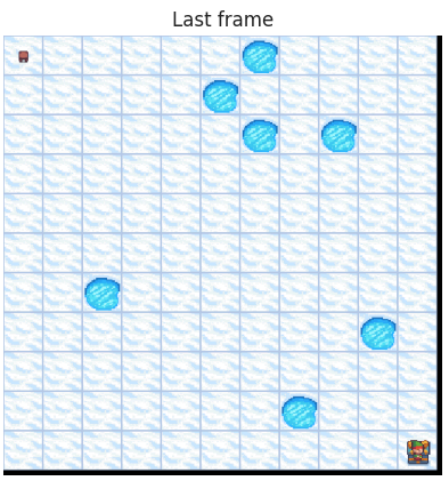
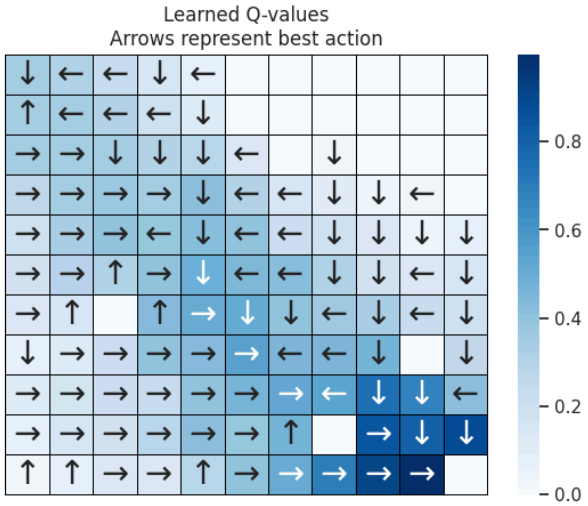
LAB 6, V5: Frozen Lake

GR 5

Introduction

This report presents the results of applying Q-learning to solve the FrozenLake-v1 environment with different hyperparameter settings. The FrozenLake environment represents a grid-world navigation problem where an agent must traverse a frozen lake from a starting point to a goal while avoiding holes. The environment uses a map size of 8x8 and the non-slippery version was selected. The Q-learning algorithm is a model-free reinforcement learning technique that learns the value of an action in a particular state. This algorithm uses an exploration-exploitation strategy to balance between trying new actions and using known good actions.





Experimental Setup

We conducted a series of experiments to investigate how different hyperparameters affect the performance of the Q-learning algorithm in the FrozenLake environment. Each experiment was run for 20,000 episodes with different configurations of the following parameters:

- **Learning rate (α)**: Controls how much new information overrides old information (0.7, 0.8, 0.9)
- **Discount factor (γ)**: Determines the importance of future rewards (0.98, 0.99, 0.995)

- **Epsilon decay rate**: Controls how quickly the exploration rate decreases (0.9995, 0.9999, 0.99995)

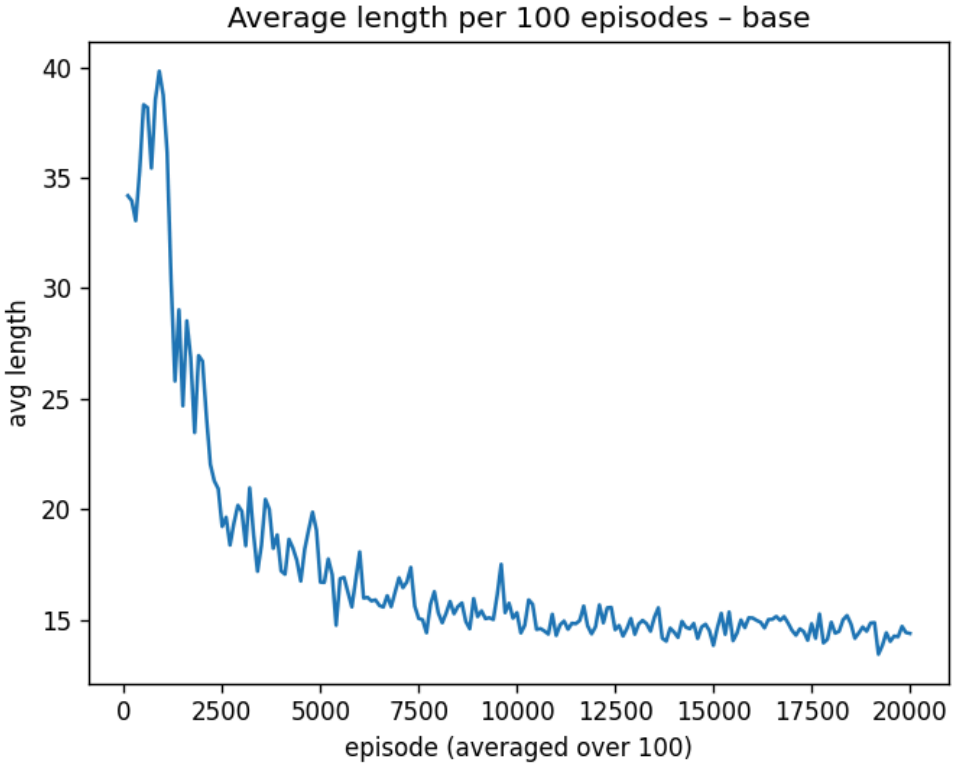
All experiments used the same starting epsilon ($\varepsilon_0 = 1.0$) and random seed (42) for reproducibility. Performance was evaluated based on:

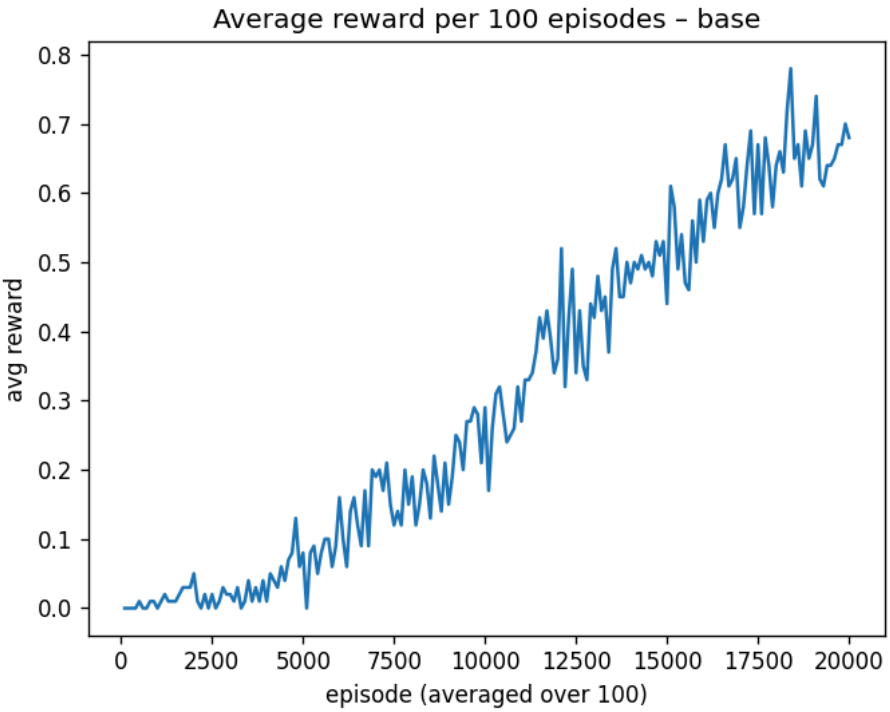
1. Reward graph
2. Average win rate per batch
3. Average length per batch
4. Final win rate during evaluation (1000 trials)

