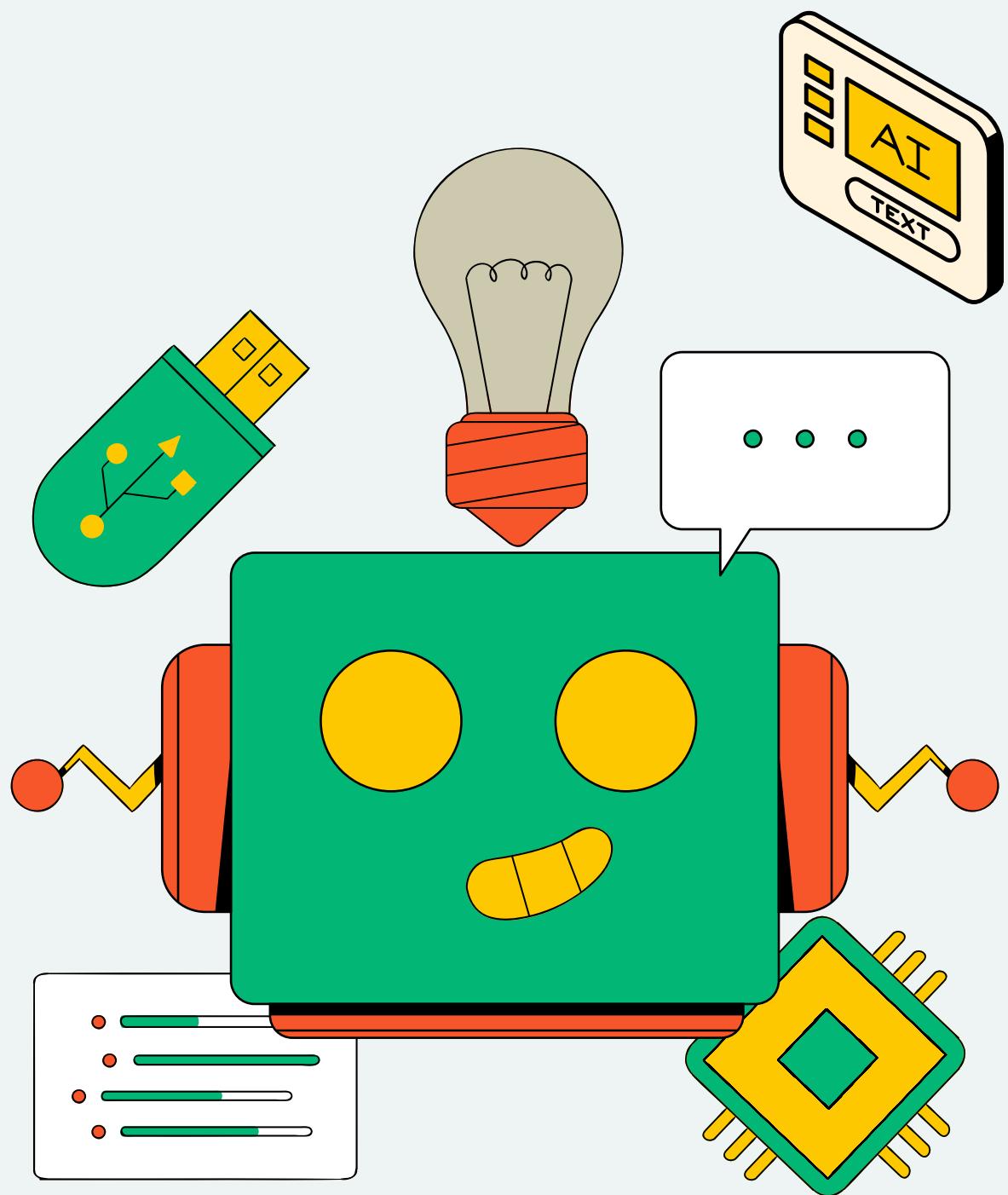


THYNK UNLIMITED
WE LEARN FOR THE FUTURE

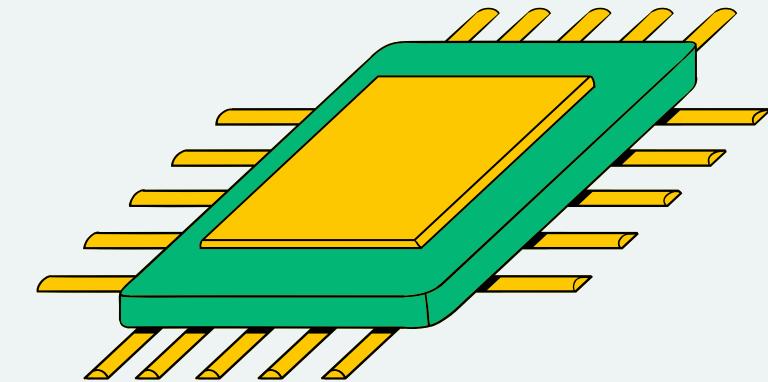


INTELLIGENZA ARTIFICIALE

TRAFFIC CONTROL

PRESENTED BY:

CAPALDO GENNARO,
SENATORE CARMELA
PIA



PRESENTAZIONE

- Introduzione
- Ambiente
- Metriche di Valutazione
- Definizione dello Stato
- Penalità e Ricompense
- Implementazione Q-Learning
- Implementazione Deep Q-Learning
- Implementazione SARSA
- Confronti
- Conclusioni



INTRODUZIONE

Il progetto ha l'obiettivo di sviluppare e implementare un sistema di controllo del traffico urbano, mirato a massimizzare l'efficienza del flusso veicolare.

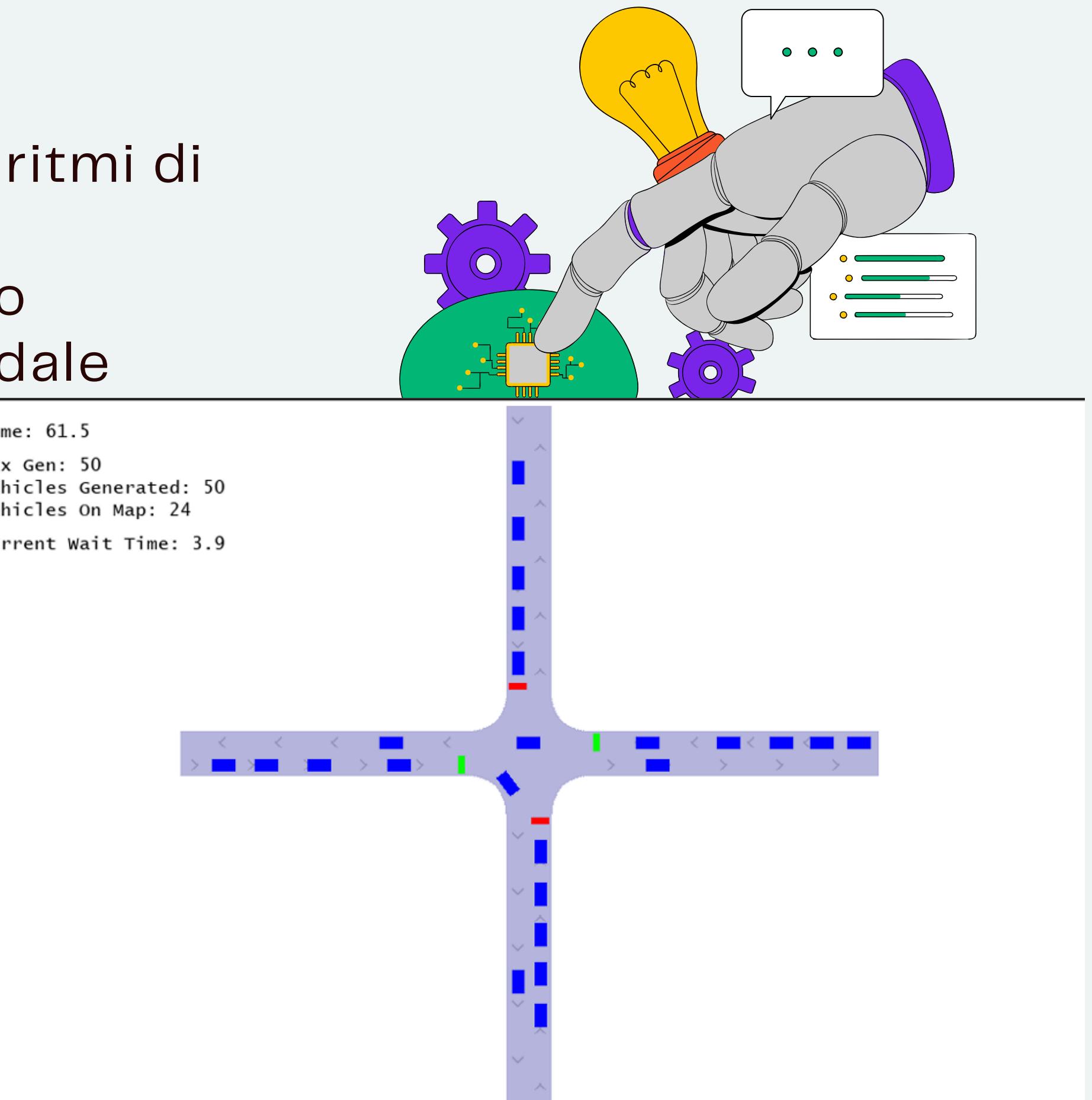
L'obiettivo verrà raggiunto utilizzando diversi modelli di reinforcement learning:

