Transfer Learning in Deep Q-Networks: A Comparative Study of DDQN and Dueling DQN Across Benchmark Environments

Fatima Dossa
Dhanani School of Science and Engineering
Habib University
Karachi, Pakistan

Iqra Ahmed
Dhanani School of Science and Engineering
Habib University
Karachi, Pakistan

Azkaa Nasir
Dhanani School of Science and Engineering
Habib Univeristy
Karachi, Pakistan

Deep reinforcement learning (DRL) has advanced significantly with improvements like Double Deep Q-Networks (DDQN) and Dueling DQN, which address overestimation bias and generalization challenges. While these architectures excel in single-task settings, their effectiveness in transfer learning—particularly in adapting to new environments—remains underexplored. This study evaluates DDQN and Dueling DQN in transfer learning between CartPole and LunarLander, two benchmark environments with differing complexities. Our results show that Dueling DQN achieves faster convergence in CartPole due to its decoupled value-advantage learning, while DDQN offers greater training stability. However, transferring learned policies to LunarLander proved ineffective due to hardware constraints, environmental disparities, and computational inefficiencies. Despite these challenges, our findings highlight key trade-offs between learning speed and robustness, providing insights for future work on cross-domain transfer in DRL.

I. Introduction

Reinforcement learning (RL) has made remarkable progress with deep neural networks, enabling agents to solve complex tasks in dynamic environments. Among value-based methods, Deep Q-Networks (DQN) and its extensions—Double DQN (DDQN) and Dueling DQN—have been pivotal in addressing challenges such as overestimation bias and inefficient generalization. DDQN mitigates overestimation by decoupling action selection and evaluation, while Dueling DQN enhances generalization by separating state-value and advantage estimation. While these improvements have been studied in isolation, their comparative advantages in transfer learning—where pretrained models adapt to new tasks—remain unclear.

This study investigates the transfer learning efficiency of DDQN and Dueling DQN across CartPole and LunarLander, two environments with distinct complexities. CartPole, a simple balancing task, serves as the source domain, while LunarLander, a more complex control problem, tests the agents' ability to transfer knowledge. We analyze key metrics such as convergence speed, reward stability, and generalization to assess which architecture better facilitates adaptation.

However, our experiments faced significant challenges, including hardware limitations, training interruptions, and the inherent difficulty of transferring policies between dissimilar environments. While Dueling DQN demonstrated faster learning in CartPole, its performance was less stable than DDQN's. Additionally, attempts to transfer learning to LunarLander were hindered by computational constraints, preventing full convergence. These limitations highlight the practical difficulties of cross-domain transfer in DRL and underscore the need for more robust training methodologies.

Our findings contribute to the broader discussion on DRL generalization by:

- Comparing DDQN and Dueling DQN in both single-task and transfer learning scenarios.
- Identifying key challenges in policy transfer between environments with differing dynamics.
- Providing insights into the trade-offs between learning speed and stability in DRL architectures.

This work informs future research directions, including hybrid architectures, meta-learning for adaptive hyperparameter tuning, and improved computational strategies for scalable RL training. By addressing these challenges, DRL can move closer to real-world applications where adaptability and robustness are critical.

II. Literature Review

A. *Critical Analysis*

The paper "Rainbow: Combining Improvements in Deep Reinforcement Learning" makes significant contributions by successfully integrating six independent extensions to the DQN algorithm into a single agent named Rainbow1. These extensions include Double DQN (DDQN) to address overestimation bias, Prioritized Experience Replay to improve sample efficiency by focusing on high-potential transitions, Dueling Networks to enhance generalization across actions, Multi-step Learning to accelerate learning through faster reward propagation, Distributional Q-learning to learn the distribution of returns for a richer signal, and Noisy Nets to facilitate exploration. A key contribution is achieving new state-of-the-art performance on the Atari 2600 benchmark

suite, demonstrating both data efficiency and high final performance. The paper also empirically validates the complementarity of these extensions and provides a detailed ablation study revealing the contribution of each component, with prioritized replay and multi-step learning found to be particularly crucial. Furthermore, the generality of Rainbow is highlighted by its use of a single set of hyper-parameters across all 57 Atari games [1].

The strengths of this work lie in its principled integration of diverse techniques with clear explanations and necessary modifications. It demonstrates a substantial performance improvement over DQN and other baselines. The in-depth ablation studies offer valuable insights into the effectiveness and interactions of each component. Additionally, the paper emphasizes reproducibility by adhering to standard evaluation procedures and providing comprehensive hyper-parameter details and learning curves. The limitations include the limited hyper-parameter tuning due to the vast search spaces, the significant computational resources required for training, its primary focus on value-based methods within the Q-learning family, and the use of domain-specific modifications for Atari games, which might affect generalizability [1].

In terms of addressing common challenges, Rainbow tackles overestimation bias with DDQN by decoupling action selection and evaluation, though its impact was limited in their specific setup possibly due to reward clipping. Instability is addressed through target networks, experience replay, prioritized experience replay, and distributional Q-learning, each contributing to a more stable learning process11. The sample efficiency is significantly improved by prioritized experience replay, multi-step learning, dueling networks, and distributional Q-learning, allowing the agent to learn more effectively from fewer experiences [1].

