Adaptive DQN: Enhancing Performance of Deep Q-Networks through Test-Time Adaptation in the Face of Covariate Shift

Sorn Chottananurak
KAIST
sorn111930@kaist.ac.kr

Adriana Staudova
KAIST
yenyena@kaist.ac.kr

Junhee Lee
KAIST
jun17@kaist.ac.kr

Abstract

Deep Q-Learning (DQN) has achieved remarkable performance in various domains, but its effectiveness is often limited when the distributions of input during training and testing are misaligned. This phenomenon, known as domain shift or covariate shift [11], is prevalent in real-world scenarios where unexpected environmental changes and noise lead to poor model performance. For instance, in autonomous driving, rapidly changing weather conditions create such misalignment. To tackle this challenge, we propose Adaptive DQN (AdaDQN), the first DQN algorithm incorporating test-time adaptation (TTA) to mitigate DQN's performance degradation during testing in the presence of covariate shift, without retraining the model or accessing additional labeled data. We evaluate AdaDQN using several environments from the Arcade Learning Environment, a framework comprising Atari 2600 games. Our findings show that standard DQN significantly degrades under covariate shift, whereas our proposed AdaDQN can restore degraded DQN performance using only current unlabeled test data. Our code is available at: https://github.com/s6007541/AdaDQN.

1 Introduction

Deep Q-Networks (DQN) have achieved remarkable performance in various reinforcement learning tasks [5, 12, 4], especially in learning optimal policies from high-dimensional sensory inputs [9]. They are particularly useful in environments with vast action and state spaces, where the performance of traditional reinforcement learning algorithms suffers. Despite their success, DQNs generally exhibit limited robustness to changing environments. Misalignment between training and testing data can lead to significant performance degradation, posing a challenge for their incorporation into the real world. Real-world environments are often changing and highly variable, meaning that training conditions often do not perfectly match conditions during deployment. For example, applications in autonomous driving require agents to perform well in various weather and lighting conditions, making a model that can accommodate such changes a necessity.

We try to address this phenomenon of covariate shift [11], where the distribution of the input data during the testing or deployment phase is significantly different than that during the training phase. Such a shift often results in decreased model efficacy, as the assumptions and strategies learned during training no longer hold true in the new environment. In practice, the inability of DQNs to adapt to a changing environment can have serious ramifications. This challenge is compounded by the fact that in many real-world applications, it is impractical or impossible to retrain models with new data to reflect environmental changes. As such, there is a pressing need for innovative solutions enabling DQN models to dynamically adapt to new environments.

Contributions. (i) We implemented the very first full Arcade Learning Environment (ALE) with covariate shift, which effectively represents covariate shift in a reinforcement learning environment. (ii) We emphasize that environmental diversity in real-world scenarios poses a significant challenge.

Our findings reveal that standard DQN experiences considerable performance degradation in environments affected by covariate shift. (iii) As a solution, we introduce AdaDQN, a novel approach that integrates test-time adaptation (TTA) into the DQN framework to enhance its robustness against covariate shift without requiring extensive retraining or accessing additional labeled data.

2 Preliminaries

Arcade Learning Environment. Central to our study is the Arcade Learning Environment (ALE) [1], a platform that provides an extensive collection of Atari 2600 games, serving as a benchmark for evaluating advancements in reinforcement learning. ALE is an ideal testbed as it includes diverse challenges requiring strategic decision-making and rapid response adaptation. Deep Q-Networks (DQN) have achieved demonstrated success on many of these tasks [9], which makes them ideal for use as environments in this work, as their performance is expected to deteriorate under the covariate shift.

It is possible to vary conditions in ALE games, mirroring real-world scenarios where conditions during deployment can drastically differ from those during training, which presents an ideal setting to investigate the robustness of DQN under covariate shift. Our use of ALE serves a dual purpose: firstly, as a rigorous testing ground to assess the performance degradation of standard DQN models under changing game dynamics; and secondly, as a proving ground for AdaDQN, our enhanced DQN model incorporating TTA. By simulating a range of covariate shifts within these Atari games, we aim to demonstrate the capability of AdaDQN not only to withstand but also to adapt and maintain performance consistency in the face of environmental variability.

