Combination Data Augmentation for Offline Rinforcement Learning in Robotics

Chin Yi Hong

S*chool of Computer Science*  
*University of Nottingham Malaysia*   
hcyyc6@nottingham.edu.my

*Abstract*— The field of computer vision has grown rapidly throughout the years, leading to the research of many data augmentations technique. In contrast, state-based inputs, which are commonly employed in robotics, have gotten comparatively less attention in field of data augmentation. Data augmentation within the context of offline reinforcement learning (ORL) has proven to be effective for acquiring useful skills for deployment in real-world settings with expensive and hazardous interactions. Nevertheless, adapting to new environments and utilizing static datasets presents difficulties for ORL. Three unique augmentation techniques will be used in this paper. This paper aims to find out the effects of combination data augmentations for offline reinforcement learning in robotics.

# Introduction

In traditional Reinforcement Learning (RL), agents are trained to learn from interactions with its environment. However, such interactions can be very costly. The recent advancement of hardware capabilities and data availability has help made ORL a powerful and cost-effective method. It enables learning complex behaviours from previously collected datasets without requiring new interactions with the environment. This makes ORL well-suited for robotics applications where real-world interactions can be prohibitively expensive, dangerous, or simply infeasible. By utilizing static, historical datasets, ORL methods alleviate many of the safety and computational burdens associated with traditional reinforcement learning approaches that learn on-policy through continued exploration. [1].

Despite the promise of ORL, the inability of offline agents to access the environment for collecting new data presents a significant limitation. ORL is met with the challenge of developing tactics to improve agents' capacity to generalise in new situations that they haven't experienced in training is the difficult part. Data augmentation is a technique that can address this problem and is being actively researched by many [2].

Currently, research efforts on effects of combined data augmentations for state-based inputs is very limited. This research aims to close the gap by examining how different augmentation techniques can work together to improve or affect an agent’s performance.

Specifically, this paper studies the following research questions: (1) Explore different augmentation combinations to determine optimal schemes for improving offline RL performance; (2) Study sample efficiency trade-offs with increasing augmentation complexity; (3) Compare combined versus stacked augmentation approaches to find the best-performing methodology.

My main contributions include the development and studies of a framework for augmentations which allows us to ascertain the best combination based on performance metrics, exploring the dynamics of sample efficiency and a thorough evaluation of the augmentation’s methods.

# Literature Review

## Offline Reinforcement Learning

The key challenge in offline RL is that the agent needs to generalize well beyond the given dataset to unseen states and actions. Naive algorithms like Q-learning tend to overestimate the values of out-of-distribution states and actions, leading to brittle policies [3]. Conservative Q-Learning (CQL) was proposed to address this overestimation issue by regularizing the Q-function on random state-action pairs [4]. However, the function approximation of deep Q-networks can still be quite poor in offline RL due to lack of environment interactions.

S4RL is a framework that combined both data augmentations and self-supervision techniques. The S4RL research paper builds on two recent lines of work - data augmentation and self-supervision in RL. Prior works like Data Regularized Q-learning (DrQ) [5] and Reinforcement Learning with Augmented Data (RAD) [6] have shown that simple data augmentations on the observation space can significantly improve performance. However, these focused on online RL from image observations. It has been unclear how to effectively perform augmentations on proprioceptive state spaces more common in robotics. S4RL focuses specifically on data augmentations for offline RL on such robotic state spaces. It outperforms prior augmentation techniques on offline RL benchmarks, showing the usefulness of state augmentations. The simplicity of S4RL also allows it to be easily combined with existing model-free offline RL algorithms [7]. As such, the best performing augmentation techniques has been selected from the S4RL for combination.

# Methdology

Initial baseline dataset for training is created using a simple Q-learning [8]. Then, 3 different augmentation techniques are applied to them by using different strategies including stacked, combined and individually. The augmented data will then be trained and tested using a Batch Deep Q-learning model (DQN) [9]. The goal of this experiment is to use data augmentations to smooth out the state-space for the static dataset.

