Assignment

Multi Agent Systems

Vișan Ionuț

This assignment explores the performance of three reinforcement learning algorithms (SARSA, Q-learning, and Double Q-learning) in navigating two grid-based environments: Gridworld A and Gridworld B. The goal is to evaluate each algorithm's ability to learn optimal paths from a start position to a goal, while dealing with different types of environmental constraints: static obstacles in Gridworld A and stochastic wind effects in Gridworld B.

1. Task 1

We vary α and ε to compare Convergence Time, Path Efficiency and Robustness. Results are visualized through comparisons between key performance metrics and learning plots.

Definitions

Steps per Episode: The total number of steps the agent takes to reach the goal in each episode. Collected to analyze learning progression.

Convergence Episode: The first episode after which the average number of steps across a fixed-size window (example - last 25 episodes) consistently remains below a predefined threshold (example - 25 steps).

Path Efficiency: The average number of steps taken to complete an episode in the final window of training episodes. Lower values indicate more efficient paths.

Robustness: The standard deviation of steps per episode within the final window. Lower values suggest more consistent performance.

Learning Curve Plot: The plot shows the number of steps taken by the agent to reach the goal in each episode.

Algorithm Comparison: To compare the algorithms, we run each of them on the same environment for the same number of episodes using identical α and ε parameters. For each, we record the metrics and the plot.

1.1. GridWorld A

```
Gridworld A
. . . . . . . . . .
. . . . . # . . . .
. . . . . # . . . .
S . . . . # . G . .
. . . . . # . . . .
. . . . . . . . . .
. . . . . . . . . .
```

1.1.1.

alpha = 0.2

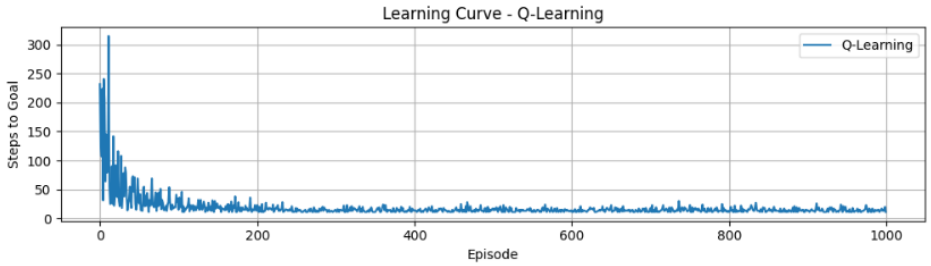
epsilon = 0.2

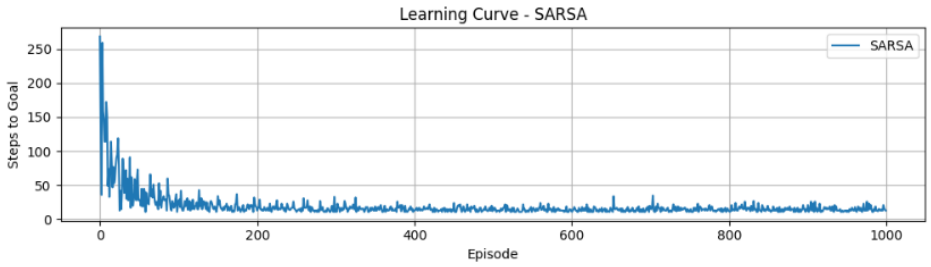
episodes = 1000

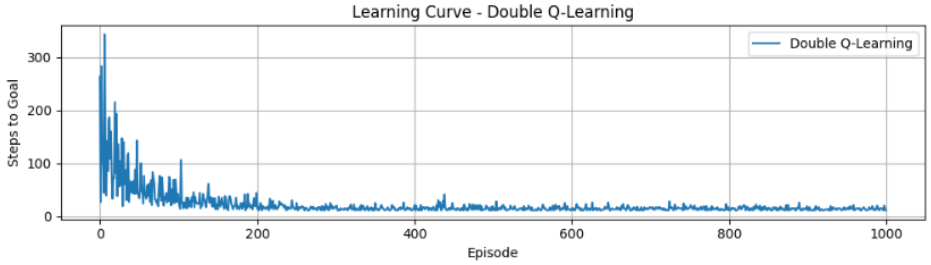
window_size = 25

convergence_threshold = 50

Algorithm	Convergence Episode	Path Efficiency	Robustness
Q-Learning	46	13.96	2.537400
SARSA	48	15.36	3.740374
Double Q-Learning	73	14.12	2.643029







Q-learning learned the fastest with stable and efficient paths, as seen in its steep learning curve. SARSA was slightly slower and more variable, reflecting its conservative, on-policy updates. Double Q-learning showed smoother but slower learning, balancing stability with caution.

1.1.2.

alpha = 0.2

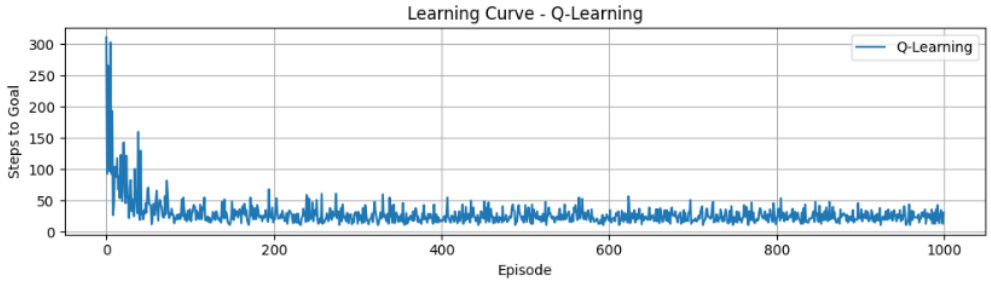
epsilon = 0.5

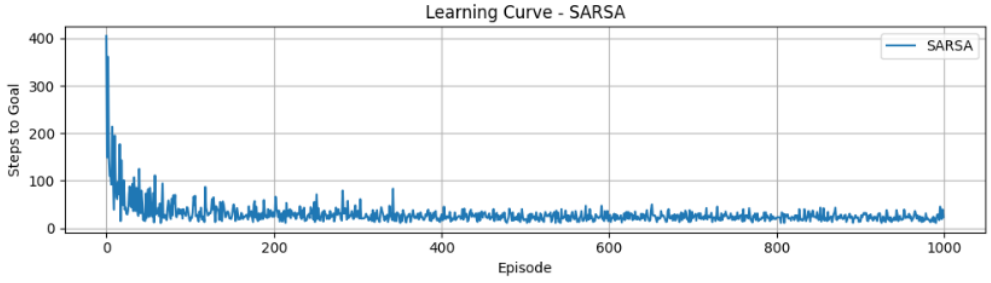
episodes = 1000

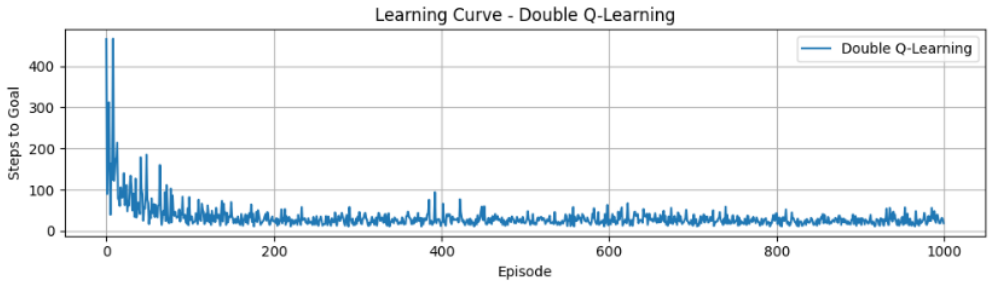
window_size = 25

convergence_threshold = 50

Algorithm	Convergence Episode	Path Efficiency	Robustness
Q-Learning	55	25.00	8.442748
SARSA	62	21.60	8.433267
Double Q-Learning	89	29.64	10.099030







With increased exploration, all algorithms converged more slowly and showed increased variability. Q-learning remained the fastest to converge, maintaining relatively stable performance. SARSA showed competitive behavior, with smoother learning and slightly more efficient paths. Double Q-learning continued to learn more cautiously, with the slowest convergence and the highest variance, but consistent progress across episodes.

1.1.3.

alpha = 0.2

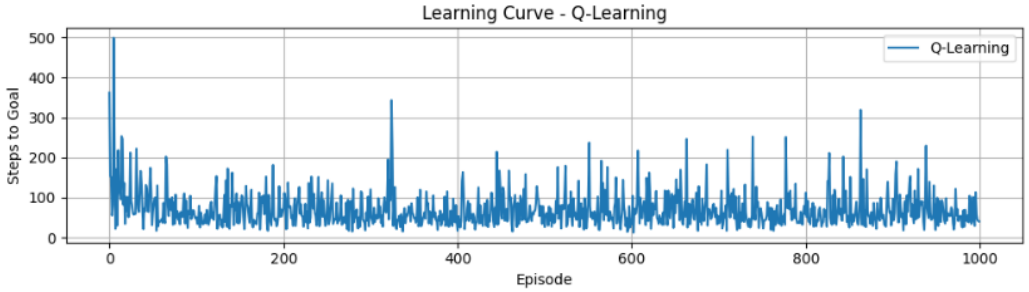
epsilon = 0.8

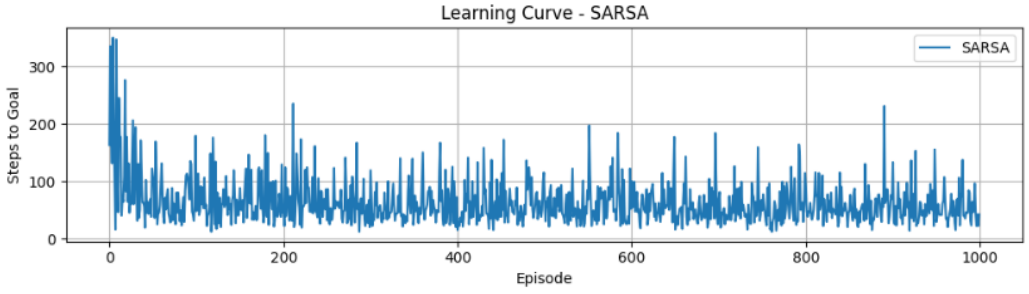
episodes = 1000

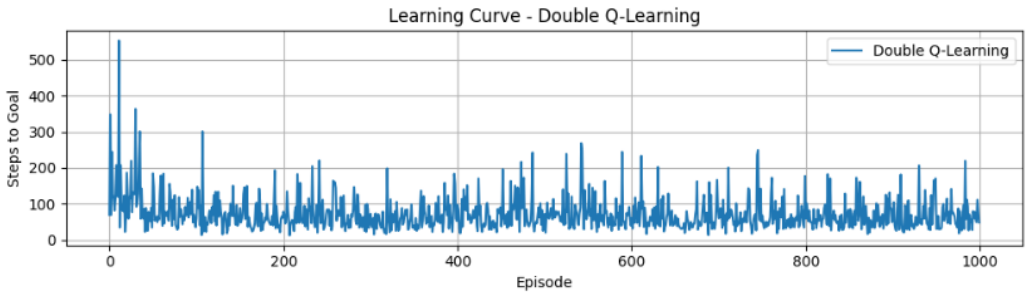
window_size = 25

convergence_threshold = 50

Algorithm	Convergence Episode	Path Efficiency	Robustness
Q-Learning	354	53.48	24.109948
SARSA	408	54.28	28.378189
Double Q-Learning	662	68.28	40.161195







With higher exploration, all algorithms showed significantly slower convergence and higher variability. Q-learning converged earlier than the others, but with noisy performance. SARSA was slightly more stable, though still inconsistent. Double Q-learning learned the slowest and had the most fluctuating behavior, struggling to stabilize within the episode limit.

Algorithm Comparison – Gridworld A (α = 0.2, varying ε)

As ε increases, exploration becomes more aggressive, impacting all algorithms. With ε = 0.2, learning was stable and efficient, showing fast convergence and low variance. At ε = 0.5, agents explored more, leading to slower convergence and increased variability. With ε = 0.8, the high exploration hindered stability, convergence was significantly delayed, and all algorithms showed noisy learning curves.

Across all ε values, Q-learning consistently converged faster, while SARSA maintained a good balance between safety and performance. Double Q-learning lagged in convergence but was more consistent at lower exploration rates. Overall, a moderate ε (e.g., 0.2–0.5) provided the best trade-off between exploration and learning stability.

1.1.4.

alpha = 0.5

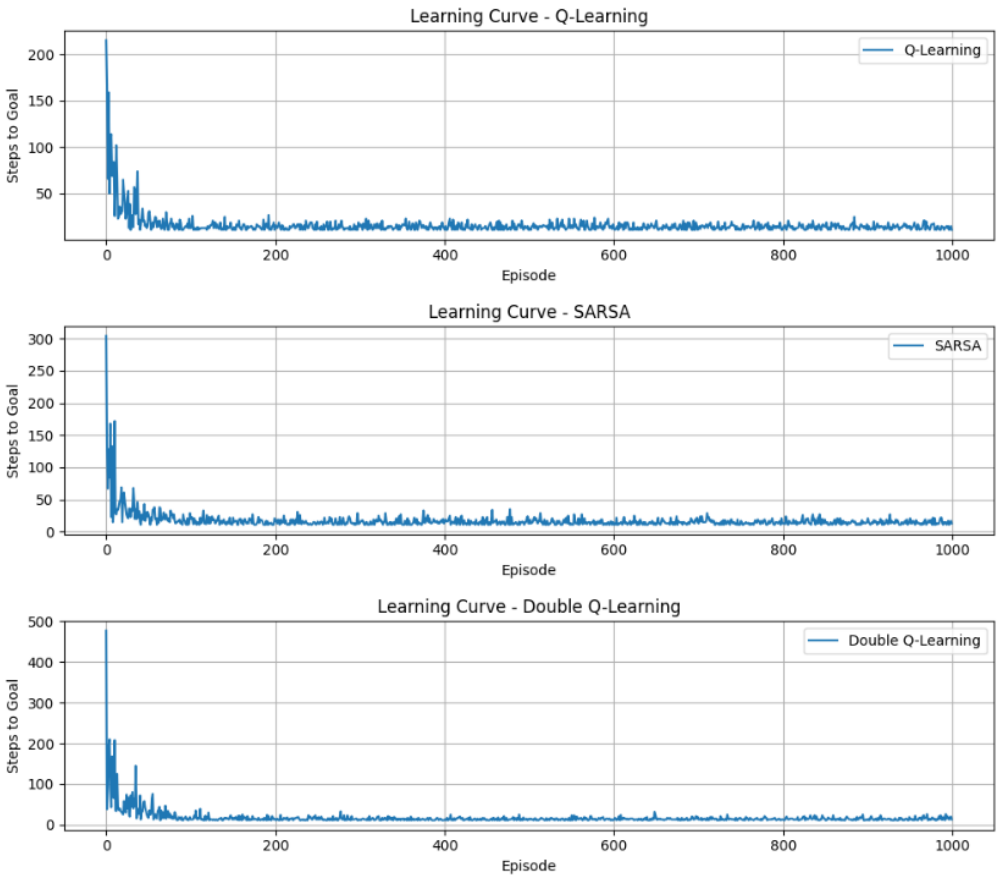
epsilon = 0.2

episodes = 1000

window_size = 25

convergence_threshold = 50

Algorithm	Convergence Episode	Path Efficiency	Robustness
Q-Learning	30	13.68	2.167395
SARSA	31	14.84	3.738235
Double Q-Learning	39	15.84	3.905688



All algorithms converged very quickly and learned efficiently. Q-learning remained the fastest and most consistent. SARSA followed closely, showing slightly higher variability. Double Q-learning converged a bit slower, with more variance, but maintained smooth progress overall.

