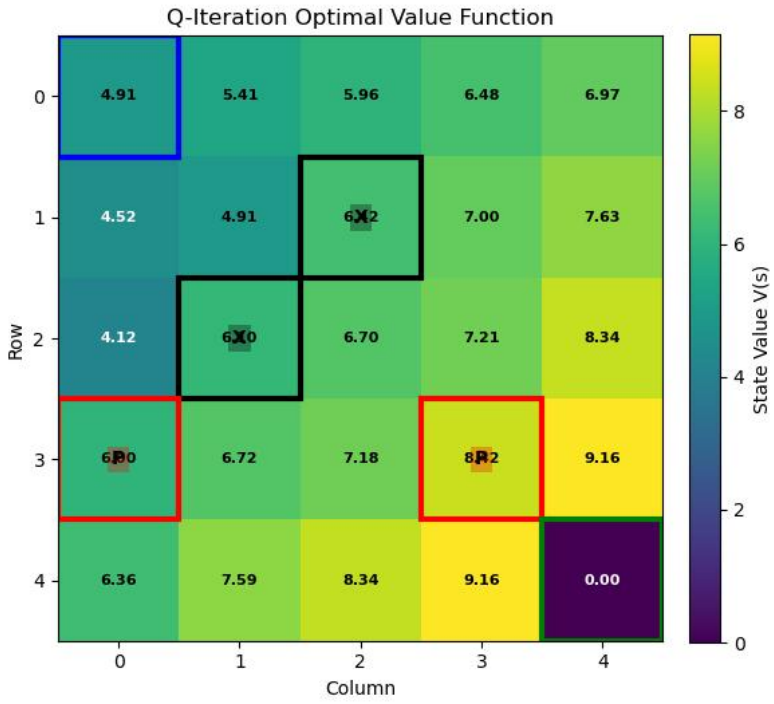
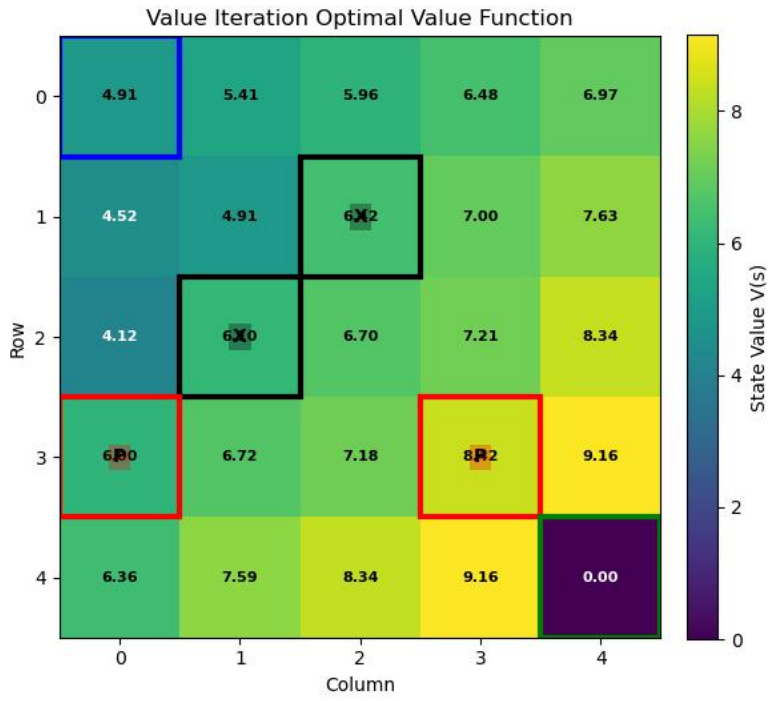
Problem1