Covariate Shift. Covariate shift refers to the misalignment between the distributions of source and target data [11]. It is formally defined where source data, denoted as $\mathcal{D}_{\mathcal{S}} = \{\mathcal{X}^{\mathcal{S}}, \mathcal{Y}\}$, comprises instance-label pairs $(\mathbf{x}_i, y_i) \in \mathcal{X}^{\mathcal{S}} \times \mathcal{Y}$ that follow a probability distribution $P_{\mathcal{S}}(\mathbf{x}, y)$. Covariate shift occurs when there is a discrepancy between this source data distribution and that of the target data, represented as $\mathcal{D}_{\mathcal{T}} = \{\mathcal{X}^{\mathcal{T}}, \mathcal{Y}\}$. In this context, each target instance-label pair $(\mathbf{x}_j, y_j) \in \mathcal{X}^{\mathcal{T}} \times \mathcal{Y}$ adheres to a different target probability distribution $P_{\mathcal{T}}(\mathbf{x}, y)$. The assumption under covariate shift is that $P_{\mathcal{S}}(\mathbf{x}) \neq P_{\mathcal{T}}(\mathbf{x})$, while $P_{\mathcal{S}}(y|\mathbf{x}) = P_{\mathcal{T}}(y|\mathbf{x})$. This implies that the relationship between inputs and outputs remains constant, but the input data distribution changes, presenting a challenge for models trained under one distribution and tested under another.

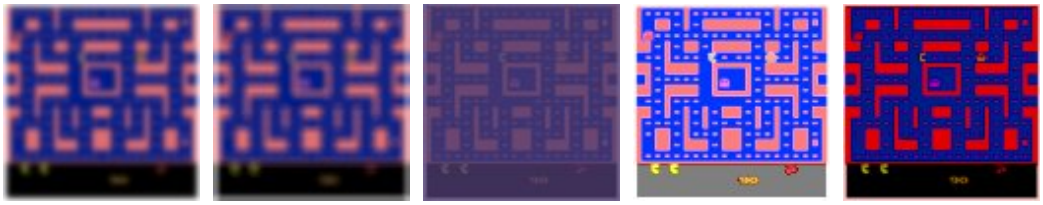
In the context of using DQN for playing Atari games under conditions of covariate shift, various types of shifts are considered. These types of covariate shifts are illustrated in Figure 1. Furthermore, the severity of these shifts is categorised into different levels, ranging from 1 to 5, with level 1 being the least severe and level 5 the most severe. The different levels of severity are depicted in Figure 2, using impulse noise as an example. Notably, all examples shown in Figure 1 represent the most severe level (level 5) of covariate shift.

Test-time adaptation. Test-Time Adaptation (TTA) is recognized as a technique developed to tackle covariate shift, enabling a model's adaptation to a new data distribution during the testing phase [10, 15]. When a model, denoted as $f(\cdot; \Theta)$, is pre-trained on source data $\mathcal{D}_{\mathcal{S}}$, TTA seeks to adapt this model to the target distribution $P_{\mathcal{T}}$ observed during testing. This adaptation uniquely occurs using only the target instances $\mathbf{x}_j$, generally without access to the corresponding labels y_j, as they are often unknown in practical applications. The adaptation process may involve modifying the model's parameters or employing strategies such as adjusting batch normalization statistics [10, 14]. The objective of TTA is to maintain or enhance the model's performance amidst changes in the input data distribution, ensuring reliable predictions even when the testing environment varies significantly from the training environment.

3 Methodology

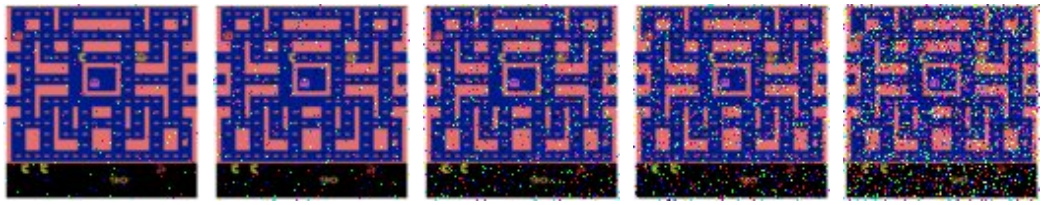
This study builds on an implementation of Deep Q-Network (DQN) as described in Huang et al. [6], adding batch normalization layers and test-time adaptation (TTA) to propose the enhanced AdaDQN. It then evaluates the AdaDQN performance under various covariate shift conditions simulated in various games from ALE.