- Q-Learning;
- Deep Q-Learning
- SARSA



AMBIENTE

Per valutare le prestazioni degli algoritmi di Reinforcement Learning (RL), `e stato utilizzato un ambiente simulato rappresentante un'intersezione stradale urbana.

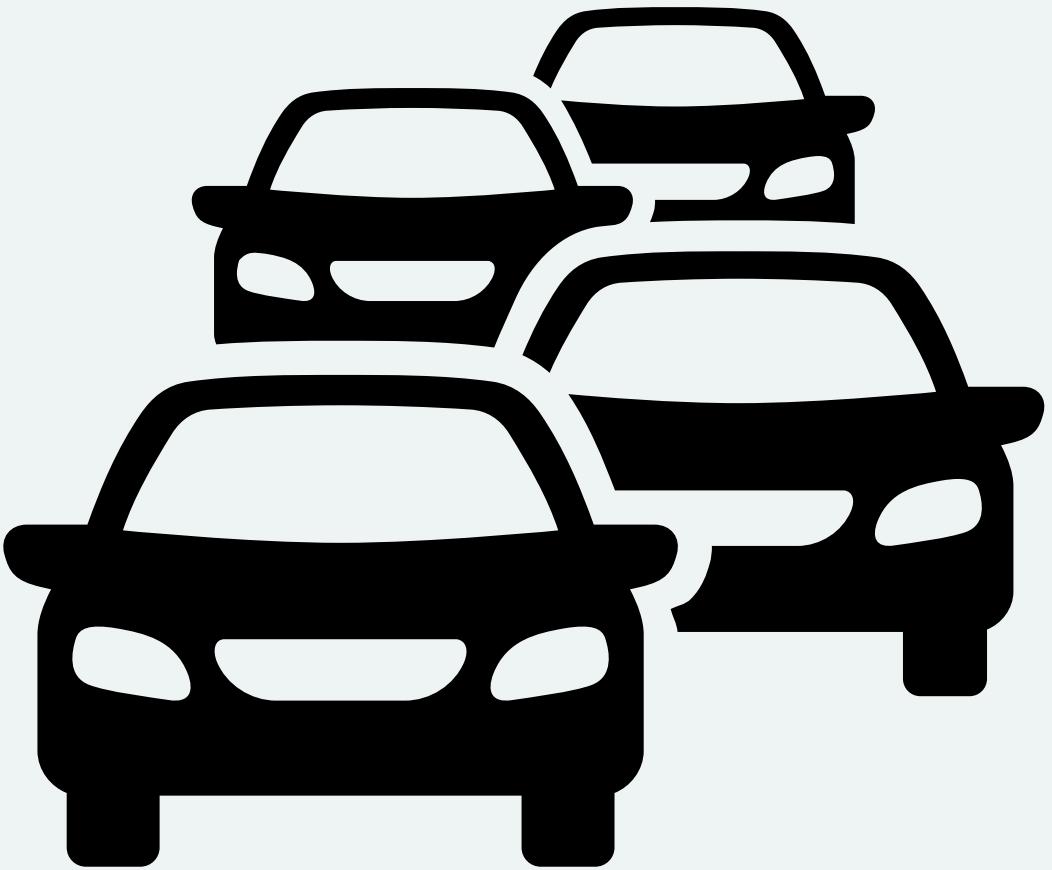


AMBIENTE: VEICOLI

Vengono generati su tutte e 4 le intersezioni fino a che non si raggiunge il limite di auto generate e a patto che non venga saturata una particolare direzione

Hanno una velocità di frenata, velocità di accelerazione, distanza di sicurezza che varia in base alle condizioni meteo

Si fermano quando si trovano in prossimità del semaforo (quando diventa giallo / rosso)



AMBIENTE: METEO

Sono stati introdotti quattro tipi di condizioni meteorologiche nell'ambiente:



Soleggiato



Pioggia

Neve

Nebbia



Il meteo cambia la velocità, distanza di sicurezza e velocità di frenata dei veicoli.

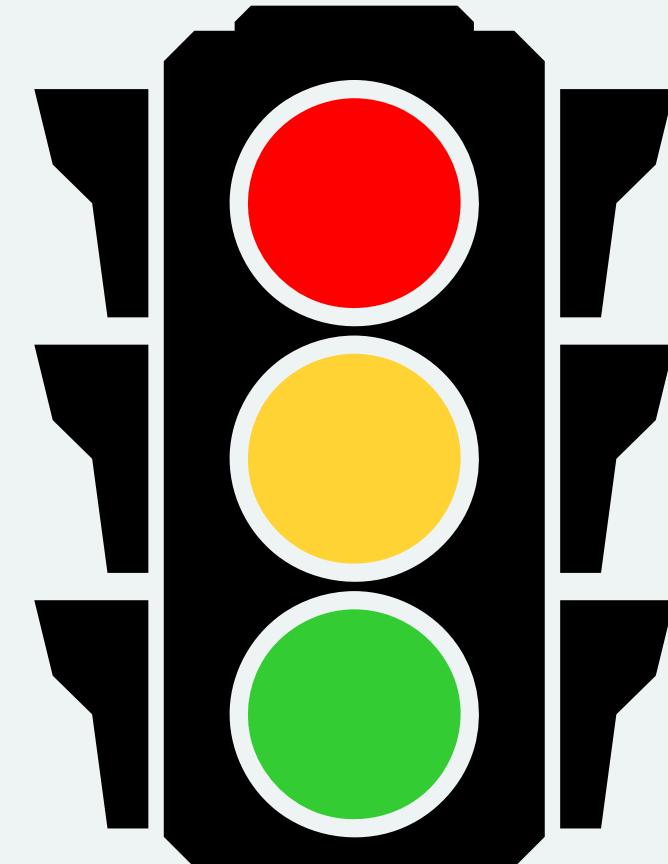


AMBIENTE: SEMAFORI

I semafori sulla mappa sono quattro

Possono avere due stati: rosso o verde

I semafori lungo la stessa direzione
hanno lo stesso stato



METRICHE DI VALUTAZIONE

Tempo di Attesa Globale

Numero di collisioni

Semaforo a Fasi

Code più lunghe



DEFINIZIONE DELLO STATO



Stato dei semafori: 0 se verde per la direzione orizzontale, 1 viceversa

Numero di veicoli presenti sulla direzione verticale

Numero di veicoli presenti sulla direzione orizzontale

Indicazione di incrocio vuoto

Condizione meteoreologica



DEFINIZIONE DELLE RICOMPENSE

Tempo medio di attesa

Il tempo medio di attesa usato come indice che attesta quanto i veicoli sulla mappa sono in attesa



Flow Change

La funzione di ricompensa si basa sulla differenza tra il numero di veicoli sulle strade in entrata dopo l'azione precedente e il numero di veicoli in dopo l'azione corrente

```
total_reward = [ flow_change - round(average_wait_time_on_map) ]
```



IMPLEMENTAZIONE Q-LEARNING

01

INTRODUZIONE

Il Q-Learning è un algoritmo di Reinforcement Learning (RL) off-policy che consente a un agente di apprendere come massimizzare una ricompensa cumulativa interagendo con l'ambiente.

