Data Regularized Q Learning Applied to Snake

Firstname1 Lastname1 [* 1] **Firstname2 Lastname2** [* 1 2] **Firstname3 Lastname3** [2] **Firstname4 Lastname4** [3]
Firstname5 Lastname5 [1] **Firstname6 Lastname6** [3 1 2] **Firstname7 Lastname7** [2] **Firstname8 Lastname8** [3]
Firstname8 Lastname8 [1 2]

Abstract

This work explores the application of Data-Regularized Q-learning (DrQ) to discrete-action reinforcement learning in the Snake game. This implements DrQ's core components—random shift augmentation, double Q-learning, and target network updates adapting them for grid-based tasks with discrete actions. The implementation includes a custom Snake environment, an augmented replay buffer, and a convolutional Q-network with dual heads for stable training. This current implementation provides a foundation for evaluating augmentation-based regularization for improving sample efficiency and training stability in pixel-based RL. The results find that DrQ when applied in a Snake environment improves training performance in both non harsh reward shaping conditions and harsh sparse reward conditions, especially outperforming in the latter.

https://anonymous.4open.science/r/ECE570-DrQSnake-9F8B/

1. Introduction

1.1. Brief Overview/Background

Reinforcement Learning (RL) is a machine learning paradigm where an agent learns optimal actions by interacting with an environment to maximize cumulative rewards. Traditional RL techniques have been extensively used in control problems, but training RL agents directly from pixel-based inputs introduces significant challenges. High sample complexity, slow convergence, and instability due to overfitting to environment-specific features make it difficult to generalize policies across environments. Moreover, RL algorithms often require extensive hyperparameter tuning and a significant amount of training data to achieve optimal performance.

To address these issues, recent research has introduced data augmentation techniques as a form of regularization for RL. DrQ (Data-Regularized Q-learning) proposes an approach where random shift augmentations are applied to visual inputs, helping to enforce Q-value consistency across visually similar states. Unlike conventional RL methods that modify architectures or introduce additional losses, DrQ operates purely at the data level, leading to improvements in sample efficiency, stability, and generalization. The method is simple yet highly effective, making it an attractive choice for improving RL performance in environments where agents learn from high-dimensional pixel inputs.

DrQ has demonstrated state-of-the-art performance in continuous control tasks, but its effectiveness in discrete-action RL environments remains largely unstudied. This paper aims to answer the question of whether augmentation-based regularization can benefit grid-based environments with discrete actions. This project explores the application of DrQ to the Snake game, investigating whether augmentation can improve sample efficiency, generalization, and training stability in a structured discrete-action RL task. Additionally, this research aims to compare different augmentation strategies and analyze their effectiveness in discrete reinforcement learning scenarios.

1.2. Problem Definition

The selected papers address critical challenges in Q-learning frameworks for vision-based reinforcement learning, particularly focusing on **sample inefficiency** and **overfitting risks** when training agents on high-dimensional pixel inputs. Traditional Q-learning methods struggle with two core issues in such settings: (1) **instability in Q-value estimation** due to sparse rewards and correlated high-dimensional observations, and (2) **overfitting to environment-specific features**, which limits policy generalization. These challenges could be even larger in dynamic environments like the Snake game, where the agent must learn from raw pixels while adapting to an evolving state space (e.g., growing

[*] Equal contribution [1] Department of XXX, University of YYY, Location, Country [2] Company Name, Location, Country [3] School of ZZZ, Institute of WWW, Location, Country. Correspondence to: Firstname1 Lastname1 <first1.last1@xxx.edu>, Firstname2 Lastname2 <first2.last2@www.uk>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

snake body). The research questions focus on whether data augmentation and regularization techniques can stabilize Q-learning, improve sample efficiency, and reduce overfitting in discrete-action tasks without requiring architectural modifications or excessive hyperparameter tuning.

Key technical challenges include:

- **Q-value overestimation bias:** Double Q-learning mitigates this but introduces complexity in maintaining dual networks, risking slower convergence.
- **Overfitting to pixel patterns:** High-dimensional inputs encourage networks to memorize noise or static features rather than learning robust representations of game dynamics.
- **Sparse reward propagation:** Delayed consequences of actions (e.g., eventual collisions) complicate temporal credit assignment in Q-value updates.
- **Generalization gaps:** Policies trained on limited visual variations fail in unseen environments, necessitating regularization at the data or representation level.

