REINFORCEMENT LEARNING IN ROBOTICS-ASSIGNMENT 1

Group members:
Teja Vishnu Vardhan Boddu
Vijayramsriram Sathananthan

Q-LEARNING SOLUTION FOR GRIDWORLD NAVIGATION

Objective: Solve the 3×4 stochastic gridworld problem using reinforcement learning

Approach: Online Q-Learning

Agent learns through direct interaction with environment

Updates Q-values after each action (no prior model needed)

Balances exploration (trying new actions) vs exploitation (using learned knowledge)

Algorithm: Q-Learning with ε-greedy policy

Update Rule: $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \cdot \max Q(s',a') - Q(s,a)]$

Training: 1000-2000 episodes per experiment

Hyperparameters Tested:

- Learning rate (α): 0.01, 0.1, 0.5, 1.0
- Discount factor (γ): 0.5, 0.8, 0.95, 0.99
- Exploration rate (ε): 0.0, 0.1, 0.3, 0.5

Environment Specifications:

Grid: 3 rows × 4 columns

Start: (1,1) | Goal: (4,3) [+1] | Trap: (4,2) [-1] | Wall: (2,2)

Stochastic transitions: 80% intended, 10% perpendicular left, 10% perpendicular right

Step cost: -0.04 for all non-terminal states

What This Presentation Covers:

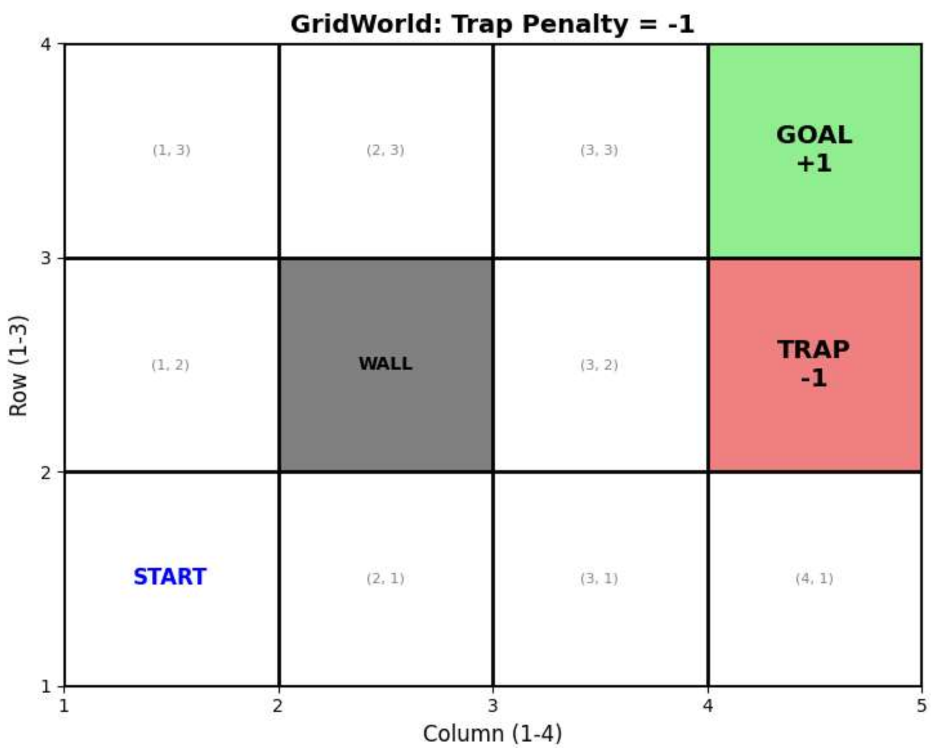
Effect of hyperparameters (α, γ, ε) on learning performance

Q-value vs policy convergence analysis

Impact of extreme penalty changes (-1 vs -200)