The next paper ”Enhancing Two-Player Performance Through Single-Player Knowledge Transfer: An Empirical Study on Atari 2600 Games” contributes by demonstrating that knowledge transfer from single-player Atari environments (Gymnasium) to corresponding two-player environments (PettingZoo) can improve performance and reduce training time. The paper also proposes a method for quantifying and visualizing the complexity of the Atari RAM state and explores its correlation with transferred agent performances [2].

The strengths of this study include the empirical demonstration of effective transfer learning across ten Atari games, offering a potential solution for training in complex two-player settings. It also highlights the reduction in training time achieved through transfer learning and freezing initial network layers and introduces a novel method for RAM complexity analysis. The use of standard libraries like Gymnasium and PettingZoo enhances the accessibility and reproducibility of the work [2]. However, limitations include its restriction to ten Atari games with visible player avatars and mostly symmetric graphics, potential limitations due to hardware-constrained training steps, its focus solely on the DQN algorithm, and the weak to moderate correlation found between RAM complexity and transfer performance [2].
As for common challenges, this work addresses instability in two-player training, which often arises from the non-stationary nature of self-play, by transferring knowledge from a stable single-player environment where the target policy is fixed. The sample efficiency is potentially improved by pre-training in the single-player setting, allowing for faster learning of useful features, and by freezing initial layers during two-player training. The challenge of agent indication in multi-agent settings is tackled by proposing a method to annotate and swap RAM parts related to each player [2].

The next paper is a survey paper titled ”Transfer Learning in Deep Reinforcement Learning”. It provides a comprehensive and up-to-date review of transfer learning approaches within deep reinforcement learning over the last decade. Its main contribution is offering a systematic framework for categorizing these approaches based on the nature of the transferred knowledge. The survey reviews TL methods applicable to evolved RL tasks and newer schemes like representation disentanglement and policy distillation, reflecting on developments and suggesting future research directions. It analyzes TL approaches based on their goals, methodologies, compatible RL backbones (including DQN and its variants), and practical applications, while also summarizing important metrics for TL evaluation [3].

The strengths of this survey paper lie in its comprehensiveness and up-to-date nature, systematic categorization, broad coverage of various TL techniques, insightful discussion of future directions, provision of an analysis framework, and consideration of evaluation metrics [3].

A limitation is that while broad, the deep dive into specific algorithms might be limited due to the survey’s scope. The survey compares different approaches to transfer learning by categorizing them based on the format of transferred knowledge, including Reward Shaping, Learning from Demonstrations, Policy Transfer (relevant to DQN through policy distillation), Inter-Task Mapping, and Representation Transfer [3].

The survey paper also examines how various transfer learning approaches address common RL challenges. Learning from Demonstrations (LfD) enhances sample efficiency and exploration by leveraging expert knowledge as a more informed starting point. Policy Transfer techniques, such as policy distillation, enable knowledge transfer from ’teacher’ to ’student’ policies, accelerating learning in target tasks where training from scratch would be inefficient. While not explicitly connecting transfer learning to overestimation issues

(as addressed by Double DQN), the survey paper implies that leveraging prior knowledge might indirectly mitigate instability and convergence problems. It also highlights how recent deep learning breakthroughs have empowered transfer learning, suggesting that modern RL algorithms (like those in Rainbow, including target networks and experience replay) contribute to training stability [3].

Transfer learning fundamentally addresses sample efficiency by providing rich initial states, policies, or representations that reduce the need for extensive exploration. The survey paper's categorization of transfer approaches (reward shaping, demonstrations, policies, mappings, representations) offers a framework for understanding how prior knowledge can address these core RL challenges.

B. Thematic Discussion

Following the critical analysis of the selected literature, several key trends emerge within the field of deep reinforcement learning, particularly concerning advancements and applications within environments such as those provided by Gymnasium.

A prominent theme is the synergistic combination of independent algorithmic improvements to achieve state-of-the-art performance, exemplified by the Rainbow agent [1]. Hessel et al [1] successfully integrated six distinct enhancements to the Deep Q-Network (DQN) architecture, including Double DQN, Prioritized Experience Replay, Dueling Networks, Multi-step Learning, Distributional Q-learning, and Noisy Nets, demonstrating the power of modularity in algorithm design. This trend suggests a move towards building more sophisticated and effective agents by carefully selecting and combining techniques that address specific challenges like overestimation bias and sample inefficiency [1].

Another significant trend highlighted across the literature is the increasing importance of transfer learning to enhance data efficiency and generalization in deep RL [2].Saadat and Zhao [2] empirically demonstrated the benefits of transferring knowledge learned in single-player Gymnasium environments to corresponding two-player PettingZoo environments, showcasing improved performance and reduced training time. This aligns with the broader focus in the field, as discussed by the survey paper [3] , on leveraging prior knowledge to tackle the inherent sample inefficiency of deep RL algorithms. The survey emphasizes the diverse forms of knowledge that can be transferred, including demonstrations, policies, and representations, underscoring the versatility of transfer learning as a paradigm [3].