02

Q-VALUE

Si basa sul concetto di Q-value, ovvero una funzione che stima il valore atteso di una data azione A eseguita in uno stato S.

03

CONVERGENZA

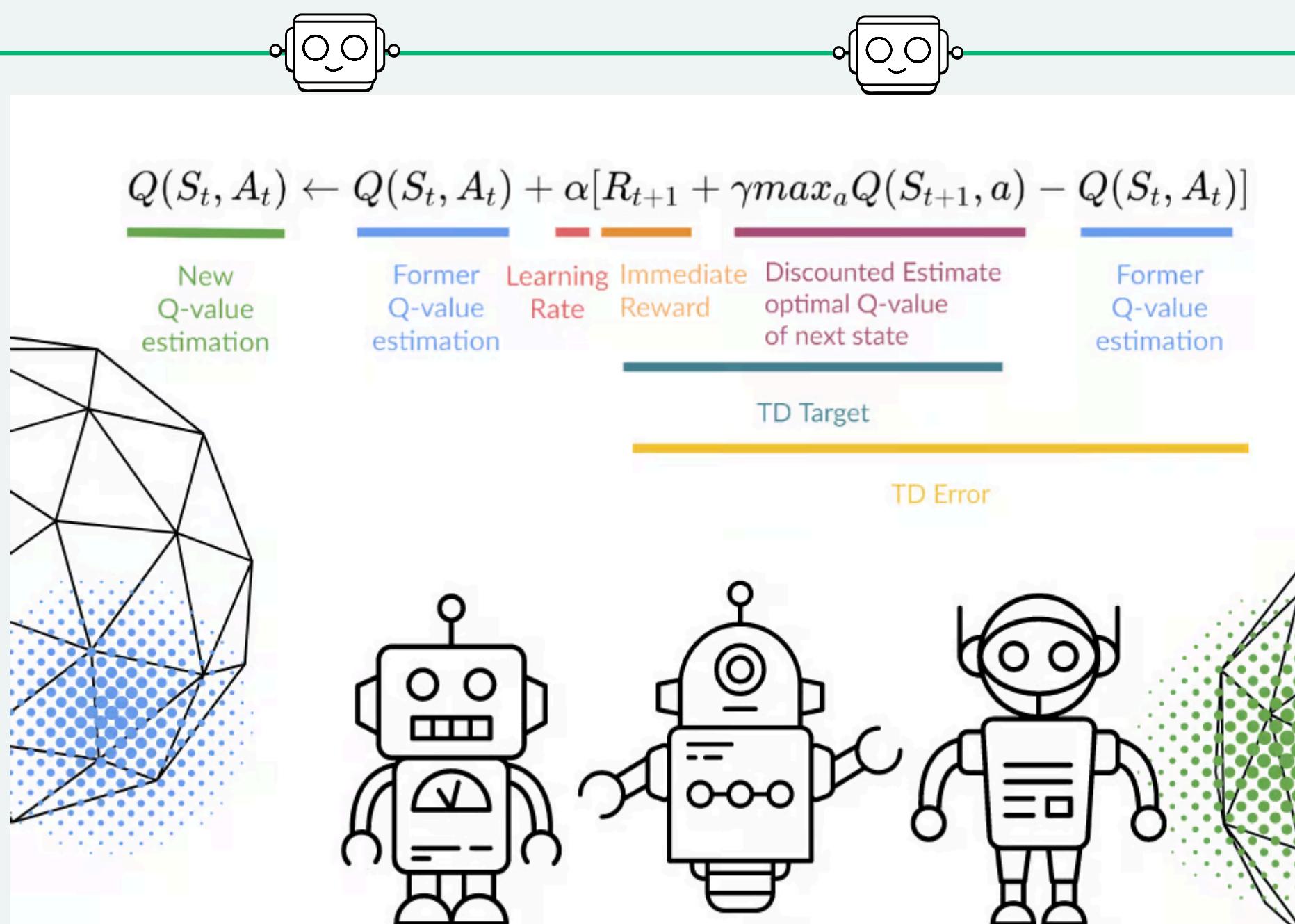
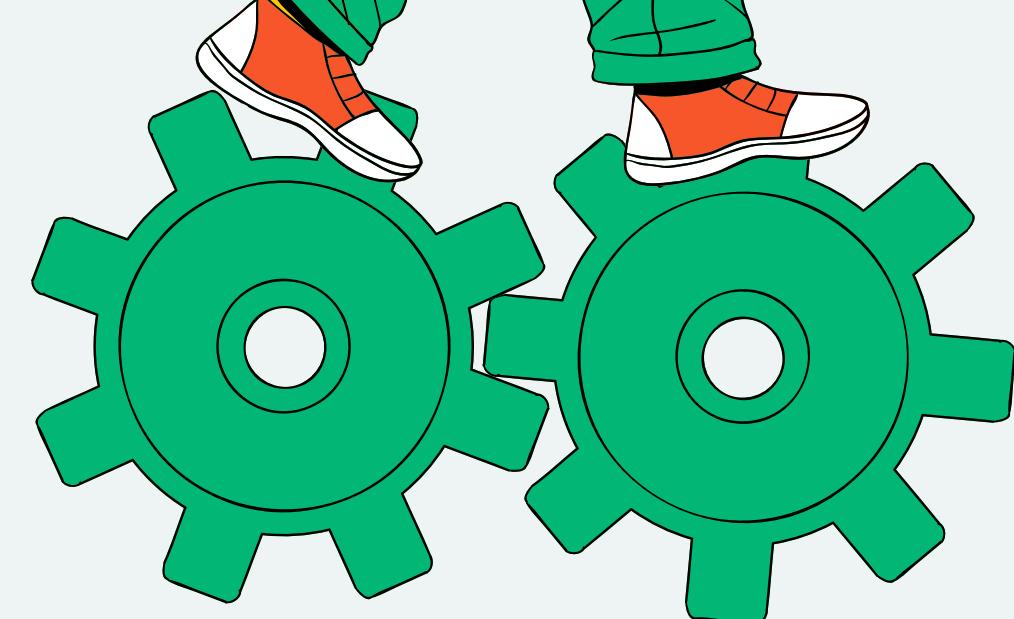
L'obiettivo dell'algoritmo è trovare la politica ottimale π ovvero una mappa che associa ad ogni stato S l'azione migliore A, massimizzando la ricompensa.



