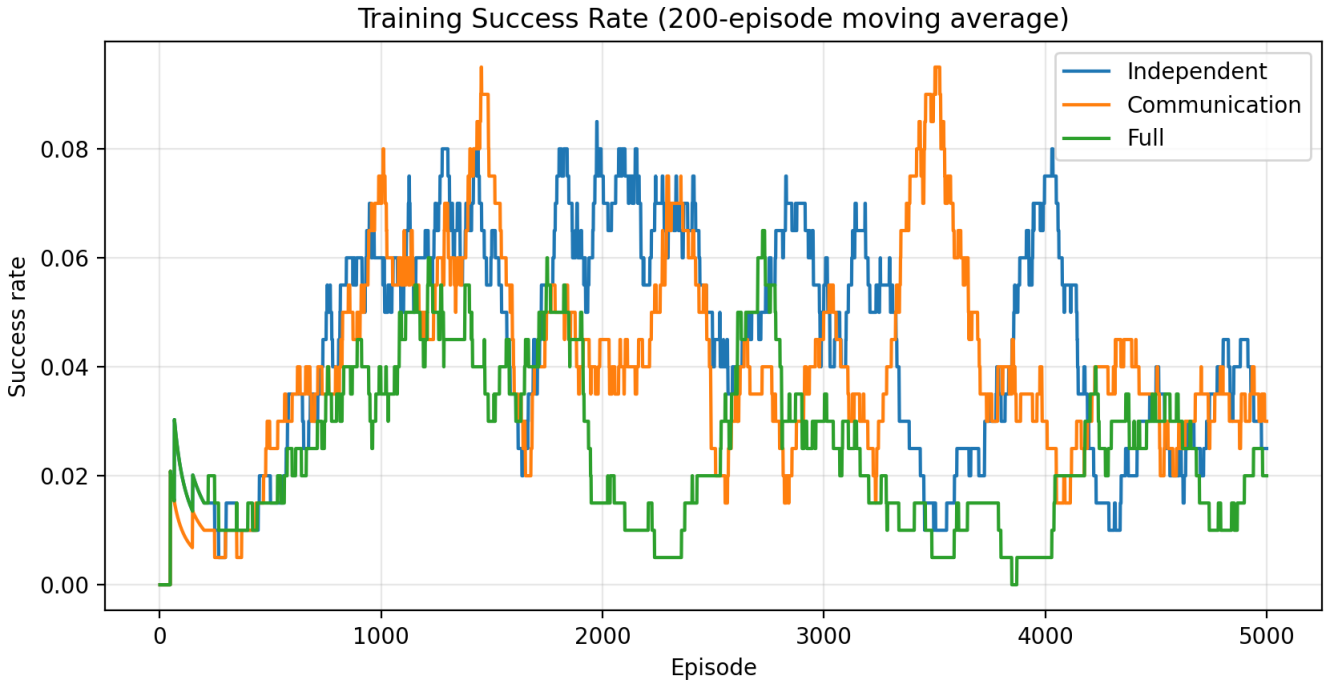
Training Hyperparameters

Shared across `results_independent`, `results_comm`, and `results_full` (see `train.py`).

- Episodes: 5,000 per configuration with seed 641
- Optimizer: Adam, learning rate 1e-3
- Batch size: 32;
- replay buffer: 50,000 transitions
- Discount factor: 0.99; target network update every 100 episodes
- Epsilon schedule: $\epsilon_0=1.0$ decayed each episode by a factor of 0.9995 down to ϵ_{min}=0.05 (give agents more opportuniteis to explore the trajectories)

Training Curves (Avg Reward & Success Rate)





Final Evaluation Metrics

Config	Mean reward	Success rate	Avg steps
independent	10.67	**0%**	50.0
comm	11.17	**0%**	50.0
full	23.15	**4%**	48.5

Only the full-information agents ever deliver simultaneous arrivals (2/50 episodes). Independent and comm agents routinely earn positive reward by reaching the goal individually and timing out while waiting, which never yields the +10 cooperative bonus.

Performance Comparison Across Configurations

Config	Extra info available	Training overall reward	Training overall success	Eval success	Notes
independent	None	8.60	4.4%	0%	Relies purely on a 3×3 egocentric view; successes disappear once ε decays.
comm	Learned comm scalar (distance masked)	12.20	4.1%	0%	Comm outputs cluster near 0.70 with 0.68 correlation, so signals add little beyond faster exploration.
full	Comm scalar + normalized inter-agent distance	11.28	2.5%	4%	Distance lets the first agent camp on the target and occasionally guide the partner, boosting evaluation reward and rare successes.

Impact of Distance Information & Communication

- `results_comm/evaluation_results/communication_analysis.json` shows both agents emitting tightly clustered signals (means ≈0.70, correlation 0.68). The network mostly learns to broadcast “target confidence,” not the direction to travel, so the second agent still wanders.
- `results_full/evaluation_results/communication_analysis.json` features higher variance (correlation 0.24). Coupled with the distance scalar, a strong signal roughly indicates “partner already at target,” allowing the follower to treat the distance cue as a homing metric—hence the modest 4% success rate.
- When all shared signals are masked (`results_independent/...`), correlation drops to -0.20 and success collapses to 0% because each agent lacks any hint that the partner is waiting on the goal.

Learned Strategy Discussions

- **Independent:** Agents execute short biased random walks around their spawn tiles. Rewards spike only when both accidentally bump into the target simultaneously; otherwise they time out with -0.1 penalties (50-step average). No agent willingly stays once it reaches the goal because there is no shared state to indicate progress.
- **Comm-only:** One agent often reaches the goal first and idles, but the emitted scalar (mean 0.70) carries no spatial encoding. The trailing agent reacts by slightly enlarging its search radius yet still circles obstacles blindly, so episodes still end at 50 steps with no cooperative bonus.
- **Full:** Distance input gives the leader a gradient toward the target. After arrival it tends to “camp” (reward spikes of 47–100). The follower monitors both the distance scalar and the elevated comm value, occasionally inferring the correct direction, which explains the only observed successes (4%) and mean reward 23.15.

Overall, slowing the epsilon decay let agents discover the goal faster, but coordination remains limited because communication lacks semantics. Future work should either encode explicit “target found” messages or broaden the observation so trailing agents can triangulate the goal once the leader arrives.