Ablation Study of Deep Reinforcement Learning in ViZDoom: Algorithm Selection and Extensions for Visual Control and Combat Tasks

Hasan Arı
Department of Computer Engineering
Izmir Institute of Technology
Izmir, Turkey
hasanari@iyte.edu.tr

***Abstract*—This paper presents a comparative ablation study of deep reinforcement learning algorithms in the ViZDoom environment, a visually complex first-person shooter platform. I compare Deep Q-Network (DQN) against Deep SARSA across three scenarios of increasing complexity: basic navigation, survival defense, and deathmatch combat. I further investigate the impact of n-step returns as a bridge between temporal difference and Monte Carlo methods, and evaluate modern DQN extensions including Double DQN, Prioritized Experience Replay, and Dueling architectures. The experiments, conducted over 33 training runs totaling 13.2 GPU hours, reveal that off-policy DQN consistently outperforms on-policy Deep SARSA. Furthermore, results indicate that n-step returns provide mixed results highly dependent on task complexity, and architectural extensions yield modest improvements of 3-7% on simpler tasks. These findings provide empirical guidance for algorithm selection in visual reinforcement learning domains under computational constraints.**

***Index Terms*—Deep reinforcement learning, DQN, ViZDoom, ablation study, visual navigation**

I. Introduction

Reinforcement learning (RL) has achieved remarkable success in game-playing domains, from Atari games [1] to complex strategy games. However, applying RL to visually rich, three-dimensional environments presents unique challenges that tabular methods cannot address due to the high-dimensional state spaces involved. The ViZDoom platform [2] provides an ideal testbed for studying deep RL in first-person shooter (FPS) environments. Unlike simpler 2D domains, ViZDoom requires agents to process raw pixel observations, handle partial observability, and learn complex behaviors such as navigation, target acquisition, and survival strategies.

Tabular methods such as Q-learning and SARSA are theoretically well-understood but fundamentally limited in visual domains. A single 84×84 grayscale frame contains over 7,000 dimensions, making state enumeration infeasible. This motivates the use of deep neural networks as function approximators, leading to algorithms like Deep Q-Network (DQN) [1].

This paper presents a systematic ablation study examining:

1) **Off-policy vs. On-policy**: Comparing DQN (off-policy) against Deep SARSA (on-policy) to understand how learning paradigm affects performance in visual domains.
2) **TD vs. MC Trade-off**: Investigating n-step returns ($n = 3$) as a bridge between temporal difference (TD) and Monte Carlo (MC) methods.
3) **DQN Extensions**: Evaluating Double DQN [3], Prioritized Experience Replay (PER) [4], and Dueling architectures [5] against baseline DQN.

The contributions of this work include: (1) empirical comparison of fundamental RL paradigms in visual FPS domains, (2) analysis of how task complexity affects algorithm performance, and (3) practical recommendations for algorithm selection in similar domains.

II. Related Work

A. Deep Reinforcement Learning

Mnih et al. [1] introduced DQN, combining Q-learning with deep neural networks using experience replay and target networks to stabilize training. This breakthrough demonstrated human-level performance on Atari 2600 games directly from pixel inputs. Several extensions have improved upon vanilla DQN. Van Hasselt et al. [3] proposed Double DQN to address overestimation bias by decoupling action selection from value estimation. Schaul et al. [4] introduced Prioritized Experience Replay, sampling transitions based on TD-error magnitude. Wang et al. [5] developed the Dueling architecture, separating state-value and advantage streams to improve learning efficiency.

B. ViZDoom Platform

Kempka et al. [2] introduced ViZDoom as a research platform based on the Doom game engine. The platform provides various scenarios ranging from simple navigation to complex multiplayer combat. Wydmuch et al. [6] extended ViZDoom with additional scenarios and improved API support. Lample and Chaplot [7] demonstrated that DQN-based agents can learn effective policies for FPS navigation and combat, outperforming rule-based bots on certain tasks. Their work highlighted the importance of reward shaping and curriculum learning in complex visual domains.

C. On-policy vs. Off-policy Learning

The distinction between on-policy (e.g., SARSA) and off-policy (e.g., Q-learning) methods has been extensively studied in tabular settings [8]. Off-policy methods can learn from any experience, enabling better sample efficiency through replay, while on-policy methods learn directly from the behavior policy, potentially offering more stable learning in certain domains.