IMPLEMENTAZIONE Q-LEARNING



AGGIORNAMENTO Q- VALUE



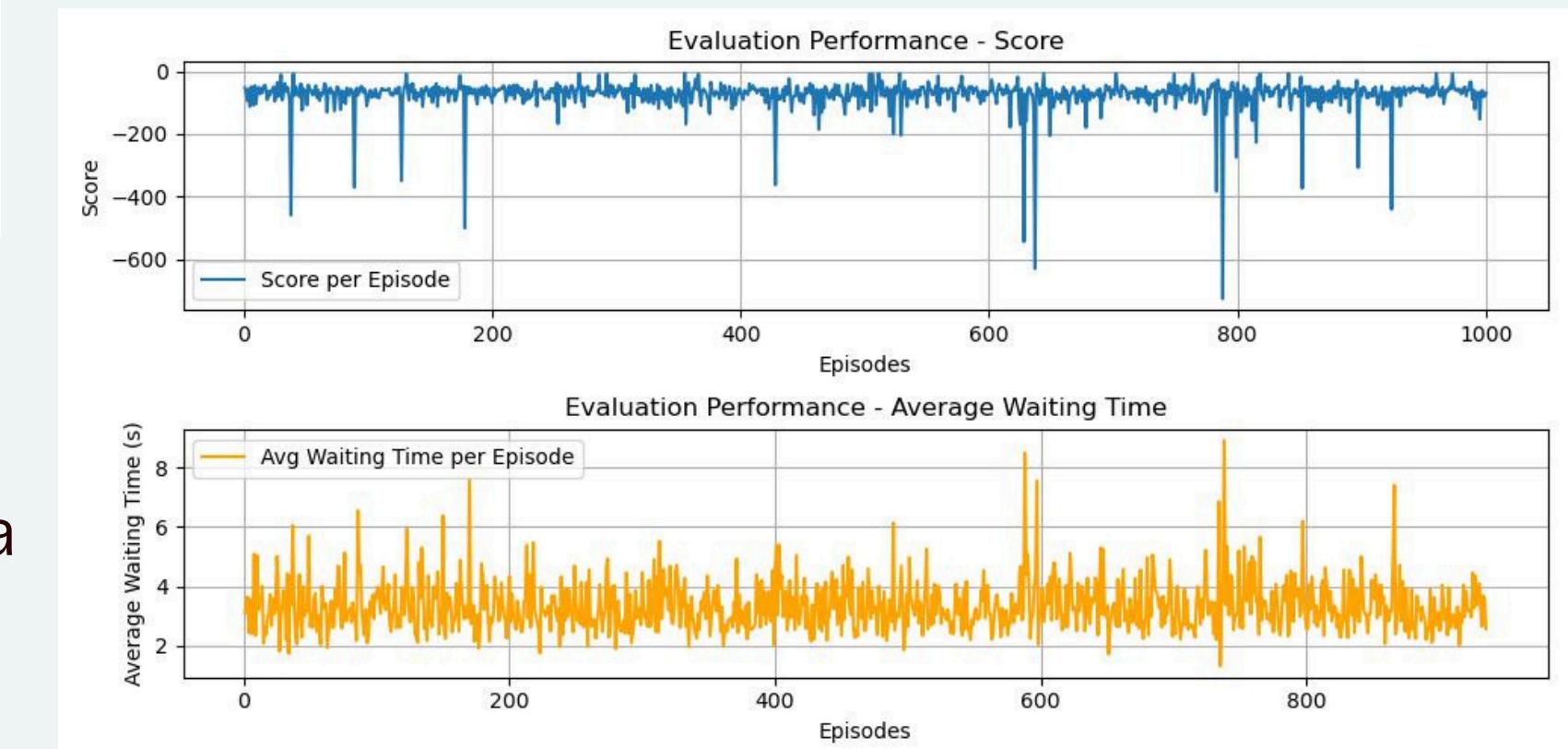
Aggiorna il valore $Q(s, a)$, che rappresenta la qualità di un'azione A in uno stato S , combinando la ricompensa immediata $R+1$ e la previsione futura, rappresentata da $\max Q(s, a')$. Il termine di aggiornamento, noto come errore di previsione (TD error), misura la differenza tra il valore attuale di Q e la nuova stima basata sulla ricompensa e sul valore futuro. Questo errore viene corretto tramite il tasso di apprendimento α e il fattore di sconto γ , che rispettivamente controllano quanto il nuovo valore influenza il precedente e quanto le ricompense future sono importanti.

TRAINING E VALIDATION



Validazione dell'algoritmo su 1000 episodi. Il punteggio è vicino allo 0 con alcuni picchi molto bassi. Il tempo medio di attesa oscilla fra i 2 e i 4 secondi.

Training su 300 mila episodi. I punteggi e il tempo medio di attesa misurato sono abbastanza instabili.



IMPLEMENTAZIONE DEEP Q-LEARNING

01

INTRODUZIONE

Il Deep Q-Learning è un algoritmo off-policy, il che significa che l'agente apprende una politica ottimale basandosi su esperienze passate, indipendentemente dalle azioni effettivamente scelte.

02

RETE NEURALE

In Deep Q-Learning, una rete neurale approssima la funzione $Q(s, a)$, permettendo di gestire spazi di stati e azioni troppo grandi per una tabella. La rete stima il valore di ogni azione per uno stato dato e viene aggiornata durante l'apprendimento.

03

CONVERGENZA

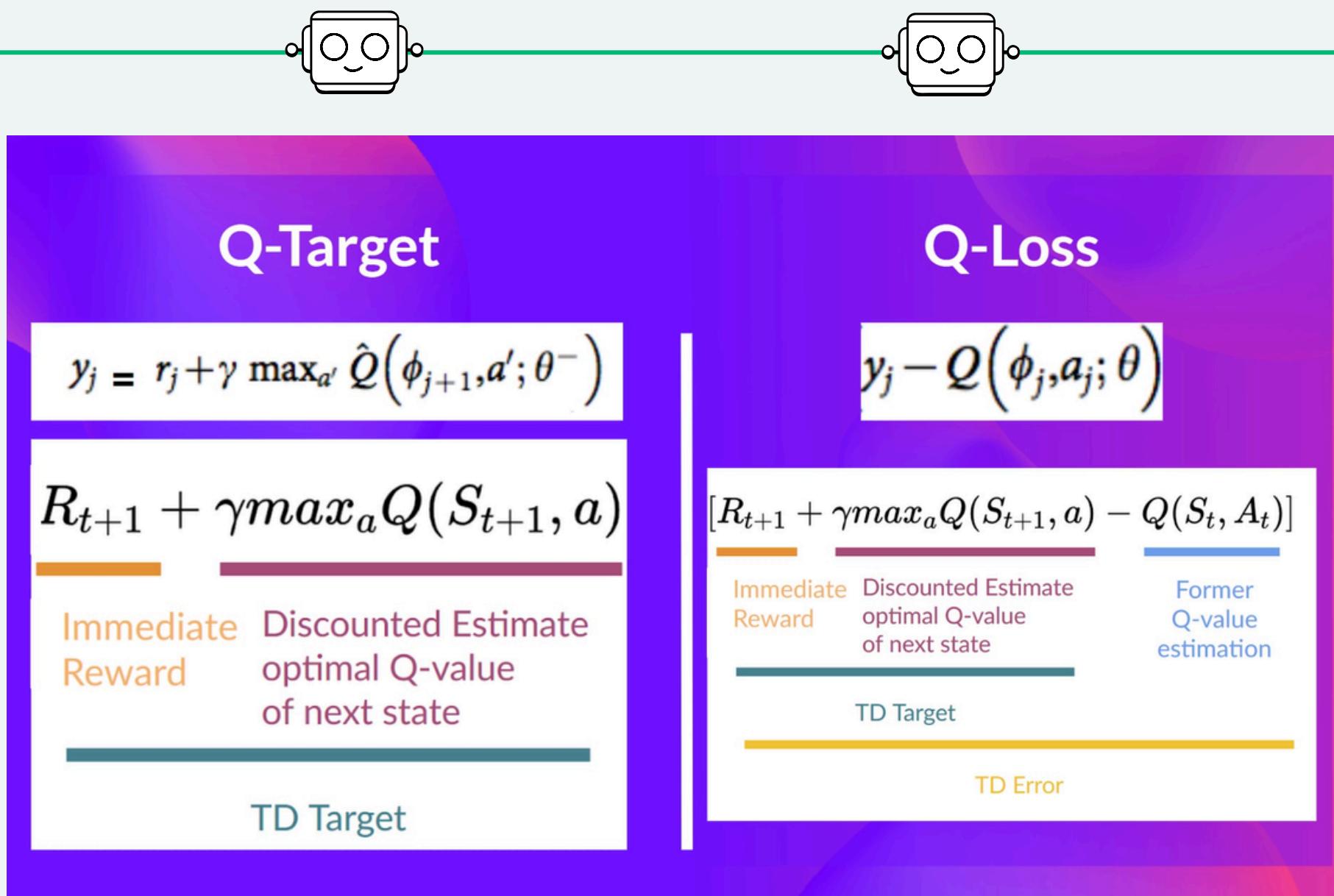
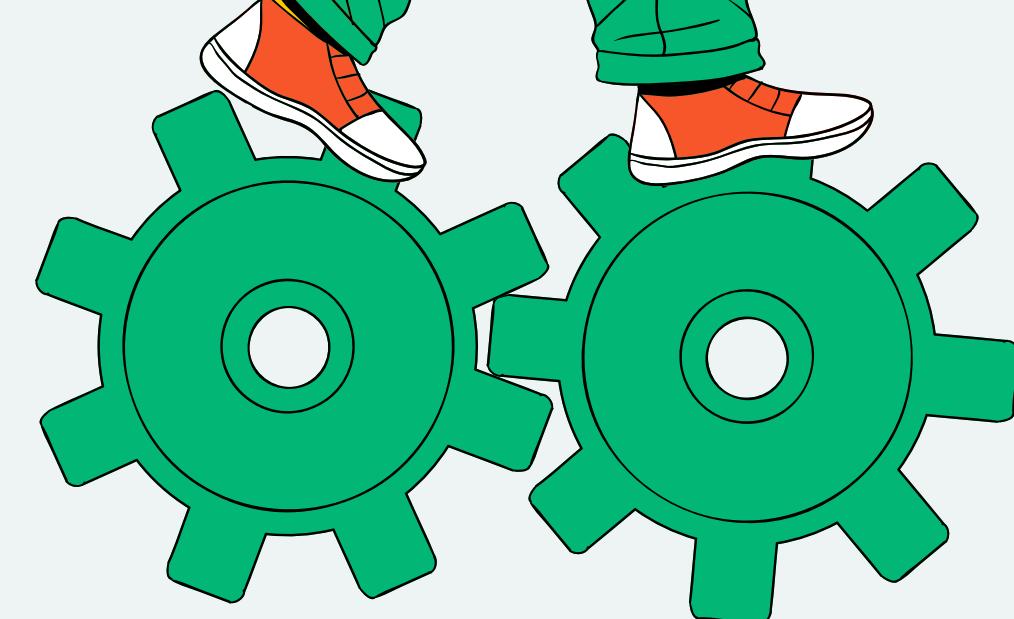
L'obiettivo del Deep Q-Learning è apprendere la migliore politica per massimizzare la ricompensa cumulativa a lungo termine basandosi sulle stime della funzione $Q(s, a)$. L'algoritmo cerca di affinare questa politica aggiornando continuamente la rete neurale che approssima la funzione Q , riducendo l'errore tra il valore attuale delle azioni e il valore target derivante dalle ricompense future.



