consisted of a hidden layer with 512 neurons activated by ReLU and an output layer with one neuron. The output of the critic branch estimated the state-value function $V(s)$.

3.4.1.2 Discriminator Network

Like the agent network, the discriminator network consisted of a feature extraction and a fully connected module. The feature extraction module in this network followed a similar layout to the agent network, which consisted of three convolutional layers with the same number of kernels, stride lengths, and attention locations. However, instead of using ReLU to activate the convolutional layers, it used a leaky ReLU, following Gulrajani et al. (2017) and Miyato et al. (2018).

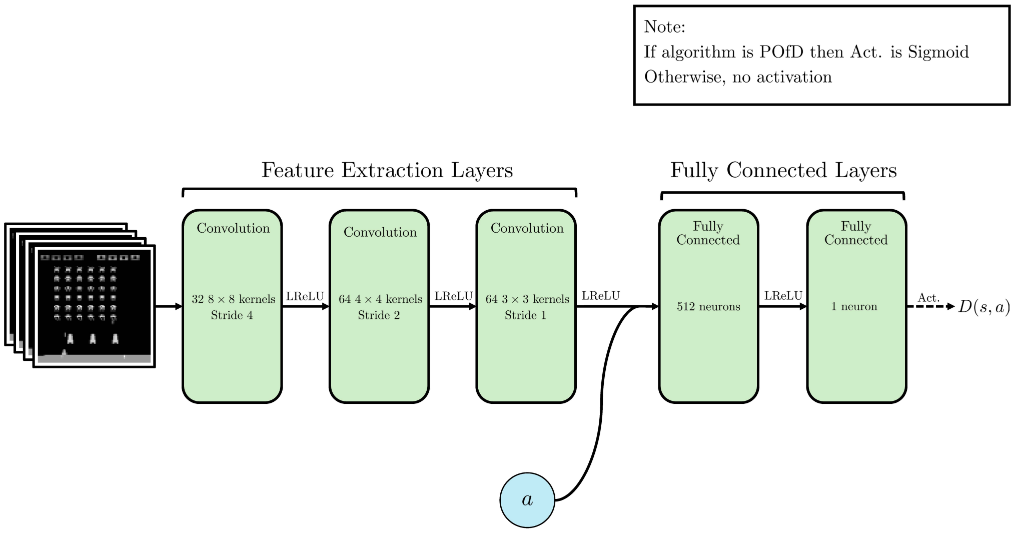


Figure 3.4: *A visualization of the implemented discriminator network.*

As for the fully connected module in the network, it accepted the feature maps from the feature extraction layers and concatenated it with the corresponding action. These concatenated features were passed into a sequence of fully connected layers with a layout consisting of a hidden layer with 512 neurons, activated with a leaky ReLU, and an output layer with one neuron. For POfD, this output neuron implemented a sigmoid activation function. Meanwhile, for SNGAL, the output neuron was not activated. Either way, the output produced the discriminator value $D_w(s, a)$.

3.4.2 Implemented Hyperparameters

The experiments implemented PPO, POfD, and SNGAL with almost identical hyperparameters. These hyperparameters were taken from the previous experiments of Schulman et al. (2017) and are listed in Table 3.1. However, it should be noted that one hyperparameter was implemented differently in the experiments, which was the trade-off coefficients for POfD and SNGAL. This was done because of the different formulations of the reshaped rewards in the two algorithms, where POfD can only perform guided reinforcement learning and reinforcement learning; SNGAL can perform all three types of learning: reinforcement learning, imitation learning, and guided reinforcement learning.

Table 3.1: *The implemented hyperparameters for PPO, POfD, and SNGAL. α is linearly annealed from an inital value of 1 to* 0*, at the final timestep. β is annealed from an initial value of 1 to 0, based on the cosine scheduler.*

Algorithm	Hyperparameter	Value
PPO, POfD, SNGAL	Step size (T)	128
	Minibatch size	32×8
	GAE parameter	0.95
	Number of agents	8
	Generator epochs	4
	Discount factor (γ)	0.99
	Agent learning rate	$2.5 \times 10^{-4} \times \alpha$
	Number of updates (I)	10000
	Clipping coefficient (ϵ)	$0.1 \times \alpha$
	Entropy coefficient (c_2 or λ_2)	0.01
	Value function coefficient (c_1)	0.5
POfD, SNGAL	Discriminator epochs	5
	Discriminator learning rate	2.5×10^{-4}
POfD	Trade-off Coefficient (λ_1)	0.01
SNGAL	Trade-off Coefficient (λ_1)	$1 \times \beta$

For the selected values, a constant trade-off coefficient value of 0.01 was used for POfD. In contrast, SNGAL implemented a scheduled trade-off coefficient that was initialized to 1 at the beginning of training ($i = 0$) and gradually decreased to 0, based on a cosine function, by the final update ($i = I$):

$$\lambda_1(i) = \cos\left(\frac{i}{I}\frac{\pi}{2}\right). \tag{3.3}$$

The intuition behind the cosine annealing was that the agent should learn as much as possible from the demonstrations before attempting to learn by itself.