1.1.5.

alpha = 0.5

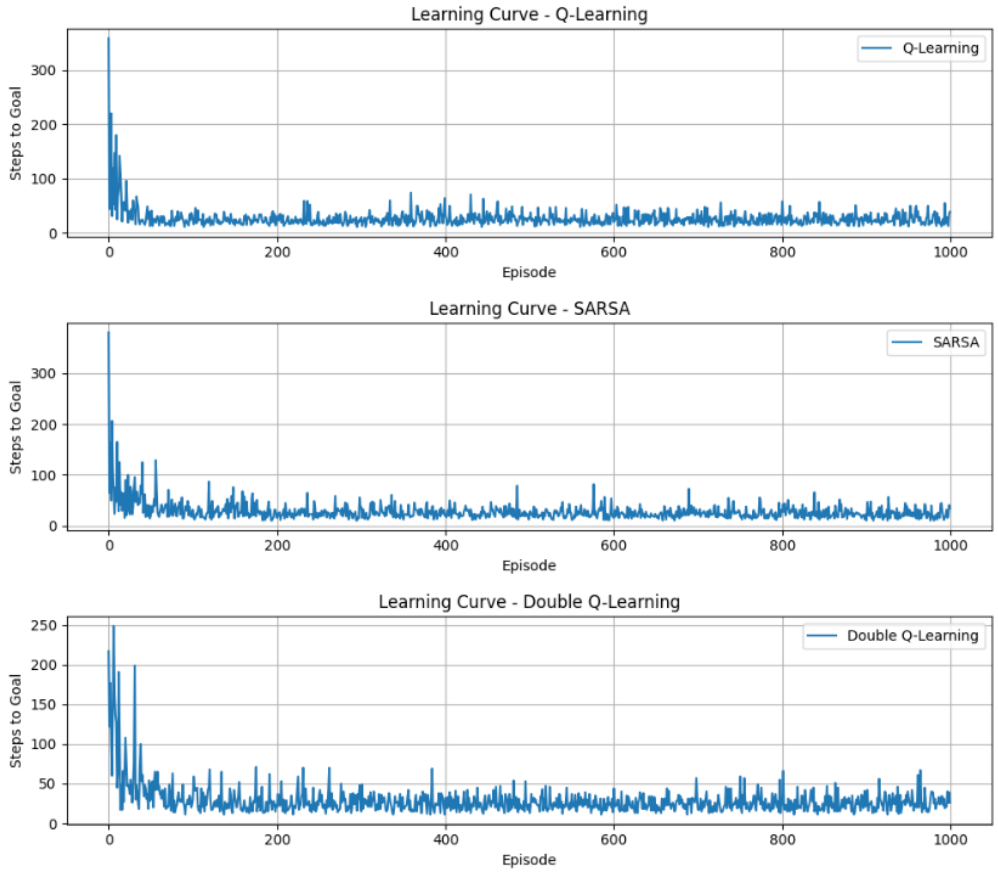
epsilon = 0.5

episodes = 1000

window_size = 25

convergence_threshold = 50

Algorithm	Convergence Episode	Path Efficiency	Robustness
Q-Learning	38	25.00	10.334409
SARSA	52	23.08	10.287546
Double Q-Learning	57	28.72	6.948496



All algorithms converged relatively quickly, with Q-learning leading in speed and SARSA achieving the most efficient paths. Double Q-learning was slightly slower but showed the most stable behavior, with less variation in later episodes.

1.1.6.

alpha = 0.5

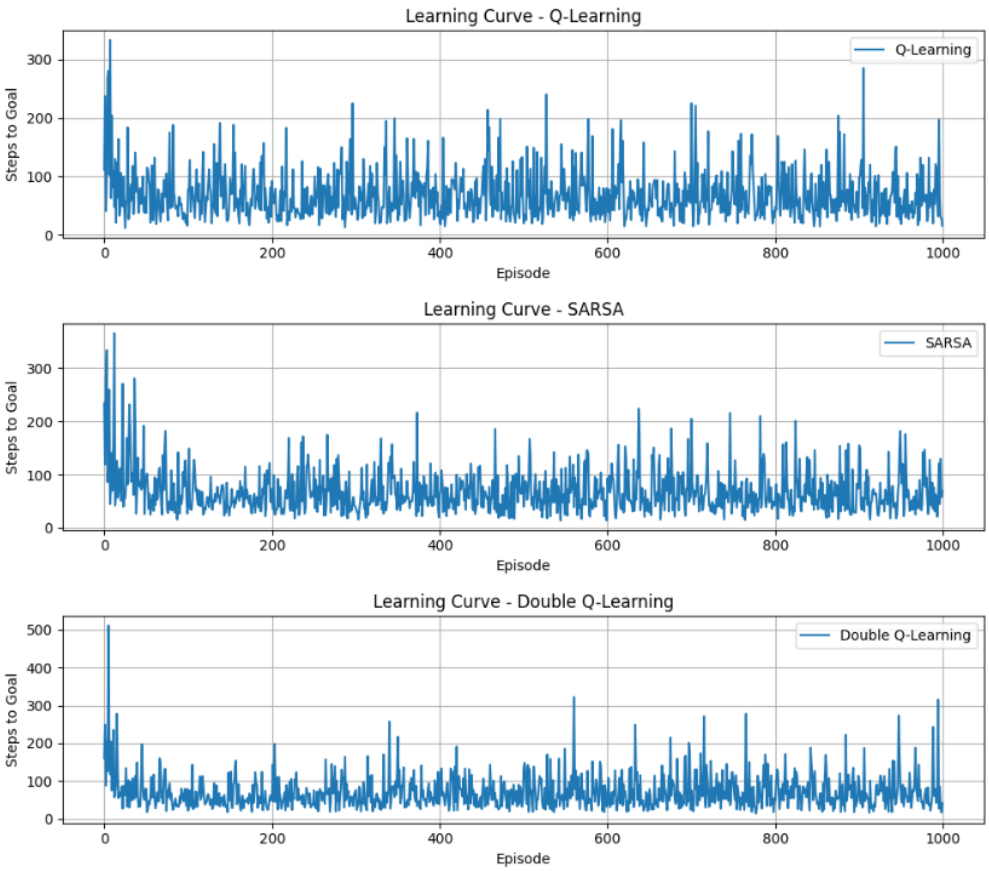
epsilon = 0.8

episodes = 1000

window_size = 25

convergence_threshold = 50

Algorithm	Convergence Episode	Path Efficiency	Robustness
Q-Learning	243	66.52	41.166122
SARSA	305	71.56	36.157522
Double Q-Learning	145	69.44	65.843499



High exploration caused all algorithms to converge later and perform less efficiently. Double Q-learning converged faster than the others, but with the most unstable behavior. Q-learning had moderate performance and stability, while SARSA showed the highest path efficiency but also frequent fluctuations in learning.

Algorithm Comparison – Gridworld A (α = 0.5, varying ε)

At ε = 0.2, all algorithms learned quickly and stably, with Q-learning leading in speed and consistency. Increasing ε to 0.5 introduced more exploration, slightly slowing convergence and increasing variability, especially for SARSA and Double Q-learning. At ε = 0.8, learning became much more unstable across all algorithms. Q-learning maintained balanced performance, while Double Q-learning converged faster but was highly erratic. SARSA remained the most efficient but struggled with stability.

1.1.7.

alpha = 0.8

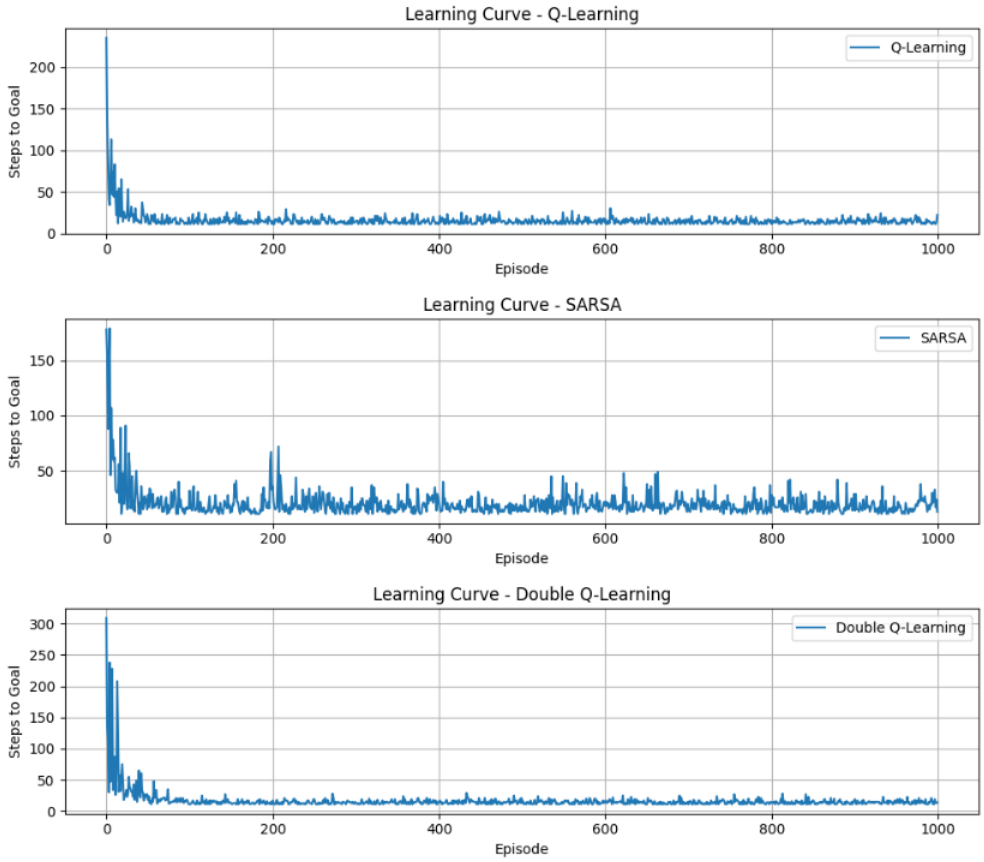
epsilon = 0.2

episodes = 1000

window_size = 25

convergence_threshold = 50

Algorithm	Convergence Episode	Path Efficiency	Robustness
Q-Learning	26	13.88	2.746926
SARSA	30	22.16	5.787435
Double Q-Learning	35	14.44	3.020993



With a high learning rate, updates were fast and decisive, while the low exploration encouraged early exploitation of good paths. Q-learning performed best, converging quickly with efficient and stable paths due to its aggressive off-policy updates. Double Q-learning followed, slightly slower but more consistent, as its decoupled updates helped avoid overestimation. SARSA was more conservative, taking longer and producing less efficient paths, as it learns based on actual behavior rather than optimal actions.

1.1.8.

alpha = 0.8

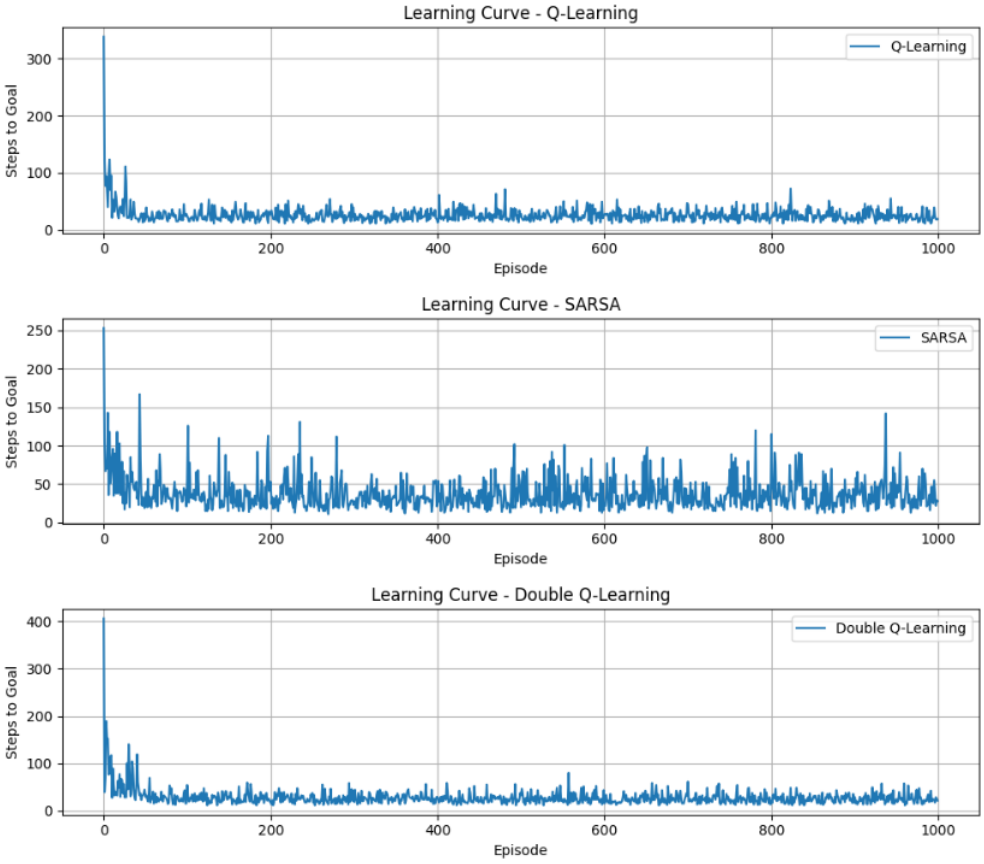
epsilon = 0.5

episodes = 1000

window_size = 25

convergence_threshold = 50

Algorithm	Convergence Episode	Path Efficiency	Robustness
Q-Learning	32	21.68	8.403428
SARSA	43	36.40	15.200000
Double Q-Learning	54	23.80	7.904429



A higher learning rate combined with moderate exploration led to fast learning but increased variability. Q-learning converged the fastest with efficient paths, though robustness was moderate. Double Q-learning was slightly slower but more stable, benefiting from reduced overestimation. SARSA, being more cautious, had the least efficient paths and highest variance, reflecting its on-policy sensitivity to exploration.