(a) Original. (b) Gaussian Noise. (c) Shot Noise. (d) Impulse Noise. (e) Speckle Noise.



(f) Gaussian Blur. (g) Defocus Blur. (h) Contrast. (i) Brightness. (j) Saturate.

Figure 1: Comparison between 9 different types of level 5 covariate shifts in MS Pacman



(a) Level 1. (b) Level 2. (c) Level 3. (d) Level 4. (e) Level 5.

Figure 2: Comparison between 5 different level of covariate shift (Impulse Noise) in MS Pacman

The primary modification to the traditional DQN architecture involves the inclusion of batch normalization layers [7]. These layers are placed within the DQN's neural network to stabilise and accelerate the learning process. Batch normalization achieves this by normalizing the inputs to each layer, ensuring that the distribution of the inputs to each layer remains consistent across different mini-batches during training. This normalization is mathematically given as follows:

$$y = \frac{x - \mathbb{E}[x]}{\sqrt{\mathbb{V}[x] + \epsilon}} \times \gamma + \beta,$$

where x is the input to the layer, $\mathbb{E}[x]$ and $\mathbb{V}[x]$ are the mini-batch mean and variance, ϵ is a constant for numerical stability, and γ and β are the layer's learnable parameters. The rationale behind this addition is twofold: first, to improve the training efficiency of the DQN by reducing internal covariate shift, and second, to enhance the model's ability to generalize to new data distributions, which is critical under covariate shift conditions.

To simulate realistic scenarios where the DQN model might face covariate shift, various types of image distortions were applied to the Atari game frames during testing. These distortions include Gaussian noise, shot noise, impulse noise, speckle noise, Gaussian blur, defocus blur, and changes in contrast, brightness and saturation, each applied at varying severity levels from 1 to 5, as described in Section 2, and can be viewed in Figure 1. The intention behind these manipulations is to assess the DQN model's resilience and adaptability under conditions that deviate from the training environment.

Our AdaDQN integrates TTA which involves adjusting the batch normalization layers during inference to adapt to the new data distribution encountered at test time. Unlike the training phase, where batch normalization statistics are learned from the training batches, during the testing phase, these statistics are dynamically adjusted based on the incoming test batch. This adjustment allows the model to recalibrate its internal representations to better suit the new data distribution, thereby aiming to maintain performance consistency even under the covariate shift [10, 14].

The performance evaluation of the AdaDQN model was conducted using four different Atari games from the ALE (Phoenix, Space Invader, Ms Pacman, Air Raid), comparing the performance of AdaDQN and the original DQN in covariate shift conditions. The key metric for evaluation was the

Table 1: Total episodic return on 4 different Atari games under 9 types, 5 levels of covariate shifts. Original contains only unshifted target samples, while other scenarios include both unshifted and each type of noisy samples specified. **Bold** numbers are the highest episodic return. Averaged over 1,000 evaluation episodes.