1.3. Related Work

1.3.1. Paper 1: Image Augmentation is All You Need: Regularizing Deep Reinforcement Learning from Pixels

- **Contributions:** Proposes DrQ, a data augmentation framework that applies random shifts to pixel observations to enforce Q-value consistency across augmented states. Introduces a **data-level regularization** approach without architectural changes.
- **Methods:** Uses twin Q-networks (Q1, Q2) with a shared encoder. Augmented states are dynamically generated during training, and Q-values are averaged across augmentations to reduce variance.
- **Results:** Achieves state-of-the-art sample efficiency on continuous control tasks (e.g., Walker2D), with a 2.5× improvement over SAC in sample-limited settings.
- **Critique:** While methodologically elegant, DrQ's reliance on random shifts may limit its applicability to grid-based games where shifts could disrupt critical spatial relationships (e.g., snake body segments). The paper also lacks ablation studies on augmentation diversity and does not address discrete-action settings, leaving its broader utility unproven.

1.3.2. Paper 2: Reinforcement Learning with Augmented Data (RAD)

- **Contributions:** Systematically evaluates multiple augmentation strategies (cropping, flipping, cutout) in DQN, demonstrating their effectiveness in reducing overfitting and improving generalization.
- **Methods:** Augmentations are applied to both current and target states during Q-learning updates. Evaluates performance on Procgen and DeepMind Control Suite benchmarks.
- **Results:** RAD achieves 40% higher success rates in procedurally generated environments compared to unaugmented DQN, with cropping and cutout showing the strongest gains.
- **Critique:** Although RAD provides a comprehensive comparison, it omits explicit mechanisms for Q-value consistency (unlike DrQ), potentially leading to inconsistent target estimates. The computational overhead of multiple augmentations is also unaddressed, raising scalability concerns for real-time tasks.

1.3.3. Paper 3: Self-Predictive Representations (SPR) for Reinforcement Learning

- **Contributions:** Introduces a self-supervised auxiliary loss where the agent predicts future latent states, enhancing representation learning and reducing overfitting.
- **Methods:** Extends Q-networks with a prediction head that forecasts future embeddings using contrastive learning. Integrates the auxiliary loss with TD-learning.
- **Results:** Improves sample efficiency by 30% on Atari benchmarks, with notable gains in sparse-reward games like Montezuma's Revenge.
- **Critique:** SPR's dual-objective training increases computational complexity and hyperparameter sensitivity. The paper does not explore combining SPR with data augmentation, leaving open whether the two methods synergistically address overfitting. Additionally, the focus on continuous tasks limits insights into discrete-action grid environments.

2. Problem Definition

2.1. Preliminary Implementation Details

2.1.1. Experimental Motivation

This implementation adapts the DrQ (Data-Regularized Q-learning) framework to discrete-action reinforcement learning in the Snake game. The primary objectives are to: (1) evaluate whether data augmentation techniques developed for continuous control tasks generalize to grid-based discrete environments, (2) address sample inefficiency and overfitting challenges in pixel-based Q-learning, and (3) establish a baseline for comparing augmentation strategies in structured discrete-action spaces. The Snake environment serves as an ideal testbed due to its dynamic state space, sparse rewards, and need for temporal abstraction.

2.1.2. Experimental Setup

- **Environment:** Custom 12x12 grid Snake game with discrete actions (0=Up, 1=Right, 2=Down, 3=Left). Observations are 84x84 RGB images generated by upsampling grid states using 7x7 pixel blocks.
- **Network Architecture:** Three-component system:
  - *ConvEncoder:* 3-layer CNN (32 filters, kernel sizes 5-5-3) → Linear layer (50D features) with LayerNorm and Tanh
  - *Double Q-Heads:* Two 128-unit MLPs operating on encoded features
  - *Target Network:* Periodically updated via Polyak averaging (=0.01)
- **Training Parameters:**
  - Replay buffer capacity: 100k transitions
  - Batch size: 128 with dual augmentation
  - Optimizer: Adam (lr=1e-3), =0.99
  - Exploration: -greedy (1.0→0.1 over 5k steps)

