universität freiburg

Exploring DQN Extensions on MinAtari-Breakout

RL Laboratory

Group 2

Presentors : Renu Pal, Thejaswini Raju, Julian Wiedermann

Freiburg, 6th February 2026

INTRODUCTION

DQN - Deep Q-Network

- DQN combines Q-learning with deep convolutional neural networks and experience replay
- Handles large and complex state spaces.
- **Problem:** Overestimation Bias, Training Instability
- Extensions:
    - Double DQN
    - Multi-step DQN
    - Dueling DQN

DDQN - Double Deep Q-Network

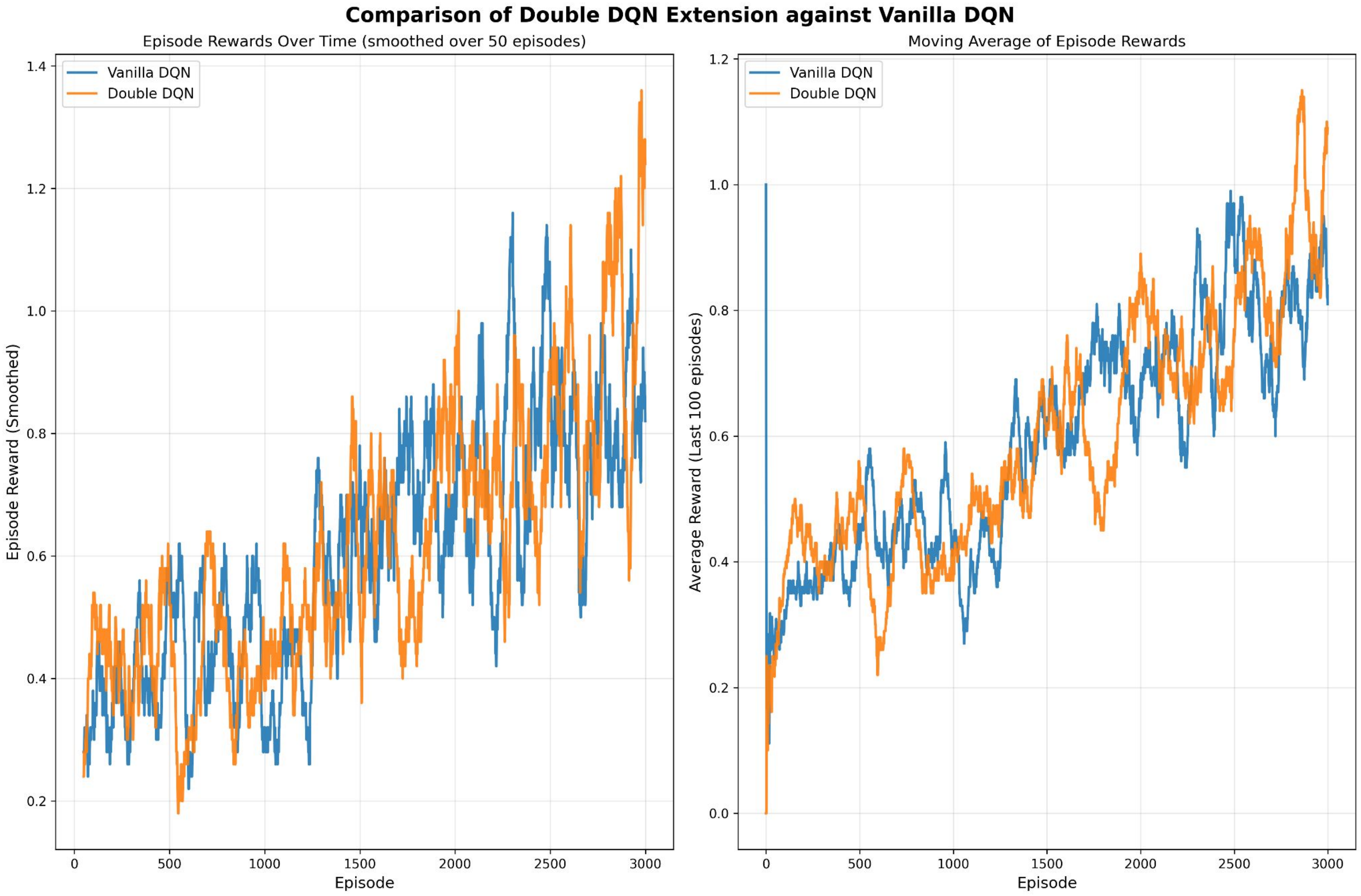
- Fixes overestimation bias in DQN
- Key idea:
    - **Online Network** to select actions
    - **Target Network** to evaluate actions