- **Number of iterations until convergence for both algorithms**

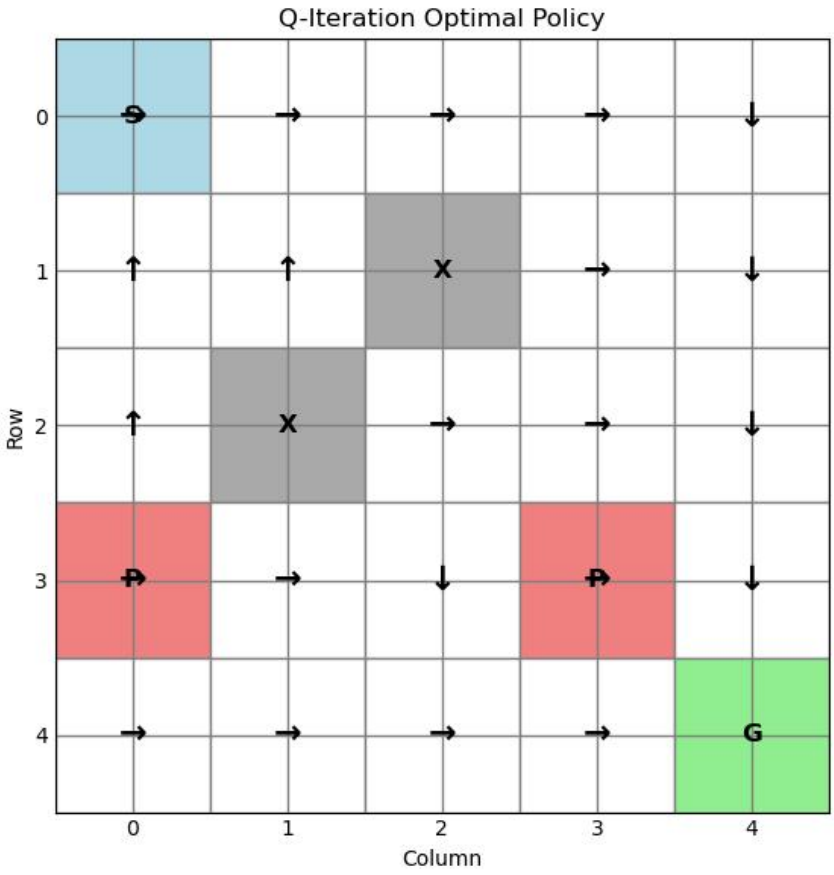
1.Value Iteration (VI): 31 iterations
2.Q-Iteration (QI): 31 iterations

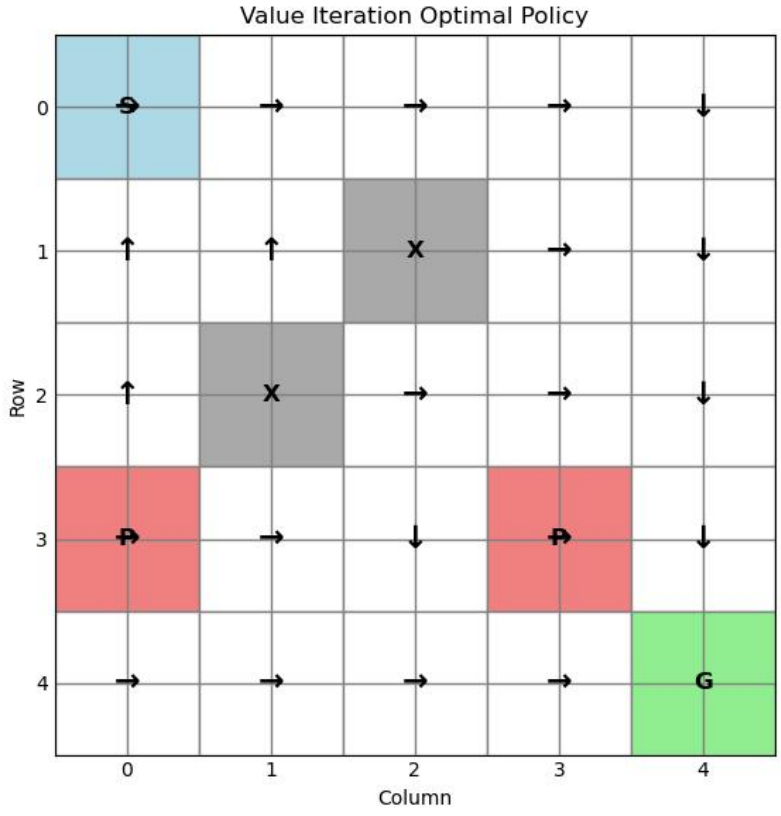
- **Visualization of the final value function (heatmap)**