				Covariate Shift									
Shift Level	Environment	Method	Original	Gaussian Noise	Shot Noise	Impulse Noise	Speckle Noise	Gaussian Blur	Defocus Blur	Contrast	Brightness	Saturate	Avg.
1	Phoenix-v5	DQN	**2648.2**	1675.3	2608.8	**1139.0**	**2679.0**	2346.9	2317.1	1947.9	85.8	102.6	1755.1
		Ours	2625.6	**2380.7**	**2767.2**	1054.4	2635.9	**2674.7**	**2717.2**	**2087.4**	**2742.4**	**2485.9**	**2417.1**
	SpaceInvader-v5	DQN	519.2	173.6	518.9	131.5	527.3	406.4	400.5	511.9	170.5	77.7	343.7
		Ours	**548.3**	**423.0**	**549.5**	**168.5**	**544.1**	**558.0**	**564.4**	**533.5**	**532.2**	**543.7**	**496.5**
	MSPacman-v5	DQN	2419.0	1708.2	1806.6	1237.1	1978.4	1688.8	**2121.5**	**2084.9**	75.6	296.3	1541.6
		Ours	**2523.0**	**1835.9**	**1896.2**	**1799.3**	**2047.2**	**1970.0**	2082.2	2038.0	**1859.2**	**1790.0**	**1984.1**
	AirRaid-v5	DQN	**8164.3**	400.5	4526.3	282.8	**7686.0**	9375.8	8715.0	8726.0	8594.0	50.3	5652.1
		Ours	8004.5	**1775.0**	**4634.5**	**901.5**	7441.0	**9403.3**	**8884.3**	**8956.5**	**9281.0**	**7609.5**	**6689.1**
2	Phoenix-v5	DQN	2629.1	557.4	2569.9	508.8	2436.7	1165.5	1959.2	1263.6	110	88.4	1328.9
		Ours	**2691.2**	**1584.5**	**2795.4**	**697.4**	**2620.8**	**2108.9**	**2387.7**	**1383.9**	**2340.2**	**2564.9**	**2117.5**
	SpaceInvader-v5	DQN	511.9	117.4	502.7	85.4	506.6	80.6	80.4	495.6	51.9	259.3	269.2
		Ours	**519.8**	**281.3**	**519.6**	**120.4**	**510.6**	**253.3**	**487.2**	**512.4**	**549.9**	**557.0**	**431.1**
	MSPacman-v5	DQN	2414.6	1304.1	1743.3	985.6	1863.8	365.9	683.5	1488.7	60.0	70.0	1098.0
		Ours	**2439.2**	**1809.6**	**1786.3**	**1714.1**	**1907.1**	**2051.3**	**1786.8**	**1610.2**	**1833.9**	**1847.7**	**1878.6**
	AirRaid-v5	DQN	**8373.0**	119.3	1755.3	94.5	3906.5	4442.0	7012.8	4390.8	6081.3	252.0	3642.7
		Ours	8211.3	**717.5**	**3109.5**	**451.0**	**4543.8**	**4536.3**	**8657.8**	**5927.5**	**7606.0**	**8017.8**	**5177.8**
3	Phoenix-v5	DQN	2537.9	133.4	2465.4	174.7	2417.0	844.8	1598.5	977.7	113.0	34.7	1129.7
		Ours	**2653.2**	**1172.7**	**2644.5**	**650.4**	**2664.4**	**1778.5**	**2512.0**	**1131.2**	**2578.1**	**2472.2**	**2025.7**
	SpaceInvader-v5	DQN	519.0	126.7	497.5	92.1	535.3	4.1	64.0	484.5	48.3	429.7	280.1