## Data augmentations

Data augmentations is commonly used in the field of computer vision to expand the dataset size while preserving majority semantics of the data by using transformation like rotation and blur. Data augmentations for ORL in robotic is used by performing transformation on the states. The value of the state after transformation must be within the state space of the environment, only can it be considered a valid transformation. In ORL, data augmentations can be used to enhance the generalization of a dataset. However, the data in robotics is very delicate as aggressive augmentation strategies may remove the reward and state relationship, hindering the learning process. This is because the reward for original state may not line up with the rewards in augmented state as augmentation is only performed on the state. In other words, . Therefore, a fundamental assumption when performing data augmentations on state is that the reward model is smooth, implying that .

As such, the choice of transformation function must be one that perform local perturbation without changing the semantics. This is particularly more crucial in ORL, as an agent cannot revisit past states during training unlike traditional RL. The goal of using data augmentations is hoping that it could leverage the smoothness property of the reward model in the Markov Decision Process (MDP). By doing so, we can artificially explore and visit states that might exist in the broader state space but may not be present in the original dataset [7].

The following data augmentation techniques are taken from S4RL to carry out the experiments:

1. **Zero-Mean Gaussian Noise**:

,

where is drawn from a zero-mean Gaussian distribution , σ is a hyperparameter (standard deviation).

1. **Zero-Mean Uniform Noise**:

,

where is drawn from a zero-mean uniform distribution , is a hyperparameter (half the range of the uniform distribution).

1. **Adversarial State Training**:

,

This augmentation involves taking the gradient with respect to the value function to obtain a new state. is a hyperparameter to control the size of the gradient.

## Experiment Setup

2 environments have been chosen for evaluation which is the cartpole and mountain car problems from OpenAI Gym. 10 datasets will be used for evaluation and comparison in the experiment including baseline, 3 files consisting of baseline + augmented data, 2 files consisting of baseline + stacked augmented data, and 4 files consisting of baseline + combinations of augmented data. For the stacked files, Adversarial State Training is first applied onto baseline file followed by noise injection. For Cartpole and Mountain car, the baseline dataset consists of 50k and 130k transitions using q learning respectively. A cap of 600 episodes is set for cartpole and 300 episodes for Mountain car each trial. For data augmentations, the parameters used for σ, and ϵ are 0.0003, 0.0003 and 0.0001 respectively (taken from S4RL). Hyperparameters tuning are performed for DQN using grid-based search. The chosen parameters value is as follow: gamma: 0.99, learning rate: 0.0001, RMSprop optimizer and RELU activation function. The DQN used is a single hidden layer feed forward neural network for cartpole and 2 hidden layers for mountain car. CQL loss is also used to ensure smooth loss throughout the training [4]. To ensure reproducibility of the experiment, multiple trials are conducted for each of the dataset over 5 random seeds. The Mann-Whitney U tests are also employed for result analysis.

# Result

An important caveat should be understood around the experiment setup themselves before analyzing the result. Baseline success rates in Cartpole average around 70%, meaning most trials do reach convergence. Meanwhile in Mountain Car, only 5% of baseline attempts succeed. As such, apparent data augmentation gains may stem partly from the higher convergence likelihood alone rather than efficacy of the techniques themselves in simpler Cartpole. As for Mountain Car, low baseline convergence rates significantly reduce ability to evaluate the performance of data augmentations.

The results reveal that while data augmentation techniques led to modest raw performance gains in Cartpole, statistical testing shows these are not statistically significant improvements over baseline. The Mann-Whitney U tests find no significant differences in mean rewards or episodes to solution at the p < 0.05 level for any augmentation method when compared to baseline. This indicates the gains, including a 16% better mean reward for C\_AdvGaus, fail to rise above random chance. Similarly, none of the augmentations were able to meaningfully improve on the baseline's -195 average reward in Mountain Car despite substantially having more training data.