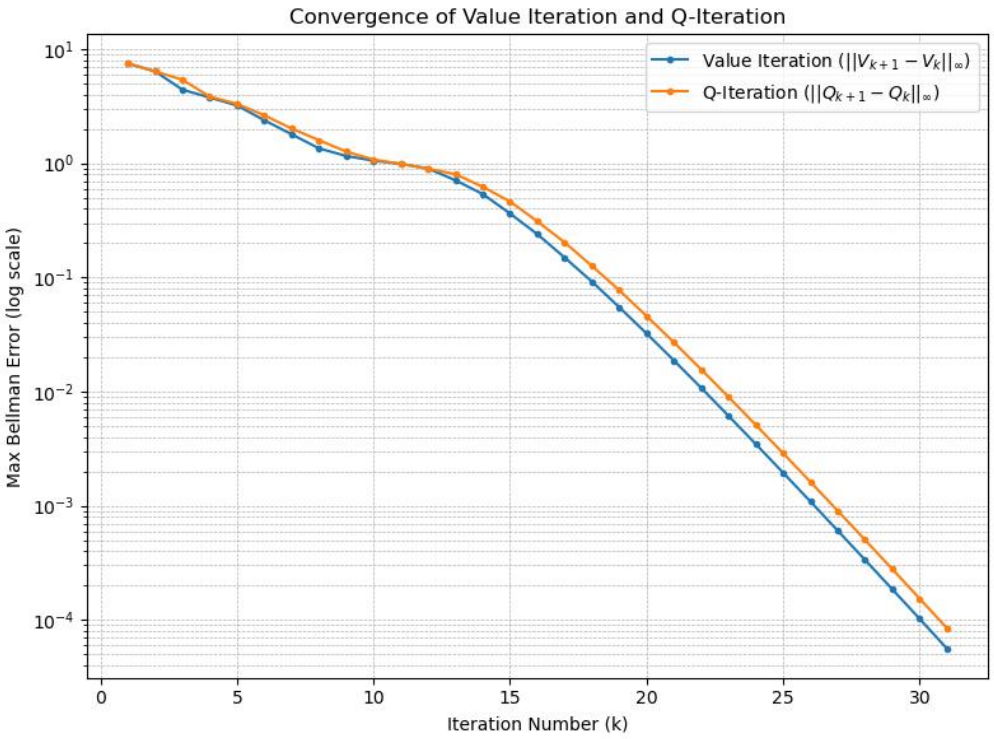


- Visualization of the optimal policy (arrows on grid)





- **Brief comparison of Value Iteration vs Q-Iteration convergence rates**




```
{
"vi_iterations": 31,
"qi_iterations": 31,
"vi_convergence_time": 0.011943817138671875,
"qi_convergence_time": 0.027086734771728516,
"policies_match": true,
"max_value_diff": 0.0
}
```


As observed from the convergence plot, both Value Iteration (VI) and Q-Iteration (QI) converged in 31 iterations, indicating they require the same number of steps to reach the optimal solution's convergence threshold. However, a comparison of the actual computational time reveals that Value Iteration's convergence time (0.0119s) is significantly lower than that of Q-Iteration (0.0271s), making VI faster in practice. This difference in speed is attributed to the internal computational complexity of each algorithm's update step: Q-Iteration updates a value for every state-action pair (Q(s,a)), whereas Value Iteration updates only a single value per state (V(s)). Since QI processes a larger number of values (S×A vs. S) in each iteration, its per-iteration computational cost is higher. Consequently, for this small-scale problem, this increased complexity leads to VI achieving a faster practical convergence speed despite the identical number of iterations required.

- **Discussion of how the stochastic transitions affect the optimal policy**

1. In a deterministic environment, every action taken by the agent directly moves it toward the goal. However, in a stochastic environment, the policy must account

for the 20% chance of unintended movement. Because of this risk, the optimal policy avoids taking actions that keep the agent too close to obstacles or grid boundaries. This helps prevent the 10% sideways drift from causing collisions (which incur penalties) or trapping the agent in loops.

2. Since the goal state T provides a large positive reward, states near the goal typically have very high value. In many of these states, the optimal policy chooses the "Stay" action. Although this may look inefficient, it is actually safer: the agent avoids the risk of drifting away from the goal due to stochastic movement. Because the discount factor is high (γ=0.95), remaining in a high-value state can yield a higher long-term return than taking a risky step toward the goal.
3. Stochastic transitions make the value function V(s) smoother. Instead of following the strict "shortest path," the value propagation reflects the "lowest expected risk" path. Even actions that deviate from the optimal route do not lose value sharply, because there is always a 20% chance of drifting back to a favorable position. As a result, the differences in value between neighboring states become less abrupt.