Code Repository: GitHub:

https://github.com/btvvardhan/Gridworld_Qlearning

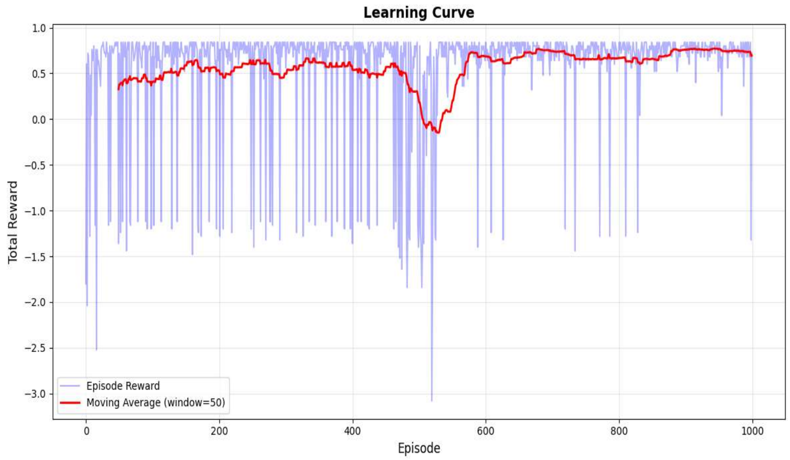


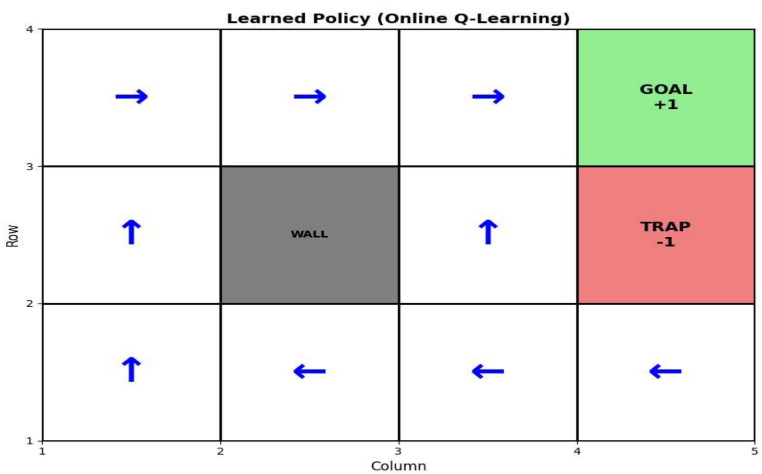
PROJECT SETUP & ENVIRONMENT

```
# Import necessary modules
from gridworld import GridWorld
from qlearning_agent import QLearningAgent
from experiments import *

# Set random seeds for reproducibility
np.random.seed(42)
random.seed(42)

# Create the gridworld environment
env = GridWorld(trap_penalty=-1)
env.visualize_grid()
```





BASELINE SOLUTION - TRAINED AGENT

Training Results:

Initial performance: Avg reward ~0.367 (Episode 100)

Final performance: Avg reward ~0.729 (Episode 1000)

Improvement: ~97% increase in average reward

Q-table: 9 non-terminal states successfully learned

Key Observations:

Agent learns to navigate toward goal (right arrows in top row)

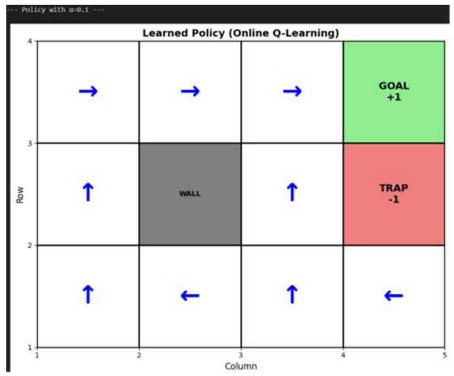
Avoids trap at (4,2) by moving up at (3,2)

Bottom row moves left to avoid unnecessary steps toward trap

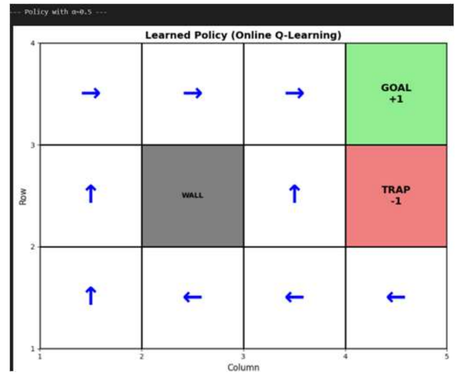
Policy shows clear path optimization despite stochastic transition