2.1.3. Main Implementation

The implementation realizes DrQ's core components with discrete-action adaptations:

- **Augmented Replay Buffer:** Implements random shift augmentation via border replication and random cropping (4-pixel padding). Each sampled batch contains two independently augmented views of current and next states.
- **Double Q-Learning:** Twin Q-heads with shared encoder, using minimum target values for stability. The critic loss enforces consistency across four prediction paths:

  $$\mathcal{L} = \sum_{i=1}^{2}\sum_{j=1}^{2}\mathbb{E}[(Q_i(s_{\text{aug}j}, a) - y)^2]$$

  where $y = r + \gamma(1-d)\frac{1}{2}(\min Q_{\text{target}}(s'_{\text{aug1}}, a^*) + \min Q_{\text{target}}(s'_{\text{aug2}}, a^*))$
- **Frame Stacking:** Though not yet implemented, the observation structure supports temporal stacking through channel concatenation.

2.2. Problem Statement

This project aims to reimplement and evaluate the effectiveness of Data-Regularized Q-learning (DrQ) in the context of the Snake game, a dynamic and structured discrete-action environment. The primary focus is to investigate how DrQ's data augmentation techniques, specifically random shift augmentation, perform in a setting where the environment and the agent's state space evolve over time (e.g., the snake grows as it consumes food). The project will explore whether DrQ can improve sample efficiency, training stability, and generalization in such environments compared to traditional Q-learning methods. By systematically comparing these approaches, the project seeks to provide insights into the role of data augmentation and representation learning in improving reinforcement learning performance for grid-based games.

3. Methodology

The methodology for this project involves a systematic reimplementation of DrQ in the Snake environment, followed by a series of experiments to benchmark its performance against other augmentation strategies and baseline methods. The key steps are outlined below:

- **Reimplementation of DrQ:**
  - Adapt the DrQ framework to the Snake game, implementing random shift augmentation, double Q-learning, and target network updates.
  - Develop a custom Snake environment with pixel-based observations (84x84 RGB images) and discrete actions (up, right, down, left).
  - Implement a convolutional Q-network with twin heads to mitigate overestimation bias and enforce Q-value consistency across augmented states.
  - Train the agent using an augmented replay buffer, where each sampled batch includes two independently augmented views of the current and next states.
- **Baseline Comparisons:**
  - **DrQ (Random Shifts):** The core DrQ implementation with random shift augmentation, serving as the primary baseline for evaluating the impact of data augmentation.
  - **Vanilla DQN (No Augmentation):** A standard Deep Q-Network without any data augmentation, serving as a control to measure the baseline performance in the absence of regularization techniques.
- **Experimental Design:**

  - Trained each model (DrQ and Vanilla DQN) on the Snake environment for a fixed number of episodes, tracking key metrics such as episode reward, collision rate, and generalization performance.
  - Conduct ablation studies to isolate the effects of individual components (e.g., random shifts, rotations, cutout, and latent prediction) on overall performance.
  - Evaluate the models on both training and test environments to measure generalization gaps and overfitting tendencies.

3.1. Challenges and Technical Hurdles

    * Hyperparameter tuning was quite difficult at first and found different results. Too high a reward for survival and the snake would converge to a suboptimal path that prioritized staying alive but not eating fruit. making the fruit reward too high led to the snake being too dangerous and not caring about collisions. Eventually upon adding reward shaping which uses a calculation of the manhattan distance to the fruit from the snake head to either reward points or take them away from the snake.
    * Plateaus and spikes would occur in training sets that used wrong hyperparameters. Most sets seemed to plateau when using the full run, but showed linear progression for most of the smoke test runs with less episodes

  - **Augmentation Artifacts:** Random shifts occasionally obscure critical snake body segments. Partial solution: Constrained shifts to 2-pixel maximum during early training phases.
  - **Discrete Action Smoothing:** Argmax policy created abrupt Q-value jumps. Introduced soft action selection during target computation:

$$a^* = \text{argmax}_a \frac{Q_1(s', a) + Q_2(s', a)}{2}$$

4. Experimental Results

- **Initial Results:**

  - Upon gathering results, they seem to match what was expected and the DrQ batches ended up performing better on average in similar conditions as the DQN batches. I ran initial smoke tests and Full runs for DrQ and DQN based on a harsher environment with no reward shaping and instant death from reversal. Next I Tried smoke tests and full runs for DrQ and DQN with slightly easier and more tuned environments that include rewards for movement towards fruit and the inability to turn 180 degrees and instantly die.