IMPLEMENTAZIONE DEEP Q-LEARNING

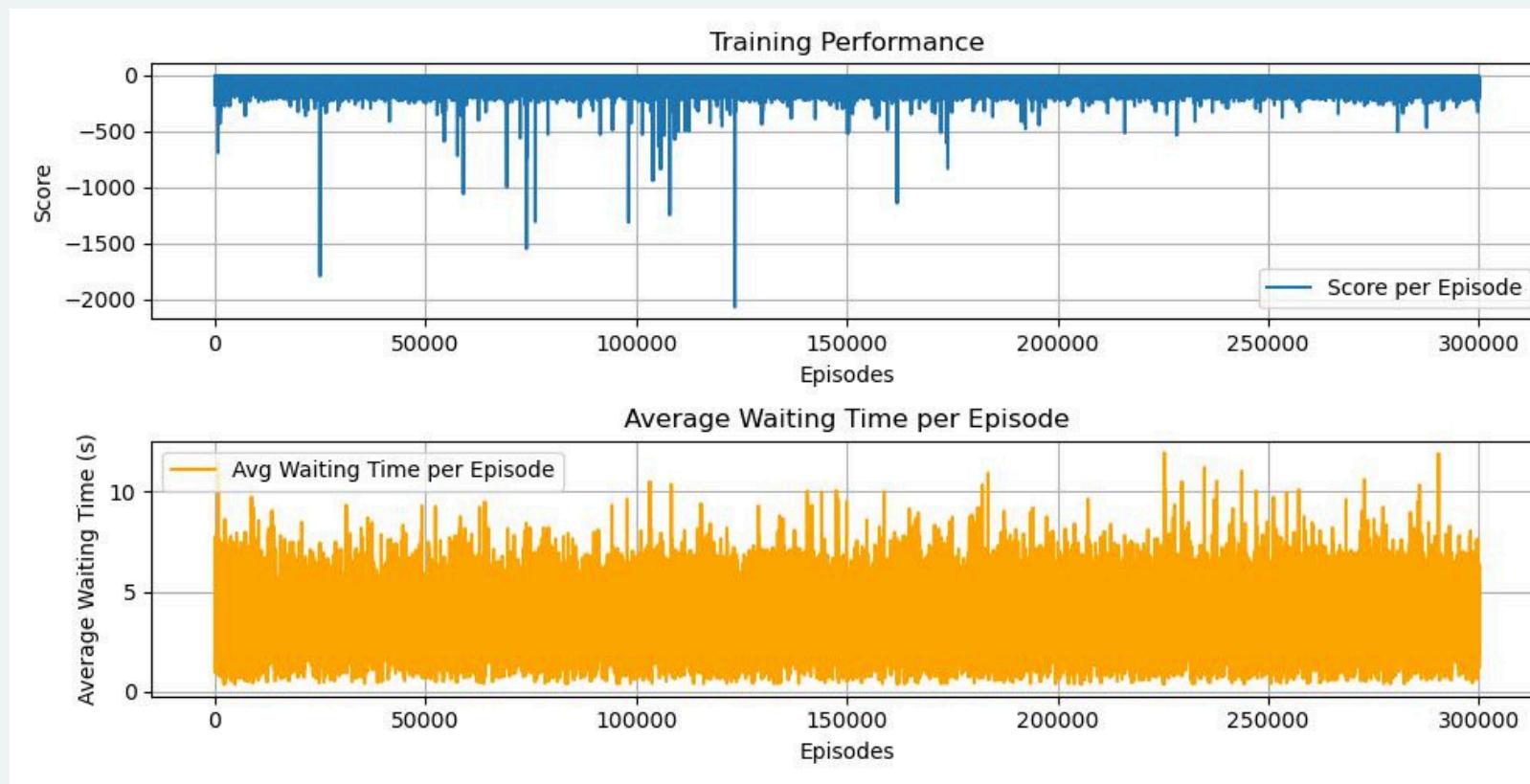


AGGIORNAMENTO DELLA RETE



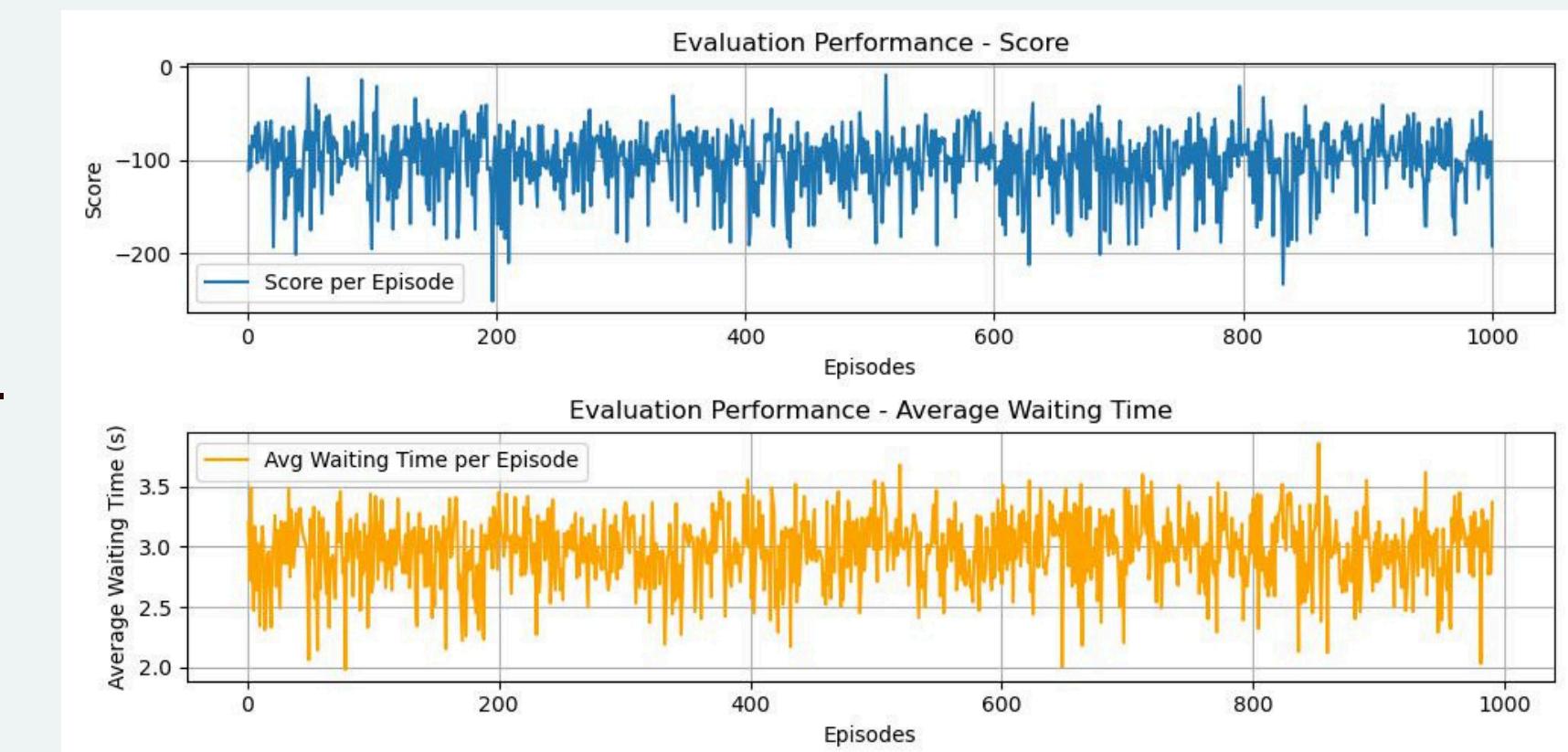
Nel Deep Q-learning, il Q-target è la stima della ricompensa futura massima che un agente può ottenere a partire da un dato stato e una data azione. La loss misura la differenza tra il valore Q predetto dalla rete neurale e il Q-target. L'algoritmo cerca di minimizzare questa differenza aggiornando i pesi della rete, in modo che la previsione fatta dalla rete si avvicini sempre di più al Q-target. Questo procedimento aiuta l'agente ad avvicinarsi alla politica ottimale perché, minimizzando la loss, la rete neurale impara a prevedere correttamente il valore delle azioni in ogni stato. Aggiornando continuamente i pesi in base alla differenza tra la previsione e il Q-target, l'agente diventa in grado di selezionare le azioni che massimizzano la ricompensa cumulativa futura scegliendo così la migliore strategia.

TRAINING E VALIDATION



Validazione dell'algoritmo su 1000 episodi. Il punteggio è vicino ai -100. Il tempo medio di attesa per episodio è molto stabile, si muove fra i 2.5 e 3.5 secondi.