1.1.9.

alpha = 0.8

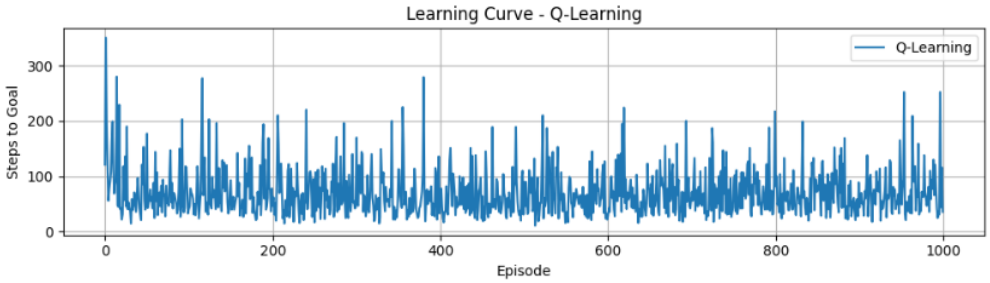
epsilon = 0.8

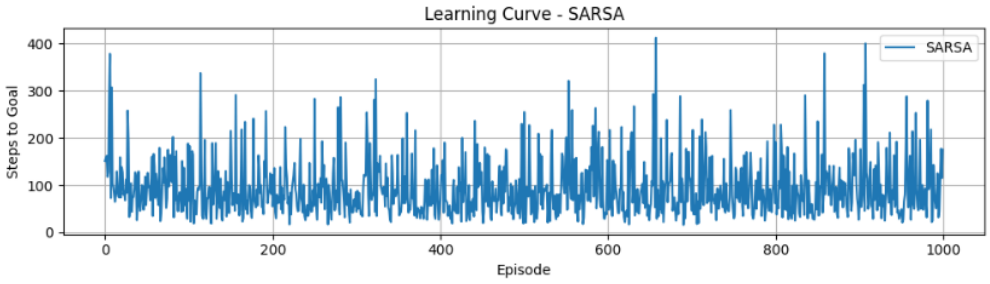
episodes = 1000

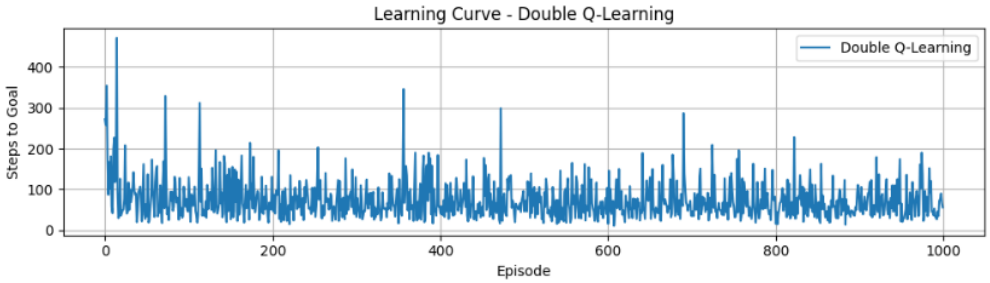
window_size = 25

convergence_threshold = 50

Algorithm	Convergence Episode	Path Efficiency	Robustness
Q-Learning	None	77.56	47.486486
SARSA	None	100.04	62.253983
Double Q-Learning	None	63.16	34.326293







With both high learning rate and high exploration, none of the algorithms reached convergence. The excessive randomness from ε = 0.8 caused agents to explore heavily, preventing them from consistently exploiting optimal paths. Double Q-learning performed relatively better in terms of efficiency and stability, likely due to its reduced overestimation bias. Q-learning showed moderate efficiency but high variance, while SARSA was the least efficient and most unstable, struggling with its on-policy strategy under high exploration.

alpha = 0.8

epsilon = 0.8

episodes = 1000

window_size = 25

convergence_threshold = 60

Algorithm	Convergence Episode	Path Efficiency	Robustness
Q-Learning	66	70.76	45.595859
SARSA	80	97.60	69.623559
Double Q-Learning	53	68.08	36.372429

Due to the increased exploration (ε = 0.8), learning became significantly more variable, making it harder for agents to maintain consistently low step counts across episodes. As a result, the original threshold (50 steps) was too strict, and none of the models converged. Raising the threshold to 60 allowed for a fairer evaluation of convergence under high-exploration conditions, reflecting realistic agent behavior in a more stochastic policy setting.

Algorithm Comparison – Gridworld A (α = 0.8, varying ε)

At low exploration (ε = 0.2), all algorithms converged quickly and stably, with Q-learning leading in efficiency. As ε increased to 0.5, learning became noisier—Q-learning and Double Q-learning remained effective, while SARSA's variance increased significantly. With ε = 0.8, exploration dominated learning; agents struggled to exploit optimal paths, requiring a relaxed convergence threshold. Even then, convergence was slower and performance less stable, especially for SARSA. Double Q-learning handled high exploration best, balancing learning speed and robustness.

Algorithm Comparison – Gridworld A (varying α, varying ε)

Across all tested configurations, Q-learning consistently achieved the fastest convergence, especially with low ε values, benefiting from its off-policy updates. However, its performance became more unstable under high exploration. SARSA showed greater variability and slower convergence, particularly at high ε, due to its on-policy nature, but occasionally produced more cautious and efficient paths in low-noise settings. Double Q-learning performed best in terms of robustness, especially in high-ε scenarios, thanks to its reduced overestimation bias. Although slower to converge in some cases, it maintained balanced learning and stability across different parameter combinations.

1.2. GridWorld B

```
Gridworld B
. . . . . . . . . .
. . . . . . . . . .
. . . . . . . . . .
S . . . . . . G . .
. . . . . . . . . .
. . . . . . . . . .
. . . . . . . . . .

0 0 0 1 1 1 2 2 1 0
```

1.2.1.

alpha = 0.2

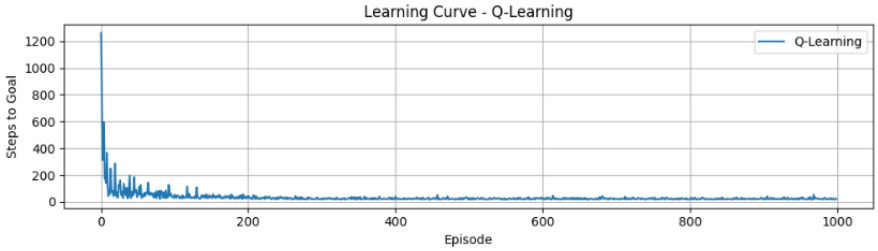
epsilon = 0.2

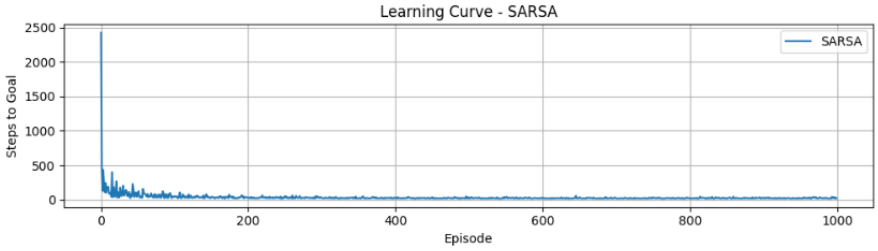
episodes = 1000

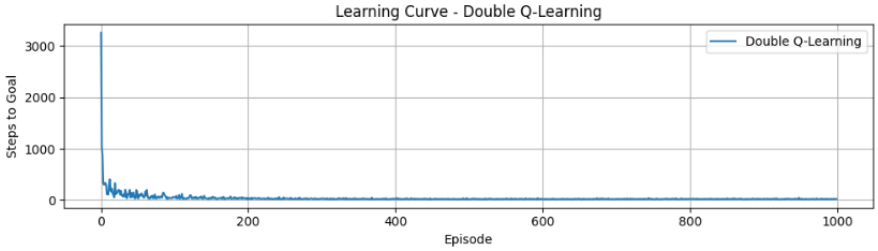
window_size = 25

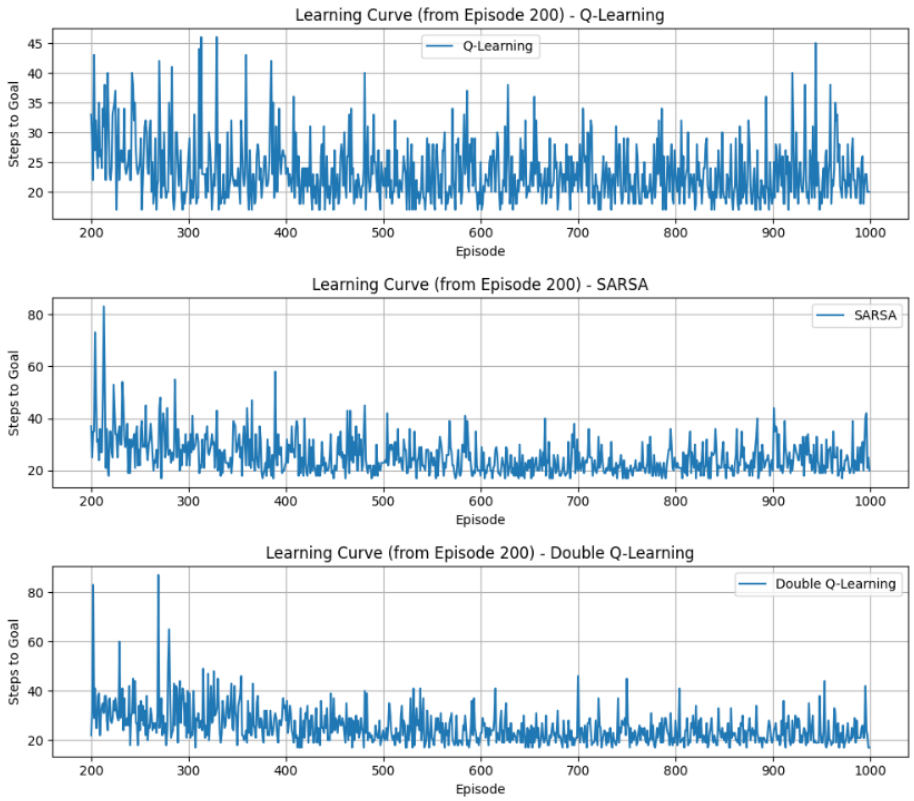
convergence_threshold = 50

Algorithm	Convergence Episode	Path Efficiency	Robustness
Q-Learning	107	21.56	3.837499
SARSA	110	25.68	6.169084
Double Q-Learning	114	21.80	3.136877









In the stochastic environment with wind dynamics, all algorithms successfully converged, though more gradually compared to Gridworld A. Q-learning showed the fastest convergence and most efficient navigation, leveraging its off-policy updates to quickly adapt to the wind-affected transitions. Double Q-learning maintained stable learning and robust performance, minimizing overestimation even in the presence of environmental noise. SARSA, being more conservative, converged slightly slower and was more sensitive to the wind effect, leading to longer paths and higher variability.

Notably, the zoomed-in learning curves (from episode 200 onward) highlight that Double Q-learning achieved smoother and more consistent performance over time, whereas SARSA exhibited more fluctuation in step counts.

1.2.2.

alpha = 0.2

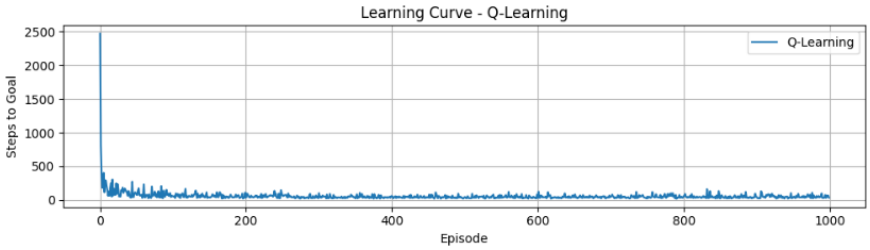
epsilon = 0.5

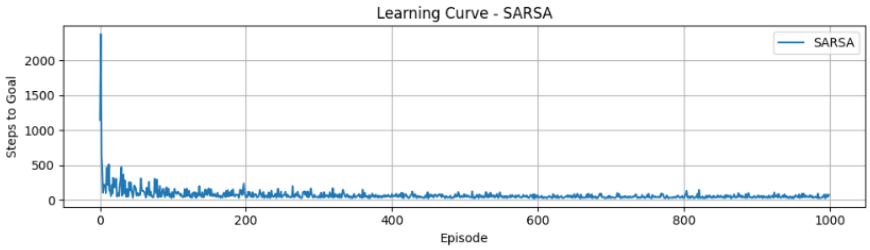
episodes = 1000

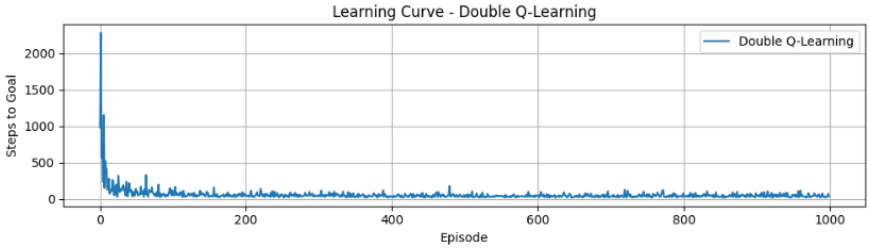
window_size = 25

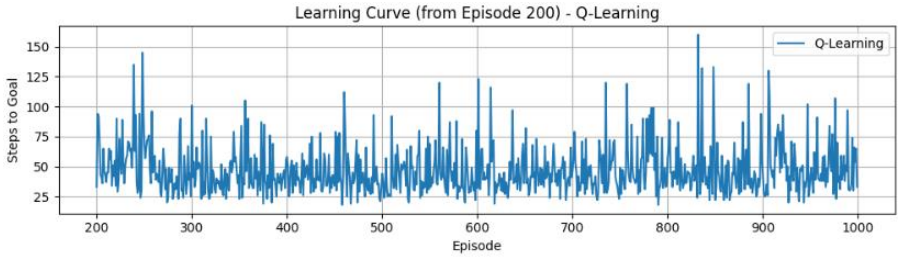
convergence_threshold = 50

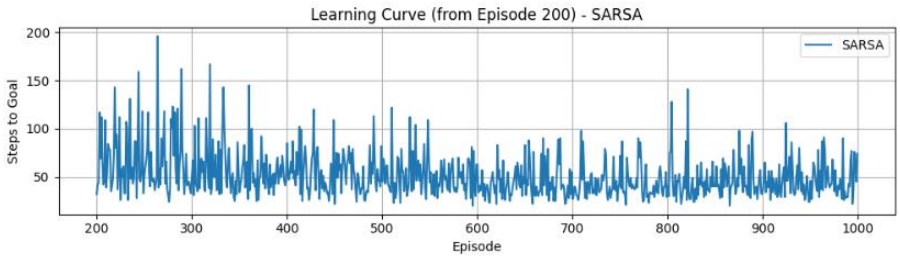
Algorithm	Convergence Episode	Path Efficiency	Robustness
Q-Learning	191	49.28	20.843263
SARSA	389	45.04	18.666505
Double Q-Learning	173	45.60	16.107141