Learning rate =0.01

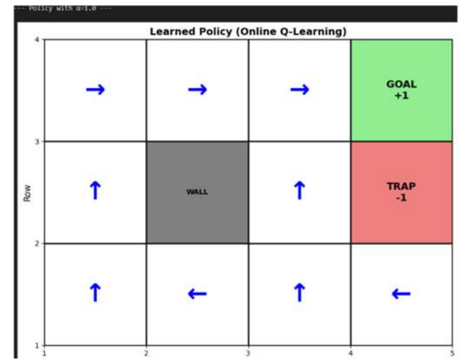
Learning rate =0.1

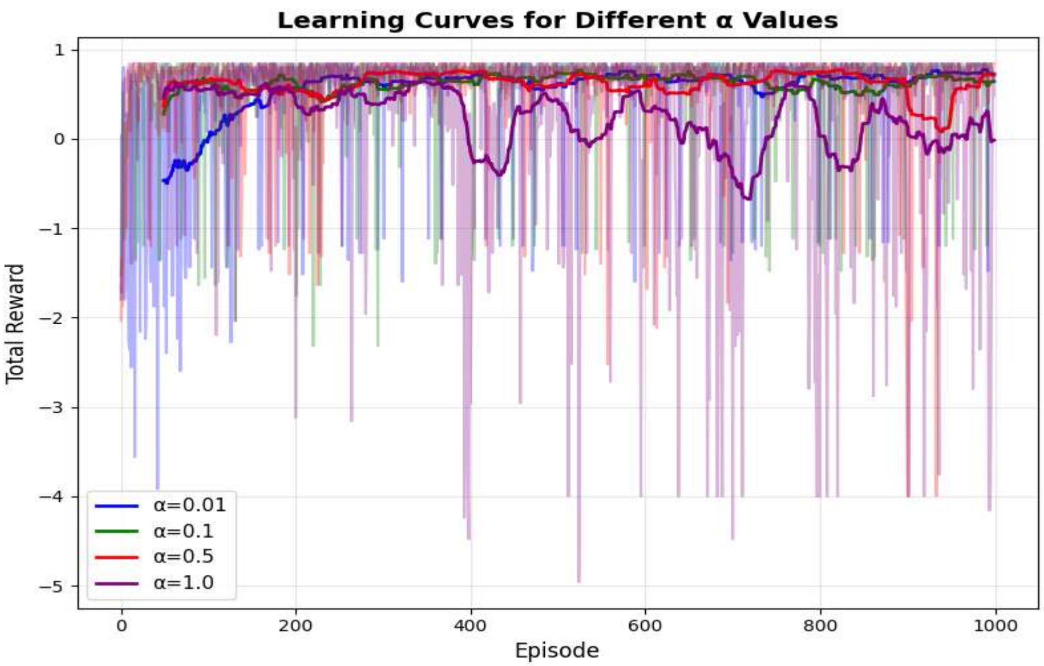


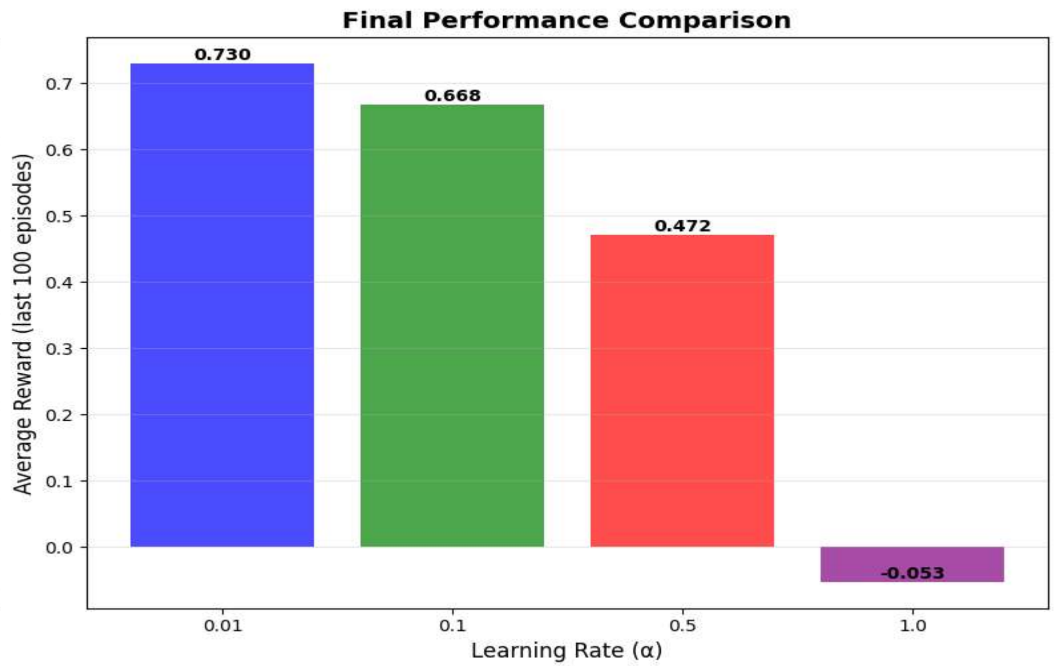
Learning rate =0.5



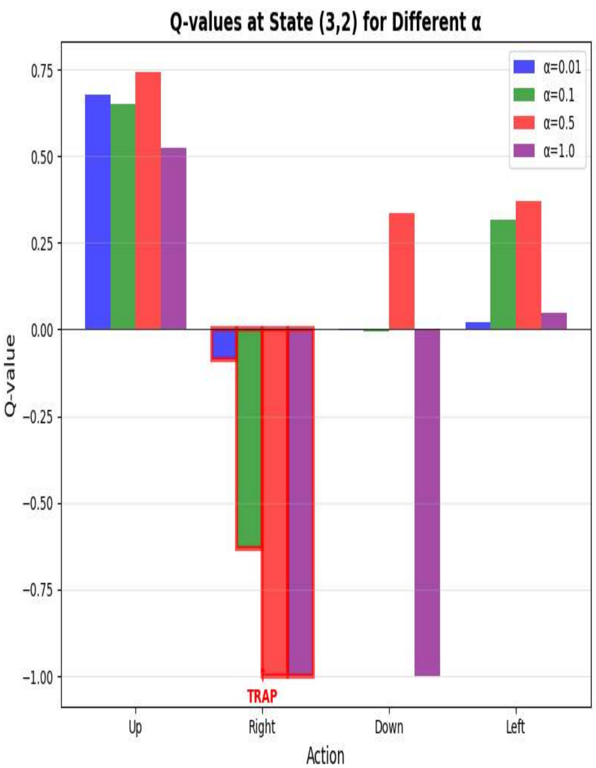
Learning rate =1

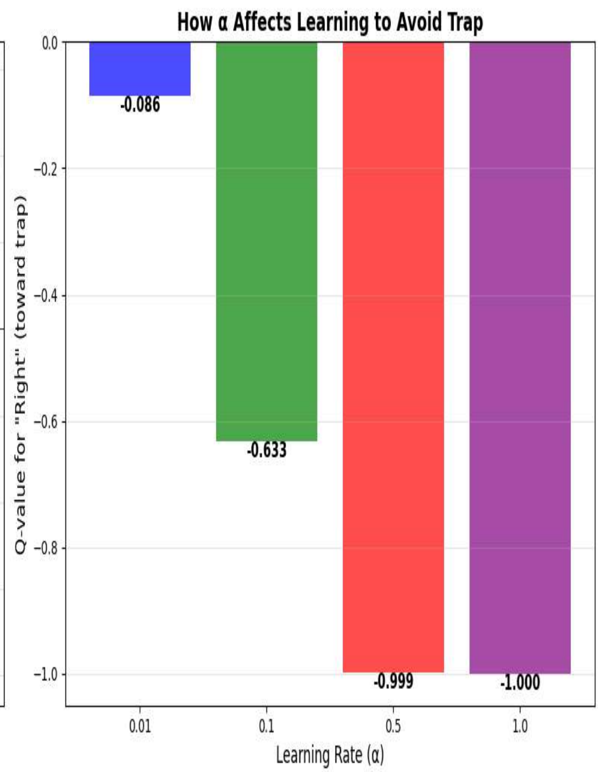






LEARNING RATE EFFECT ON RISK ASSESSMENT AT (3,2)





Critical Insights:

Trap Detection Accuracy:

- α=0.01: Mild negative Q-value (-0.086) - slow to learn danger
- α=0.1: Strong negative (-0.633) - good risk assessment
- α=0.5, 1.0: Maximum negative (-1.0) - recognizes trap but unstable overall

All agents correctly choose "Up" as best action - policy is consistent

Trade-off:

- Low α: Gradual, stable learning but slower trap recognition
- High α: Fast trap recognition but unstable Q-values elsewhere

Conclusion: α=0.1 provides optimal balance - strong trap avoidance with stable convergence.