		Ours	**548.7**	**191.5**	**499.8**	**137.7**	**555.6**	**50.8**	**241.1**	**494.2**	**544.6**	**541.6**	**380.5**
	MSPacman-v5	DQN	2399.7	1126.9	1546.7	780.1	1700.1	142.7	201.0	**1256.0**	60.0	60.0	927.3
		Ours	**2418.1**	**1714.1**	**1796.1**	**1693.5**	**1809**	**1560.8**	**1576.4**	1177.4	**1850.4**	**1762.5**	**1735.8**
	AirRaid-v5	DQN	7903.3	66.0	373.8	84.8	2751.5	3465.0	5012.0	2889.5	3142.5	4991.0	3068.0
		Ours	**7941.0**	**437.5**	**1671.8**	**307.3**	**4400.0**	**4096.0**	**5327.8**	**3865.5**	**6654.8**	**7413.8**	**4211.6**
4	Phoenix-v5	DQN	2633.6	103.6	2470.2	108.8	2584.9	560.9	1311.8	667.2	109.4	120.2	1067.1
		Ours	**2608.6**	**1005.7**	**2676.3**	**421.9**	**2697.9**	**1470.9**	**2290.2**	**806.2**	**2524.7**	**2862.6**	**1936.5**
	SpaceInvader-v5	DQN	**511.1**	123.8	439.3	85.4	496.7	0.0	0.0	441.5	105.0	75.4	227.8
		Ours	510.9	**159.0**	**458.4**	**167.1**	**505.8**	**60.0**	**62.0**	**467.2**	**518.4**	**555.2**	**346.4**
	MSPacman-v5	DQN	2338.3	1047.8	1246.0	316.2	1481.0	109.3	103.4	**1074.3**	60.0	60.0	783.6
		Ours	**2501.0**	**1757.4**	**1709.9**	**1441.7**	**1781.9**	**1209.6**	**1129.9**	955.4	**1760.8**	**1129.7**	**1537.7**
	AirRaid-v5	DQN	7902.0	58.0	234.0	133.3	1591.3	2845.3	3891.5	1659.5	2106.25	1456	2187.7
		Ours	**8059.5**	**368.5**	**950.5**	**235.3**	**3243.0**	**3058.5**	**4035.8**	**1938.3**	**7329.0**	**7611.0**	**3682.9**
5	Phoenix-v5	DQN	2635.9	92.0	2395.1	96.6	2317.5	216.7	230.3	415.8	115.2	117.2	863.2
		Ours	**2679.2**	**1016.9**	**2692.3**	**308.8**	**2643.1**	**1090.3**	**1600.9**	**458.4**	**2509.7**	**2474.1**	**1747.4**
	SpaceInvader-v5	DQN	510.1	119.6	429.6	81.5	490.6	0.0	0.0	401.2	0.2	0.9	203.4
		Ours	**513.5**	**143.3**	**440.8**	**197.8**	**496.9**	**86.2**	**79.3**	**404.0**	**545.9**	**522.9**	**343.0**
	MSPacman-v5	DQN	2453.2	957.2	990.1	212.2	1105.8	95.4	90.0	725.7	60.0	60.0	675.0
		Ours	**2460.3**	**1621.1**	**1755.6**	**1119.6**	**1788.5**	**818.0**	**971.7**	**718.0**	**1781.6**	**1493.5**	**1452.8**
	AirRaid-v5	DQN	7879.5	65.0	108.75	146.0	527.5	2302.0	2966.5	638.5	1196.8	624.3	1645.5
		Ours	**7941.3**	**309.8**	**755.3**	**224.3**	**2006.0**	**2970.5**	**3908.8**	**1321.8**	**6891.5**	**7480.5**	**3381.0**