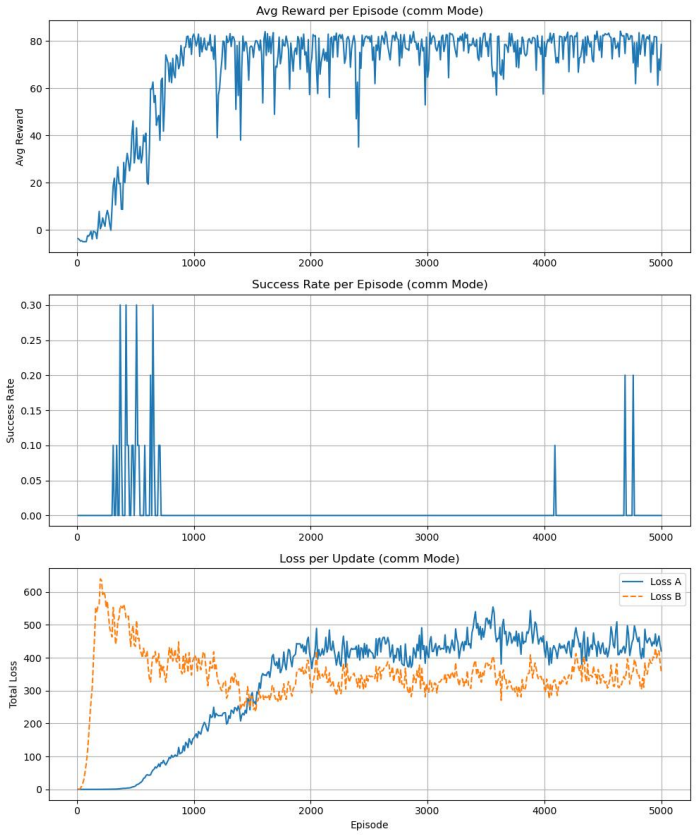
Problem2

- **Training hyperparameters (learning rate, batch size, epsilon schedule, replay buffer size)**

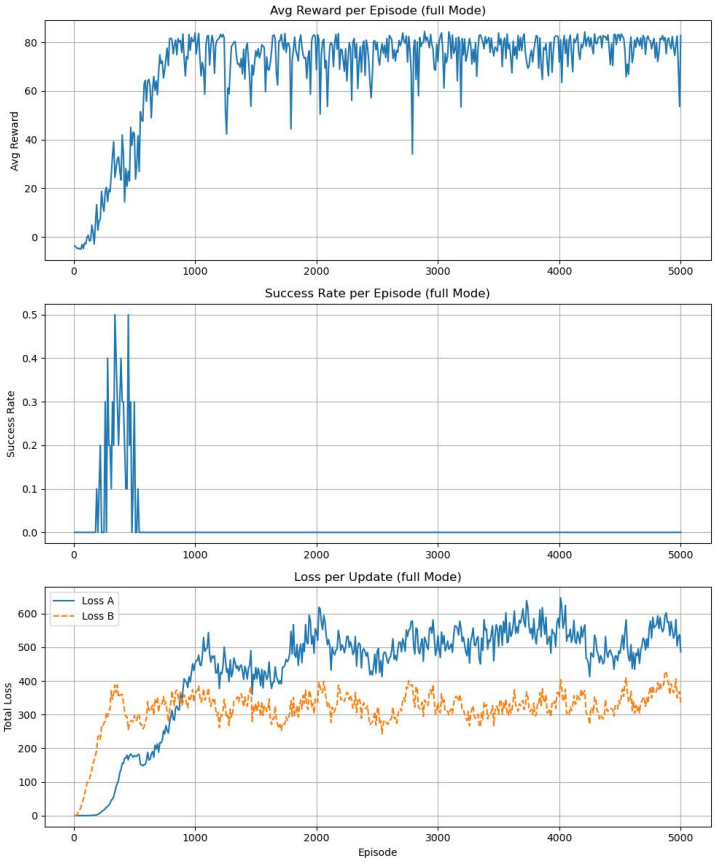
Parameter	Value
learning rate	1e-3
batch size	32
epsilon schedule	5000
replay buffer size	10000

- **Training curves for all three configurations showing average reward and success rate**