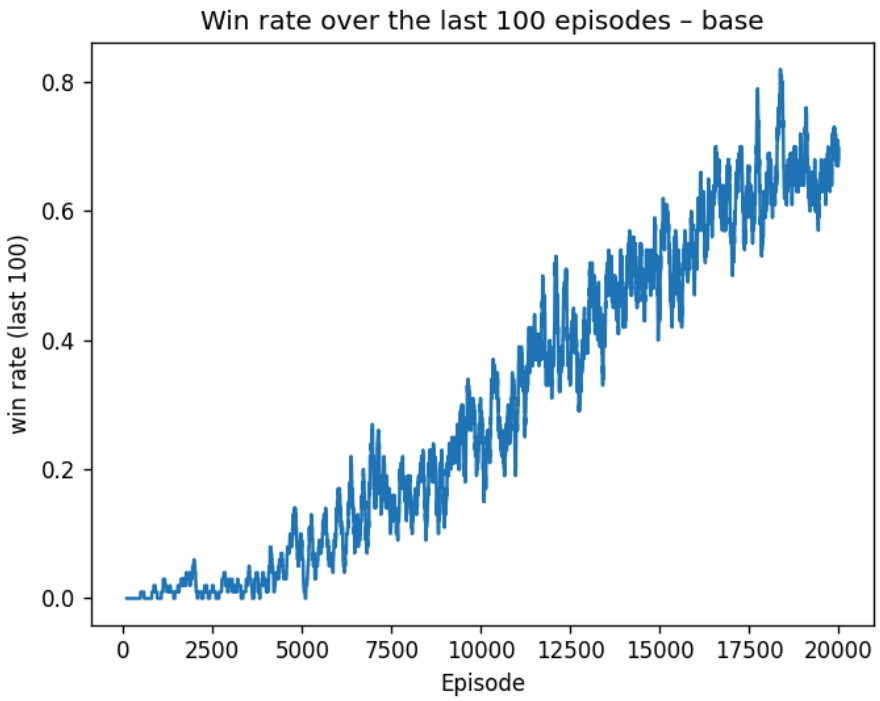
Results

Learning Rate Variation

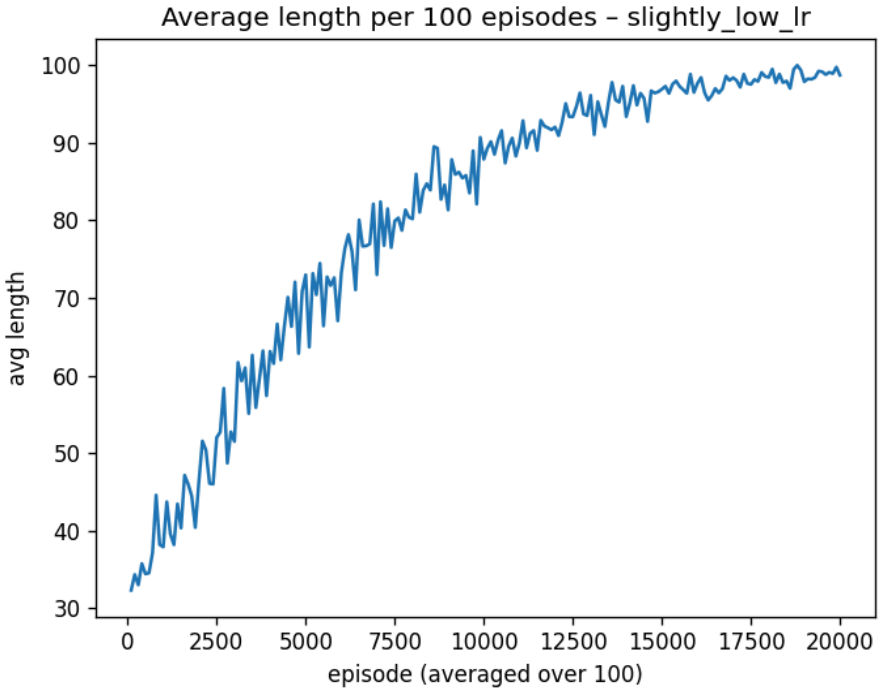
Base Case ($\alpha = 0.8$)


Average length per 100 episodes – base
avg length
40
35
30
25
20
15
0
2500
5000
7500
10000
12500
15000
17500
20000
episode (averaged over 100)

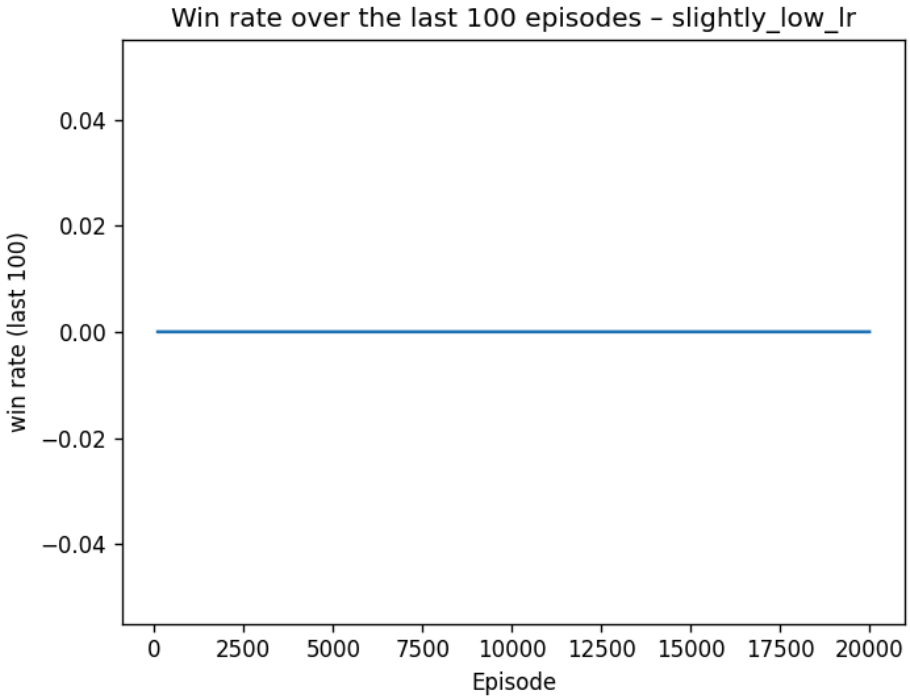

Average reward per 100 episodes – base
avg reward
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
2500
5000
7500
10000
12500
15000
17500
20000
episode (averaged over 100)



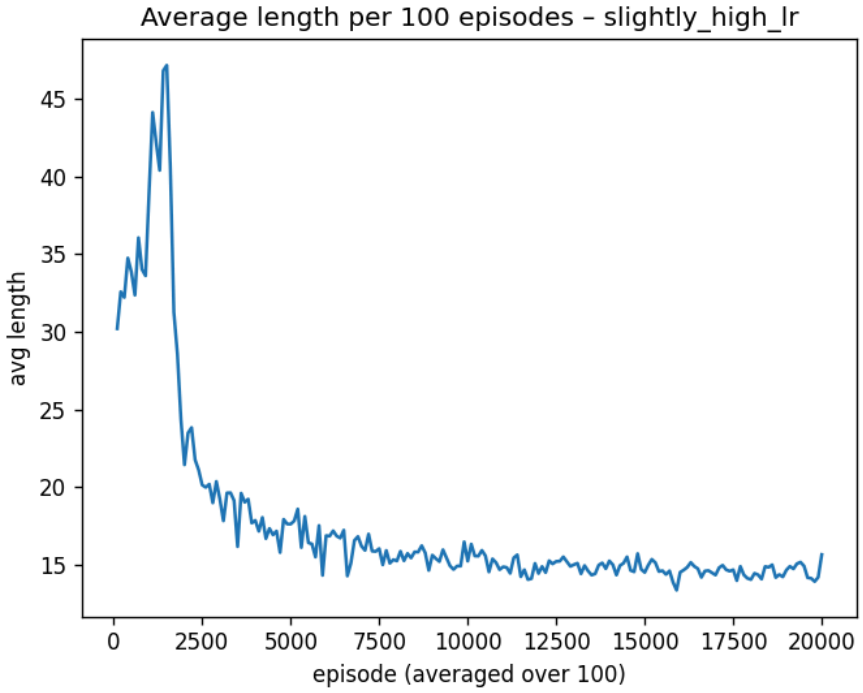
Slightly Lower Learning Rate (α = 0.7)