Action	α=0.01	α=0.1	α=0.5	α=1.0
Up	0.6783	0.6518	0.7447	0.5132
Right (→TRAP)	-0.086	-0.633	-0.999	-1.000
Down	-0.002	-0.005	0.3476	-0.014
Left	0.021	0.315	0.0486	0.0486

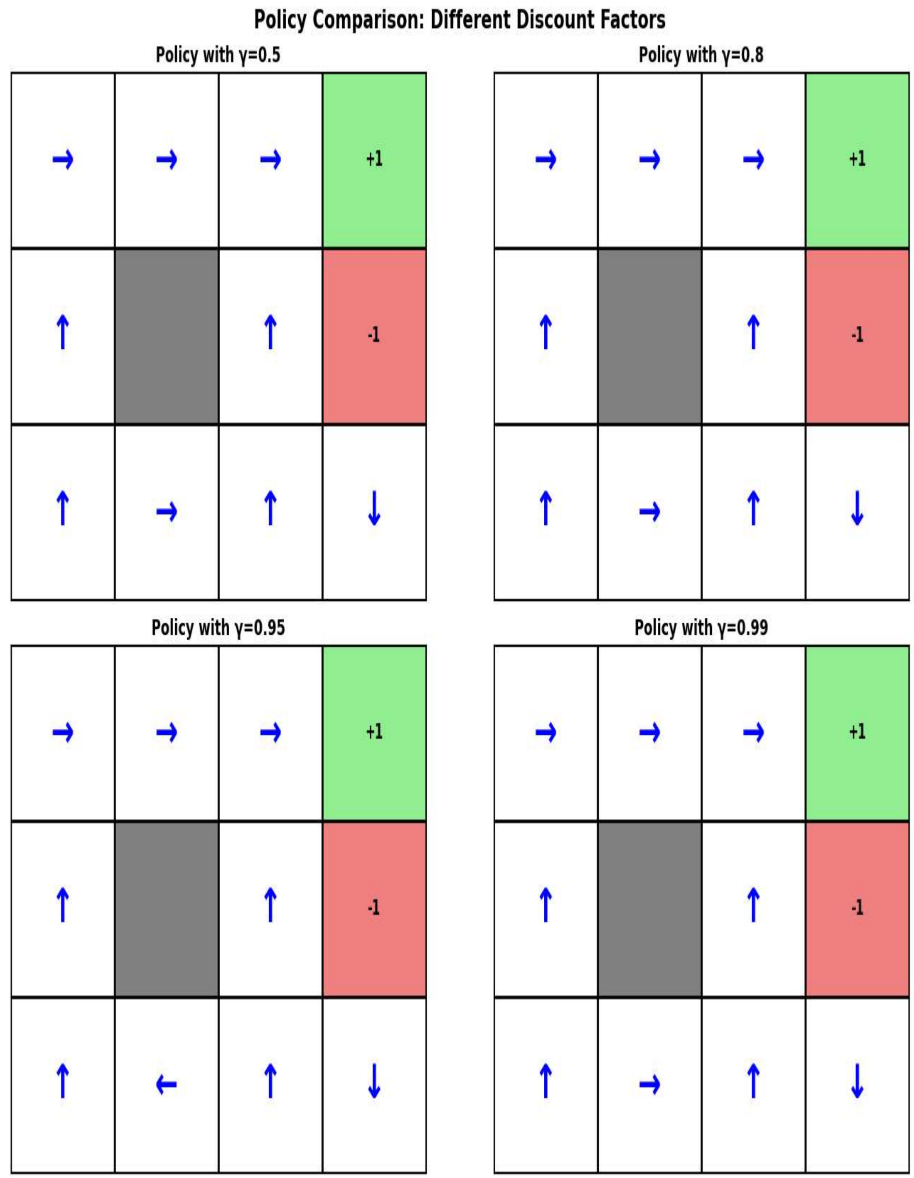
EFFECT OF DISCOUNT FACTOR

Key Observations:
All policies are identical! - Despite different γ values, the optimal actions are the same
Performance difference: Higher γ yields better rewards (0.6516 → 0.7432)
Why identical policies?