models' ability to maintain high performance levels as measured by the average of 1,000 episodic return.

4 Results and Discussion

This section describes the results of our experiments with the AdaDQN model on four different Atari games under various covariate shift scenarios. The experiments are designed to evaluate the effectiveness of AdaDQN in adapting to changes in data distribution between training and deployment, as compared to the vanilla DQN.

We examine the performence of both AdaDQN and the original DQN under 5 different levels of covariate shift, as described in Section 2. The shift conditions include Gaussian noise, shot noise, impulse noise, speckle noise, Gaussian blur, defocus blur, and changes in contrast, brightness and saturation, each applied at severity levels ranging from 1 to 5. As demonstrated in Table 1, which details the episodic returns for each model under each condition, AdaDQN not only consistently outperforms the original DQN under various shifted conditions but also maintains a robust performance in unshifted environments. This finding suggests that AdaDQN could potentially serve as a more versatile and reliable model for general reinforcement learning tasks, beyond just those subjected to covariate shift.

Table 2: Statistical analysis of total rewards in the Phoenix game under various covariate shifts. The table includes Welch's t-test p-values and Hedges' g values. Averaged over 1000 evaluation episodes.

			Covariate Shift								
Shift Level	Measure	Original	Gaussian Noise	Shot Noise	Impulse Noise	Speckle Noise	Gaussian Blur	Defocus Blur	Contrast	Brightness	Saturate
1	p-value	3.7e-3	0.0	1.3e-123	4.7e-53	1.6e-8	9.5e-302	0.0	1.2e-105	0.0	0.0
	Hedge's g	0.1	-5.2	-1.1	0.7	0.3	-2.0	-2.4	-1.0	-36.2	-19.4
2	p-value	3.9e-17	0.0	6.8e-143	8.2e-131	5.1e-148	0.0	0.0	1.1e-68	0.0	0.0
	Hedge's g	-0.4	-12.6	-1.2	-1.2	-1.3	-7.5	-2.6	-0.8	-19.1	-21.3
3	p-value	3.8e-66	0.0	9.3e-152	0.0	8.4e-195	0.0	0.0	8.2e-168	0.0	0.0
	Hedge's g	-0.8	-11.3	-1.3	-3.8	-1.5	-7.4	-6.2	-1.4	-18.3	-23.8
4	p-value	4.1e-3	0.0	4.0e-146	0.0	1.4e-50	0.0	0.0	0.0	0.0	0.0
	Hedge's g	0.1	-8.8	-1.3	-2.5	-0.7	-6.8	-9.0	-3.7	-26.4	-25.0
5	p-value	4.9e-9	0.0	1.1e-276	0.0	2.7e-234	0.0	0.0	0.0	0.0	0.0
	Hedge's g	-0.3	-7.9	-1.9	-12.5	-1.7	-37.6	-12.5	-2.1	-21.4	-18.1

To assuredly demonstrate the superiority of AdaDQN performance under all levels of covariate shift, we perform a statistical analysis of the average episodic returns of each model in the Phoenix game. The results can be viewed in Table 2. The statistical analysis includes p-values of Welch's t-test and Hedge's g, the variations of the commonly used Student's t-test and Cohen's d for distributions with significant differences in variance [13, 8]. The generally accepted p-value for statistical significance is $p < 0.05$ and Hedge's g is generally reported as showing a large effect size if $|g| > 0.75$ [3, 2]. As evident from this analysis, the improvements of AdaDQN are both statistically significant and demonstrate very large effect sizes in most instances. This underscores the model's effectiveness in handling the challenges posed by covariate shift, which is a common concern in real-world applications of reinforcement learning.

One notable observation is the relative stability of AdaDQN's performance across increasing levels of shift severity. While the original DQN shows a marked decline in performance under higher levels of shift, AdaDQN demonstrates a remarkable resilience. This can be attributed to the TTA, which inherently grants it adaptive capabilities.

However, our study is not without limitations. Our algorithm did not consider various temporal distribution shifts, such as temporally-correlated streams and domain changes which are common in domain shift research area. Towards developing a TTA algorithm robust to any test streams in the wild, more comprehensive and realistic considerations should be taken into account, which we believe is a meaningful future direction.

5 Conclusion

We investigate the problem of having noisy samples and the performance degradations caused by those samples. To tackle the problem, we propose AdaDQN, a DQN enhanced with adaptive batch normalization and TTA. Our evaluation with four noisy environments demonstrates that our model significantly outperforms the original DQN model in those scenarios. Its ability to maintain high performance under a variety of conditions suggests a promising direction for future research and applications in dynamic environments.

References

[1] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, June 2013.

[2] Christopher R Brydges. Effect size guidelines, sample size calculations, and statistical power in gerontology. *Innovation in Aging*, 3(4), August 2019.

[3] T Dahiru. P-value, a true test of statistical significance? a cautionary note. *Annals of Ibadan Postgraduate Medicine*, 6(1), March 2011.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Commun. ACM*, 63(11):139–144, oct 2020.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[6] Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João G.M. Araújo. Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274):1–18, 2022.

[7] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

[8] Lifeng Lin and Ariel M. Aloe. Evaluation of various estimators for standardized mean difference in meta-analysis. *Statistics in Medicine*, 40(2):403–426, November 2020.

[9] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013.

[10] Zachary Nado, Shreyas Padhy, D Sculley, Alexander D'Amour, Balaji Lakshminarayanan, and Jasper Snoek. Evaluating prediction-time batch normalization for robustness under covariate shift. *arXiv preprint arXiv:2006.10963*, 2020.

[11] Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2008.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[13] Graeme D. Ruxton. The unequal variance t-test is an underused alternative to student's t-test and the mann–whitney u test. *Behavioral Ecology*, 17(4):688–690, May 2006.

[14] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 11539–11551. Curran Associates, Inc., 2020.

[15] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021.