Training su 300 mila episodi. I punteggi e il tempo medio di attesa misurato sono abbastanza instabili.



IMPLEMENTAZIONE SARSA

01

INTRODUZIONE

SARSA (State–Action–Reward–State–Action) è un algoritmo di reinforcement learning che utilizza un approccio on–policy. Quindi aggiorna i valori Q in base all'azione che l'agente effettivamente sceglie nel prossimo stato, seguendo la propria politica attuale.

02

Q VALUE

In SARSA, l'aggiornamento dei Q-value avviene considerando l'azione effettivamente scelta dall'agente nel prossimo stato. Il valore $Q(s, a)$ viene aggiornato in base alla ricompensa ricevuta e al valore Q dell'azione scelta nel nuovo stato, seguendo la politica corrente dell'agente.

03

CONVERGENZA

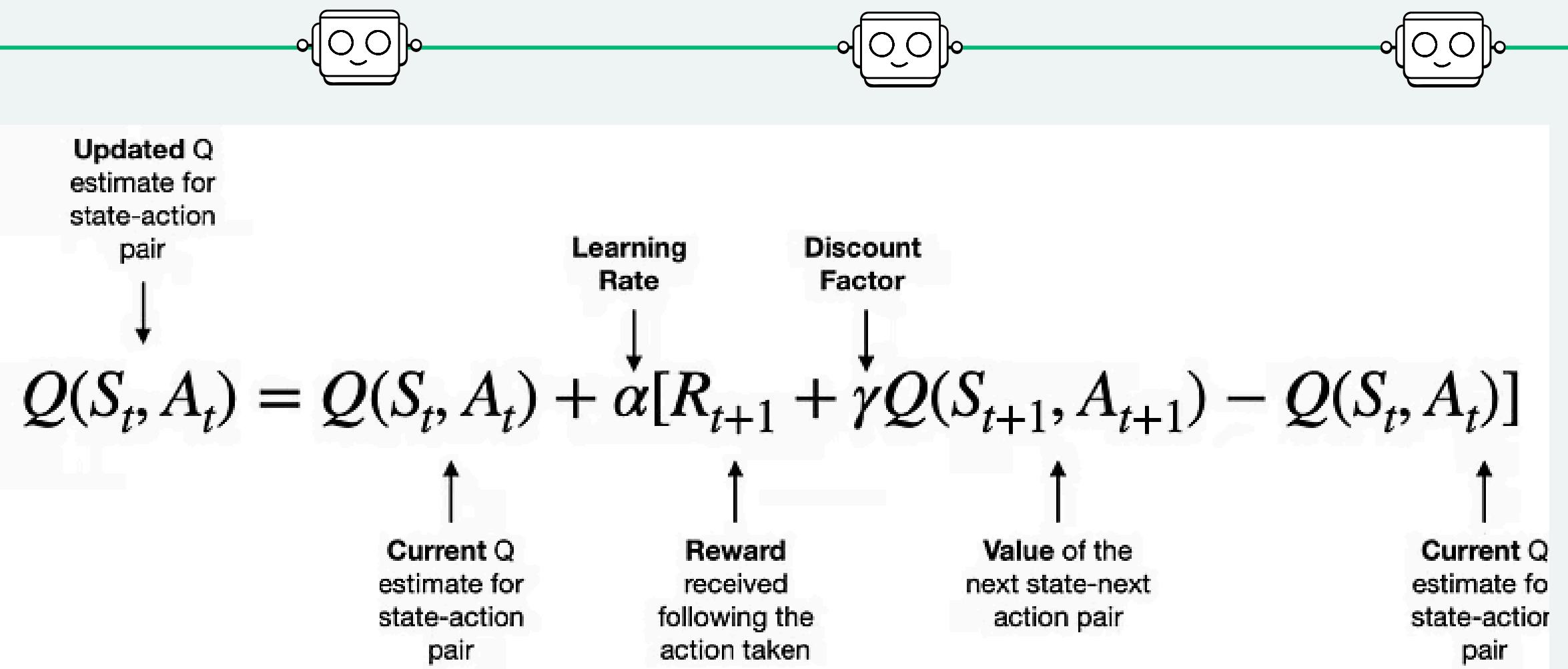
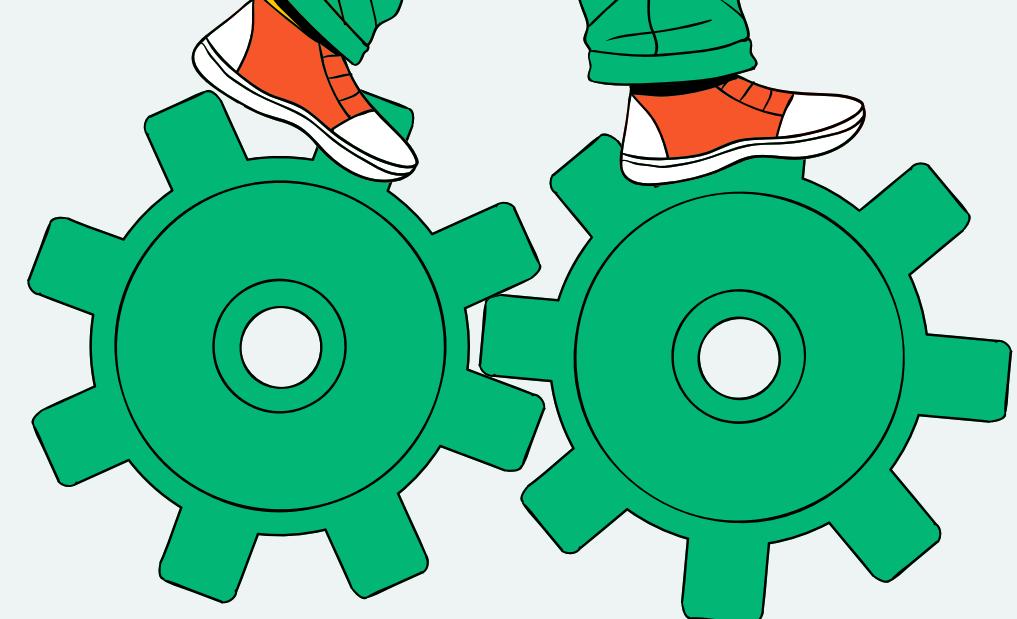
L'algoritmo mira a ottimizzare il ritorno cumulativo delle ricompense a lungo termine, aggiornando progressivamente i valori Q per trovare una politica che massimizza le ricompense attese.