Building upon these observed trends and considering the diverse and flexible environment suite offered by Gymnasium, several promising avenues for future research emerge:

• **Further Exploration of Algorithmic Integration:** The success of Rainbow [1] suggests that future work could investigate the integration of additional cutting-edge improvements with existing high-performing architectures. This could involve combining Rainbow with policy-based or actor-critic methods to potentially overcome its value-based limitations [1]. Furthermore, exploring the synergistic effects of incorporating more advanced exploration techniques beyond Noisy Nets, as well as memory and sequence-based learning mechanisms, holds significant potential for enhancing performance in complex and partially observable Gymnasium environments [1].

• **Deepening the Understanding and Application of Transfer Learning:** Given the benefits demonstrated by Saadat and Zhao [2] and the comprehensive overview provided by [3] , further research is warranted to explore the boundaries and mechanisms of transfer learning within Gymnasium environments. This includes investigating the effectiveness of different transfer learning techniques (e.g., policy transfer, representation transfer) across a wider range of tasks and environment complexities [3]. As well as analyzing the types of knowledge that are most effectively transferred and how to quantify the similarity between source and target Gymnasium environments are also crucial directions [2]. Moreover, exploring transfer learning in more challenging multi-agent Gymnasium settings, beyond the scope of symmetric two-player Atari games, could yield valuable insights [2].

• **Addressing Practical Challenges in Deep RL:** The limitations identified in the literature also point towards important future research directions. The significant computational cost associated with training complex agents like Rainbow [1] necessitates investigating more efficient training methodologies, potentially leveraging the parallel environment capabilities of Gymnasium [1]. Also, reducing the reliance on manual hyper-parameter tuning [1] , perhaps through the application of meta-learning or automated optimization techniques, would also enhance the practicality and accessibility of advanced deep RL algorithms within the Gymnasium framework [1].

• **Developing More General and Robust Agents:** A long-standing goal in reinforcement learning is to develop agents that can generalize effectively to new and unseen tasks. Future research could focus on leveraging Gymnasium's diverse environment offerings to develop and evaluate transfer learning methods that promote better generalization [3]. Then, exploring the learning of disentangled representations that capture task-invariant features could be particularly beneficial in this regard [3]. Furthermore, adapting and evaluating state-of-the-art agents like Rainbow on a broader spectrum of Gymnasium environments, including those with continuous action spaces and different reward structures, is crucial for assessing their generality and identifying necessary modifications [1].

By pursuing these research directions within the context

of Gymnasium environments, the deep reinforcement learning community can continue to advance the field towards the development of more powerful, efficient, and general-purpose learning agents [1].

III. Research Proposal

A. *Research Topic*

How do Double DQN and Dueling DQN architectures compare in terms of transfer learning efficiency (e.g., data efficiency during transfer, speed of adaptation) and generalization ability (e.g., performance on novel, related tasks) within deep reinforcement learning?

B. *Rationale/Justification*

The Deep Q-Networks (DQN) algorithm was a significant advancement in reinforcement learning (RL) by combining Q-learning with deep neural networks and experience replay. Since its inception, several extensions have been proposed to improve its speed and stability. Two prominent extensions are Double DQN (DDQN) which addresses the overestimation bias inherent in Q-learning by decoupling the action selection and evaluation process, and Dueling DQN which aims to improve generalization across actions by separating the representation of state values and action advantages. Double DQN (DDQN) was developed to tackle the overestimation bias inherent in conventional Q-learning. This bias arises from the maximization step in the target value calculation, which can lead to the selection and evaluation of suboptimal actions. DDQN decouples the action selection and evaluation processes using two separate value functions (often the online and target networks), resulting in a more stable and accurate learning process. As noted in the sources, DDQN reduces harmful overestimations and improves performance. Prioritized DDQN and Dueling DDQN have both successfully incorporated double Q-learning, indicating its compatibility and potential for synergy with other improvements.

Dueling DQN introduces a network architecture that explicitly separates the representation of the state value and the action advantages. By having two separate streams that share a common convolutional encoder, the dueling architecture allows the agent to learn which states are valuable without necessarily having to learn the effect of each action in those states. This separation is hypothesised to improve generalisation across actions, as the learned state value can inform the value of different actions within that state, even if those specific state-action pairs have not been frequently visited. The dueling network architecture is specifically designed for value-based RL and has been shown to improve performance. It has also been combined with prioritized experience replay, further demonstrating its adaptability.

Transfer learning in RL is a crucial technique for overcoming the sample inefficiency of RL algorithms and tackling complex problems. It involves leveraging knowledge gained from a different domain to improve learning in a target domain. The survey "Transfer Learning in Deep Reinforcement Learning: A Survey" emphasizes the recent progress in transfer learning in RL, particularly with the advancements in deep learning. This survey categorizes transfer learning approaches based on what knowledge is transferred. While it provides a comprehensive overview of transfer learning in DRL, it does not provide a direct comparative analysis of how different fundamental DRL architectures, such as DDQN and Dueling DQN, inherently affect the transfer learning process in terms of efficiency and generalization. Our research question directly addresses this by focusing on these two influential architectural improvements to DQN.