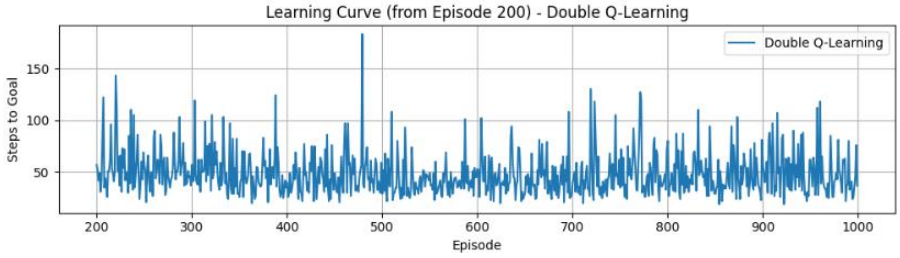












With increased exploration, learning slowed across all algorithms, but convergence was still achieved. Double Q-learning converged fastest, likely due to its ability to stabilize learning in the noisy wind environment by reducing overestimation. Q-learning followed closely, but exhibited slightly higher variance. SARSA, although stable in some settings, was the slowest to converge and showed the highest path inefficiency under these conditions. From episode 200 onward, Double Q-learning maintained the most consistent performance, while SARSA struggled with greater fluctuations.

1.2.3.

alpha = 0.2

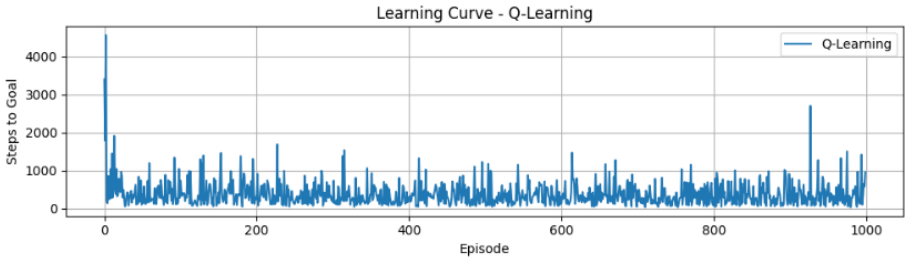
epsilon = 0.8

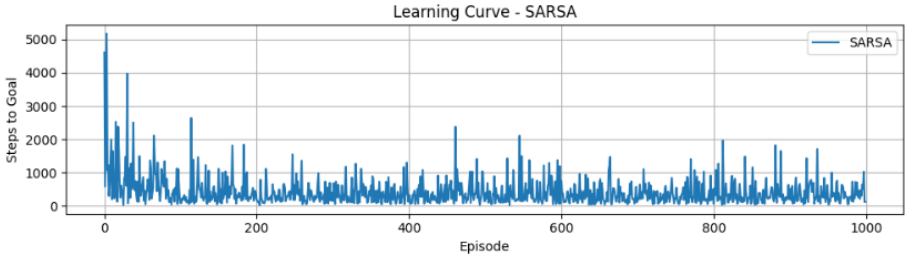
episodes = 1000

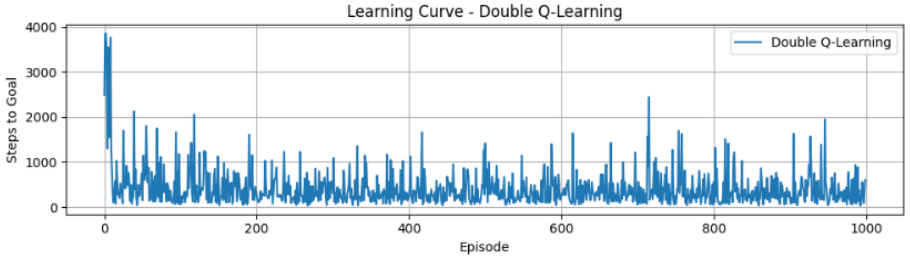
window_size = 25

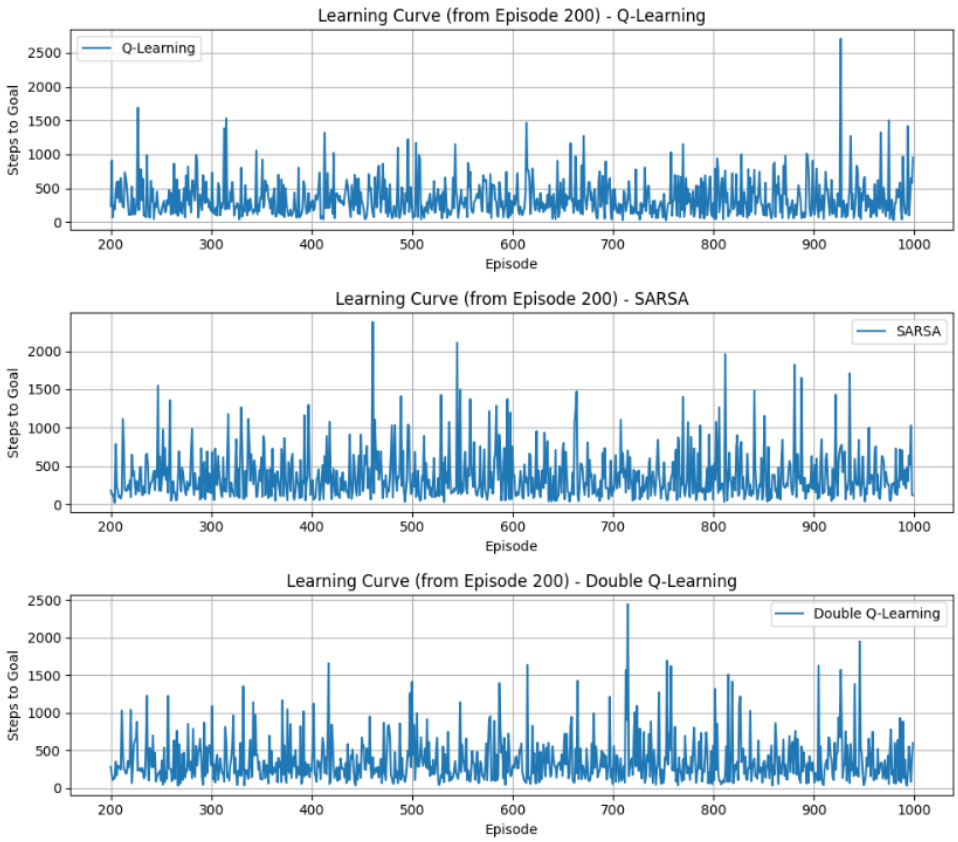
convergence_threshold = 50

Algorithm	Convergence Episode	Path Efficiency	Robustness
Q-Learning	None	427.56	395.262857
SARSA	None	360.12	241.074150
Double Q-Learning	None	339.40	273.652919









High exploration significantly impacted learning stability. None of the algorithms reached convergence, as agents explored too much and failed to consistently exploit good paths. SARSA performed worst in terms of both efficiency and robustness, due to its on-policy updates that are sensitive to noisy policies. Q-learning had the highest path cost and variance, as it aggressively updated values without reliable exploitation. Double Q-learning handled the instability better, maintaining relatively lower path cost and variance, likely due to its balanced update strategy. Still, performance remained highly inconsistent across episodes, confirming the limitations of excessive exploration in wind-affected environments.

alpha = 0.2

epsilon = 0.8

episodes = 1000

window_size = 25

convergence_threshold = 240

Algorithm	Convergence Episode	Path Efficiency	Robustness
Q-Learning	255	363.52	296.872649
SARSA	700	308.24	193.213515
Double Q-Learning	890	265.56	168.096896

Due to the high exploration rate (ε = 0.8), agent behavior remained highly stochastic, even in later training episodes. As a result, the standard convergence threshold (50 steps) was too restrictive for capturing meaningful stabilization. Increasing the threshold to 240 allowed the evaluation to reflect slower, yet valid convergence trends in a noisy environment, especially for Double Q-learning, which converged late but consistently.

Algorithm Comparison – Gridworld B (α = 0.2, varying ε)

At low exploration (ε = 0.2), all algorithms converged reliably, with Q-learning and Double Q-learning showing fast, stable learning and SARSA being slightly less efficient. As ε increased to 0.5, exploration caused learning to slow down, but convergence was still achieved; Double Q-learning stood out for its robustness and consistent performance. At ε = 0.8, all algorithms struggled, initial convergence thresholds failed, and agents required a more lenient threshold to reflect learning progress. Even then, performance was unstable, with Q-learning suffering from high variance, SARSA being inefficient, and Double Q-learning emerging as the most robust in noisy conditions.

1.2.4.

alpha = 0.5

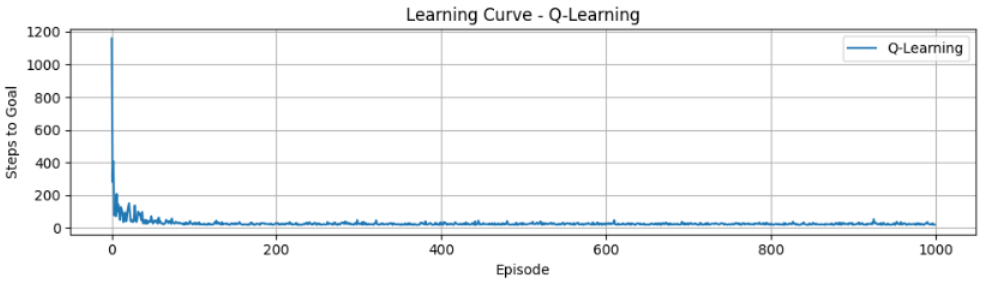
epsilon = 0.2

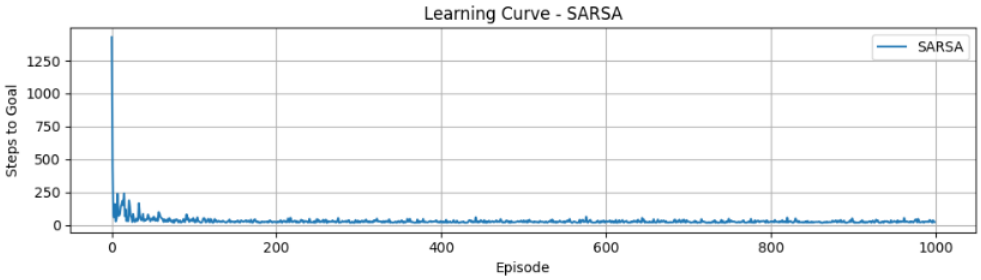
episodes = 1000

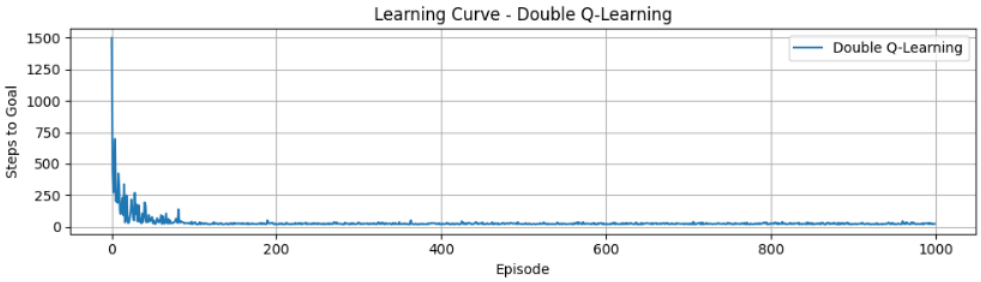
window_size = 25

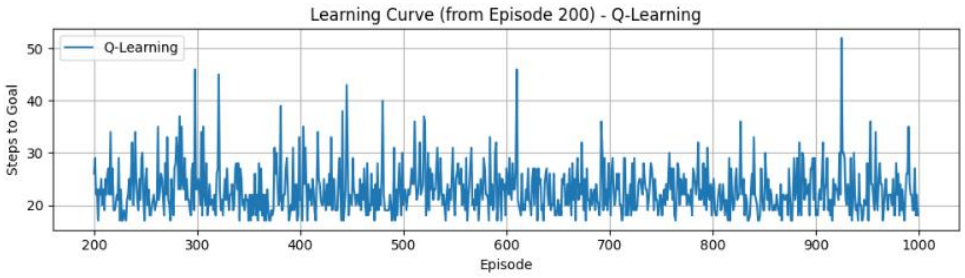
convergence_threshold = 50

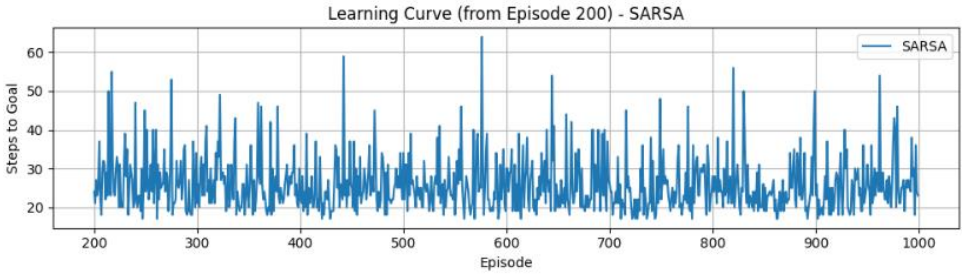
Algorithm	Convergence Episode	Path Efficiency	Robustness
Q-Learning	54	22.28	4.171523
SARSA	64	27.96	7.480535
Double Q-Learning	67	22.44	3.395055