- **Visualization:** These graphs show the reward as time goes on for different methods. The smoke tests are set for about 50,000 steps while the Full runs are usually for around 500,000 steps. These step values and limitations for amount of training were mostly due to constraints of Google Colab's monthly rates and these values led to a sustainable use of credits without going over. Smoke tests would range from 30 mins to 2 hours of training, while full tests ranged from 6-8 hours of training.

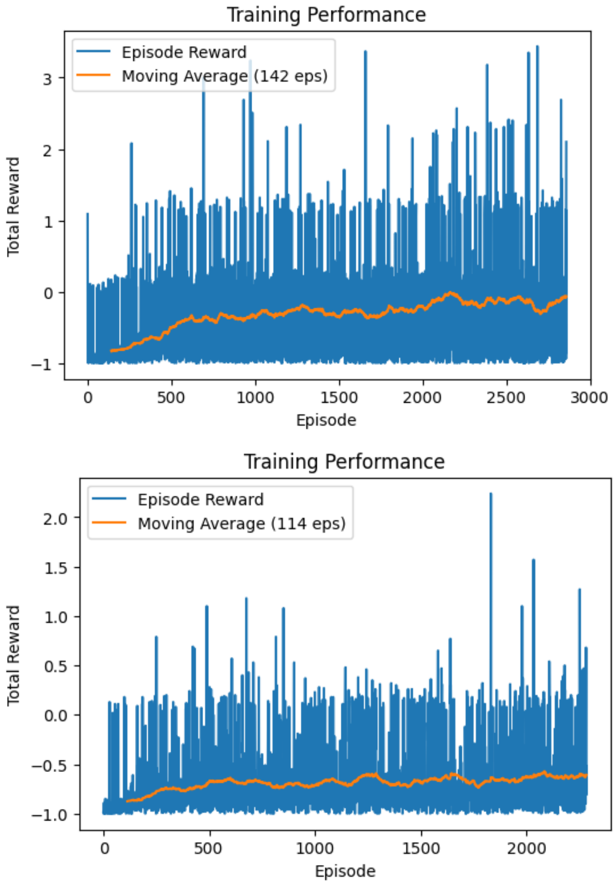

Figure 1. DrQ Smoke Test in Harsh Environment (top) v.s DQN Smoke Test in Harsh Environment (bottom). These figures show that in the initial harsh environment test, DrQ and DQN seem to both go up linearly in terms of reward however, DrQ is significantly faster. This is the most basic implementation.