This research is significant because it advances our understanding of how different Deep Q-Network (DQN) extensions—specifically Double DQN (DDQN) and Dueling DQN—facilitate transfer learning. By identifying which architecture better supports generalization across tasks or environments, this study offers practical insights for building more robust and data-efficient reinforcement learning agents. Unlike prior work that emphasizes single-task performance, this investigation centers on cross-task adaptability, a crucial yet underexplored aspect of deep reinforcement learning. It addresses a gap left by the "Rainbow" paper, which evaluates the combined benefits of various DQN improvements but does not isolate their individual contributions to transfer learning. Furthermore, it extends existing surveys on transfer learning in deep RL by focusing on the comparative effectiveness of two foundational architectures. This work also complements studies on single-to-multi-agent transfer by offering a more detailed architectural comparison, helping researchers make informed decisions about model design in transfer-based learning scenarios.

C. *Research Objectives*

- To empirically compare the transfer learning efficiency of agents employing Double DQN and Dueling DQN architectures across a suite of relevant reinforcement learning tasks or environments (e.g., related Atari games).
- To evaluate and contrast the generalisation capabilities of Double DQN and Dueling DQN agents when transferring knowledge to novel or slightly modified environments.
- To identify the specific mechanisms or properties of each architecture (e.g., reduced overestimation in DDQN, separated value and advantage streams in Dueling DQN) that contribute to or hinder their transfer learning performance.

D. *Potential Impact*

- Enhanced Understanding of DQN Extensions: It will provide a deeper understanding of the individual and potentially synergistic benefits of DDQN and Dueling DQN in the context of transfer learning, moving beyond single-task evaluations.
- Guidance for Algorithm Selection: The findings will offer practical guidance for researchers and practitioners in selecting the most appropriate DQN-based architecture for reinforcement learning tasks where transfer learning

is a key requirement for efficiency or solving complex problems.
- Improved Transfer Learning Techniques: The insights gained could inform the development of new or hybrid architectures and transfer learning strategies that leverage the strengths of both DDQN and Dueling DQN.
- Advancement of Deep RL: By focusing on the critical area of transfer learning, this research contributes to the broader goal of creating more generalisable, adaptable, and sample-efficient deep reinforcement learning agents, as highlighted by the challenges mentioned in the sources.

IV. Methodology

In this study, we implemented transfer learning techniques using both Double DQN (DDQN) and Dueling DQN (DQN) agents to evaluate their performance in the LunarLander environment, while also training each agent independently to establish a baseline comparison. The primary objective was to assess how effectively each algorithm could adapt to the more complex LunarLander task by transferring learned features and model weights from the simpler CartPole environment.

A. Independent Training of DDQN and Dueling DQN

Initially, we independently trained both the DDQN and Dueling DQN agents in the CartPole environment, where the agent's goal is to balance a pole on a cart. The CartPole task served as a simplified environment to train both models before testing transfer learning on the more complex LunarLander environment.

For the DDQN agent, we utilized the DDQN architecture to mitigate overestimation bias by employing two separate networks for action selection and evaluation, enhancing the stability of the learning process. The model consisted of an input layer for the four state variables, two hidden layers with 128 neurons each, and an output layer predicting Q-values for the two possible actions. An experience replay buffer was established to store past experiences, and an epsilon-greedy strategy was used for exploration, gradually decaying the exploration rate to balance exploration and exploitation. Training spanned 300 episodes, with regular evaluations and checkpoints implemented to monitor performance.

Similarly, the Dueling DQN agent was trained independently in CartPole using a Dueling DQN architecture, which separates the value function and the advantage function to improve learning efficiency and stability. This architecture consisted of an input layer for the four state variables, two hidden layers with 128 neurons each, and separate streams for value and advantage calculations. The Q-values were computed using a custom layer that combined these outputs. An experience replay buffer and an epsilon-greedy strategy were again employed for exploration.

After training both agents independently in the CartPole environment, we compared their performance in terms of episode rewards, training loss, and validation rewards. These metrics allowed us to assess the efficiency and effectiveness of each algorithm in this relatively simple environment, forming the basis for further analysis in LunarLander.

B. Transfer Learning to LunarLander

Following the successful training of both the DDQN and Dueling DQN agents in CartPole, we transitioned to the more complex LunarLander environment, where the goal is to control a spacecraft's thrusters to land safely on the Moon's surface. LunarLander consists of eight continuous state variables and four discrete actions.

To facilitate transfer learning, we attempted to adapt the pre-trained DDQN model from CartPole to LunarLander by modifying the input and output layers to match the new state and action dimensions. However, the transfer learning approach was not successful in this case, as detailed further in the challenges section. Despite the adjustments made to the model, the learning process did not accelerate as expected, and the agent struggled to leverage the knowledge gained from CartPole effectively. More details on the challenges encountered will be discussed later in the study.