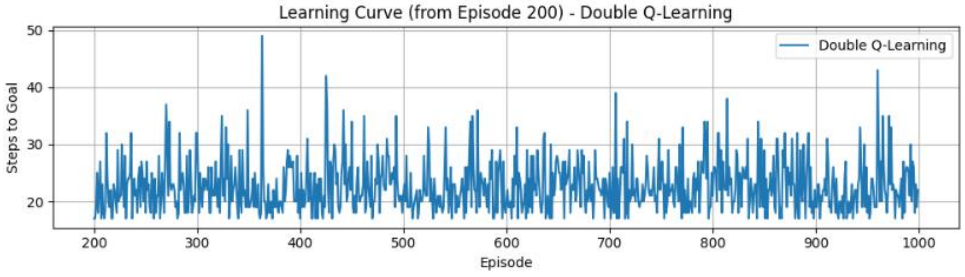

Learning Curve - Q-Learning
Q-Learning
Steps to Goal
Episode
0
200
400
600
800
1000
1200


Learning Curve - SARSA
SARSA
Steps to Goal
Episode
0
200
400
600
800
1000
1250


Learning Curve - Double Q-Learning
Double Q-Learning
Steps to Goal
Episode
0
200
400
600
800
1000
1500


Learning Curve (from Episode 200) - Q-Learning
Q-Learning
Steps to Goal
Episode
20
30
40
50
200
300
400
500
600
700
800
900
1000


Learning Curve (from Episode 200) - SARSA
SARSA
Steps to Goal
Episode
20
30
40
50
60
200
300
400
500
600
700
800
900
1000


Learning Curve (from Episode 200) - Double Q-Learning
Double Q-Learning
Steps to Goal
Episode
20
30
40
50
200
300
400
500
600
700
800
900
1000

With moderate learning and low exploration, all algorithms converged effectively. Q-learning achieved the fastest convergence and maintained stable performance, benefiting from decisive updates. Double Q-learning followed closely with the best robustness, confirming its stability. SARSA, while slower and more variable, still converged and reached decent efficiency. After convergence, Double Q-learning showed the most consistent behavior in the long run.

1.2.5.

alpha = 0.5

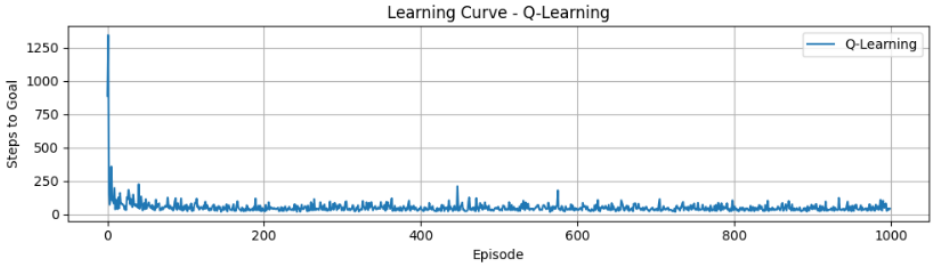
epsilon = 0.5

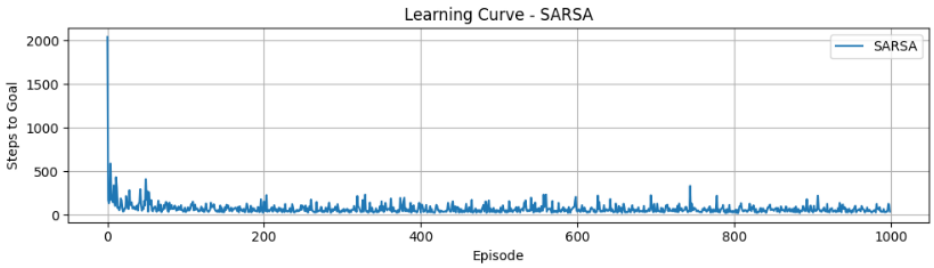
episodes = 1000

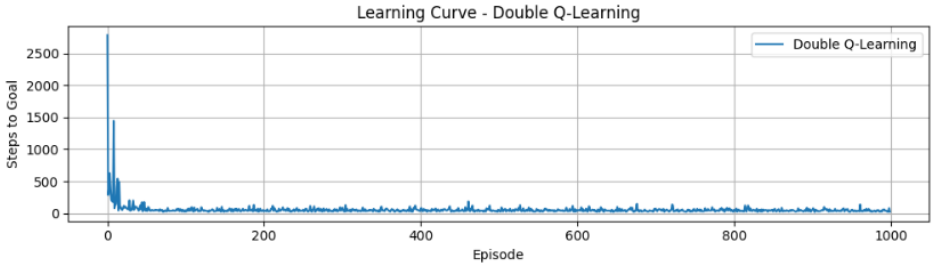
window_size = 25

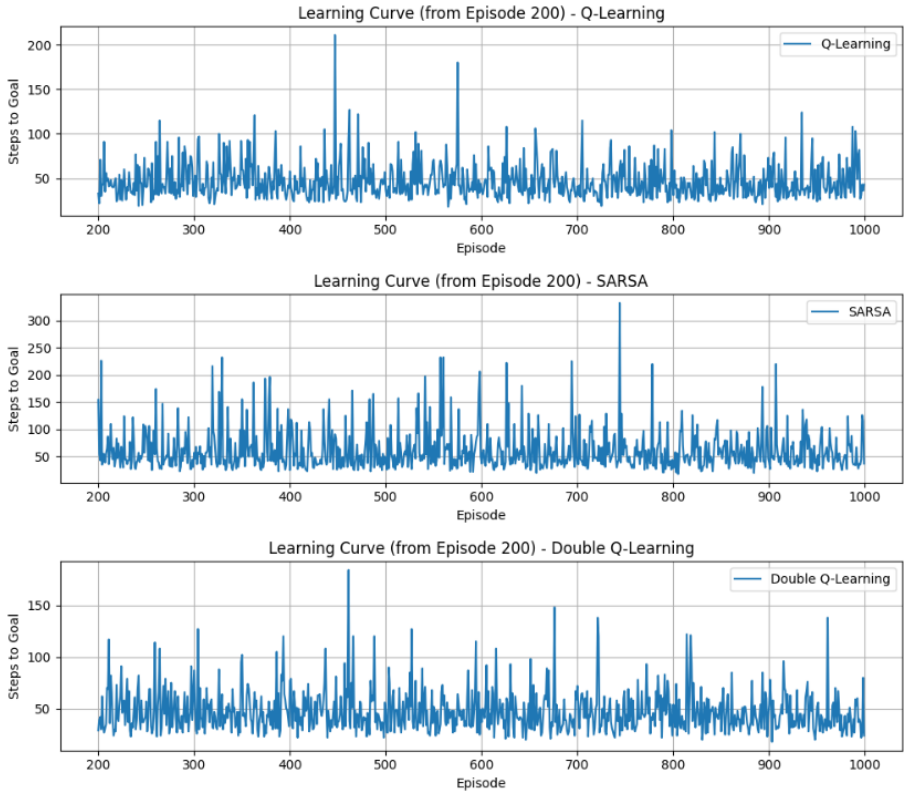
convergence_threshold = 50

Algorithm	Convergence Episode	Path Efficiency	Robustness
Q-Learning	141	51.08	23.208481
SARSA	431	54.28	29.398667
Double Q-Learning	73	38.20	13.971399









At moderate learning and exploration rates, all algorithms converged, but at different speeds and with varying stability. Double Q-learning converged fastest and was the most robust, demonstrating strong control over overestimation in the windy environment. Q-learning followed, converging decently with moderate variance. SARSA was the slowest to converge and showed high variance after convergence, likely due to its sensitivity to the stochastic transitions caused by wind.

1.2.6.

alpha = 0.5

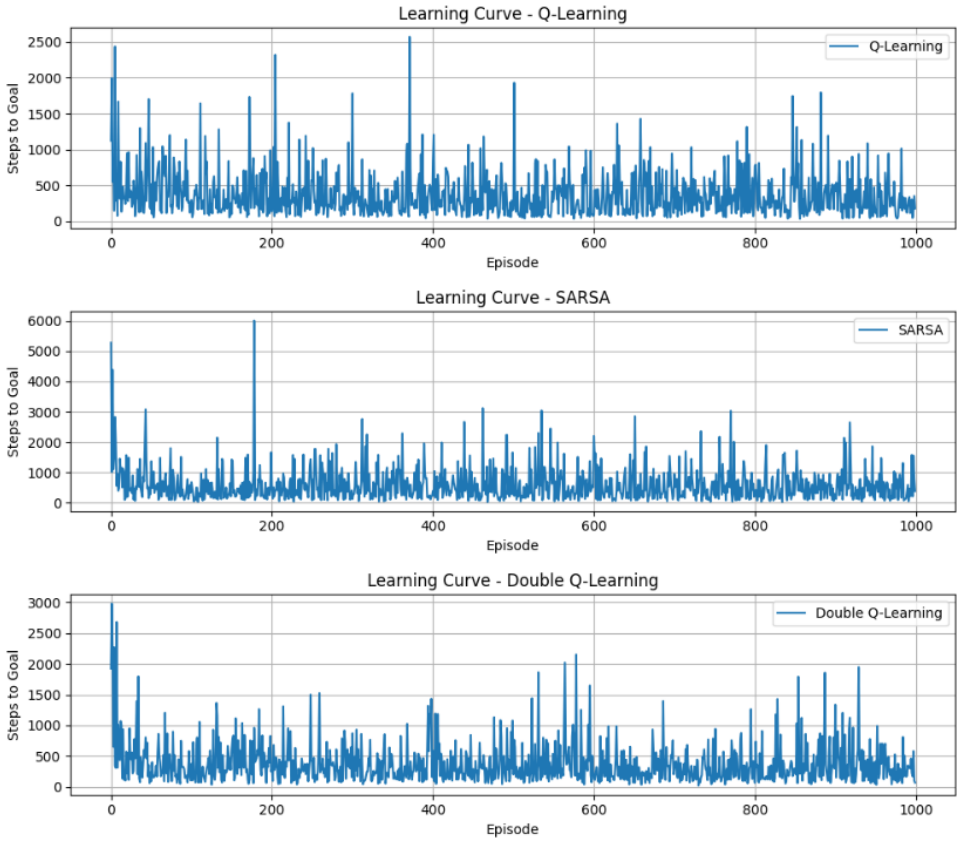
epsilon = 0.8

episodes = 1000

window_size = 25

convergence_threshold = 50

Algorithm	Convergence Episode	Path Efficiency	Robustness
Q-Learning	None	231.76	190.850367
SARSA	None	548.96	440.663271
Double Q-Learning	None	264.80	176.439451



With high exploration, none of the algorithms managed to converge. The environment's stochasticity combined with frequent exploratory actions led to unstable learning and erratic paths. SARSA was the most affected, with extremely high path cost and variability. Q-learning and Double Q-learning performed slightly better, with Double Q-learning achieving more consistent behavior. However, all agents suffered from noisy updates and difficulty exploiting learned knowledge.

alpha = 0.5

epsilon = 0.8

episodes = 1000

window_size = 25

convergence_threshold = 350

Algorithm	Convergence Episode	Path Efficiency	Robustness
Q-Learning	73	235.28	165.480759
SARSA	495	616.52	630.892391
Double Q-Learning	40	341.84	216.331353

Due to the high exploration rate, the agents continued to explore frequently throughout training, delaying stabilization. The default convergence threshold was insufficient to detect learning progress. Raising the threshold to 350 allowed convergence to be captured for all algorithms, revealing that Double Q-learning, despite the noise, adapted quickest, while SARSA remained highly unstable and inefficient in this regime.

Algorithm Comparison – Gridworld B (α = 0.5, varying ε)

When α is fixed at 0.5, the impact of varying ε becomes evident in the learning dynamics of all three algorithms. At low exploration (ε = 0.2), all methods converged reliably, with Q-learning being fastest and Double Q-learning offering the most stable post-convergence behavior. At moderate exploration (ε = 0.5), Double Q-learning clearly outperformed the others in terms of convergence speed and robustness, handling the environment's stochasticity more effectively. At high exploration (ε = 0.8), all algorithms struggled—convergence was delayed or not achieved under standard thresholds. Still, Double Q-learning maintained better control over learning, while SARSA suffered from high variance and inefficiency. Overall, Double Q-learning proved to be the most robust and adaptable across exploration levels.