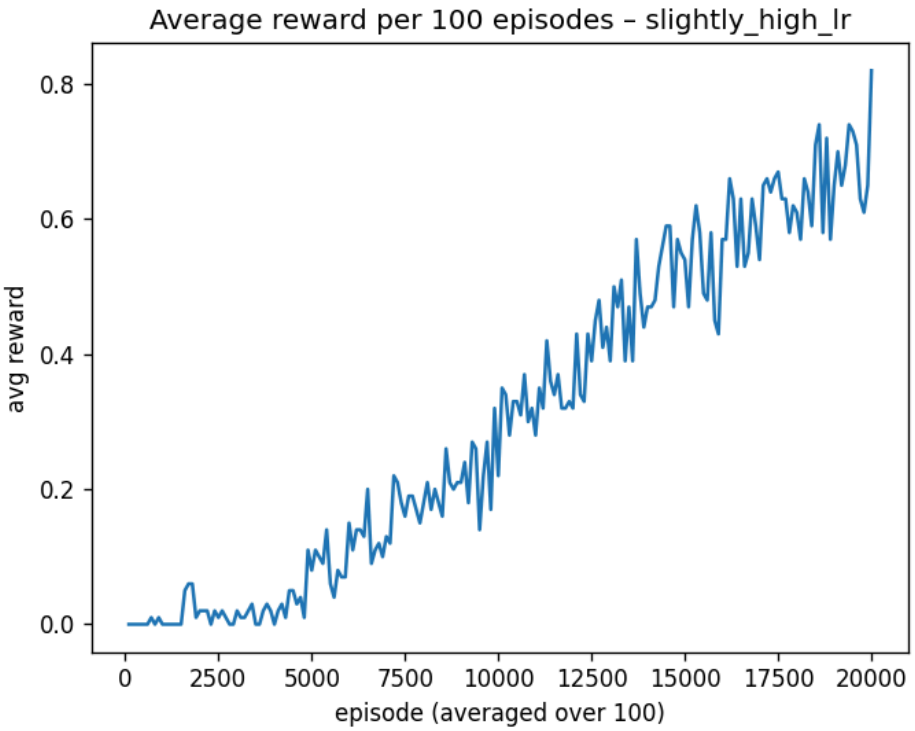

Average length per 100 episodes – slightly_low_lr
avg length
100
90
80
70
60
50
40
30
0
2500
5000
7500
10000
12500
15000
17500
20000
episode (averaged over 100)

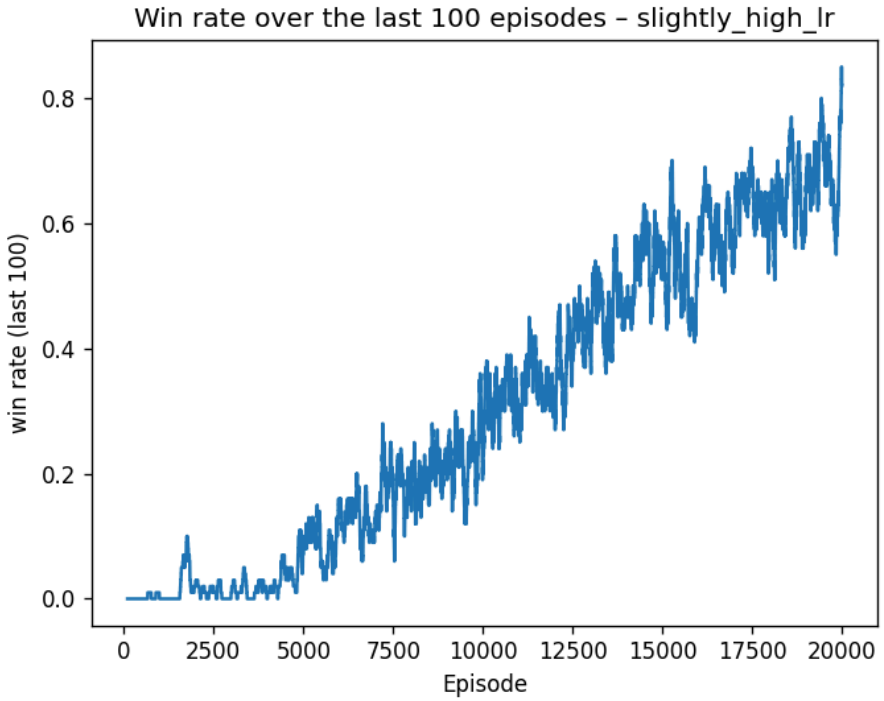

Average reward per 100 episodes – slightly_low_lr
avg reward
0.04
0.02
0.00
−0.02
−0.04
0
2500
5000
7500
10000
12500
15000
17500
20000
episode (averaged over 100)



Slightly Higher Learning Rate (α = 0.9)


Average length per 100 episodes – slightly_high_lr
avg length
45
40
35
30
25
20
15
0
2500
5000
7500
10000
12500
15000
17500
20000
episode (averaged over 100)


Average reward per 100 episodes – slightly_high_lr
avg reward
0.8
0.6
0.4
0.2
0.0
0
2500
5000
7500
10000
12500
15000
17500
20000
episode (averaged over 100)