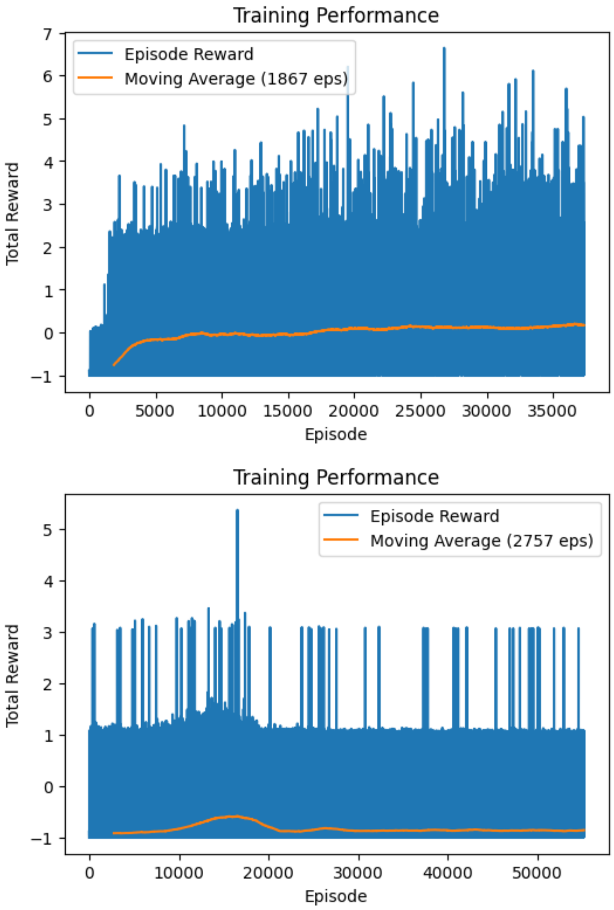

Figure 2. DrQ Full Test in Harsh Environment (top) v.s DQN Full Test in Harsh Environment (bottom). These figures show that in the full test in the harsh environment, DrQ and DQN seem to both eventually plateau for rewards, however DQN seems to taper back down due to overfitting in this harsh environment with all this training. However DrQ Seems to just plateau at a suboptimal path. This is tough for the implementation to work with heavy training in comparison to an easier environment.

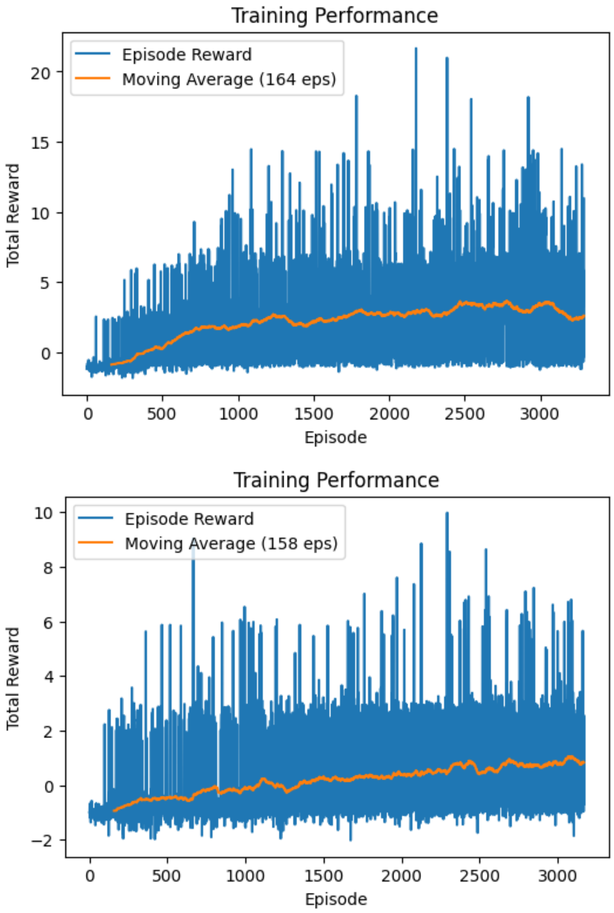

Figure 3. DrQ Smoke Test in Forgiving Environment (top) v.s DQN Full Test in Forgiving Environment (bottom). These figures show that when put into the more forgiving environment both DrQ and DQN are able to perform better than the initial smoke tests. However with this small number of episodes, the graphs look qutie similar with just linear increase in which DrQ is has slightly higher values.