C. Fine-Tuning and Transfer Learning

After unsuccessful transfer learning, we proceeded to fine-tune the pre-trained DDQN model on LunarLander. This fine-tuning process allowed the agent to continue learning from the LunarLander environment, albeit without the expected initial advantage from the transferred knowledge. Similarly, we fine-tuned the Dueling DQN model, adapting it to LunarLander and training it with an epsilon-greedy exploration strategy and experience replay buffer.

Both agents were trained for an extended period in the LunarLander environment to observe if they could achieve satisfactory performance despite the initial shortcomings of transfer learning. The training process involved managing exploration-exploitation through epsilon-greedy and periodic evaluations to assess the agent's progress.

D. Performance Comparison: DDQN vs. Dueling DQN in LunarLander

After training both agents independently and transferring their knowledge to LunarLander, we compared their performance using key metrics such as episode rewards, training loss, and validation rewards. Additionally, we recorded videos of the agents' performances to qualitatively assess their behavior during the landing task. The results were analyzed to determine which algorithm demonstrated superior efficiency in adapting to the complexities of the LunarLander environment.

While transfer learning did not yield successful results, the independent training of both DDQN and Dueling DQN in CartPole allowed us to establish a benchmark for performance. Our comparison of these two models in the CartPole environment demonstrated differences in convergence rates and overall performance, providing insights into the strengths and weaknesses of each approach when applied to a simple task.

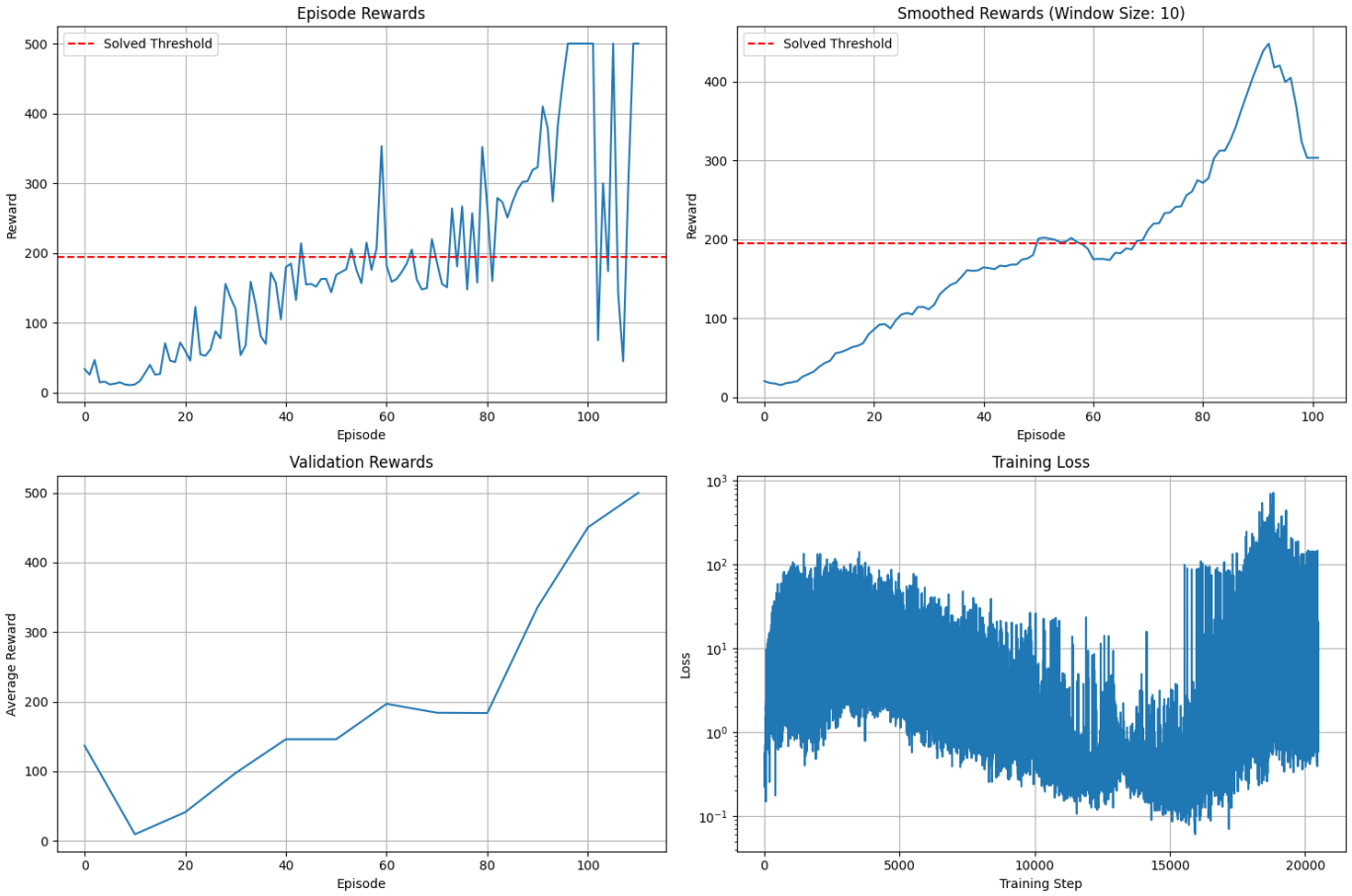

Fig. 1. Training Performance Plots for DDQN

V. Results

This section presents a comparative evaluation of two deep reinforcement learning algorithms—**Double Deep Q-Network (DDQN)** and **Dueling DQN**—on the CartPole environment. Four key performance metrics were analyzed: episode rewards, smoothed rewards, validation rewards, and training loss.