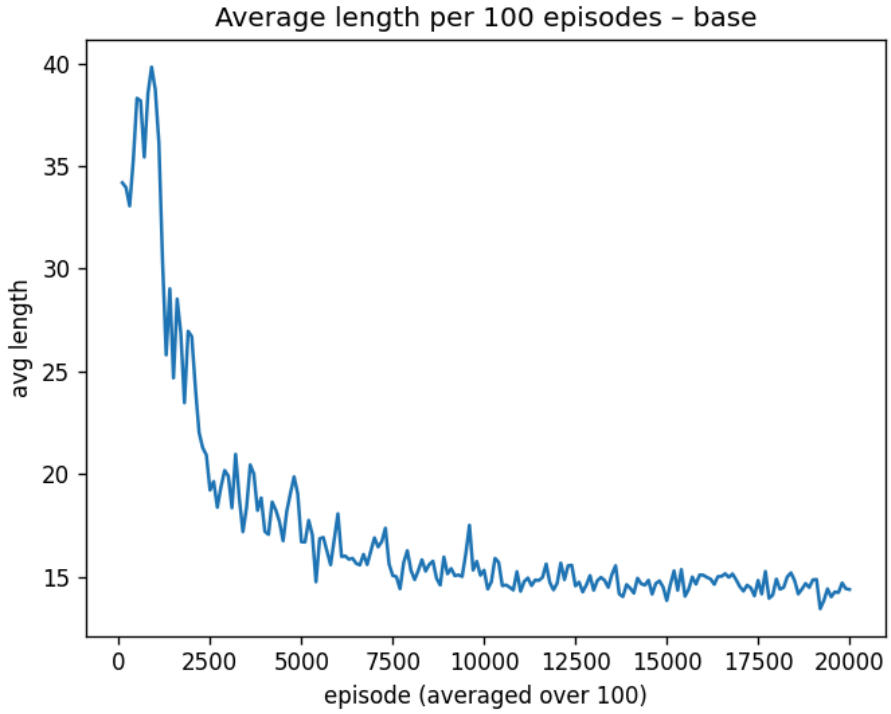


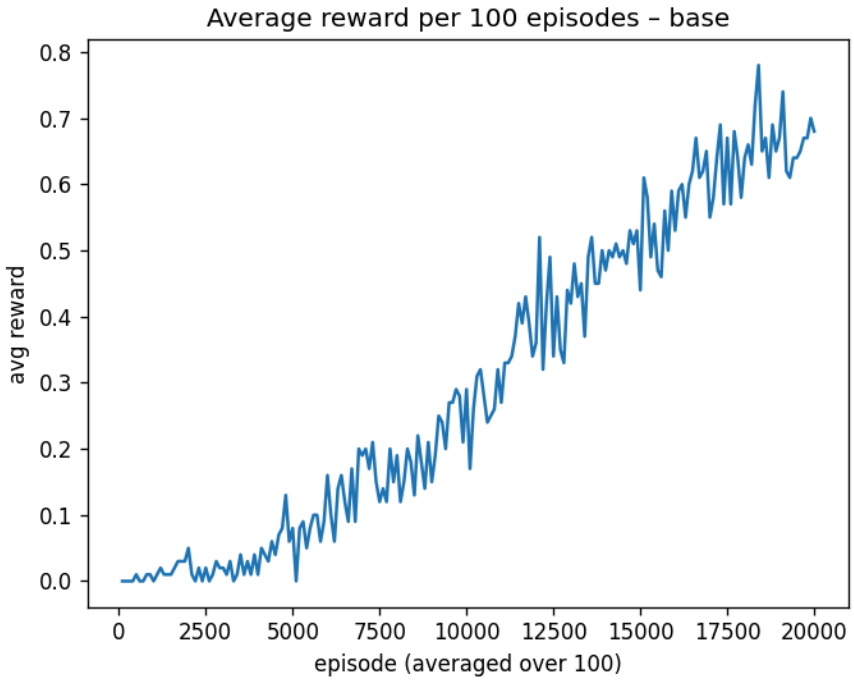
The learning rate had a dramatic effect on algorithm performance. The base case (α = 0.8) achieved a 100% win rate during evaluation with an average episode length of 14 steps. The slightly higher learning rate (α = 0.9) performed similarly, also reaching a 100% win rate during evaluation.

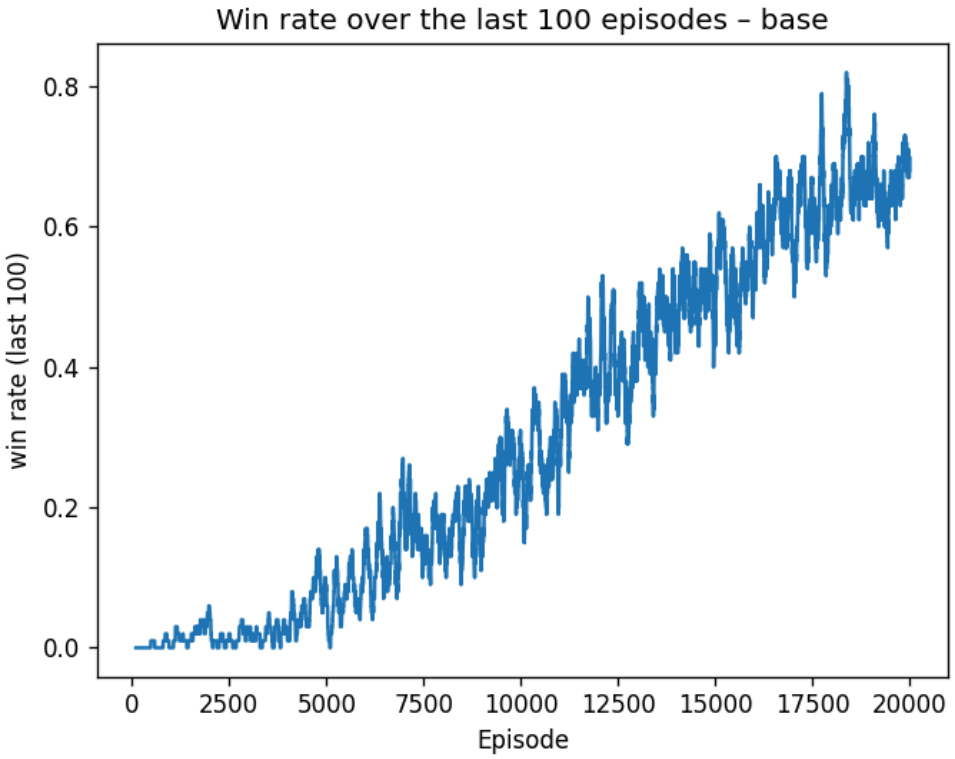
However, the slightly lower learning rate (α = 0.7) failed completely, with a 0% win rate and consistently reached the maximum episode length, indicating that the agent was unable to learn an effective policy. This suggests that in this environment, having a sufficiently high learning rate is critical for successful learning.

Epsilon Decay Rate Variation

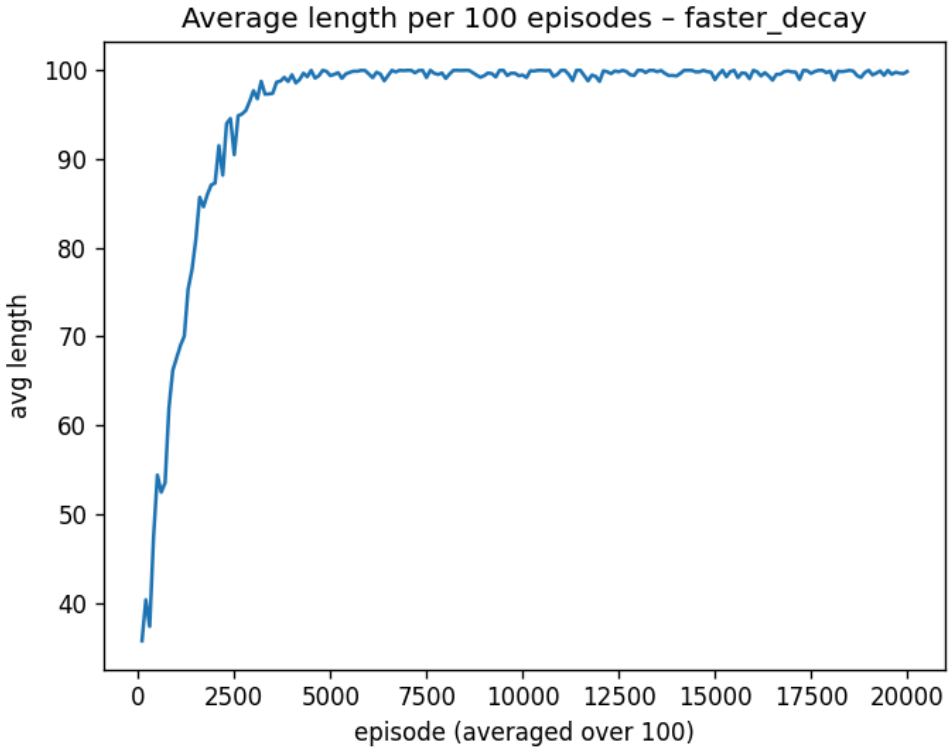
Base Case (decay = 0.9999)

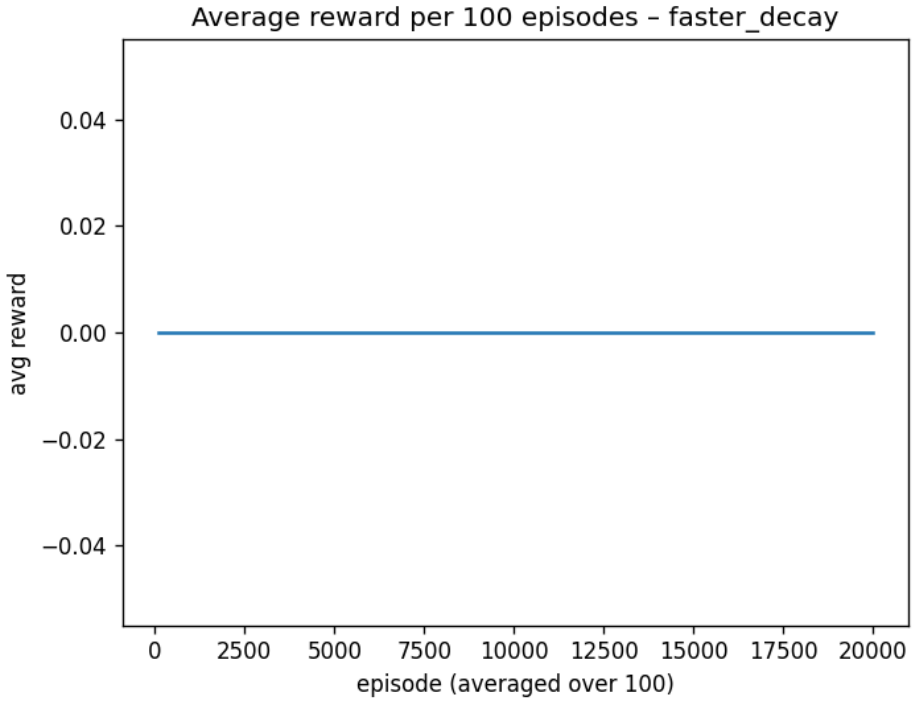

Average length per 100 episodes – base
avg length
40
35
30
25
20
15
0
2500
5000
7500
10000
12500
15000
17500
20000
episode (averaged over 100)