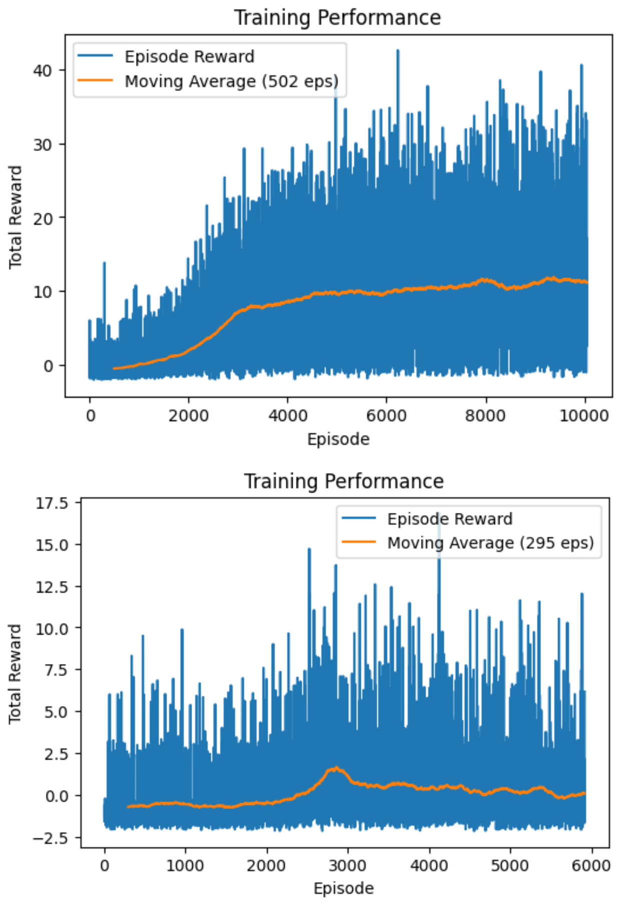

Figure 4. DrQ Full Test in Forgiving Environment (top) v.s DQN Full Test in Forgiving Environment (bottom). These final figures show what happens when a tailored correct environment with decent hyperparameters is used for both DQN and DrQ. You can see that DrQ is able to allow the model to better deal with complexity later on and continuing increasing rather than plateuing or dipping like the normal DQN model is very prone to in any long training runs

5. Contributions

- This paper focuses on a reimplementation mainly of DrQ from Kostrikov: https://github.com/denisyarats/drq.The Main contribution is the use of a completely new environment snake to test this theory on increased learning rate and ability with random shifts. Creating the whole snake environment and tailoring it to use a reimplementation of Data Regularized Q was the main concept and idea.

5.1. Conclusion

This current implementation is able to benchmark DQN vs DrQ in the snake environment and shows good results for understanding the differences in these methods and the advantages of DrQ. The results show that DrQ does seem to increase learning capability and speed in harsh as well as forgiving environments and overall helps in a significant way when dealing with pixel based Q-learning. It is able to help the model deal with weird edge scenarios and as the environment gets harsher in a game like snake as time goes along since the snake gets longer, DrQ is able to help avoid overfitting or dips in training performance.

5.2. Future Directions

The future of this project would be to optimize the snake implementation even more. This would include making the environment easier so things such as adding wall colors or adding a different color for the snake head, as well as increasing the size of the playing field could all lead to better results and a more tailored environment that Q-learning would benefit from.

Acknowledgements

Used AI in help with conversion of text into Latex formatting as well as assistance in organization and polishing of writing.

References

1. Bellemare, M. G., Dabney, W., & Munos, R. (2017). A distributional perspective on reinforcement learning. *International Conference on Machine Learning (ICML)*. https://arxiv.org/abs/1707.06887
2. Fujimoto, S., van Hoof, H., & Meger, D. (2018). Addressing function approximation error in actor-critic methods. *International Conference on Machine Learning (ICML)*. https://arxiv.org/abs/1802.09477
3. Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *International Conference on Machine Learning (ICML)*. https://arxiv.org/abs/1801.01290
4. Kostrikov, I., Yarats, D., & Fergus, R. (2021). Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/2107.09645
5. Laskin, M., Srinivas, A., & Abbeel, P. (2020). Reinforcement learning with augmented data. *Advances in Neural Information Processing Systems (NeurIPS)*. https://arxiv.org/abs/2004.14990

6. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., . . . & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature, 518*(7540), 529–533. https://doi.org/10.1038/nature14236
7. Schwarzer, M., Anand, A., Goel, R., Hjelm, R. D., & Courville, A. (2021). Data-efficient reinforcement learning with self-predictive representations. *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/2007.05929
8. Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., de Las Casas, D., . . . & Silver, D. (2018). DeepMind Control Suite. *arXiv preprint arXiv:1801.00690*. https://arxiv.org/abs/1801.00690
9. Yarats, D., Kostrikov, I., & Fergus, R. (2021). DrQ: Data-regularized Q-learning. GitHub repository. https://github.com/denisyarats/drq