- Double DQN Target Update

$$Y_t^{\mathrm{Q}} = R_{t+1} + \gamma Q(S_{t+1}, \operatorname*{argmax}_a Q(S_{t+1}, a; \boldsymbol{\theta}_t); \boldsymbol{\theta}_t)$$

- More stable and accurate Q-value estimates → reduces over optimistic updates.

DDQN vs DQN



Comparison of Double DQN Extension against Vanilla DQN
Episode Rewards Over Time (smoothed over 50 episodes)
Vanilla DQN
Double DQN
Episode Reward (Smoothed)
Episode
0
500
1000
1500
2000
2500
3000
0.2
0.4
0.6
0.8
1.0
1.2
1.4
Moving Average of Episode Rewards
Vanilla DQN
Double DQN
Average Reward (Last 100 episodes)
Episode
0.0
0.2
0.4
0.6
0.8
1.0
1.2

Multistep DQN

- Idea: Instead of using only the intermediate reward, we use the reward of the next n steps.
- Instead of using the standard TD target:

$$R_{t+1} + \gamma \max_{a'} Q_\theta(s_{t+1}, a')$$

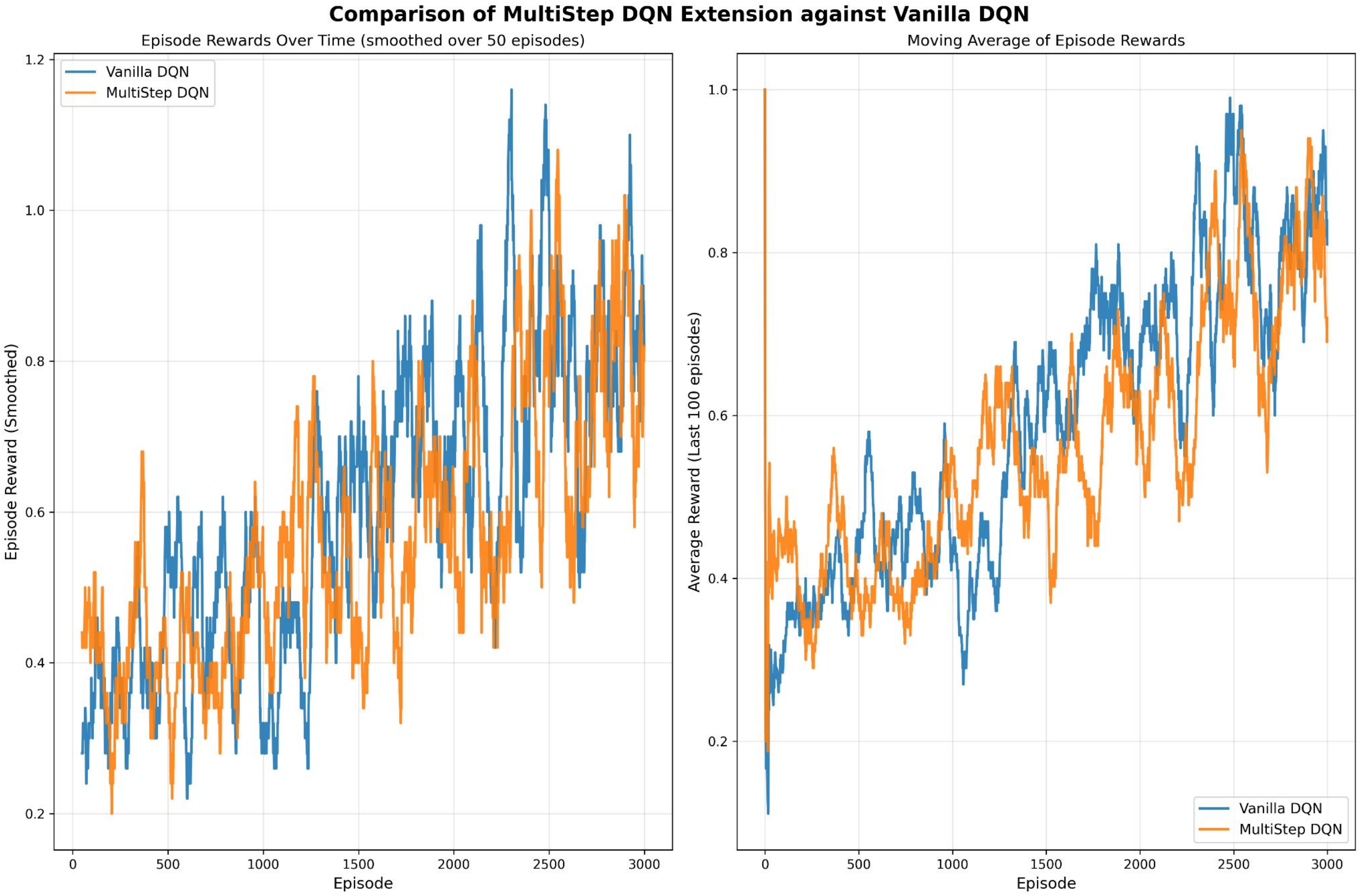
- We use the n-step return:

$$R_t^{(n)} + \gamma^{(n)} \max_{a'} Q_\theta(s_{t+n}, a')$$

- Where:

$$R_t^{(n)} = \sum_{k=0}^{n-1} \gamma^k R_{t+k+1} = R_{t+1} + \gamma R_{t+2} + \dots + \gamma^{n-1} R_{t+n}$$

Multi-step DQN vs DQN



Dueling DQN

Standard DQN estimates Q-values directly:

$$Q(s, a; \theta) \approx \text{Expected return for action } a \text{ in state } s$$

Problem: For states where action choice matters little, learning Q-values for every action separately is inefficient.

Key Idea behind Dueling :

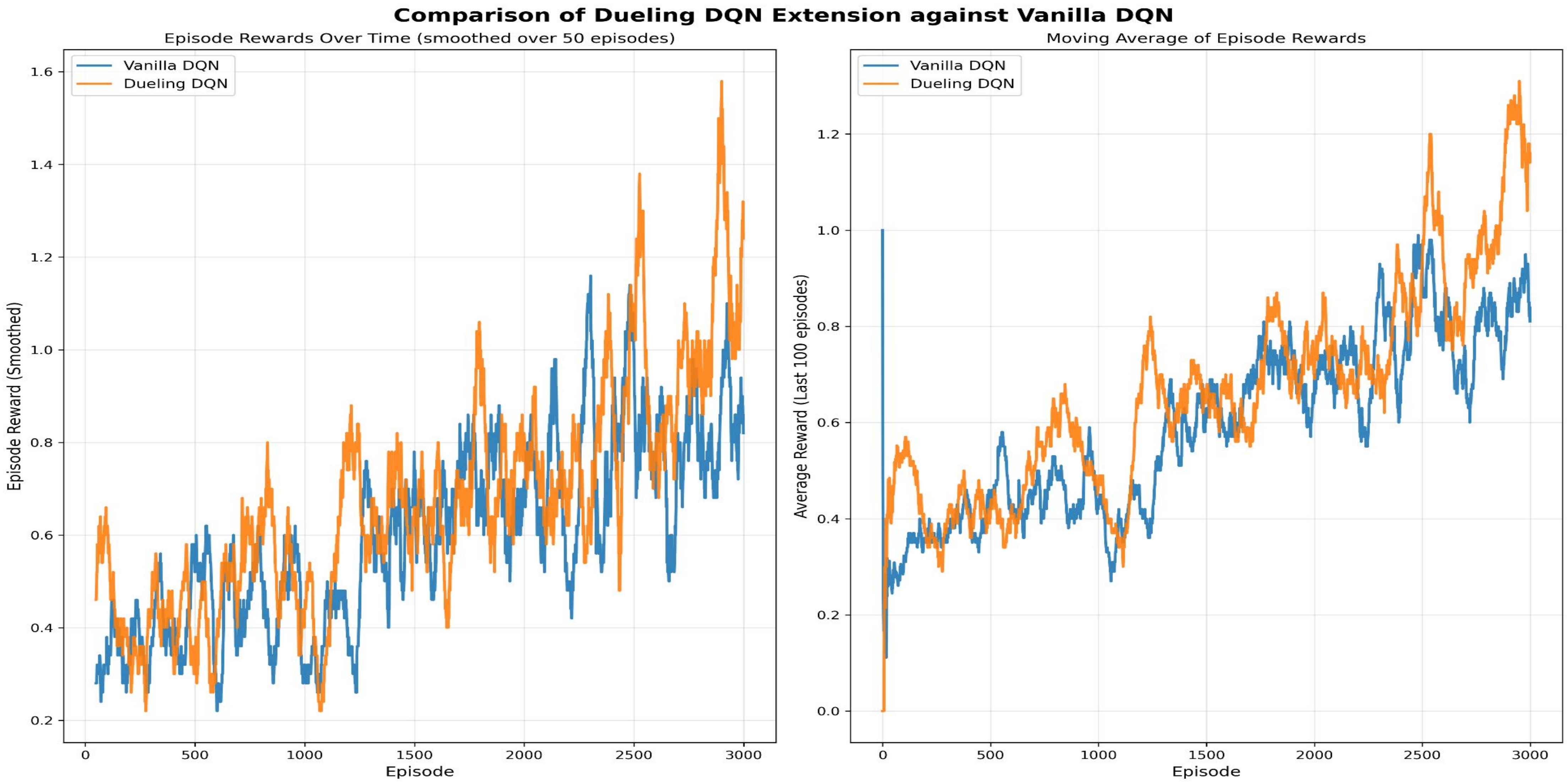
1. **Separate the Q-value into:**
   a. **State-value Stream: V(s;θ,β)** — how good it is to be in state s
   b. **Advantage Stream : A(s,a;θ,α)** — how much better action a is than the average