A. Episode Rewards

The episode reward plots (top-left of both Figures 1 and 2) show the raw reward earned by each agent per episode during training.

- **DDQN** exhibited a gradual and consistent increase in rewards, with the maximum reward of 500 achieved intermittently after episode ˜60. However, notable variance emerged in the later stages, indicating some policy instability despite the overall upward trend.
- **Dueling DQN** initially performed poorly but demonstrated a sharp performance increase around episode 25. It achieved high rewards, including frequent spikes above 500, significantly earlier than DDQN. However, this came with greater episode-to-episode volatility.

Observation: Dueling DQN converged faster to high-reward policies but displayed more performance variability across episodes.

B. Smoothed Rewards

The smoothed reward plots (top-right of Figures 1 and 2), averaged over a window of 10 episodes, offer clearer insight into the reward trends.

- **DDQN**'s smoothed rewards rose steadily, surpassing the solved threshold (˜200) around episode 50. The performance peaked around episode 90, followed by a slight drop.
- **Dueling DQN** showed a more abrupt increase between episodes 20–30 and quickly plateaued at higher reward levels. It sustained performance above the threshold for a longer duration.

Observation: Dueling DQN achieved early and sustained convergence, indicating more efficient policy learning during early training stages.

C. Validation Rewards

Validation rewards (bottom-left of each figure) measure policy generalization during evaluation episodes.