III. Methodology

A. Environment

This study utilizes three ViZDoom scenarios of increasing complexity:

Basic: A simple navigation task where the agent must shoot a stationary target. The agent receives +100 reward for hitting the target and -5 penalty per time step, encouraging efficient target acquisition.

TakeCover: A survival scenario where the agent must dodge incoming fireballs while avoiding damage. Rewards are based on survival time, requiring the agent to learn evasive behaviors.

Deathmatch: A complex combat scenario with moving enemies. The agent must navigate, aim, and engage multiple targets while managing health and ammunition. This scenario features sparse rewards and high partial observability.

B. State Representation

Following standard practice [1], observations are preprocessed by converting to grayscale, resizing to 84×84 pixels, and normalizing to $[0, 1]$. Four consecutive frames are stacked to provide temporal information, resulting in state tensors of shape $(4, 84, 84)$.

C. Network Architecture

All agents use the convolutional architecture from Mnih et al. [1]:

- Conv1: 32 filters, 8×8, stride 4, ReLU
- Conv2: 64 filters, 4×4, stride 2, ReLU
- Conv3: 64 filters, 3×3, stride 1, ReLU
- FC: 512 units, ReLU
- Output: $|A|$ units (action values)

The Dueling architecture modifies the final layers to separate value and advantage streams before combining them.

D. Algorithms

DQN: Off-policy algorithm using experience replay buffer (capacity 100,000) and target network (update frequency 1,000 steps). Loss function:

$$L = \mathbb{E}[(r + \gamma \max_{a'} Q_{\theta^-}(s', a') - Q_\theta(s, a))^2] \quad (1)$$

Deep SARSA: On-policy variant using the actual next action rather than the greedy action:

$$L = \mathbb{E}[(r + \gamma Q_{\theta^-}(s', a') - Q_\theta(s, a))^2] \quad (2)$$

N-step DQN: Extends DQN with multi-step returns:

$$G_t^{(n)} = \sum_{k=0}^{n-1} \gamma^k r_{t+k} + \gamma^n \max_{a'} Q_{\theta^-}(s_{t+n}, a') \quad (3)$$

Double DQN: Uses online network for action selection and target network for evaluation to reduce overestimation.

PER: Samples transitions proportionally to TD-error with importance sampling correction.

E. Training Configuration

All experiments use: learning rate $\alpha = 0.0001$, discount factor $\gamma = 0.99$, ϵ-greedy exploration decaying from 1.0 to 0.01, batch size 32, and 2,000 training episodes. Due to the high computational cost of training deep networks on pixel inputs, and to manage the scope of the study, each configuration was run with 2-3 random seeds. While this limits statistical power compared to large-scale benchmarks, it provides sufficient signal to observe relative algorithmic trends and sensitivities.

IV. Experimental Results

A. Phase 1: DQN vs. Deep SARSA

I first compare the fundamental off-policy (DQN) and on-policy (Deep SARSA) paradigms across all three scenarios.

TABLE I
Phase 1: DQN vs Deep SARSA Comparison (Best Eval Reward)

Scenario	Method	Best Eval	Final Reward
Basic	DQN	75.8 ± 5.7	60.9 ± 18.5
	Deep SARSA	67.6 ± 15.3	35.3 ± 23.8
TakeCover	DQN	1088.7 ± 194.2	861.7 ± 143.6
	Deep SARSA	1080.1 ± 62.2	783.6 ± 36.3
Deathmatch	DQN	5.4 ± 1.2	3.3 ± 1.2
	Deep SARSA	4.3 ± 1.0	2.1 ± 0.3

DQN consistently outperforms Deep SARSA across all scenarios. The advantage is most pronounced on the Basic task (+12% best eval) and Deathmatch (+24%), while TakeCover shows comparable performance (+0.8%). Notably, Deep SARSA exhibits lower variance on TakeCover, suggesting more consistent but slightly inferior learning. The off-policy advantage likely stems from experience replay enabling better sample efficiency and breaking temporal correlations, which is critical in visual domains.