2. **Aggregate streams to get Q-value:**

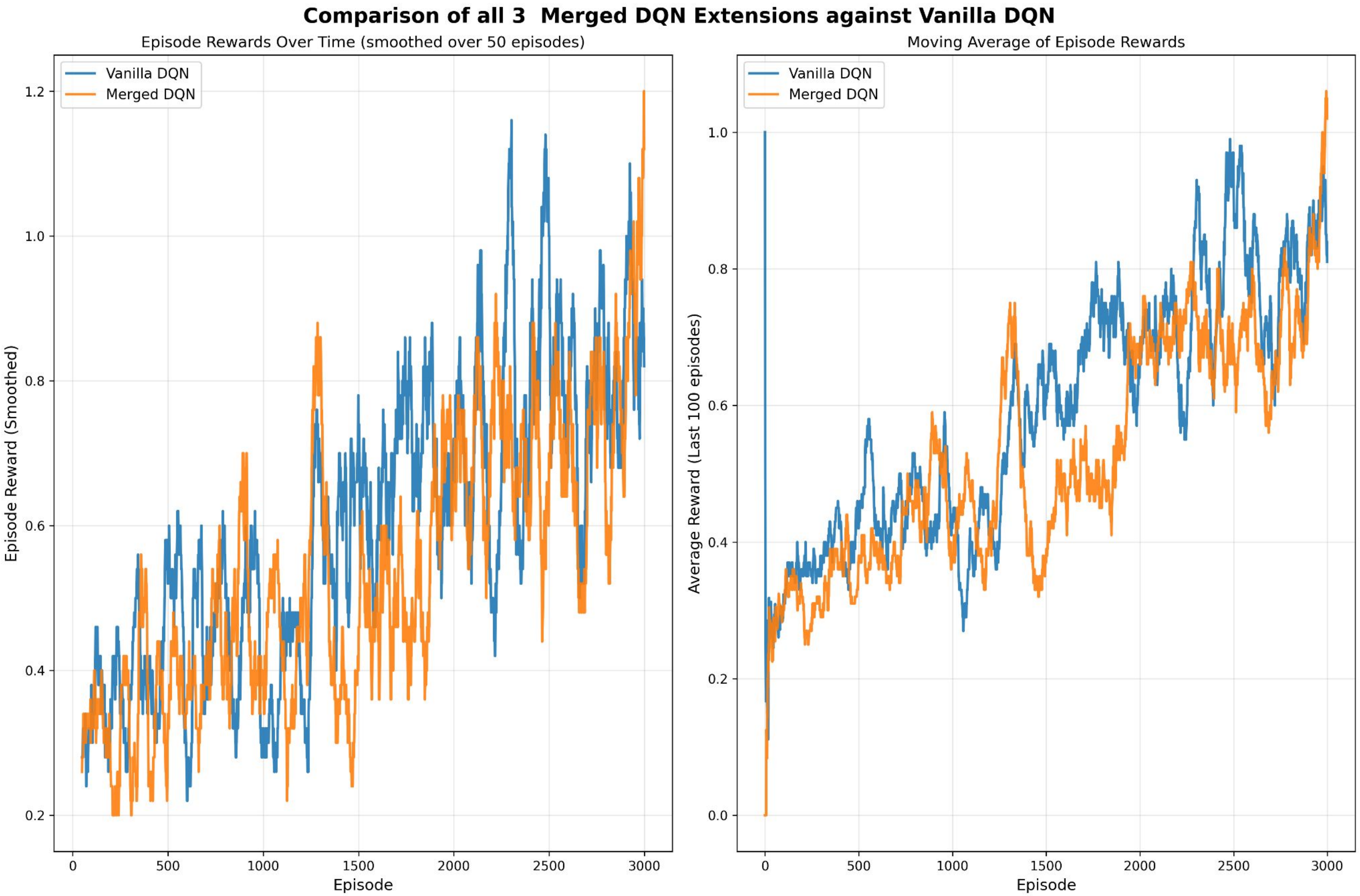
$$Q(s, a; \theta, \alpha, \beta) = V(s; \theta, \beta) + \left(A(s, a; \theta, \alpha) - \frac{1}{|\mathcal{A}|} \sum_{a'} A(s, a'; \theta, \alpha)\right)$$

where |A| is the number of possible actions.

Dueling extension Vs Vanilla DQN



Merging all three extensions



Results

Training Performance (Last 100 Episodes):