IMPLEMENTAZIONE Q-LEARNING



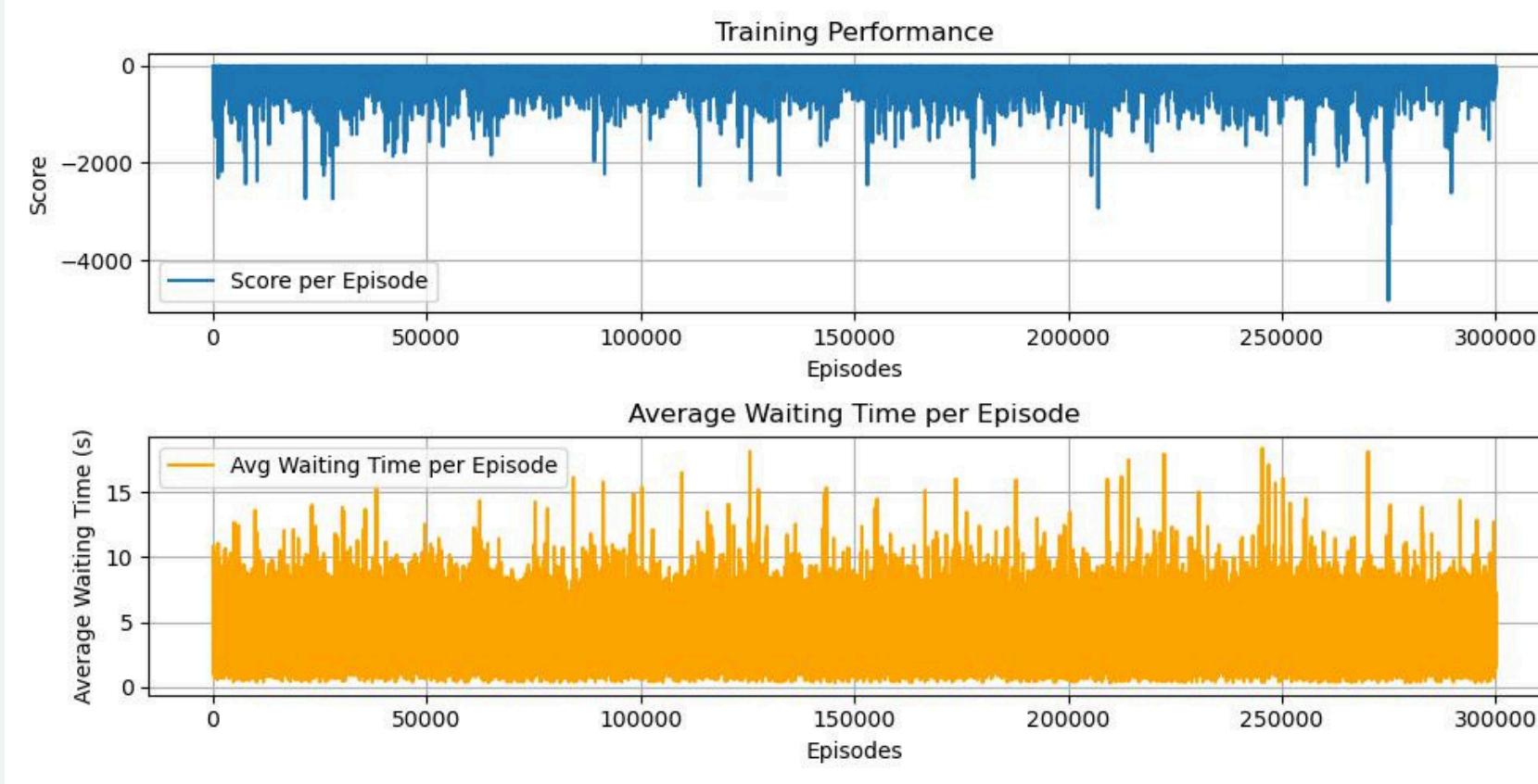
AGGIORNAMENTO Q- VALUE



L'obiettivo di SARSA è migliorare la politica dell'agente in modo on-policy, aggiornando i Q-value in base alle azioni effettivamente scelte dall'agente. Dopo aver eseguito un'azione in uno stato, l'agente osserva la ricompensa ricevuta e il nuovo stato. La prossima azione viene scelta secondo la politica corrente. Il valore Q viene aggiornato utilizzando la ricompensa immediata e il valore della futura azione scelta, affinando così progressivamente la politica in base alle esperienze reali, non a previsioni ottimali.