3.4.3 Visualizing Saliency

One of the objectives of this research was to investigate the effects of visual attention mechanisms on guided reinforcement learning models. To achieve this goal, a visual analysis was conducted by analyzing the GradCAM (Selvaraju et al., 2017) produced by the agent's policy.

The visualization of the GradCAM for the selected action, a, from a policy, π_θ, involved four steps. Firstly, the gradients of the action's score before the softmax, y^a, were computed with respect to the feature map activation A^k of layer k, with k chosen to be the final activation layer. After that, the gradients were global-average-pooled over the width and height dimensions to obtain the neuron importance weights α_k^a:

$$\alpha_k^a = \overbrace{\frac{1}{Z}\sum_i \sum_j}^{\text{global average pooling}} \quad \frac{\partial y^a}{\partial A_{ij}^k}. \tag{3.4}$$

Next, a linear combination of the importance weights and the forward feature-map activation were calculated and activated by a ReLU to obtain the GradCAM:

$$L_{\text{Grad-CAM}}^a = ReLU\left(\sum_k \alpha_k^a A^k\right). \tag{3.5}$$

Finally, the GradCAM was bilinearly extrapolated to the same size of the input image before being overlayed on the input image to produce a heatmap of the network's gradients.

CHAPTER IV
RESULTS AND DISCUSSIONS

This chapter will explain the results of the two experiments conducted in this research. The chapter will begin by describing the results of the first experiment performed to evaluate POfD and SNGAL in Space Invaders. After that, it will detail the results of the second experiment conducted to integrate guided reinforcement learning with attention mechanisms. Finally, the chapter will conclude with a discussion, connecting the obtained results with the previously discussed literature review.

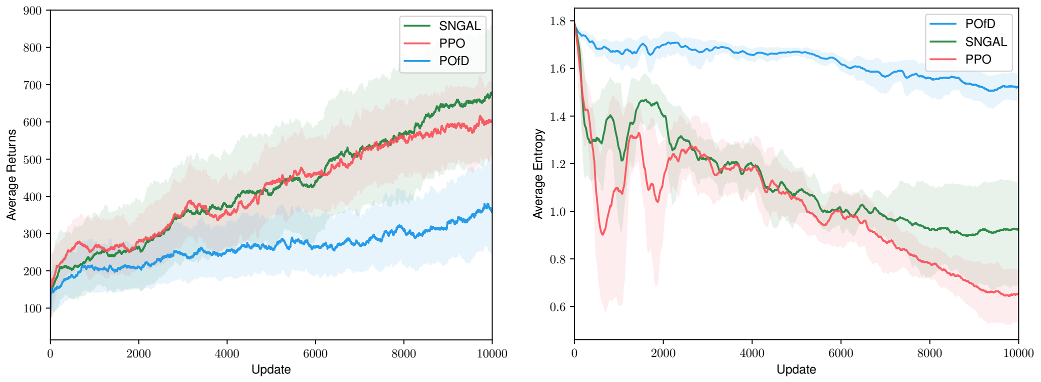
4.1 Implementing Guided Reinforcement Learning in Space Invaders

Table 4.1 presents a summary of the benchmark results obtained in the first experiment, while Figure 4.1 shows the learning curves of the models. Table 4.1 shows that SNGAL achieved the best performance in the experiment, with a pooled average return improvement of 17% over PPO and a significant 94% improvement over POfD. Additionally, SNGAL obtained the highest minimum and maximum average returns out of the three algorithms. Figure 4.1a illustrates that PPO and SNGAL had similar learning curves up to the 8000th update, but after that, SNGAL learned much faster and achieved greater returns.

On the other hand, Table 4.1 shows that POfD performed the worst in the experiment, with a 40% performance degradation compared to PPO. The high entropy of POfD, as shown in Figure 4.1b, suggests that the algorithm experienced learning difficulties.

Table 4.1: *The returns of PPO, POfD, and SNGAL obtained from 200 evaluation episodes performed in three different runs. The table shows the returns aggregated on the pooled average, pooled standard deviation, minimum average, and maximum average. Here, minimum average and maximum average refers to the minimum and maximum average returns obtained among the three different runs, respectively.*

Method	Returns			
	Pooled Average	Pooled Std. Dev.	Minimum Average	Maximum Average
PPO	603.14	207.71	531.68	702.53
POfD	361.35	191.27	263.43	477.28
SNGAL	**701.06**	206.70	**651.75**	**748.00**



(a) *(b)*

Figure 4.1: *The learning curves of PPO, POfD, and SNGAL averaged on three different runs. (a) shows the average returns, while (b) shows the average entropy.*

4.2 Integrating Attention into Guided Reinforcement Learning

After showing that SNGAL obtained the highest returns, the second experiment attempted to integrate attention with the algorithm. In this experiment three different attention modules were attempted: non-local block, convolutional block attention module (CBAM), and dual attention.