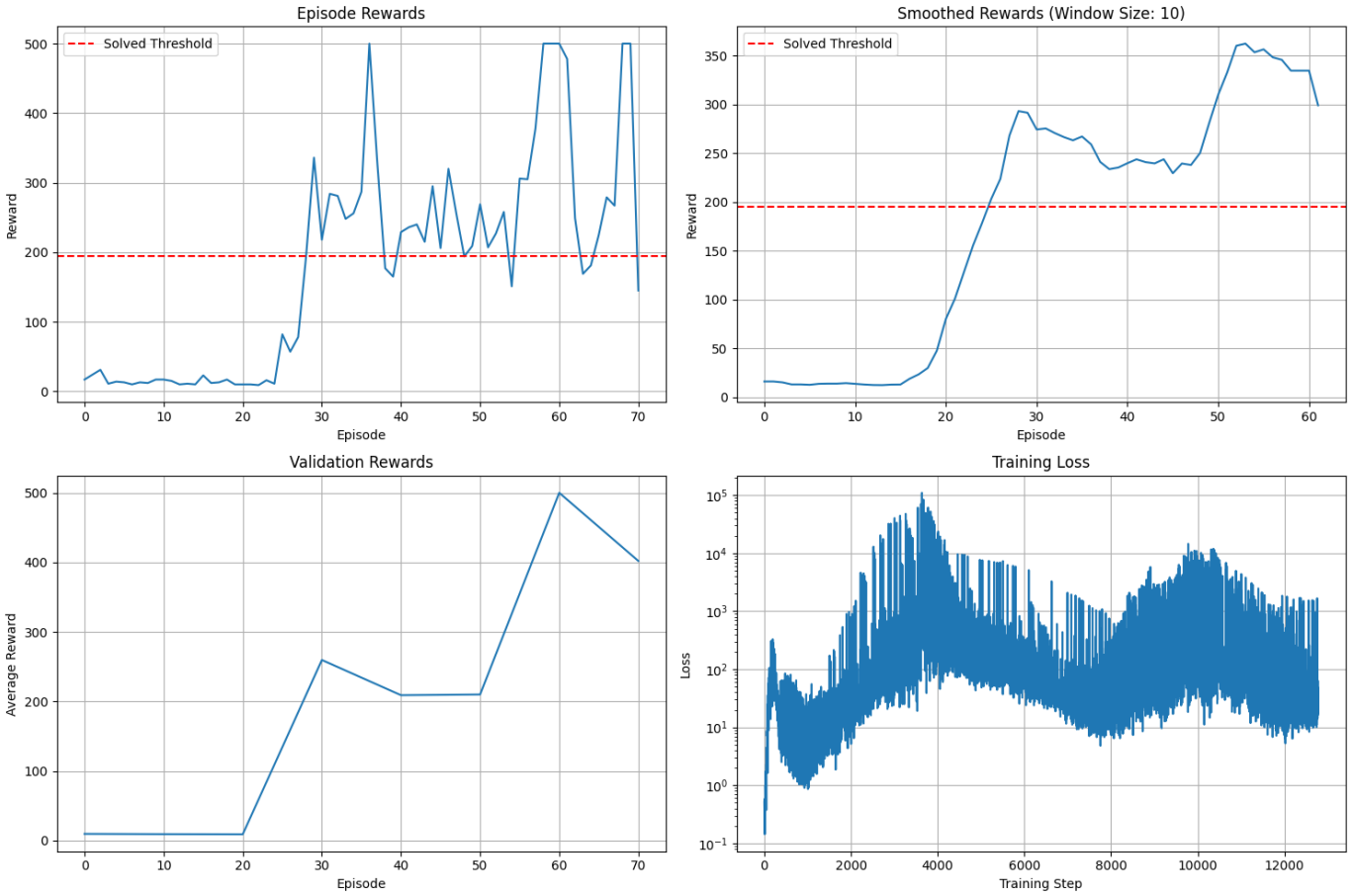

Fig. 2. Training Performance Plots for Dueling DQN

- **DDQN** showed a gradual improvement, peaking at around 500 rewards after 100 episodes. However, the curve featured intermediate plateaus near rewards of 200 and 300, suggesting a slower generalization process.
- **Dueling DQN** reached peak validation rewards more quickly, hitting 500 by episode 60. A slight decline was observed afterward, indicating possible overfitting or unstable performance at later stages.

Observation: Dueling DQN exhibited stronger early generalization but also showed sensitivity to performance degradation beyond peak training.

D. Training Loss

Training loss (bottom-right of Figures 1 and 2) was plotted per optimization step on a logarithmic scale.

- **DDQN** maintained a mostly downward loss trajectory with visible noise and occasional spikes. A slight increase in variance toward the end (˜18,000–20,000 steps) may indicate overfitting or unstable Q-value updates.
- **Dueling DQN** began with significantly higher losses and exhibited more pronounced variance, especially in the initial ˜4,000 steps. The curve transitioned into a decreasing trend until ˜9,000 steps before experiencing another moderate rise.

Observation: While both agents encountered noisy training loss, DDQN appeared more stable overall. Dueling DQN's variance may stem from its more complex network architecture.

E. Comparative Summary

Table I provides a concise comparison of the two algorithms across key performance metrics. Dueling DQN outperforms DDQN in most areas, demonstrating faster convergence, better early generalization, and higher smoothed rewards. However, DDQN exhibits greater training stability, which may be advantageous in environments where learning volatility must be minimized. While both reach optimal performance levels, Dueling DQN does so earlier, albeit with some post-peak fluctuations.

VI. Discussion

The comparative performance of DDQN and Dueling DQN on the CartPole task highlights the trade-offs between learning efficiency and stability when using advanced Q-learning architectures. While both agents ultimately achieve high rewards

and demonstrate the capacity to solve the task, their learning trajectories and training characteristics differ in meaningful ways.

Dueling DQN clearly demonstrates a faster convergence rate, evidenced by its early spike in episode rewards and quick surpassing of the solved threshold (200). This suggests that its architectural enhancement—decoupling the estimation of state values from advantages—enables the agent to more effectively identify valuable states independent of specific actions. This capability seems especially beneficial in the early learning phase, where sparse experience limits action-value clarity. The dueling structure essentially allows the agent to make better use of limited early data, which aligns with its steeper reward growth and sharper improvements in smoothed and validation rewards.

However, the greater variance and volatility in Dueling DDQN's performance—particularly visible in raw episode rewards and training loss—raises concerns about its stability. The higher and noisier loss patterns suggest that while the model learns quickly, it may be more prone to overfitting or erratic updates, possibly due to the increased expressiveness and complexity of its network. This is further evidenced by a slight decline in validation rewards post-peak, indicating potential instability in policy generalization despite early success.

In contrast, DDQN exhibits a more gradual and consistent learning curve, with smoother training loss trends and steadier reward progression. The agent benefits from its core design principle—mitigating Q-value overestimation using two separate networks for selection and evaluation—which seems to promote more stable learning dynamics. Although DDQN reaches optimal performance later, its path is more predictable, and its convergence is arguably more sustainable in the long run. This makes it potentially more reliable in environments where erratic actions can lead to catastrophic outcomes or where training must be highly robust to varying initial conditions.

An additional dimension of this comparison lies in generalization. Dueling DQN's superior performance on validation rewards suggests that its policies, once trained, transfer better across unseen trajectories. This aligns with the theoretical benefits of its architecture in estimating global state values, which may generalize better than specific action-based Q-values. However, the post-peak decline in these rewards also cautions that this generalization may not persist indefinitely without careful tuning or regularization.

Taken together, the results point to contextual suitability as a critical factor in selecting between these methods. Dueling DQN appears best suited for scenarios where rapid policy acquisition is desired and where early-stage performance is critical. Meanwhile, DDQN remains a strong candidate for tasks that demand training stability, lower variance, and greater control over Q-estimation dynamics.

VII. Challenges and Limitations

While implementing Double DQN (DDQN) for both CartPole and LunarLander, we encountered several limitations and issues that affected training efficiency and model performance. These challenges stemmed from hardware constraints, environmental complexities, and unexpected disruptions during the training process.