*Table 1. Experiment Results*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Environment | CartPole-v1 | | | | MountainCar-v0 | | | |
| Condition | Mean Episode | Std Dev Episode | Mean Reward | Std Dev Reward | Mean Episode | Std Dev Episode | Mean Reward | Std Dev Reward |
| BASELINE | 243 | 291.5 | 182.9 | 146.3 | 240.2 | 119.6 | -196 | 8 |
| GAUSSIAN | 242.6 | 291.8 | 210.5 | 177 | 300 | 0 | -200 | 0 |
| ADV | 249 | 286.6 | 194.1 | 110.8 | 300 | 0 | -200 | 0 |
| UNIFORM | 242.6 | 291.8 | 212.1 | 179.2 | 300 | 0 | -200 | 0 |
| ADVUNI | 249 | 286.6 | 196.7 | 114.6 | 300 | 0 | -200 | 0 |
| ADVGAUS | 249 | 286.6 | 193.7 | 110.7 | 300 | 0 | -200 | 0 |
| C\_GAUSUNI | 363.2 | 290 | 191 | 161.2 | 300 | 0 | -200 | 0 |
| C\_ADVSUNI | 362.2 | 291.2 | 204.9 | 162.1 | 300 | 0 | -200 | 0 |
| C\_ADVGAUS | 363.2 | 290 | 191 | 161.2 | 300 | 0 | -200 | 0 |
| C\_ALL | 250.8 | 285.4 | 160.9 | 122.2 | 300 | 0 | -200 | 0 |

# Disccusions

##### The inability of data augmentation techniques to yield measurable improvements may point to underlying issues in implementation methodology rather than deficiencies in the techniques themselves.

##### A key limitation was the use of simple DQN architectures, which likely fail to exploit the expanded state space distributions from augmentation. Advanced deep learning architectures such as Batch-Constrained deep Q-learning (BCQ), Quantile Regression DQN (QR-DQN) and Conservative Q-Learning (CQL) tailored to generalization capabilities may prove essential [11][12][4]. More advanced data augmentation techniques like Maximum-Entropy Adversarial and K-Mixup can also be considered when applying combination data augmentation [13][14]. Similarly, assessing performance in more complex offline robot learning environments such as MetaWorld and RoboSuite could better reveal benefits [15][16].

##### The lack of generative modeling beyond simple noise injection also restricts diversity of synthetic data. As seen in table 1, standard deviations remain high across techniques, especially for gaussian and uniform noise which is expected. Interestingly, stacked combinations ADVUNI and ADVGAUS have lower standard deviation values. This could indicate that the adversarial component pushes it toward edge cases, while noises add random perturbations which may be beneficial for generalization. Future work could explore combining data augmentation with distributional shift detection to ensure fidelity and coverage of generated samples [17].

##### Overall, results highlight that to unlock returns, data augmentation must be coordinated with innovations in network design, simulation complexity, and transition modeling. Rather than a standalone tool, it serves as an ingredient that when combined strategically with modern deep RL advancements, may still enhance offline and simulation-based autonomous system development. While results here were inconclusive, ample opportunity remains for advancement.

# Conclusion

In this paper, I evaluate combinations of data augmentation techniques. Despite performing extensive simulations across two environments, augmentation fails to yield statistically significant gains over baseline performance. As such, limitations in experiment protocols warrant further research into the potential synergies between combined data augmentations.