- The gridworld is small with short paths to goal
- Optimal strategy doesn't change, but Q-values do
- Higher γ better captures long-term value of avoiding trap

Conclusion: γ = 0.99 is optimal - maximizes future reward consideration without sacrificing performance.

Discount Factor (γ)	Final Avg Reward	Future Vision
0.5	0.6516	Short-term
0.8	0.6760	Medium-term
0.95	0.7324	Long-term
0.99	0.7432	✓ Optimal



EFFECT OF EXPLORATION RATE

Exploration Rate (ε)	Final Avg Reward	Behavior
0.0 (No exploration)	**0.7052**	Pure exploitation
0.1 (Recommended)	0.6760	Balanced
0.3	0.5584	Too much exploration
0.5 (Too Much!)	0.3284	Random behavior

Q-VALUE VS POLICY CONVERGENCE

Convergence Results:

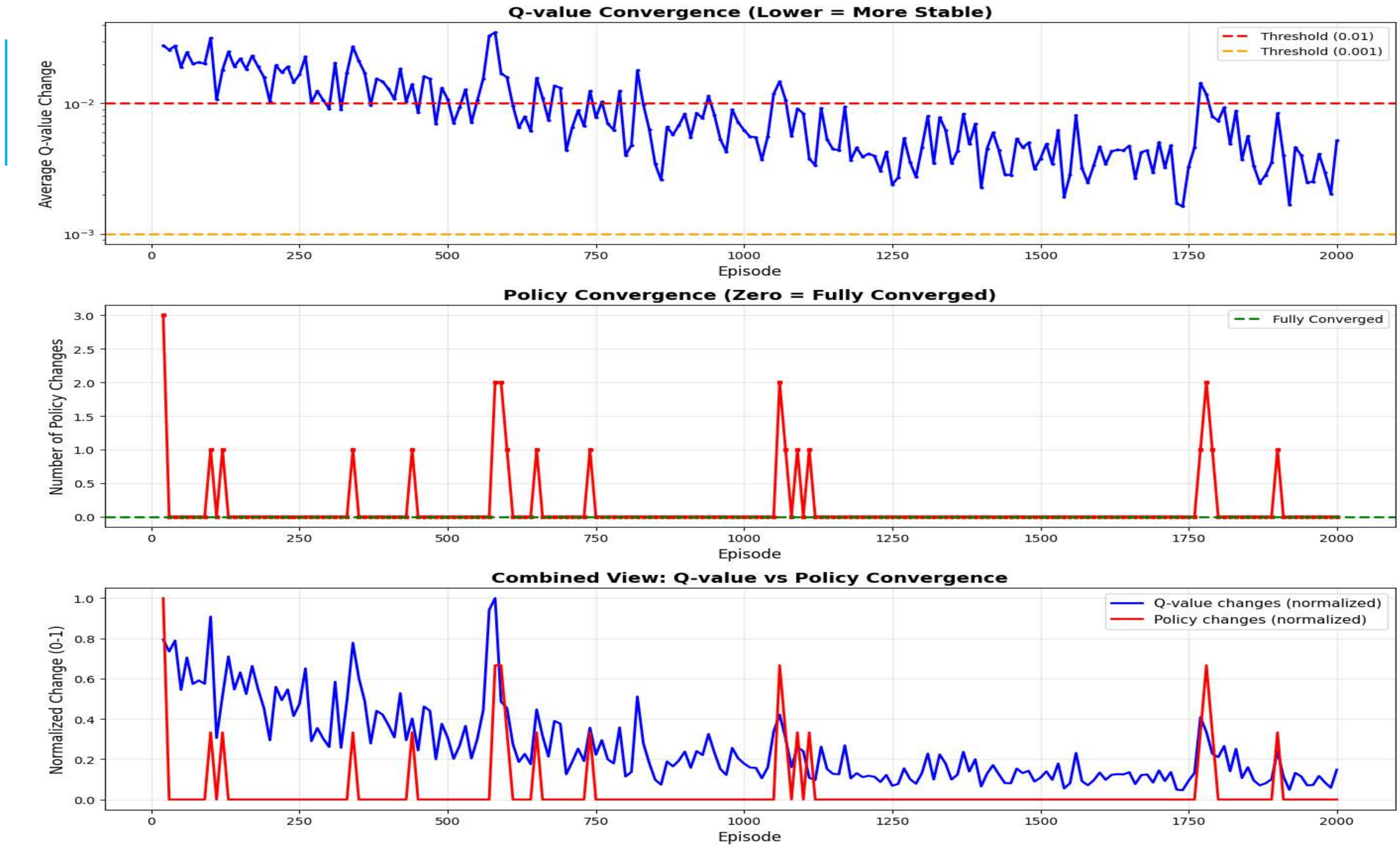
1. **Policy Convergence (Episode 130):**
   - Red line drops to zero and stays there
   - Agent found optimal actions early
   - Only occasional single-state changes due to ε-exploration
2. **Q-Value Convergence (Episode 300):**
   - Blue line continues fluctuating after policy stabilizes
   - Values keep refining even though policy is fixed
   - Never fully stabilizes (change < 0.001) due to stochasticity
3. **Combined View:**
   - Clear divergence: policy (red) settles while Q-values (blue) fluctuate
   - Policy depends only on *relative* Q-values (which action is best)
   - Exact Q-values less critical than action ranking