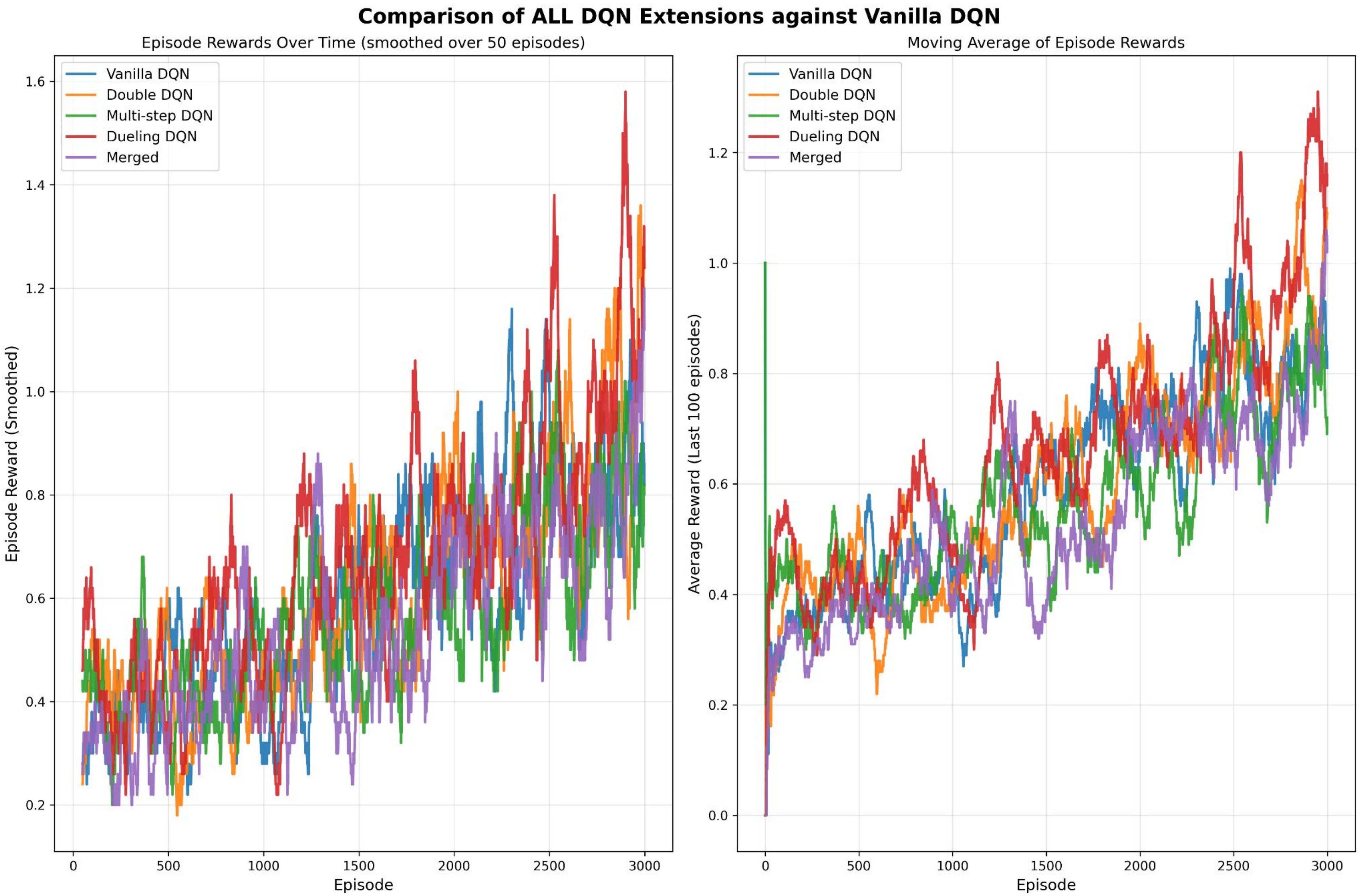
Agent	Mean	Std	Min	Max
Vanilla DQN	0.81	1.05	0.00	6.00
Double DQN	1.09	1.33	0.00	6.00
Multi-step DQN	0.72	0.97	0.00	4.00
Dueling DQN	1.16	1.26	0.00	6.00
Merged	1.02	1.09	0.00	5.00

Test Performance:

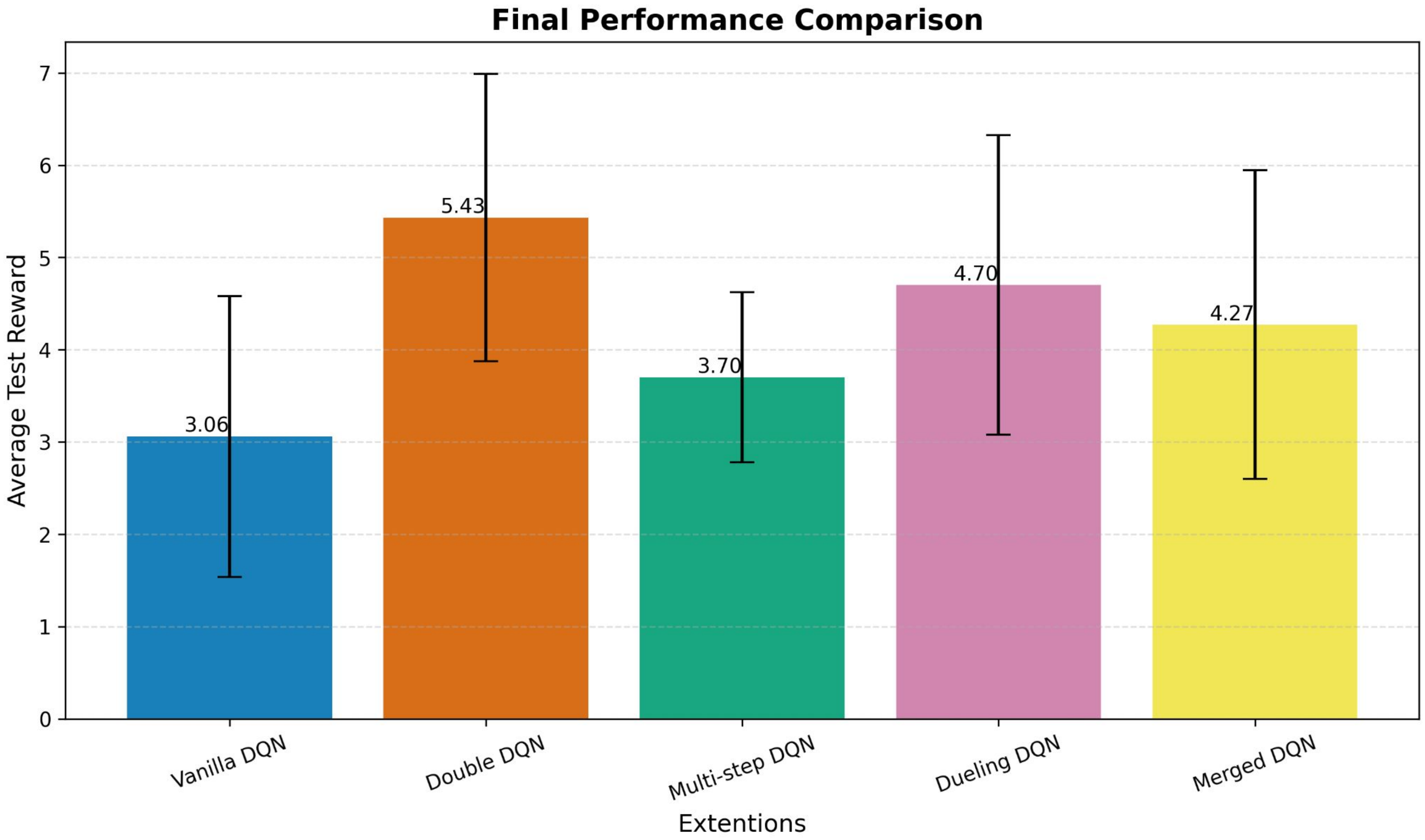
Agent	Mean	Std	Min	Max	
Vanilla DQN	3.06	1.52	0.00	6.00	
Double DQN	5.43	1.56	0.00	9.00	(+77.5%)
Multi-step DQN	3.70	0.92	2.00	7.00	(+20.9%)
Dueling DQN	4.70	1.62	2.00	7.00	(+53.6%)
Merged DQN	4.27	1.67	0.00	9.00	(+39.5%)

- Dueling DQN achieved the highest average reward during **training,**
- Double DQN achieved the best **test performance** with a 77.5% improvement over Vanilla DQN.
- Multi-step DQN showed lower variance but smaller gains, indicating improved stability rather than peak performance.
- Combining extensions did not lead to additive improvements.

Results


Comparison of ALL DQN Extensions against Vanilla DQN
Episode Rewards Over Time (smoothed over 50 episodes)
Vanilla DQN
Double DQN
Multi-step DQN
Dueling DQN
Merged
Episode Reward (Smoothed)
Episode
Moving Average of Episode Rewards
Average Reward (Last 100 episodes)

Conclusion



- Double DQN outperforms in average reward.
- Dueling DQN and Merged DQN also improve over Vanilla DQN.

Thank you

Questions ?