1.2.7.

alpha = 0.8

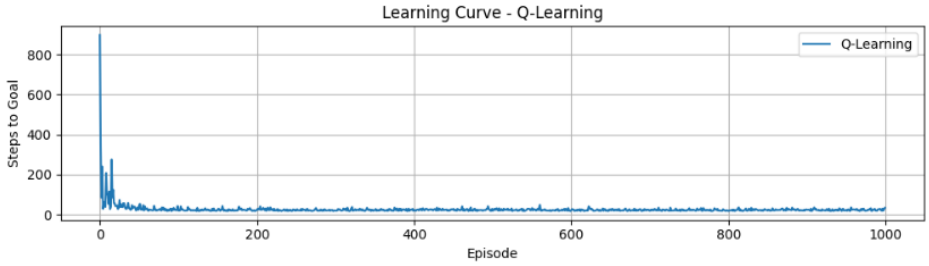
epsilon = 0.2

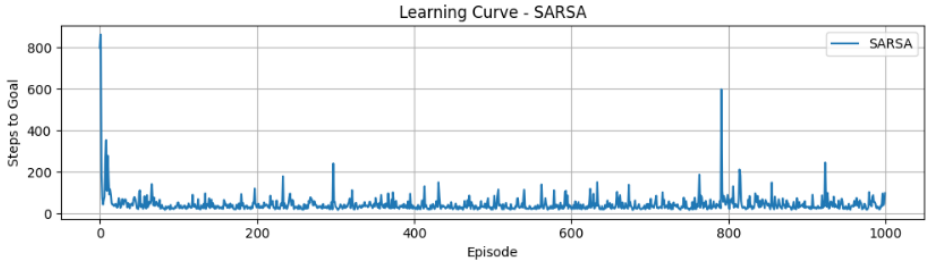
episodes = 1000

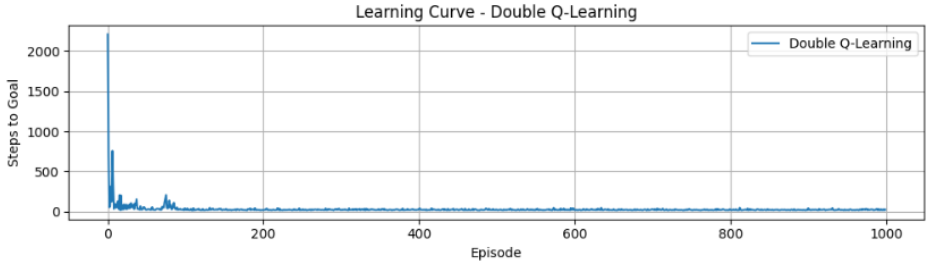
window_size = 25

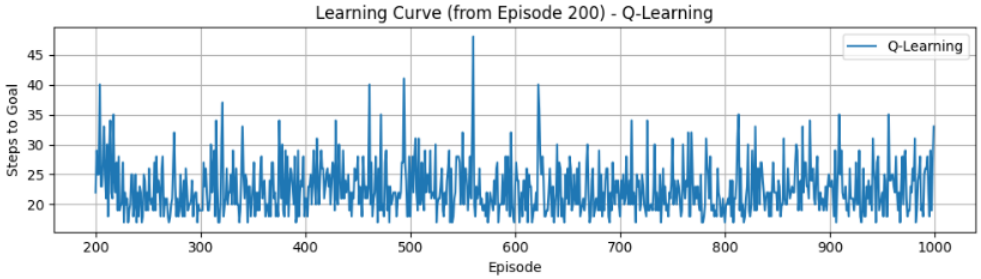
convergence_threshold = 50

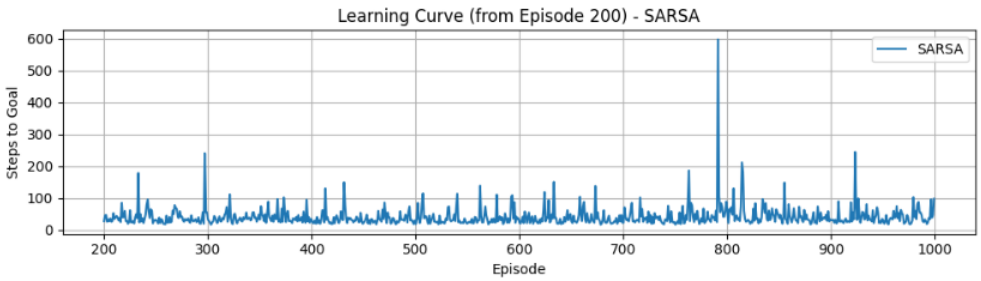
Algorithm	Convergence Episode	Path Efficiency	Robustness
Q-Learning	41	23.48	4.355410
SARSA	39	46.40	26.086012
Double Q-Learning	56	25.64	5.469040

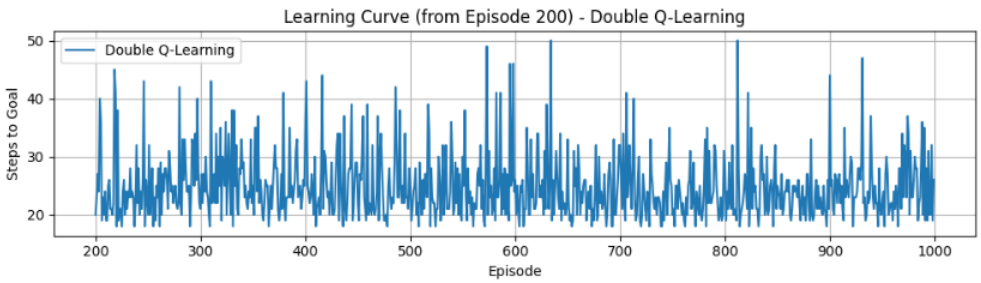

Learning Curve - Q-Learning
Q-Learning
Steps to Goal
Episode


Learning Curve - SARSA
SARSA
Steps to Goal
Episode


Learning Curve - Double Q-Learning
Double Q-Learning
Steps to Goal
Episode


Learning Curve (from Episode 200) - Q-Learning
Q-Learning
Steps to Goal
Episode


Learning Curve (from Episode 200) - SARSA
SARSA
Steps to Goal
Episode


Learning Curve (from Episode 200) - Double Q-Learning
Double Q-Learning
Steps to Goal
Episode

For Gridworld B with α = 0.8 and ε = 0.2, all three algorithms reached convergence, but their behavior differs notably. Q-Learning and Double Q-Learning maintained more stable performance, while SARSA displayed higher path efficiency at the cost of increased variance. This suggests that SARSA was more exploratory, which may have led to improved solutions in some cases, but less consistent outcomes overall.

1.2.8.

alpha = 0.8

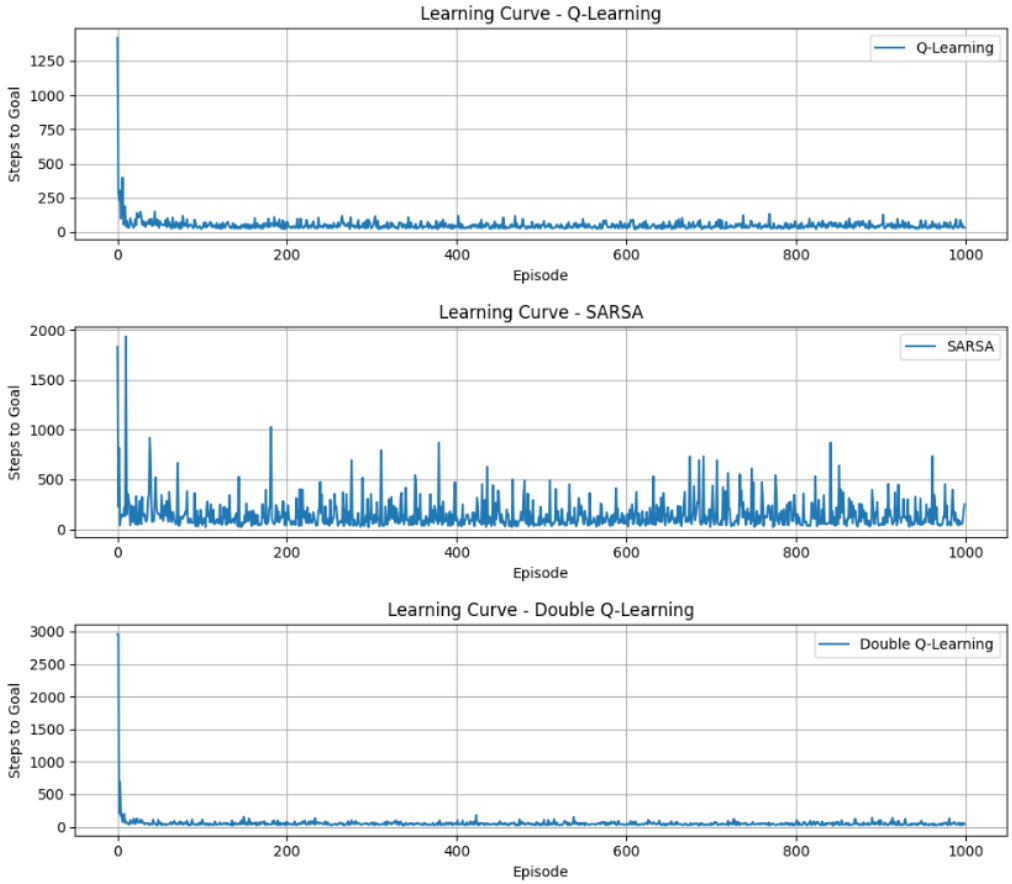
epsilon = 0.5

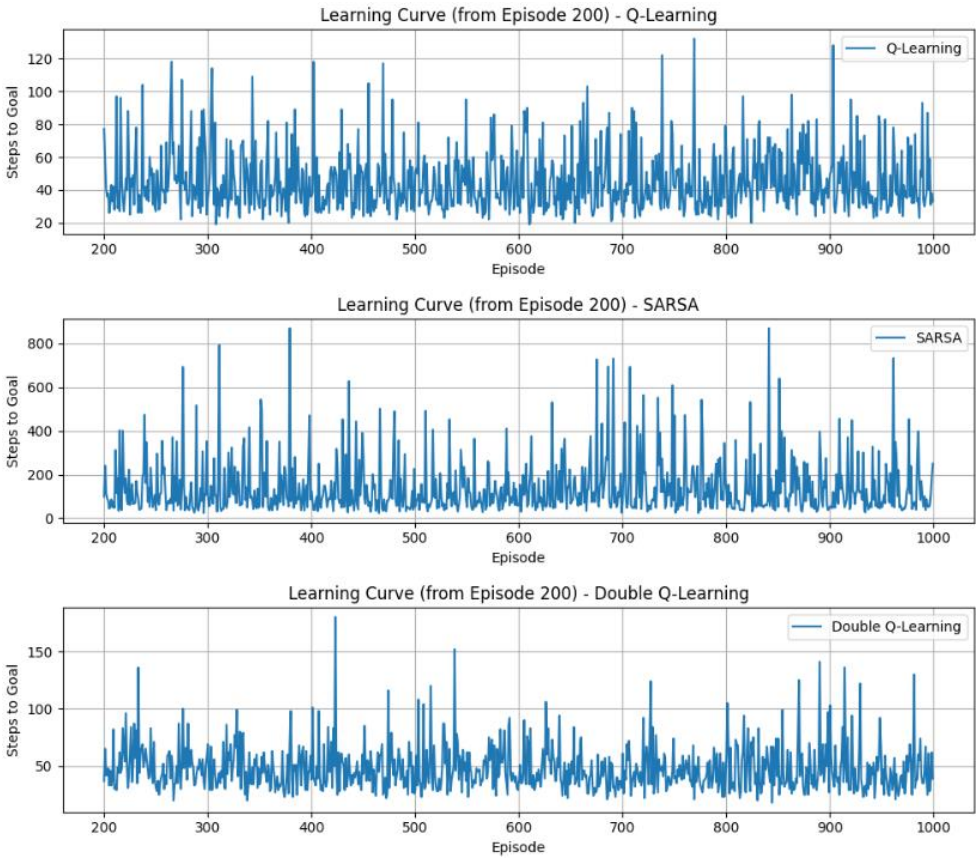
episodes = 1000

window_size = 25

convergence_threshold = 50

Algorithm	Convergence Episode	Path Efficiency	Robustness
Q-Learning	100.0	46.68	18.628408
SARSA	NaN	136.16	107.526994
Double Q-Learning	57.0	49.72	21.472811





In the configuration with $\alpha = 0.8$ and $\varepsilon = 0.5$, we observe that Double Q-Learning converged the fastest and maintained low variance in its learning curve, indicating stable learning. Q-Learning also achieved convergence but exhibited more fluctuation, suggesting less consistent performance. Meanwhile, SARSA failed to converge and showed the highest path efficiency and robustness, likely due to its on-policy nature, which helps it adapt better to the environment but at the cost of slower or unstable convergence.

This setup highlights a trade-off: Double Q-Learning offers stability and early convergence, Q-Learning provides moderate performance with some variance, and SARSA is more explorative but less reliable in convergence under this parameter combination.

alpha = 0.8

epsilon = 0.5

episodes = 1000

window_size = 25

convergence_threshold = 100

Algorithm	Convergence Episode	Path Efficiency	Robustness
Q-Learning	29	49.56	20.199168
SARSA	370	163.28	157.433293
Double Q-Learning	33	51.80	25.051946

Increasing the convergence threshold to 100 allowed SARSA to finally meet convergence criteria, highlighting its slower learning pace compared to Q-Learning and Double Q-Learning, which had already stabilized under a lower threshold.

1.2.9.