# References

1. Rafael Figueiredo Prudencio, Marcos, and Esther Luna Colombini. “A Survey on Offline Reinforcement Learning: Taxonomy, Review, and Open Problems.” IEEE transactions on neural networks and learning systems, Jan. 2023, doi: https://doi.org/10.1109/tnnls.2023.3250269.
2. Mumuni and F. Mumuni. “Data augmentation: A comprehensive survey of modern approaches.” Array, p. 100258, Nov. 2022, doi: https://doi.org/10.1016/j.array.2022.100258.
3. Jang, M. Kim, G. Harerimana, and J. W. Kim. “Q-Learning Algorithms: A Comprehensive Classification and Applications.” IEEE Access, vol. 7, pp. 133653–133667, 2019, doi: https://doi.org/10.1109/access.2019.2941229.
4. K. Aviral, Z. Aurick, T. George, and L. Sergey. “Conservative Q-Learning for Offline Reinforcement Learning.” Advances in Neural Information Processing Systems, vol. 33, 2020, Available: https://proceedings.neurips.cc/paper/2020/hash/0d2b2061826a5df3221116a5085a6052-Abstract.html.
5. Kostrikov, D. Yarats, and R. Fergus. “Image Augmentation Is All You Need: Regularizing Deep Reinforcement Learning from Pixels.” arXiv:2004.13649 [cs, eess, stat], Mar. 2021, Available: https://arxiv.org/abs/2004.13649.
6. M. Laskin, K. Lee, A. Stooke, L. Pinto, P. Abbeel, and A. Srinivas. “Reinforcement Learning with Augmented Data.” arXiv:2004.14990 [cs, stat], Nov. 2020, Available: https://arxiv.org/abs/2004.14990.
7. S. Sinha, A. Mandlekar, and A. Garg. “S4RL: Surprisingly Simple Self-Supervision for Offline Reinforcement Learning in Robotics.” proceedings.mlr.press, Jan. 11, 2022. https://proceedings.mlr.press/v164/sinha22a.html.
8. J. C. H. Watkins and P. Dayan. “Q-learning.” Machine Learning, vol. 8, no. 3–4, pp. 279–292, May 1992, doi: https://doi.org/10.1007/bf00992698.
9. J. Fan, Z. Wang, Y. Xie, and Z. Yang. “A Theoretical Analysis of Deep Q-Learning.” proceedings.mlr.press, Jul. 31, 2020. https://proceedings.mlr.press/v120/yang20a.
10. J. Han. “Data augmentation of state-based inputs for efficient offline reinforcement learning of robotic systems.” koasas.kaist.ac.kr, 2023, Accessed: Nov. 21, 2023. [Online]. Available: https://koasas.kaist.ac.kr/handle/10203/307724.
11. S. Fujimoto, E. Conti, M. Ghavamzadeh, and J. Pineau. “Benchmarking Batch Deep Reinforcement Learning Algorithms.” arXiv (Cornell University), Oct. 2019, doi: https://doi.org/10.48550/arxiv.1910.01708.
12. W. Dabney, M. Rowland, M. G. Bellemare, and R. Munos. “Distributional Reinforcement Learning with Quantile Regression.” arXiv:1710.10044 [cs, stat], Oct. 2017, Available: https://arxiv.org/abs/1710.10044.
13. L. Zhao, T. Liu, X. Peng, and D. Metaxas. “Maximum-Entropy Adversarial Data Augmentation for Improved Generalization and Robustness.” Neural Information Processing Systems, 2020. https://proceedings.neurips.cc/paper\_files/paper/2020/hash/a5bfc9e07964f8dddeb95fc584cd965d-Abstract.html (accessed Dec. 06, 2023).
14. J. Jang, J. Han, and J. Kim. “K-mixup: Data augmentation for offline reinforcement learning using mixup in a Koopman invariant subspace.” Expert Systems with Applications, vol. 225, p. 120136, Sep. 2023, doi: https://doi.org/10.1016/j.eswa.2023.120136.
15. T. Yu et al. “Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning.” arxiv.org, Oct. 2019, Available: https://arxiv.org/abs/1910.10897.
16. Y. Zhu, J. Wong, Ajay Mandlekar, and R. Martínez-Martín. “robosuite: A Modular Simulation Framework and Benchmark for Robot Learning.” arXiv (Cornell University), Sep. 2020, doi: https://doi.org/10.48550/arxiv.2009.12293.
17. Z. Liu et al., “An Empirical Study on Distribution Shift Robustness From the Perspective of Pre-Training and Data Augmentation,” arXiv.org, May 25, 2022. https://arxiv.org/abs/2205.12753 (accessed Dec. 07, 2023).