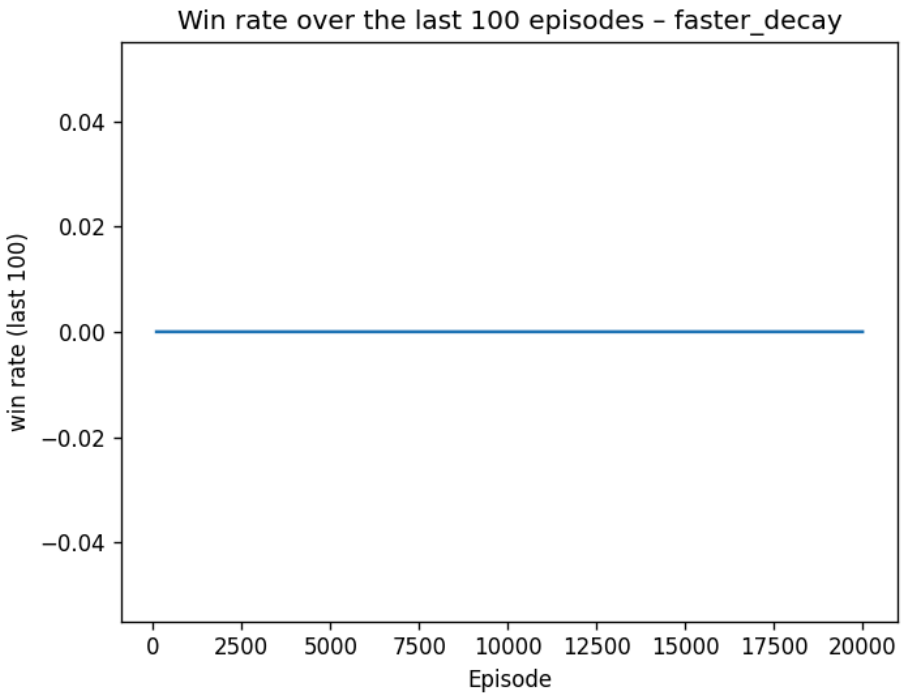

Average reward per 100 episodes – base
avg reward
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
2500
5000
7500
10000
12500
15000
17500
20000
episode (averaged over 100)



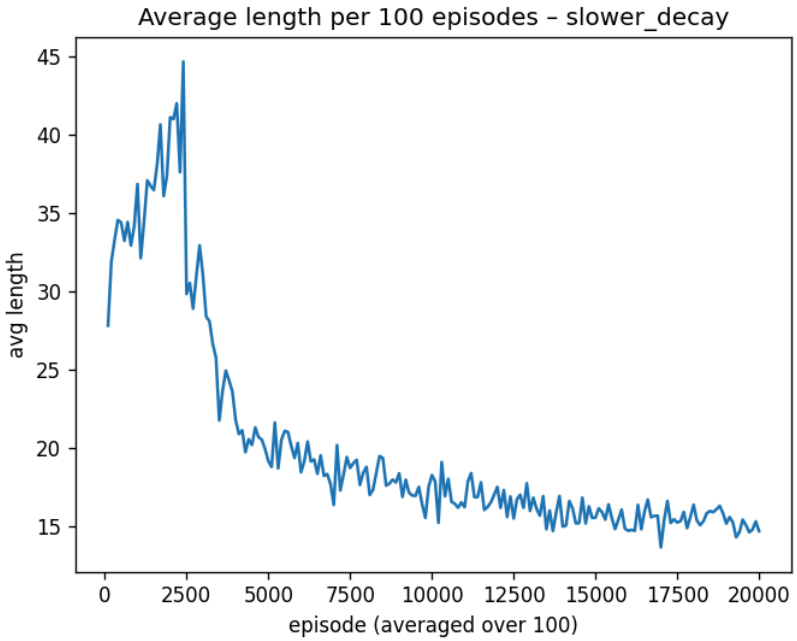
Faster Decay (decay = 0.9995)

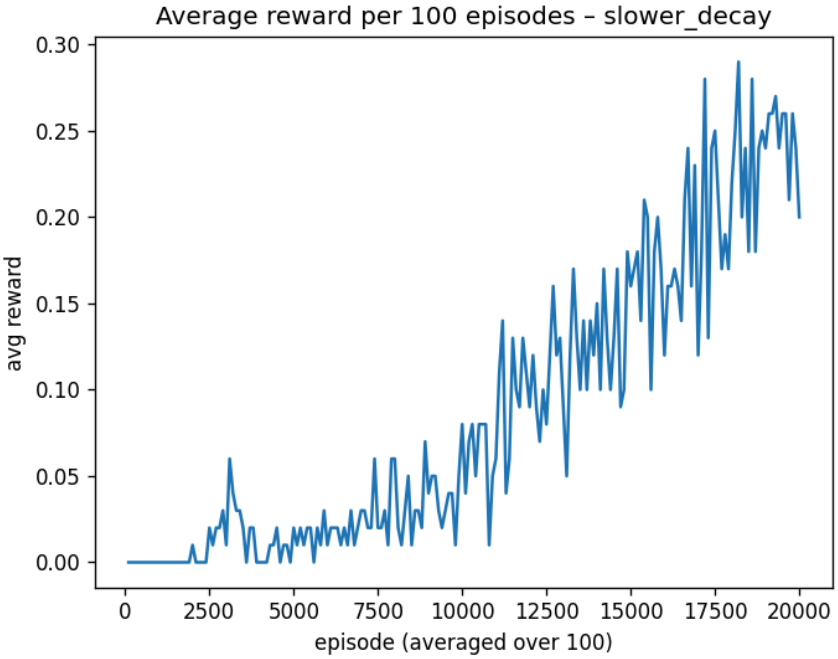

Average length per 100 episodes – faster_decay
avg length
100
90
80
70
60
50
40
0
2500
5000
7500
10000
12500
15000
17500
20000
episode (averaged over 100)