alpha = 0.8

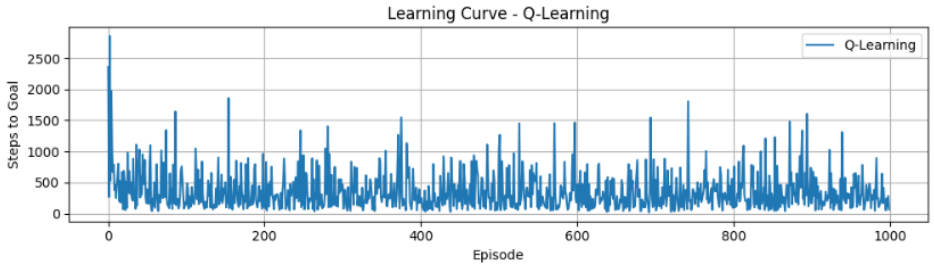
epsilon = 0.8

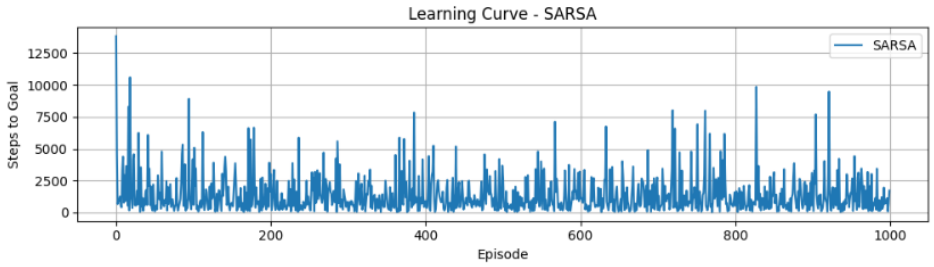
episodes = 1000

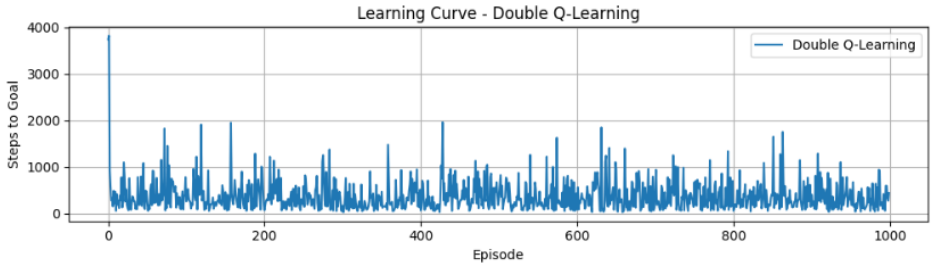
window_size = 25

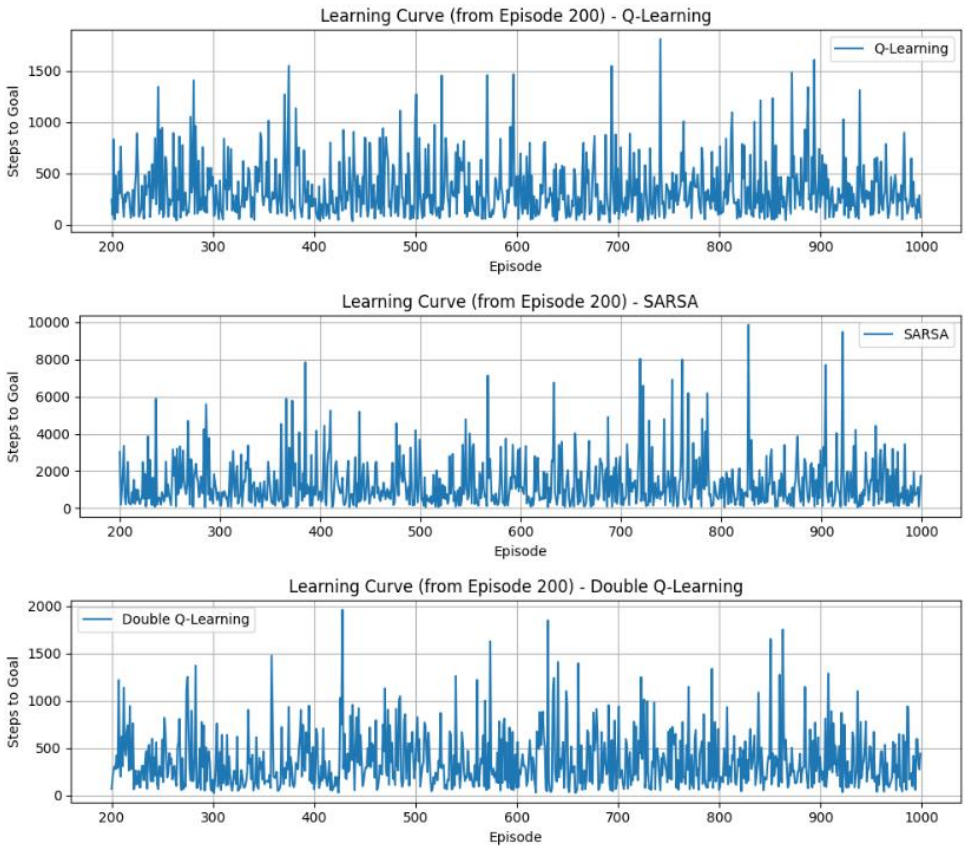
convergence_threshold = 50

Algorithm	Convergence Episode	Path Efficiency	Robustness
Q-Learning	None	264.00	188.060416
SARSA	None	876.64	842.630637
Double Q-Learning	None	335.44	263.174175









When both α (learning rate) and ε (exploration rate) are high, all algorithms struggle to converge due to overly aggressive learning and frequent exploratory actions. This combination introduces instability: updates change rapidly and the agent explores too often, preventing consistent policy refinement. As a result, none of the algorithms achieve convergence, and performance (in terms of path efficiency and robustness) deteriorates significantly.

alpha = 0.8

epsilon = 0.8

episodes = 1000

window_size = 25

convergence_threshold = 900

Algorithm	Convergence Episode	Path Efficiency	Robustness
Q-Learning	26	396.00	380.063995
SARSA	190	1232.60	819.228369
Double Q-Learning	26	350.96	305.350354

Raising the convergence threshold to 900 revealed that even under high exploration (ε = 0.8) and aggressive learning (α = 0.8), Q-Learning and Double Q-Learning managed to converge very early (episode 26), albeit with moderate path efficiency and robustness. However, SARSA converged much later (episode 190) but achieved significantly higher path efficiency and robustness—likely due to its on-policy nature, which helps refine policies better under noisy conditions. This suggests SARSA benefits more from extended training when randomness is high.

Algorithm Comparison – Gridworld B (α = 0.8, varying ε)

In Gridworld B with a high learning rate (α = 0.8), the comparison across varying ε values reveals that while all algorithms show improved responsiveness to rewards, their stability diverges. Q-Learning and Double Q-Learning benefit from this high α, managing to converge even at higher exploration levels (ε = 0.8) when an appropriate convergence threshold is used. In contrast, SARSA becomes increasingly erratic as ε grows, showing poor convergence and extremely high path efficiency and robustness values, indicating inefficient learning. This suggests that SARSA is more sensitive to exploration under aggressive learning conditions, while Q-Learning variants maintain better balance between exploration and stability.

Algorithm Comparison – Gridworld B (varying α, varying ε)

In Gridworld B, across all variations of α and ε, distinct behavioral patterns emerge among the algorithms. Q-Learning consistently achieves fast convergence and maintains relatively balanced path efficiency and robustness, especially under moderate ε values. Double Q-Learning shows better stability than Q-Learning in high-noise settings, but converges slightly slower. SARSA, while sometimes achieving high path efficiency, tends to be the least stable, especially under high exploration (ε) or aggressive learning (high α), often failing to converge or showing excessive variability. Overall, Q-Learning and Double Q-Learning demonstrate better adaptability across a wide range of configurations, while SARSA is more sensitive to parameter tuning.

2. Task 2

In the second task, we extend the experiment to a multi-agent scenario in Gridworld B, where three agents use the same Q-learning algorithm. Each agent maintains its own Q-table and learns independently through an ε-greedy strategy. However, the environment introduces cooperative rewards, granting a bonus if all agents reach the goal simultaneously.

This shared incentive encourages implicit coordination, even though learning is decentralized, and allows us to observe how cooperation emerges through reward-driven behavior.

2.1.

alpha = 0.2

epsilon = 0.2

episodes = 1000

window_size = 25

convergence_threshold = 50

```
Convergence: 147
Path efficiency: 25.56
Robustness: 3.9504936400404445
```

Under the configuration with α = 0.2 and ε = 0.2, the multi-agent Q-learning algorithm shows stable and smooth learning behavior, achieving convergence by episode 147. The path efficiency is solid at 25.56 steps, while robustness remains low (3.95), indicating minimal variability. The learning curves further support this stability, with a rapid drop in steps early on and consistent performance in the latter episodes. This setting reflects a balanced trade-off between exploration and exploitation, making it ideal for steady multi-agent coordination.

2.2.

alpha = 0.2

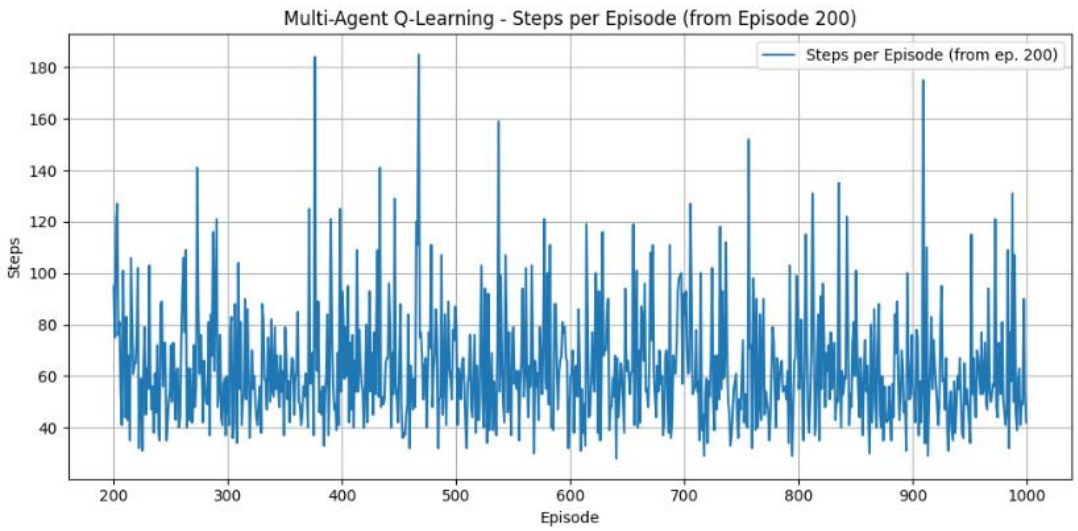
epsilon = 0.5

episodes = 1000

window_size = 25

convergence_threshold = 50

```
Convergence: 951
Path efficiency: 62.24
Robustness: 24.28872989680605
```



The Multi-Agent Q-Learning setup demonstrates very slow convergence, reaching the threshold only by episode 951. Although it eventually stabilizes, the path efficiency (62.24 steps) is considerably less optimal compared to lower ε settings. The robustness is also much higher (24.29), indicating significant variance in agent performance.

This configuration suggests that higher exploration (ε = 0.5) with low learning rate (α = 0.2) causes the agents to take longer to converge and leads to more erratic behavior across episodes. Reducing ε could improve both convergence speed and consistency.

2.3.

alpha = 0.2

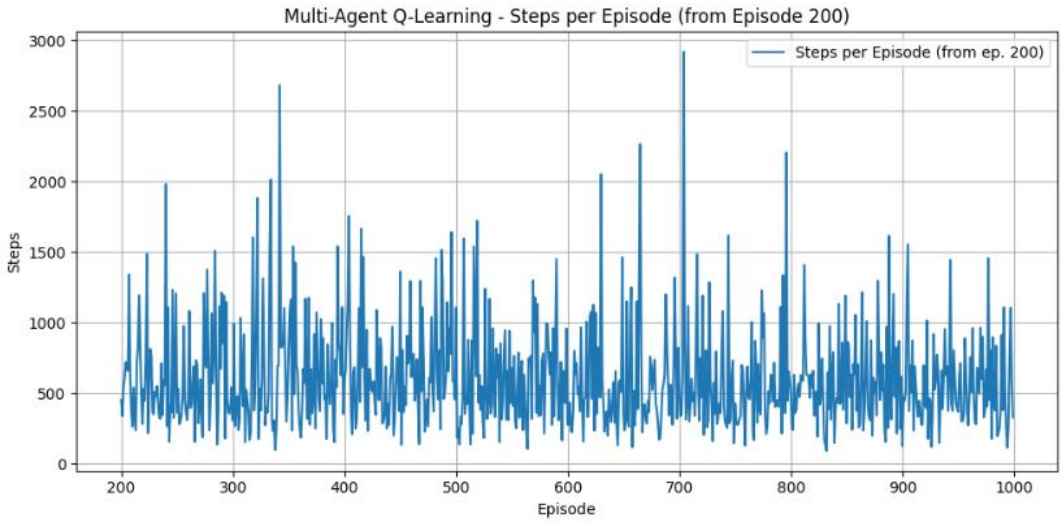
epsilon = 0.8

episodes = 1000

window_size = 25

convergence_threshold = 500

```
Convergence: 318
Path efficiency: 554.16
Robustness: 339.97666743469324
```



Under high exploration and low learning rate, the agents managed to converge relatively late (episode 318), but the learning process remained highly unstable. The very high path inefficiency (554.16) and extreme robustness value (339.98) suggest that agents frequently deviated from optimal paths, struggling to coordinate effectively. The learning curves show significant fluctuations even after convergence, indicating poor policy refinement and inconsistent behavior across episodes.

Algorithm Comparison – Multi-Agent Q-Learning (α = 0.2, varying ε)

In the multi-agent setting, cooperation and synchronization introduce an additional layer of complexity. With α = 0.2, agents update their policies slowly, making the balance between exploration and exploitation even more critical.

At ε = 0.2, agents converged quickly with low variability and efficient paths, indicating that focused exploitation allowed them to learn coordinated strategies effectively. Increasing exploration to ε = 0.5 led to very slow convergence and higher variance, suggesting that agents struggled to synchronize under moderate randomness. At ε = 0.8, convergence was somewhat faster than ε = 0.5 but resulted in highly inefficient and unstable behavior. The agents explored excessively, making it difficult to learn coordinated, reward-maximizing policies.

These results show that in cooperative multi-agent scenarios with slow learning (low α), low to moderate exploration is crucial for stable convergence and effective joint behavior, while high exploration causes coordination breakdowns and inconsistent learning.

2.4.

alpha = 0.5

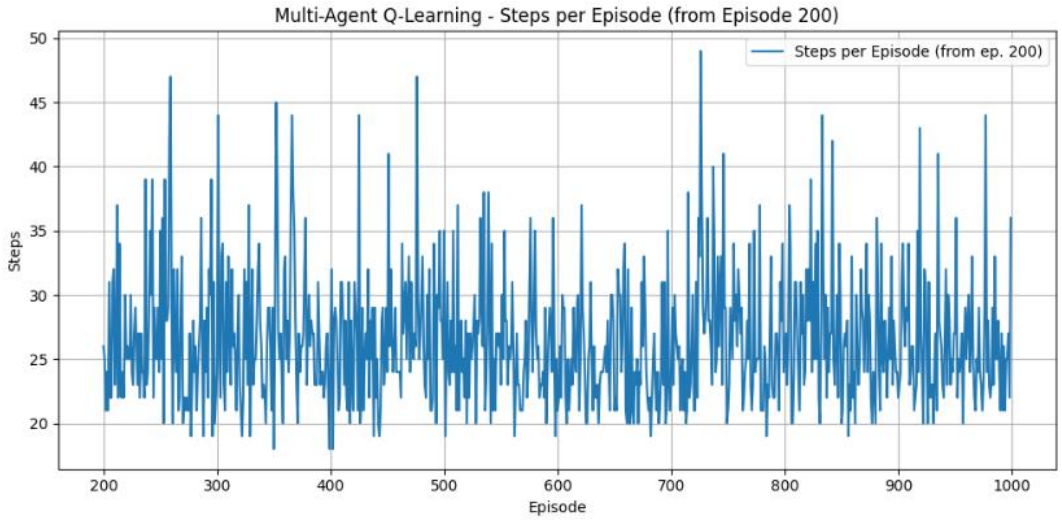
epsilon = 0.2

episodes = 1000

window_size = 25

convergence_threshold = 50

```
Convergence: 68
Path efficiency: 26.16
Robustness: 5.12
```



For the configuration $\alpha = 0.5$, $\varepsilon = 0.2$, the agent demonstrates fast convergence (episode 68), high path efficiency (26.16), and low variability (robustness = 5.12), indicating a stable and effective learning process in the multi-agent setting. The low ε fosters consistent policy exploitation, while the moderate α allows reliable updates without destabilizing learning.