Answer to Question 2: POLICY CONVERGED FIRST! (170 episodes earlier)

Why:

- Policy only needs to know which action is *best*, not exact values
- Q-values must estimate precise expected returns
- Small Q-value changes don't affect policy if action ranking unchanged

Metric	Convergence Point	Criterion
Policy	Episode 130	0 changes for 100 consecutive episodes
Q-values	Episode 300	Change < 0.01
Gap	170 episodes	Policy converged first


Q-value Convergence (Lower = More Stable)
Average Q-value Change
Threshold (0.01)
Threshold (0.001)
Episode
Policy Convergence (Zero = Fully Converged)
Number of Policy Changes
Fully Converged
Episode
Combined View: Q-value vs Policy Convergence
Normalized Change (0-1)
Q-value changes (normalized)
Policy changes (normalized)
Episode

EFFECT OF TRAP PENALTY (-1 VS -200)

Critical Observation - State (3,2):

Penalty = -1: Agent moves UP (away from trap)

Penalty = -200: Agent moves LEFT (away from trap column entirely!)

Blue spikes to -200: Agent hitting trap during exploration

Red line improvement: Gradually learning avoidance

Final performance: Still worse than -1 penalty case

Key Insights:

Policy Difference: The -200 penalty creates more conservative behavior

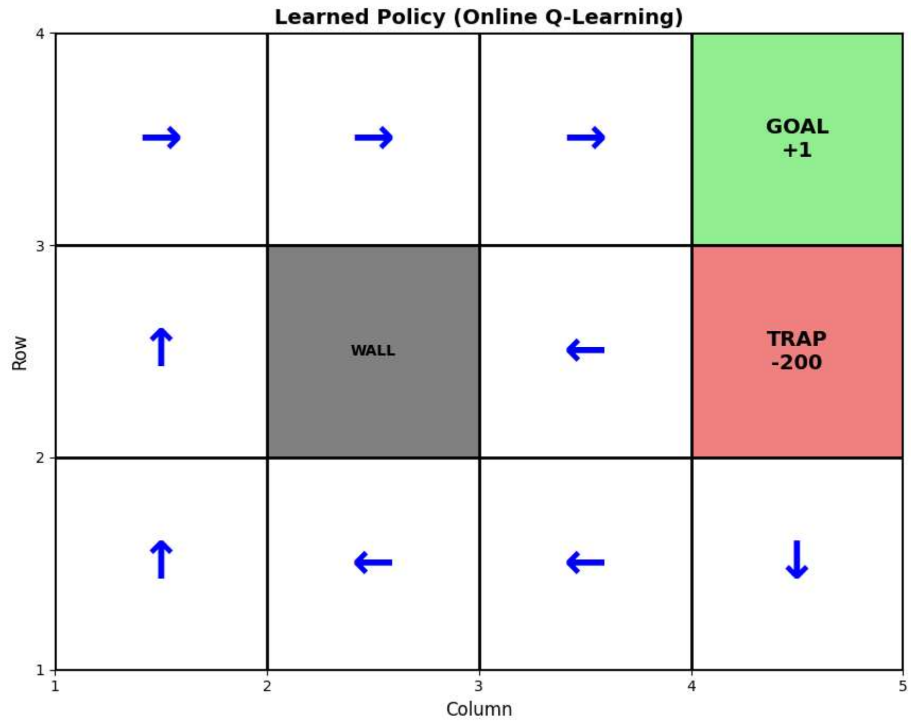
- State (3,2): LEFT instead of UP - maximizes distance from trap
- More cautious navigation throughout grid