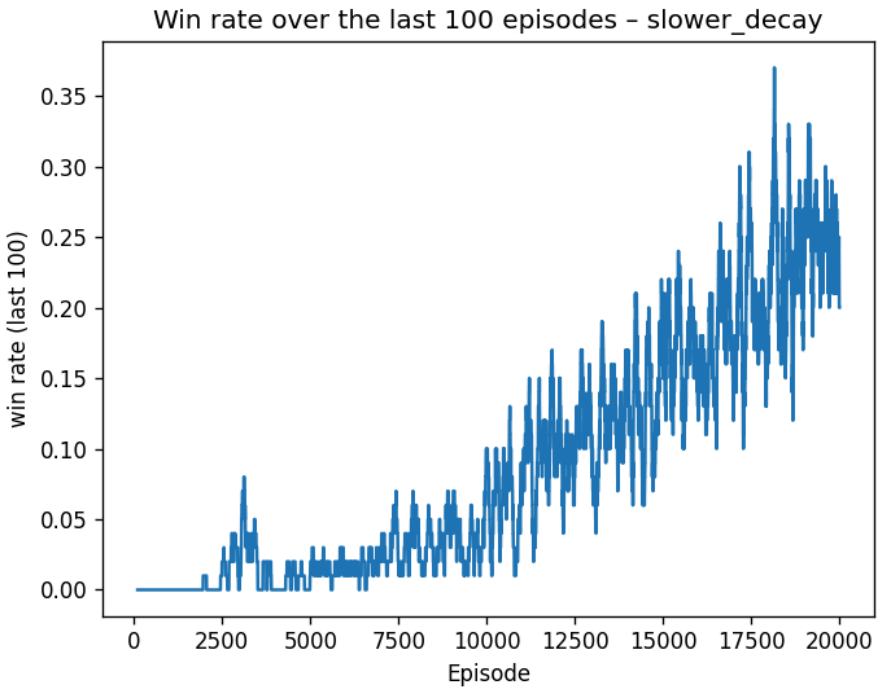

Average reward per 100 episodes – faster_decay
avg reward
0.04
0.02
0.00
−0.02
−0.04
0
2500
5000
7500
10000
12500
15000
17500
20000
episode (averaged over 100)



Slower Decay (decay = 0.99995)


Average length per 100 episodes – slower_decay
avg length
episode (averaged over 100)
45
40
35
30
25
20
15
0
2500
5000
7500
10000
12500
15000
17500
20000


Average reward per 100 episodes – slower_decay
avg reward
episode (averaged over 100)
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
2500
5000
7500
10000
12500
15000
17500
20000



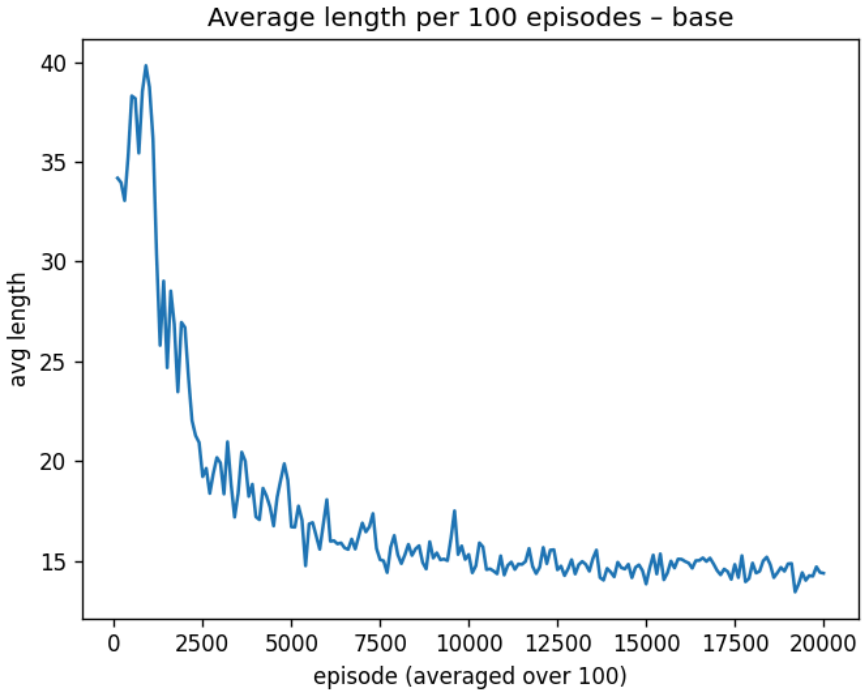
The epsilon decay rate significantly influenced exploration behavior and learning outcomes. The base decay rate (0.9999) allowed sufficient exploration while gradually transitioning to exploitation, resulting in successful learning.

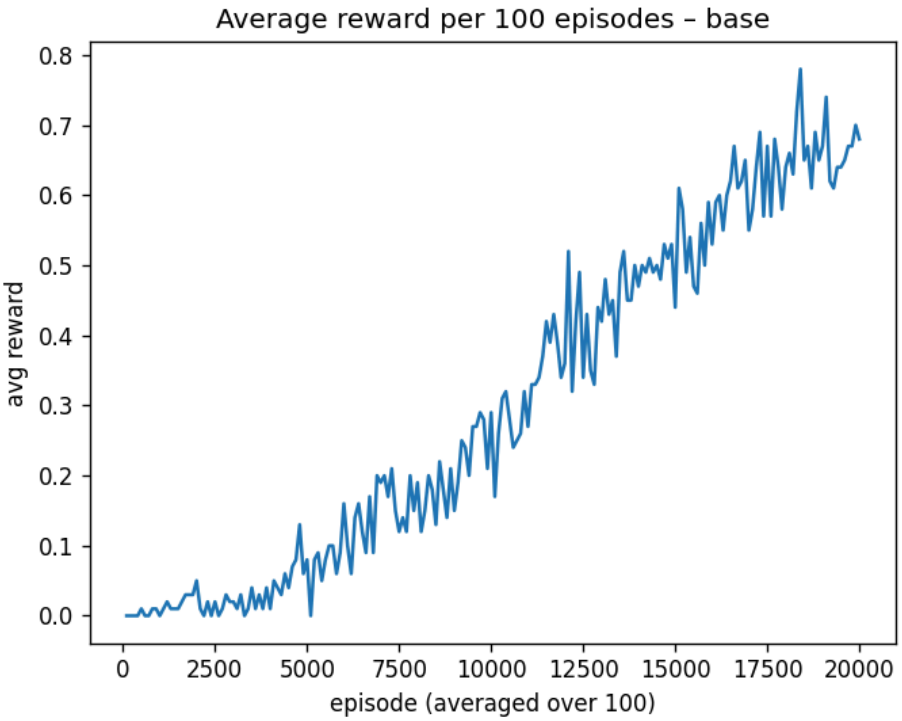
Interestingly, the faster decay rate (0.9995) led to complete failure, likely because exploration was curtailed too quickly before the agent could discover effective paths to the goal. The agent reached the minimum epsilon value (0.1) around episode 4000, much earlier than in other experiments.

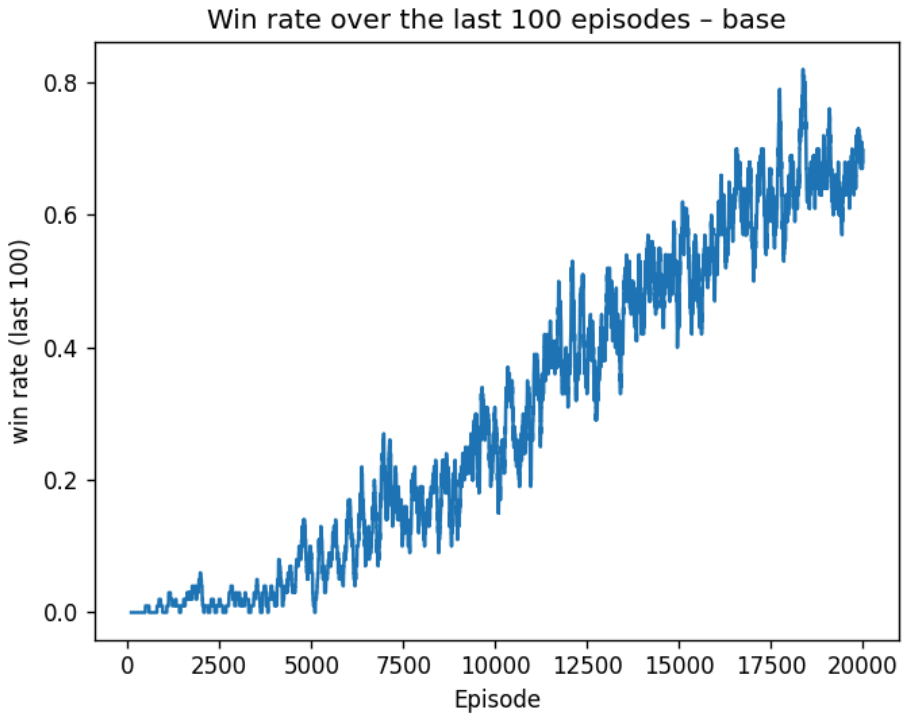
The slower decay rate (0.99995) still achieved a 100% win rate during evaluation but showed much slower convergence during training, with average rewards reaching only 0.27 by episode 20000, compared to 0.708 in the base case. This demonstrates that maintaining higher exploration rates for longer can delay convergence but still lead to optimal policies eventually.