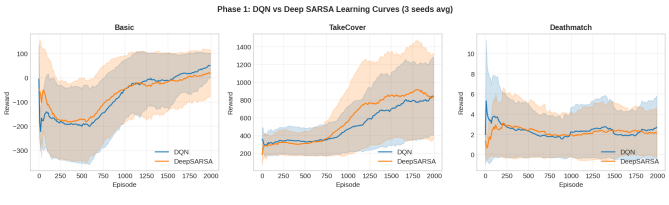

Fig. 1. Phase 1 learning curves: DQN vs Deep SARSA across three scenarios. Solid lines show mean reward (3 seeds), shaded regions indicate ± 1 std. DQN (blue) achieves higher asymptotic performance than Deep SARSA (orange) on all tasks.

B. Phase 2: N-step Returns

This phase investigates n-step returns ($n = 3$) as a bridge between TD(0) and Monte Carlo methods, comparing against baseline DQN ($n = 1$).

TABLE II
PHASE 2: N-STEP ABLATION (BEST EVAL REWARD)

Scenario	n=1	n=3	Change
Basic	75.8 ± 5.7	-175.6 ± 175.9	-332%
TakeCover	1088.7 ± 194.2	820.9 ± 267.4	-25%
Deathmatch	5.4 ± 1.2	7.0	$+30\%$

N-step returns show dramatically different effects depending on task complexity. On Basic and TakeCover, n-step learning degrades performance significantly, while on Deathmatch it provides a 30% improvement. This divergence illustrates the classic bias-variance trade-off. N-step returns reduce bias but increase variance. In simpler tasks with clear reward signals (Basic), the increased variance destabilizes learning, leading to divergence as seen in the negative scores. However, in complex tasks like Deathmatch where credit assignment is challenging, the reduced bias helps propagate rewards more effectively through longer action sequences.

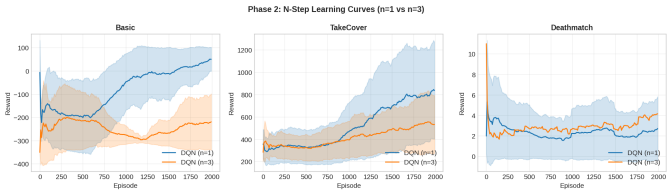

Fig. 2. Phase 2 learning curves: n-step ablation comparing n=1 (blue) vs n=3 (orange). The n=3 variant exhibits higher variance and fails to converge on Basic, while showing comparable learning on Deathmatch.

C. Phase 3: DQN Extensions

I evaluate modern DQN extensions against the baseline on Basic and TakeCover scenarios.

TABLE III
PHASE 3: DQN EXTENSIONS (BEST EVAL REWARD)

Method	Basic	TakeCover
DQN (baseline)	75.8 ± 5.7	1088.7 ± 194.2
DQN + PER	80.7 ± 3.3	1126.1 ± 103.2
Double DQN	81.4 ± 2.8	1021.5 ± 72.4
Dueling + DDQN	79.5 ± 4.0	–

On the Basic scenario, all extensions improve over baseline DQN:

- Double DQN: +7.4% (best performer)
- PER: +6.5%
- Dueling + DDQN: +4.9%

On TakeCover, results are mixed:

- PER: +3.4% improvement
- Double DQN: -6.2% degradation

The reduced variance across all extensions indicates more stable learning. PER's consistent improvement suggests that prioritizing high-error transitions benefits both scenarios.

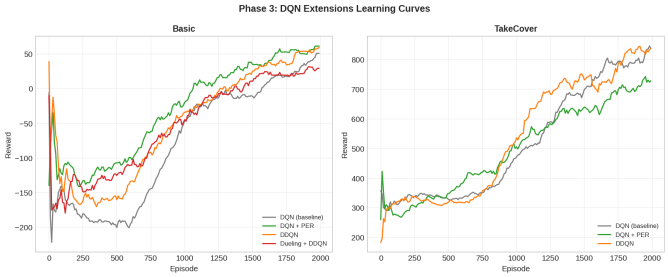

Fig. 3. Phase 3 learning curves: DQN extensions on Basic and TakeCover. All extensions show faster initial learning compared to baseline DQN (gray), with PER (green) and DDQN (orange) achieving strong asymptotic performance.