2.5.

alpha = 0.5

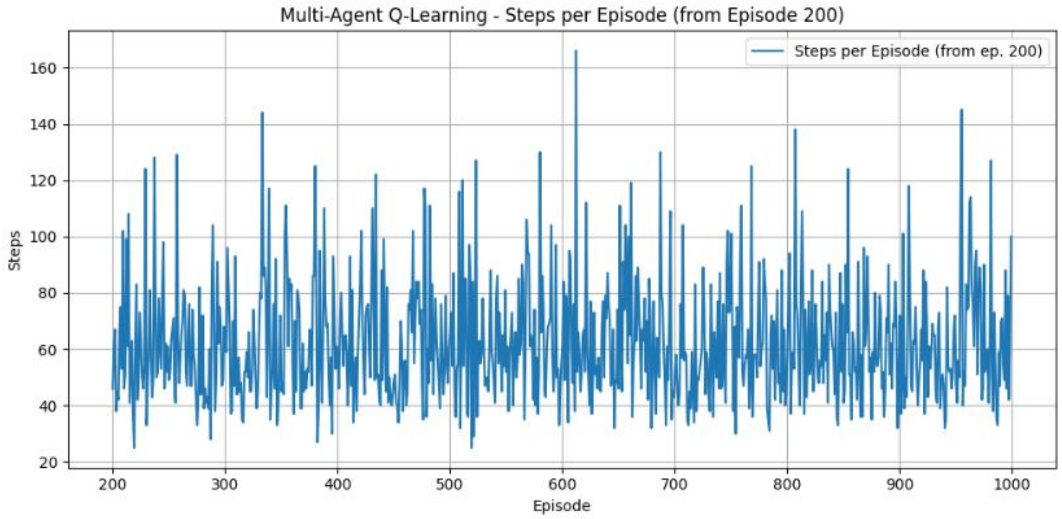
epsilon = 0.5

episodes = 1000

window_size = 25

convergence_threshold = 100

```
Convergence: 64
Path efficiency: 61.32
Robustness: 21.719521173359233
```



The agents converged early (episode 64), showing that a balanced exploration-exploitation tradeoff with a high learning rate facilitates fast learning. However, the relatively high path efficiency (61.32) and robustness (21.72) indicate that while convergence is quick, the agents' performance fluctuates more, suggesting less stable policies in the long run.

2.6.

alpha = 0.5

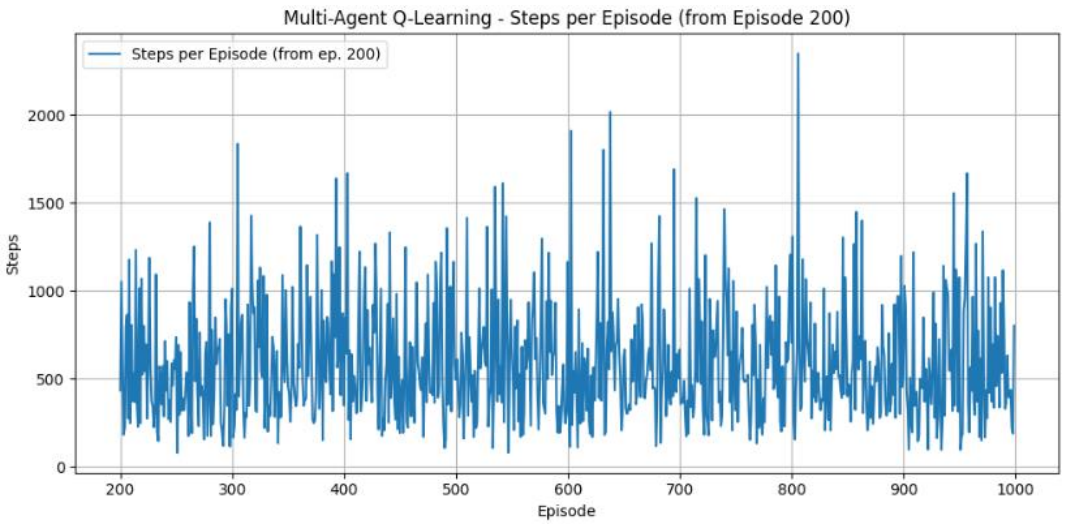
epsilon = 0.8

episodes = 1000

window_size = 25

convergence_threshold = 450

```
Convergence: 258
Path efficiency: 577.76
Robustness: 276.918078860879
```



The agent achieves convergence relatively late (episode 258), with high path inefficiency (577.76) and very high robustness (276.91), indicating unstable performance. The learning curve shows persistent fluctuations throughout training, suggesting that the high exploration rate (ε) makes it harder for the agents to consistently exploit learned policies, despite a decent learning rate.

Algorithm Comparison – Multi-Agent Q-Learning (α = 0.5, varying ε)

In the multi-agent Q-Learning setup with α = 0.5, low exploration (ε = 0.2) yields fast and stable convergence, while moderate exploration (ε = 0.5) slightly increases variability but maintains good performance. However, high exploration (ε = 0.8) significantly delays convergence and leads to much higher path inefficiency and instability, highlighting the sensitivity of the system to excessive randomness when the learning rate is already aggressive.

2.7.

alpha = 0.8

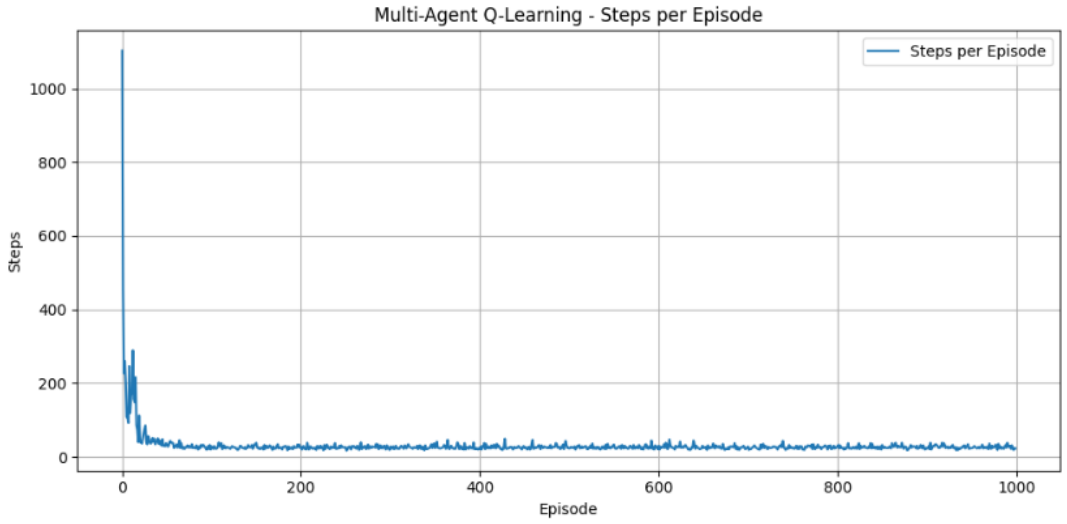
epsilon = 0.2

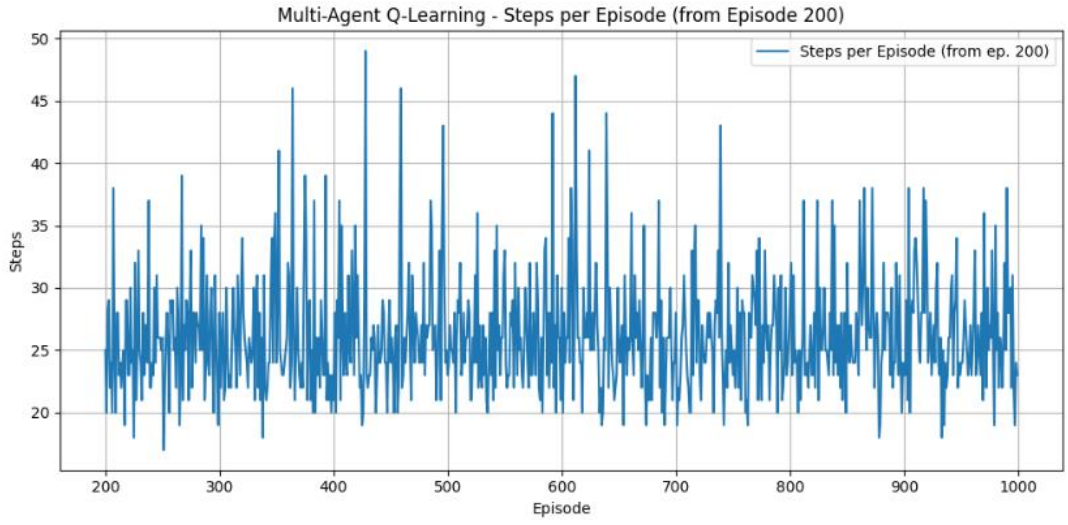
episodes = 1000

window_size = 25

convergence_threshold = 50

```
Convergence: 43
Path efficiency: 26.44
Robustness: 4.6396551595996876
```





In the multi-agent Q-learning setting with a high learning rate (α = 0.8) and low exploration (ε = 0.2), agents converged rapidly (episode 43), achieving high path efficiency (26.44) and low robustness (4.64). This indicates that under stable learning conditions, agents quickly optimize their policy with minimal variability, reflecting highly consistent behavior across episodes.

2.8.

alpha = 0.8

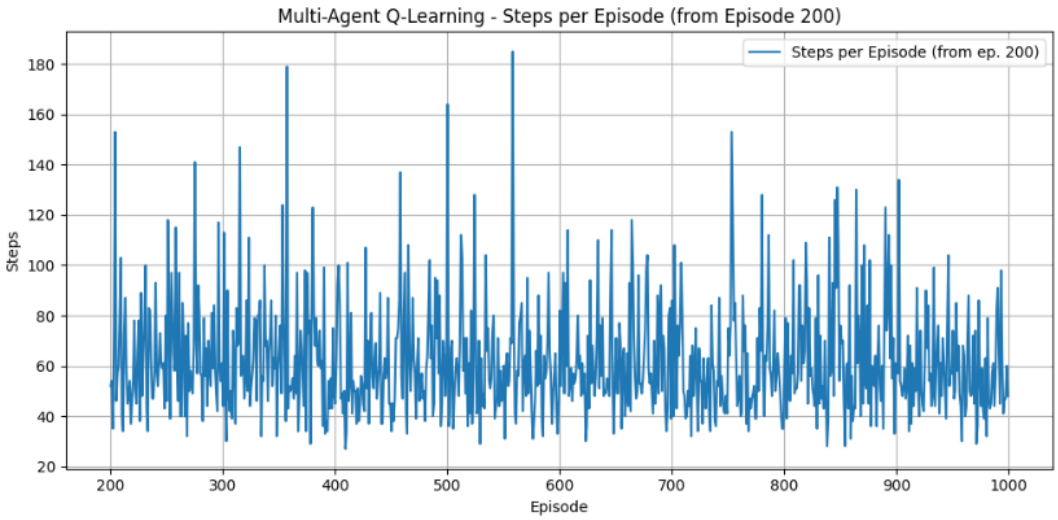
epsilon = 0.5

episodes = 1000

window_size = 25

convergence_threshold = 100

```
Convergence: 59
Path efficiency: 55.76
Robustness: 16.361613612354986
```



The agents demonstrates fast convergence at episode 59, coupled with solid path efficiency (55.76) and moderate robustness (16.36). The learning curves confirm a steep initial improvement followed by consistent performance, indicating a strong balance between rapid learning and controlled exploration.

2.9.

alpha = 0.8

epsilon = 0.8

episodes = 1000

window_size = 25

convergence_threshold = 450

```
Convergence: 203
Path efficiency: 610.96
Robustness: 304.1868478419144
```

The agents manage to converge by episode 203, but the learning process is highly unstable. The very high path inefficiency (610.96) and robustness (304.19) indicate erratic behavior and significant variance in performance, suggesting that the combination of aggressive learning and exploration hinders consistent coordination among agents.

Algorithm Comparison – Multi-Agent Q-Learning (α = 0.8, varying ε)

In the Multi-Agent Q-Learning setup with a high learning rate (α = 0.8), increasing ε reveals a clear trade-off between exploration and stability. At low ε (0.2), agents converge quickly with low variance and efficient paths. As ε increases to 0.5 and 0.8, learning becomes more unstable, convergence slows, path efficiency worsens, and robustness drops significantly.

This suggests that while a high α accelerates learning, excessive exploration undercuts consistency and coordination in multi-agent environments.

Algorithm Comparison – Multi-Agent Q-Learning (varying α, varying ε)

When comparing Multi-Agent Q-Learning across varying α and ε values, results show that low ε (e.g., 0.2) consistently supports stable convergence, especially when paired with a moderate-to-high α (0.5–0.8). High α accelerates learning, but if coupled with high ε (e.g., 0.8), performance degrades due to excessive exploration, leading to high variability and delayed convergence. The best balance is achieved with $\alpha = 0.5$ or 0.8 and $\varepsilon = 0.2$, which ensures fast convergence, efficient paths, and low variance. Overall, effective learning requires careful tuning of exploration to match the learning rate in multi-agent settings.

3. Conclusions

Across both Gridworld A (static environment) and Gridworld B (stochastic wind), results show that Q-Learning and Double Q-Learning consistently outperform SARSA in terms of convergence speed and robustness. While SARSA occasionally achieves higher path efficiency, it is significantly more sensitive to exploration and environmental variability.

In simpler environments like Gridworld A, all algorithms perform well with proper tuning. However, in Gridworld B, which introduces stochasticity and complexity, Double Q-Learning demonstrates superior stability, especially under higher exploration rates.

In the multi-agent scenario (Task 2), effective coordination is strongly influenced by the balance between learning rate and exploration. Low exploration ($\varepsilon = 0.2$) and moderate-to-high learning rates ($\alpha = 0.5$–0.8) enable agents to converge efficiently and act in sync. In contrast, excessive exploration leads to instability and poor path efficiency. Overall, off-policy methods (especially Double Q-Learning) paired with controlled exploration offer the most robust and adaptable learning across both single and multi-agent reinforcement learning tasks.