Discount Factor Variation

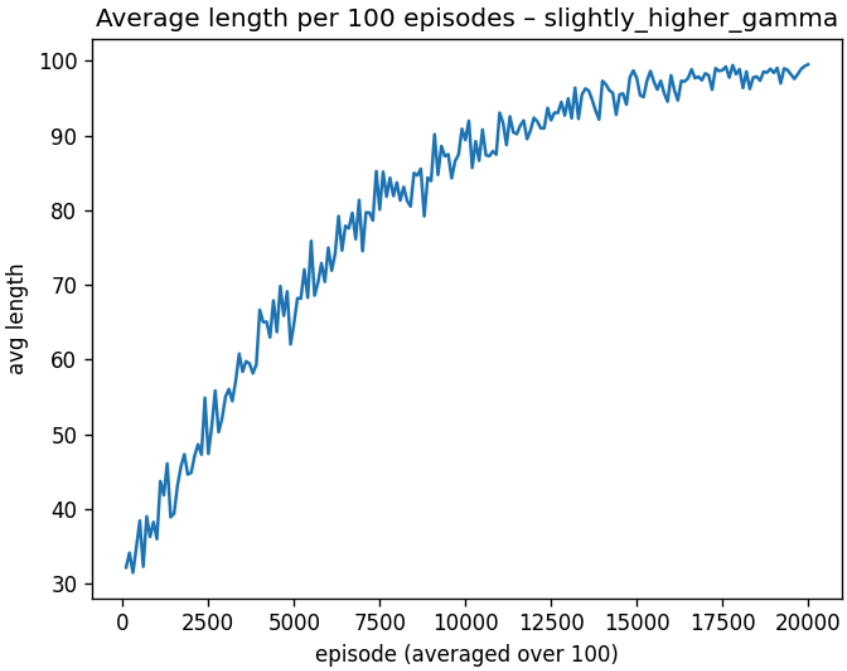
Base Case (γ = 0.99)

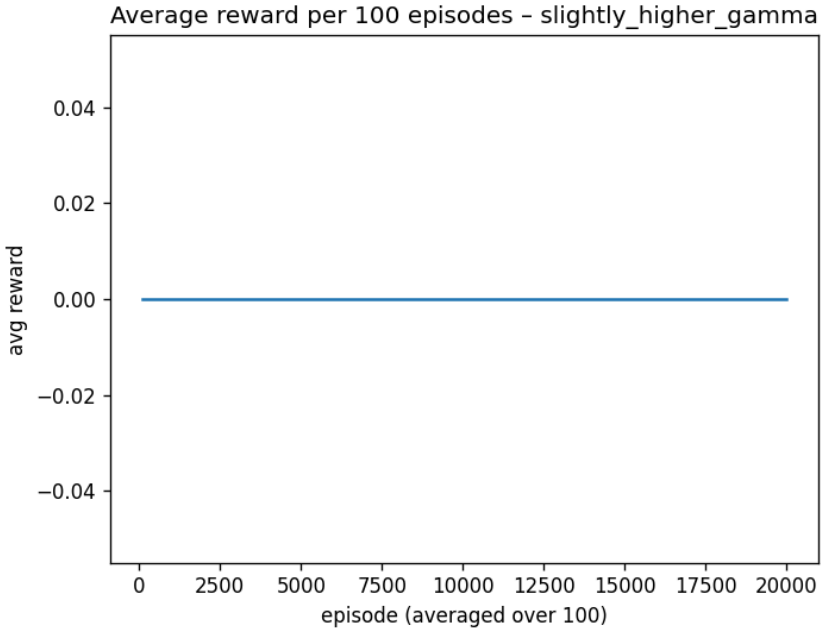

Average length per 100 episodes – base
avg length
40
35
30
25
20
15
0
2500
5000
7500
10000
12500
15000
17500
20000
episode (averaged over 100)

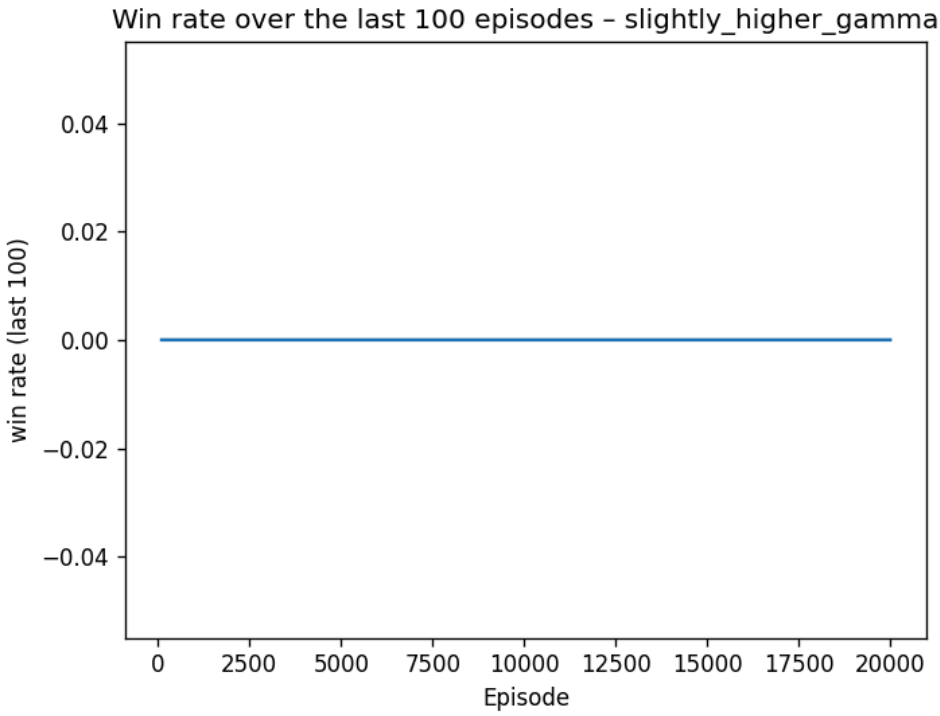

Average reward per 100 episodes – base
avg reward
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
2500
5000
7500
10000
12500
15000
17500
20000
episode (averaged over 100)



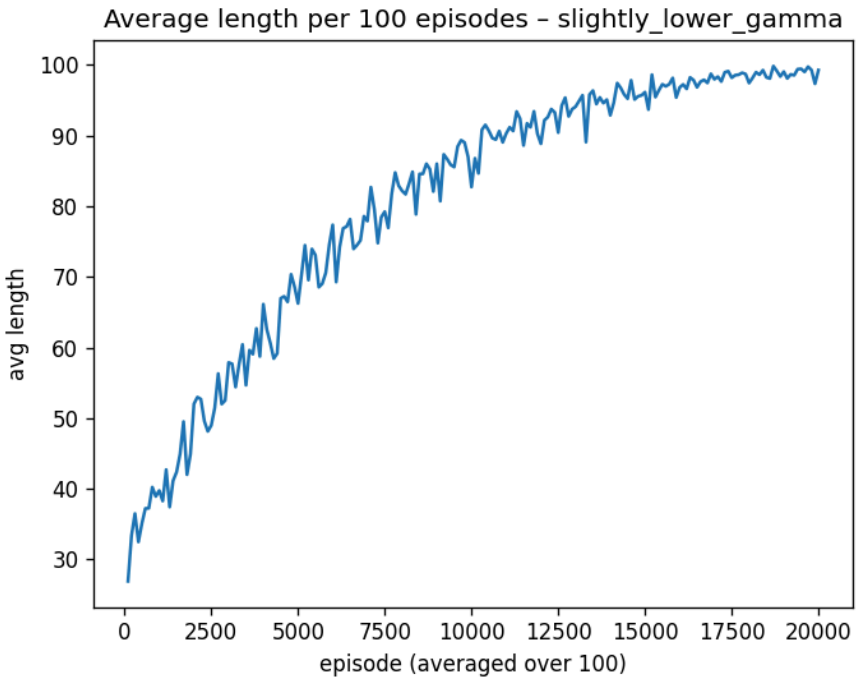
Slightly Higher Discount Factor (γ = 0.995)