A. CartPole Training Limitations

Training the DDQN model for CartPole was relatively efficient due to the environment's simplicity. However, achieving optimal performance typically requires training for up to 300 episodes, which was impractical given our limited computational resources. To mitigate this, we implemented an early stopping condition: training would halt after three consecutive validation checks (each averaging 10 episodes) where rewards consistently exceeded 195. While this allowed us to confirm convergence without completing all episodes, the trade-off was a potential loss in final performance robustness. The results showed that the model crossed the target threshold (195) after approximately 110 episodes, with rewards continuing to rise steadily. However, this workaround underscores the challenge of balancing computational efficiency with training thoroughness in resource-constrained settings.

B. LunarLander Training Challenges

Transitioning to LunarLander introduced significant hurdles, despite prior research suggesting its suitability for transfer learning from CartPole. The environment's complexity—gravity, thruster dynamics, and continuous state-action spaces—demanded far more training episodes than anticipated.

1) Extended Training Duration: LunarLander required 300+ episodes for meaningful progress, even with relaxed validation criteria. Our hardware limitations made such extended training sessions infeasible, forcing us to operate with suboptimal episode counts.

Metric	DDQN	Dueling DQN	Superior
Episode Reward Growth	Slower but consistent	Rapid rise after episode ˜25	Dueling DQN
Smoothed Rewards	Gradual convergence	Early convergence, sustained rewards	Dueling DDQN
Validation Rewards	Gradual improvement	Fast generalization, peaks early	Dueling DQN
Training Loss Stability	More stable	Noisier, higher peak loss	DDQN
Final Performance	Hits 500 late	Hits 500 earlier, slight decline later	Dueling DQN (caveat)

TABLE I
Performance Comparison of DDQN and Dueling DQN

2) Hardware and Power Constraints: Training on laptops with limited GPU capabilities slowed progress, and frequent electricity outages disrupted sessions. Additionally, reliance on Google Colab introduced instability, as sessions often disconnected after a few hours, erasing progress. For instance, an 11-hour training run was lost after a Colab timeout at 130 episodes, necessitating checkpointing—a process complicated by Colab's volatile runtime.

3) Framework Inefficiencies: While Keras simplified implementation, its slower execution compared to PyTorch—a more efficient choice for deep RL—prolonged training times. Project constraints prevented a framework switch, further reducing efficiency.

4) Incomplete Training and Transfer Learning Failures: Due to these bottlenecks, neither the transferred CartPole model nor the baseline LunarLander model reached full convergence. This compromised our ability to rigorously evaluate transfer learning efficacy, as both models operated below their potential performance ceilings.

C. General Limitations

1) Hyperparameter Sensitivity: Both DDQN and Dueling DQN required meticulous tuning of learning rates, exploration schedules, and replay buffer sizes. Minor misconfigurations led to divergent training behaviors, highlighting the fragility of DQN-based methods in transfer scenarios.

2) Generalization Gaps: While Dueling DQN showed promising validation performance in CartPole, its post-peak reward decline suggested overfitting or insufficient exploration. This raises concerns about its reliability in more complex or dissimilar environments.

These challenges emphasize the need for more robust transfer learning techniques, better computational resources, and framework optimizations to ensure consistent and scalable DRL training.

VIII. Conclusion

This study compared DDQN and Dueling DQN in transfer learning contexts, revealing critical trade-offs between convergence speed and stability. Dueling DQN's rapid reward growth in CartPole suggests superior early-stage generalization, likely due to its decoupled value-advantage architecture. However, its higher training variance and post-peak performance drops indicate potential instability in long-term deployment. In contrast, DDQN exhibited steadier learning curves, making it a safer choice for applications requiring reliability.

The failed transfer to LunarLander underscores a key limitation: architectural improvements alone cannot bridge vastly dissimilar tasks without additional techniques like feature adaptation or hierarchical learning. Hardware and framework constraints further restricted our ability to train models to convergence, highlighting the practical challenges of scaling DRL experiments.

Future work should explore hybrid architectures combining the strengths of both methods, alongside meta-learning for hyperparameter adaptation. Additionally, investing in more robust computational setups and leveraging efficient frameworks like PyTorch could mitigate training bottlenecks. By addressing these limitations, DRL can advance toward more generalizable and deployable solutions for complex real-world problems.

References

[1] M. Hessel, J. Modayil, H. van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. G. Azar and D. Silver, "Rainbow: Combining improvements in deep reinforcement learning," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1, 2018. doi: https://doi.org/10.48550/arXiv.1710.02298.

[2] K. Saadat and R. Zhao, "Enhancing Two-Player Performance Through Single-Player Knowledge Transfer: An Empirical Study on Atari 2600 Games," arXiv preprint arXiv:2410.16653, 2024. doi: https://arxiv.org/abs/2410.16653v1

[3] Z. Zhu, K. Lin, A. K. Jain, and J. Zhou, "Transfer Learning in Deep Reinforcement Learning: A Survey," IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 11, pp. 13344-13362, 1 Nov. 2023. doi: 10.1109/TPAMI.2023.3292075.