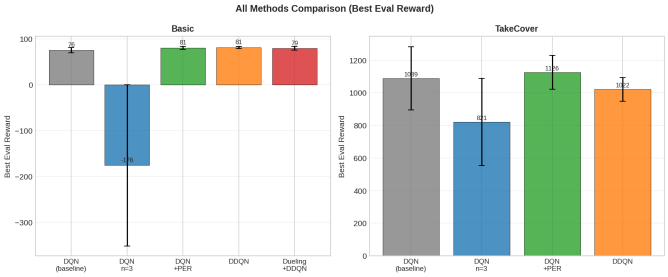

Fig. 4. Summary comparison of all methods by best evaluation reward. Error bars indicate standard deviation across seeds. On Basic, DDQN performs best; on TakeCover, PER provides the most consistent improvement over baseline.

D. Training Efficiency

Total training time across all 33 experiments was 13.2 hours on a single GPU. Average training duration per experiment ranged from approximately 15 minutes for simple tasks to 35 minutes for complex scenarios, consistent with the total computational budget.

V. DISCUSSION

A. Off-policy Advantage

The results strongly support using off-policy methods in visual RL domains. DQN's ability to learn from replay buffer experience provides consistent advantages over on-policy Deep SARSA. This aligns with theoretical expectations: off-policy methods can learn optimal policies while following exploratory behavior policies, which is essential when the state space is high-dimensional and exploration is costly.

B. Task-dependent Algorithm Selection

A key finding is that algorithm modifications have task-dependent effects. N-step returns, while theoretically appealing, require careful tuning based on task structure. Simple tasks with immediate rewards benefit from low-variance TD updates, while complex tasks with delayed rewards may benefit from longer bootstrapping horizons despite the added variance.

C. Modest Extension Benefits

Modern DQN extensions provide modest but consistent improvements (3-7%) on simpler tasks. However, the benefits diminish or reverse on more complex scenarios. This suggests that architectural improvements may matter less than fundamental algorithm choice (e.g., off-policy vs on-policy) in challenging visual domains.

D. Limitations

This study has several limitations inherent to the scope of a course project: (1) Due to computational constraints, results are reported on a limited number of seeds (2-3), which warrants caution in interpreting small performance differences. (2) Hyperparameters were fixed across all scenarios and algorithms (based on DQN baselines), which may disadvantage on-policy methods like SARSA that might require different learning rates. (3) The study focused on value-based methods without policy gradient comparisons. Future work should address these limitations by performing extensive hyperparameter sweeping and increasing the number of trials.

VI. Conclusion

This ablation study provides empirical guidance for deep RL algorithm selection in visual FPS domains. The key findings are:

1) Off-policy DQN consistently outperforms on-policy Deep SARSA across all tested scenarios, supporting the use of experience replay in visual domains.
2) N-step returns have task-dependent effects: beneficial for complex credit assignment (Deathmatch) but harmful for simpler tasks (Basic, TakeCover) due to variance injection.
3) DQN extensions (Double DQN, PER, Dueling) provide modest improvements (3-7%) on simpler tasks with reduced variance, but benefits do not consistently transfer to more complex scenarios.

These findings highlight the importance of empirical evaluation when selecting RL algorithms for new domains. While theoretical properties provide guidance, task-specific characteristics ultimately determine algorithm effectiveness.

Acknowledgment

The author thanks the ViZDoom development team for providing an excellent research platform.

References

[1] V. Mnih et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, pp. 529–533, 2015.
[2] M. Kempka, M. Wydmuch, G. Runc, J. Toczek, and W. Jaskowski, "ViZDoom: A Doom-based AI research platform for visual reinforcement learning," in Proc. IEEE Conf. Comput. Intell. Games (CIG), 2016, pp. 1–8.
[3] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in Proc. AAAI Conf. Artif. Intell., 2016, pp. 2094–2100.
[4] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," in Proc. Int. Conf. Learn. Representations (ICLR), 2016.
[5] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Freitas, "Dueling network architectures for deep reinforcement learning," in Proc. Int. Conf. Mach. Learn. (ICML), 2016, pp. 1995–2003.
[6] M. Wydmuch, M. Kempka, and W. Jaskowski, "ViZDoom competitions: Playing Doom from pixels," IEEE Trans. Games, vol. 11, no. 3, pp. 248–259, 2019.
[7] G. Lample and D. S. Chaplot, "Playing FPS games with deep reinforcement learning," in Proc. AAAI Conf. Artif. Intell., 2017, pp. 2140–2146.
[8] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.