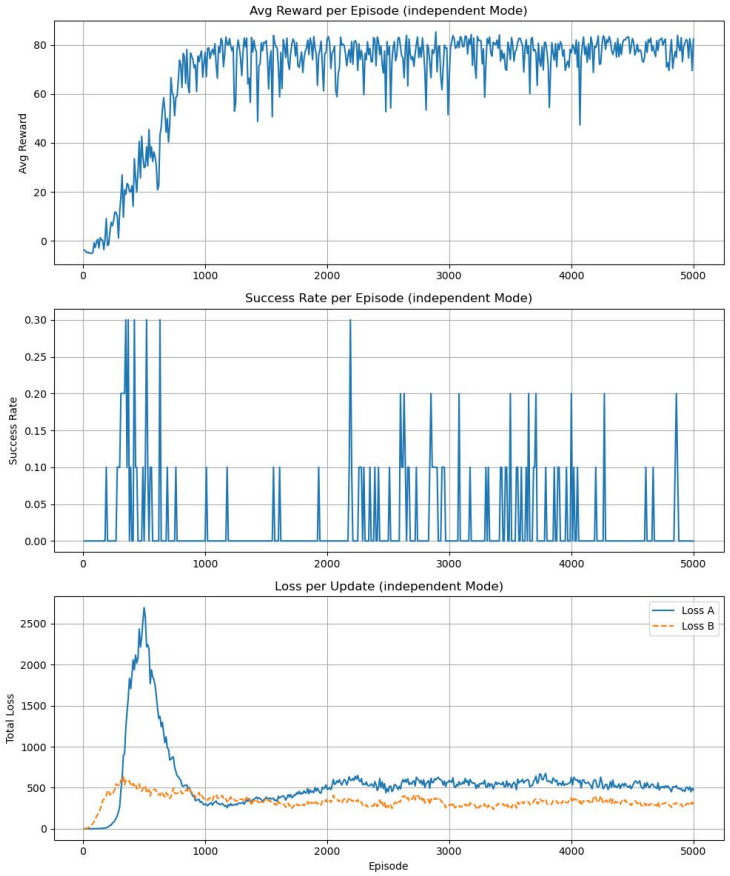
1. Communication Mode



2. Full Information Mode



3. Independent Mode



- **Final success rates for each configuration**

Configuration	Success Rate
Communication	0%
Full Information	0%
Independent	0%

All agents fail to complete even one coordinated episode under any configuration.

- **Comparison table of performance across configurations**

Configuration	Distance Provided	Communication Provided	Success Rate
Communication	no	yes	0%
Full Information	yes	yes	0%
Independent	no	no	0%

- Analysis of how distance information and communication affect coordination

Under the training conditions used in this experiment, all three configurations (Independent, Full, and Comm Only) failed to learn effective coordination, achieving a success rate of 0%. This indicates that in a partially observable multi-agent environment with extremely sparse rewards, relying solely on local observations or weak global signals (such as a distance feature or a scalar communication channel) is

insufficient for agents to establish a stable cooperative strategy. Specifically, the Independent configuration relies only on a 3×3 local observation patch, meaning that each agent spends most of the time unable to see the target or the other agent. Without any meaningful global information, the agents cannot determine whether to wait, speed up, or synchronize their movement, causing all episodes to terminate due to reaching the maximum step limit, resulting in 0% success.

In the Full configuration, we provide the agents with the normalized L2 distance in an attempt to supply explicit information about "how far the two agents are from each other," which could theoretically help them determine whether to wait or catch up. However, the results show that the agents almost completely ignore this additional global-state feature. Changes in L2 distance do not form a stable association with the final reward, and the learned trajectories remain almost identical to those in the Independent mode. The communication signal also collapses to nearly zero. Due to the non-stationarity of multi-agent Q-learning, the training process becomes unstable, preventing the explicit distance information from being effectively incorporated into the policy.

The Comm Only configuration is, in principle, designed to allow agents to share critical information through a communication signal. However, the evaluation results show that the communication channel collapses entirely to a constant zero output, meaning that no meaningful communication protocol is learned and no global information is transmitted. This collapse occurs primarily because the communication output does not directly influence the reward, making the "zero output" the most stable local optimum for the network. Combined with limited local observations, sparse rewards, and the inherent instability of independent Q-learning, the agents fail to establish any connection between communication and successful coordination. As a result, the Comm Only configuration behaves identically to the Independent configuration and also fails to achieve coordination.

- Discussion of learned strategies in each configuration

Overall, the training curves and trajectory visualizations indicate that all three configurations converge to the same "safe but ineffective" strategy: avoid collisions, avoid risky actions, and move locally until the episode ends at 50 steps. Regardless of whether distance information or communication is provided, the agents do not exhibit behaviors such as approaching the target, moving toward each other, or attempting synchronized arrival. In summary, under the current training framework (independent Q-learning, sparse rewards, and partial observability), neither distance information nor communication is successfully transformed into a usable coordination signal, making it impossible for the agents to achieve cooperative behavior.