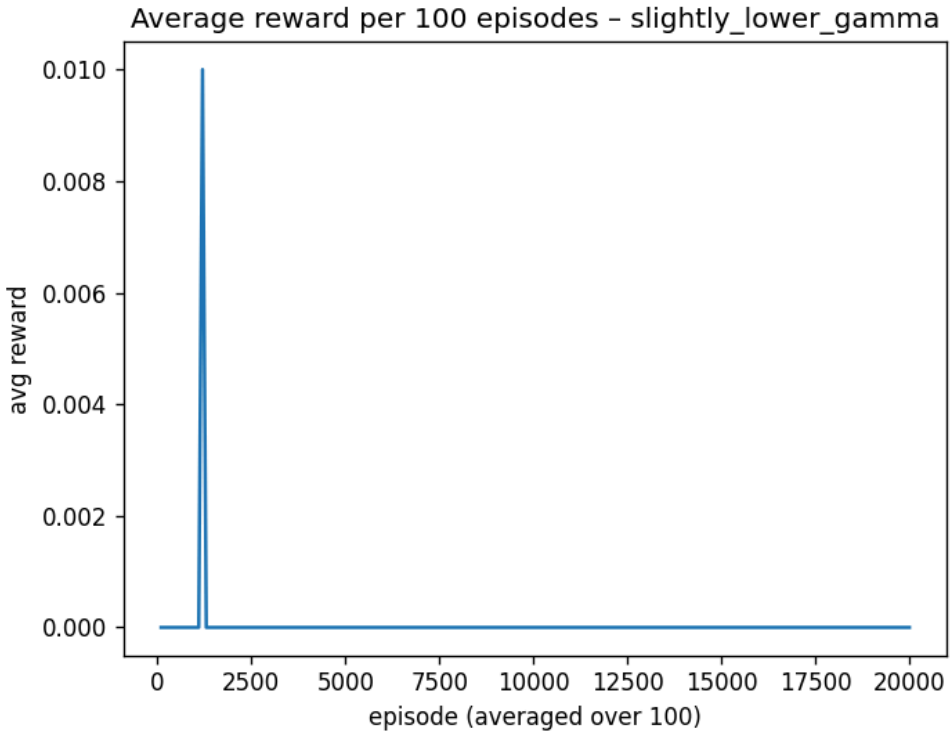

Average length per 100 episodes – slightly_higher_gamma
avg length
100
90
80
70
60
50
40
30
0
2500
5000
7500
10000
12500
15000
17500
20000
episode (averaged over 100)

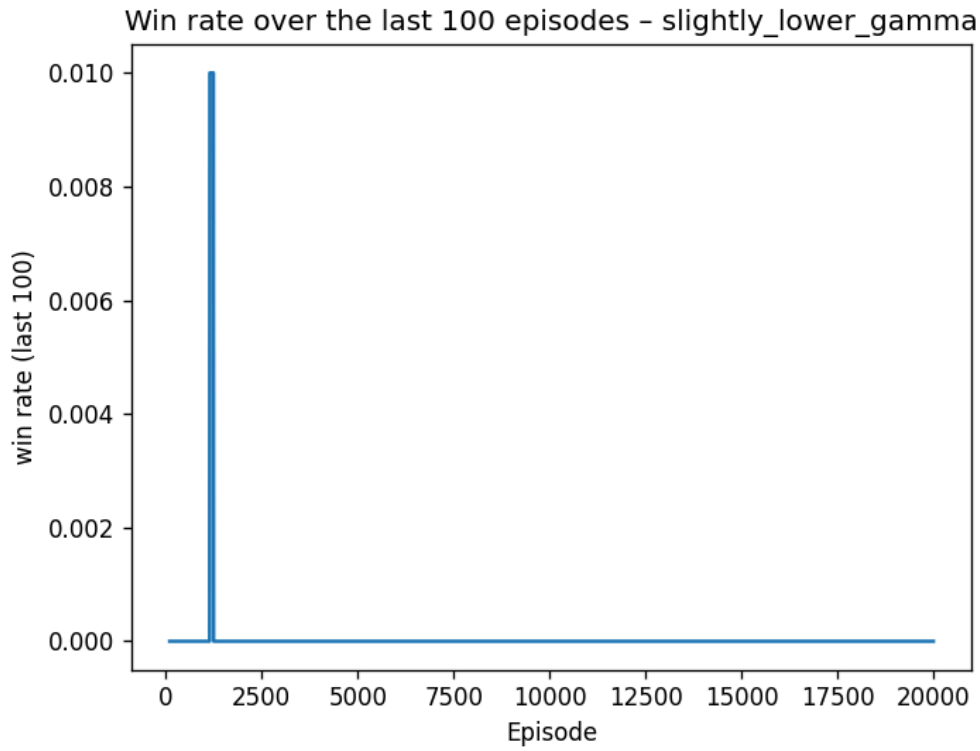

Average reward per 100 episodes – slightly_higher_gamma
avg reward
0.04
0.02
0.00
−0.02
−0.04
0
2500
5000
7500
10000
12500
15000
17500
20000
episode (averaged over 100)



Slightly Lower Discount Factor (γ = 0.98)


Average length per 100 episodes – slightly_lower_gamma
avg length
100
90
80
70
60
50
40
30
0
2500
5000
7500
10000
12500
15000
17500
20000
episode (averaged over 100)


Average reward per 100 episodes – slightly_lower_gamma
avg reward
0.010
0.008
0.006
0.004
0.002
0.000
0
2500
5000
7500
10000
12500
15000
17500
20000
episode (averaged over 100)



The discount factor determines how much the agent values future rewards. Interestingly, both the slightly higher (0.995) and slightly lower (0.98) discount factors resulted in complete learning failure (0% win rate) compared to the base case (0.99) which achieved perfect performance.

The results suggest that the optimal policy in this specific environment requires a precise valuation of future rewards, with γ = 0.99 striking the right balance.a

Key Findings

1. **Hyperparameter Sensitivity**: The Q-learning algorithm exhibits extreme sensitivity to hyperparameters in the FrozenLake-8x8 environment. Only specific combinations led to successful learning, with even small deviations causing complete failure.
2. **Critical Learning Rate Threshold**: There appears to be a critical threshold for the learning rate between 0.7 and 0.8, below which learning fails to occur effectively.
3. **Balanced Exploration-Exploitation Trade-off**: The results emphasize the importance of a properly balanced exploration-exploitation schedule. Decaying exploration too quickly or maintaining too much persistent exploration both led to suboptimal performance.
4. **Precise Discount Factor Required**: The discount factor showed surprising sensitivity, with the apparently "ideal" value of 0.99 outperforming both slightly higher and lower values significantly.

5. **Convergence Patterns**: Successful configurations showed a characteristic pattern of increasing rewards and decreasing episode lengths.

Conclusion

This study demonstrates the critical importance of careful hyperparameter tuning for Q-learning in discrete environments. The FrozenLake-8x8 environment, despite its apparent simplicity, requires precise parameter settings to achieve optimal performance.

The base configuration (α=0.8, γ=0.99, decay=0.9999, ε_min=0.1) proved most effective, achieving a perfect 100% win rate during evaluation. This suggests that these parameters provide an optimal balance of learning rate, future reward discounting, and exploration-exploitation trade-off for this particular environment.