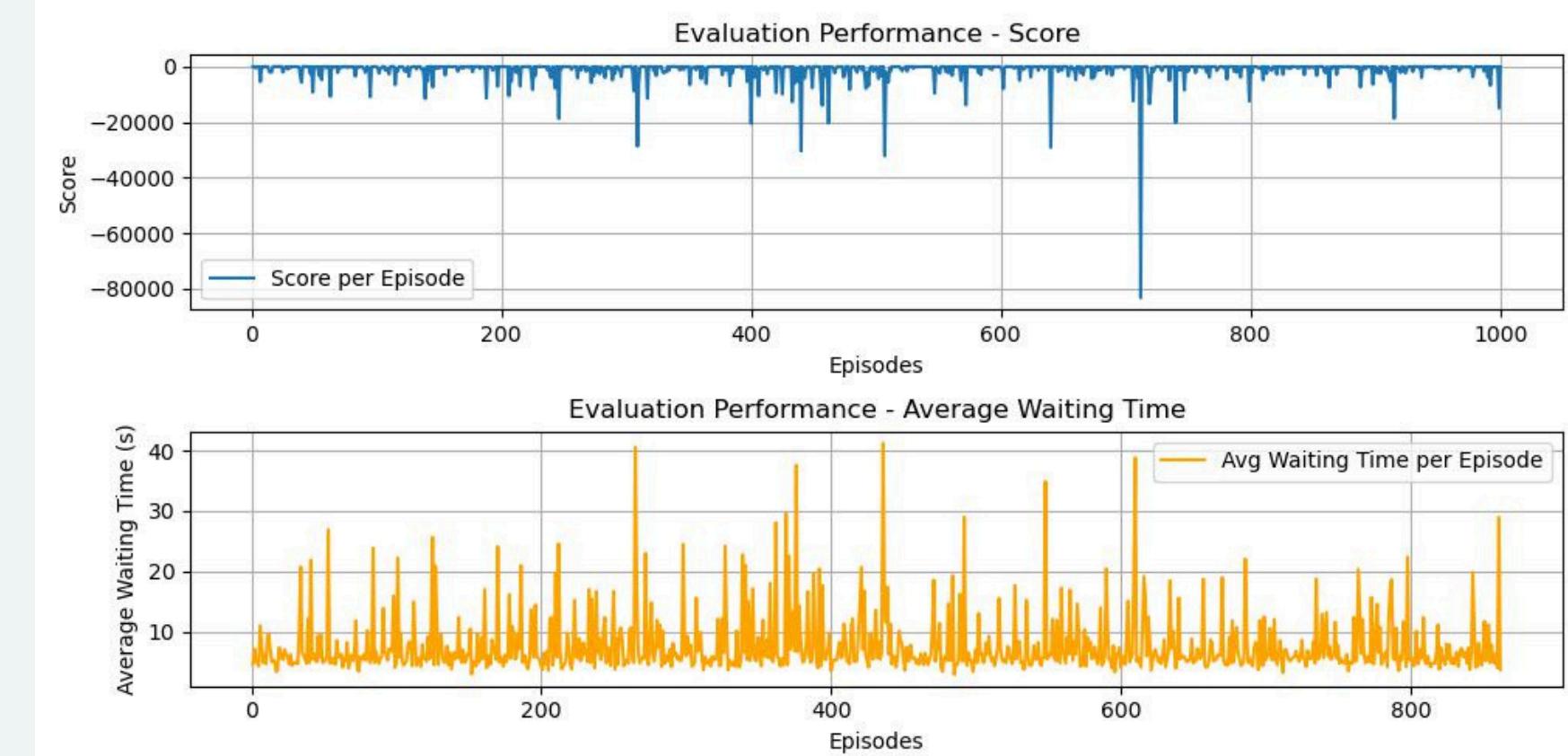


TRAINING E VALIDATION



Validazione dell'algoritmo su 1000 episodi. Il punteggio oscilla in range molto ampi. Lo stesso avviene anche per la media di attesa globale.

Training su 300 mila episodi. I punteggi e il tempo medio di attesa misurato sono del tutto instabili. I picchi sono molto ampi.

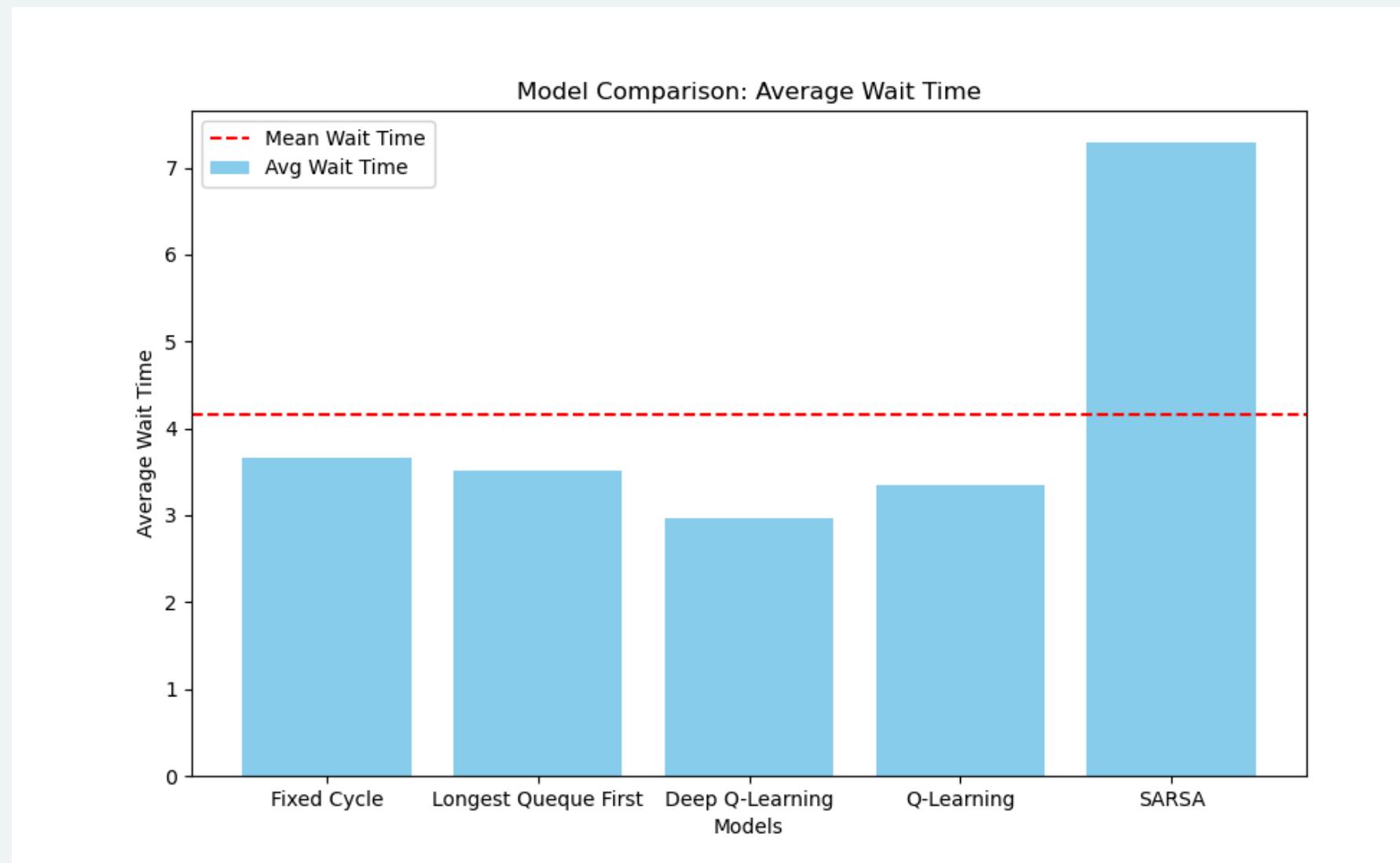


CONFRONTI

MODELLO	TEMPO MEDIO DI ATTESA	COLLISIONI
FIXED CYCLE	3.56	0.05
LONGEST QUEQUE FIRST	3.54	0.05
DEEP Q-LEARNING	2.99	0.01
Q-LEARNING	3.29	0.04
SARSA	7.80	0.13



CONCLUSIONI



Il confronto evidenzia che il Deep Q-Learning è il modello più promettente, garantendo tempi di attesa ridotti e un basso tasso di collisioni. Al contrario, il SARSA mostra prestazioni significativamente peggiori, risultando meno adeguato per l'ottimizzazione del traffico. I modelli tradizionali come Fixed Cycle e Longest Queue First offrono una stabilità accettabile ma non riescono a competere con l'efficienza dei metodi di apprendimento più avanzati.





THANK
you