Learning Difficulty: Extreme penalties make learning harder

- Early episodes dominated by catastrophic trap hits
- Takes longer to recover and find safe paths
- Exploration becomes risky - one mistake = -200 penalty

Does it Make Sense?

- YES: More severe consequences → more risk-averse policy
- Trade-off: Safety vs efficiency (longer paths to goal)



SUMMARY OF QUESTION 1

Question 1: How do hyperparameters affect learning?

Learning Rate (α):

Lower values (0.01-0.1) produce more stable convergence and better final performance

Higher values (0.5-1.0) learn faster initially but become unstable and perform poorly

Recommended: $\alpha = 0.1$ - optimal balance between learning speed and stability

Trade-off: Slow learning vs instability

Discount Factor (γ):

Higher values (0.95-0.99) achieve significantly better rewards ($0.65 \rightarrow 0.74$)

All tested values produced identical policies in this small gridworld

Higher γ better captures long-term consequences of avoiding trap

Recommended: $\gamma = 0.99$ - maximizes future reward consideration

Exploration Rate (ε):

Critical for policy quality - most impactful hyperparameter

$\varepsilon = 0.0$: Good performance but risky (no exploration, can miss better policies)

$\varepsilon = 0.1$: Balanced exploration-exploitation, reliable learning

$\varepsilon = 0.3\text{-}0.5$: Excessive exploration degrades policy quality significantly

Recommended: $\varepsilon = 0.1$ - sufficient exploration without disrupting learned behavior

Parameter	Optimal Value	Reasoning
α (Learning Rate)	0.1	Best stability-performance balance
γ (Discount Factor)	0.99	Maximizes long-term reward
ε (Exploration Rate)	0.1	Sufficient exploration, stable policy

SUMMARY OF QUESTION 2

Question 2: Does Q-value or policy converge first?

Answer: POLICY CONVERGES FIRST

Policy converged: Episode 130 (0 changes for 100+ consecutive episodes)

Q-values stabilized: Episode 300 (change < 0.01)

Gap: 170 episodes

Why this happens:

Policy depends only on relative ordering of Q-values (which action is best)

Q-values must estimate precise expected returns (absolute magnitudes)

Small Q-value fluctuations don't affect policy if action ranking stays the same

This is theoretically expected in stochastic environments

Implication: An agent can have a stable optimal policy while Q-values continue refining

Parameter	Optimal Value	Reasoning
α (Learning Rate)	0.1	Best stability-performance balance
γ (Discount Factor)	0.99	Maximizes long-term reward
ε (Exploration Rate)	0.1	Sufficient exploration, stable policy

SUMMARY OF QUESTION 3

Question 3: Effect of extreme penalty (-1 vs -200)

Performance Impact:

Penalty -1: Final reward = 0.729

Penalty -200: Final reward = -1.328

Learning with -200 shows catastrophic early failures (reward drops to -200)

Policy Differences:

At critical state (3,2):

- Penalty -1: Move UP (direct trap avoidance)
- Penalty -200: Move LEFT (maximize distance from entire trap column)

Agent with -200 penalty takes longer, more conservative paths

Does it make sense? YES

Higher stakes → more risk-averse behavior (rational)

Trade-off: Safety vs Efficiency

Extreme penalties make exploration dangerous, slowing learning

The policy difference is logically consistent with risk management