4.2.1 Where To Add the Non-Local Block?

Since Wang et al. (2018) did not suggest a specific location for the non-local block, the first part of the second experiment aimed to determine the optimal configuration for the module. This sub-experiment evaluated the non-local block at three different locations: after the first, second, or third convolutional layers. To conserve computational resources, this evaluation was only performed on one seed.

Table 4.2 summarizes the results of the sub-experiment. The table shows that the non-local block worked best at the third convolutional layer, suggesting that the module can better attend to objects in the images with higher-level feature maps.

Table 4.2: *The returns of an agent trained using SNGAL with the non-local block applied after either the first, second, or third convolutional layers. Here, "Conv." refers to the convolutional layer where the attention mechanism is applied.*

Method	Attn. Location	Returns	
		Average	Std. Dev.
SNGAL	Conv. 1	517.33	159.25
	Conv. 2	520.80	173.21
	Conv. 3	**735.25**	205.28

4.2.2 Benchmarking Visual Attention Mechansims

After discovering that the non-local block worked best at the third convolutional layer, the next experiment compared SNGAL with the three attention mechanisms: non-local block, CBAM, and dual attention. For, the non-local block, the module was applied after the third convolutional layer, following the results of the previous sub-experiment. Meanwhile, for CBAM and dual attention, the modules were applied at the locations suggested by their original papers, which were at all the convolutional layers and the final convolutional layer, respectively. The benchmark results of the experiment are summarized in Table 4.3, while Figure 4.2 illustrates the learning curves of the models.

Table 4.3: *The returns of PPO, pure SNGAL, SNGAL with non-local block (third convolutional layer), SNGAL with CBAM, and SNGAL with dual attention obtained from 200 evaluation episodes performed in three different runs. The table shows the returns aggregated on the pooled average, pooled standard deviation, minimum average, and maximum average. Here, minimum average and maximum average refers to the minimum and maximum average returns obtained among the three different runs, respectively.*

Method	Attention Type	Returns			
		Pooled Average	Pooled Std. Dev.	Minimum Average	Maximum Average
PPO	-	603.14	207.71	531.68	702.53
SNGAL	-	**701.06**	206.70	**651.75**	**748.00**
	Non-local	664.34	200.92	561.85	735.25
	CBAM	608.97	185.03	577.08	659.00
	Dual	615.63	178.03	595.95	640.58

Table 4.3 shows that pure SNGAL still achieved the highest pooled average returns. Non-local block SNGAL came in second, with a 37-point reduction in pooled average returns compared to pure SNGAL, but still performed better than PPO by 10%. In contrast, CBAM and Dual attention performed the worst, with only a slight improvement over PPO.

Figure 4.2 illustrates that all algorithms had an increasing trend in their learning curves. Among them, non-local block SNGAL was able to sustain the highest pooled average returns up to the 7000$^{\text{th}}$ update. However, after that update, pure SNGAL was able to catch up and obtain better overall performance in the end.

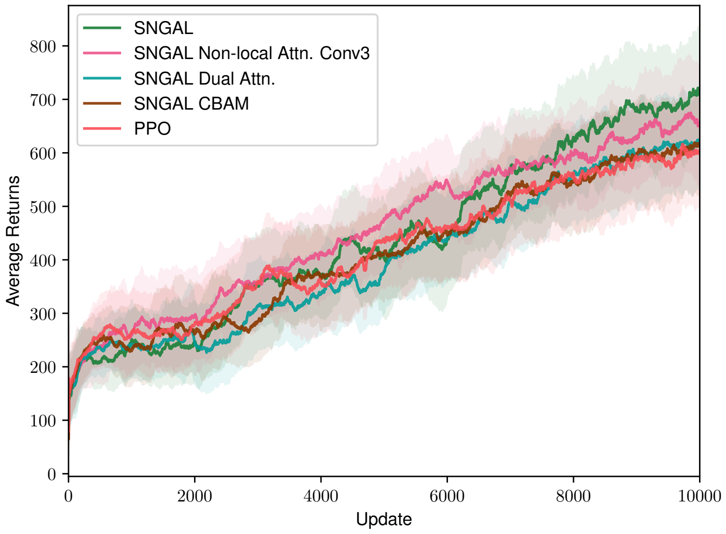
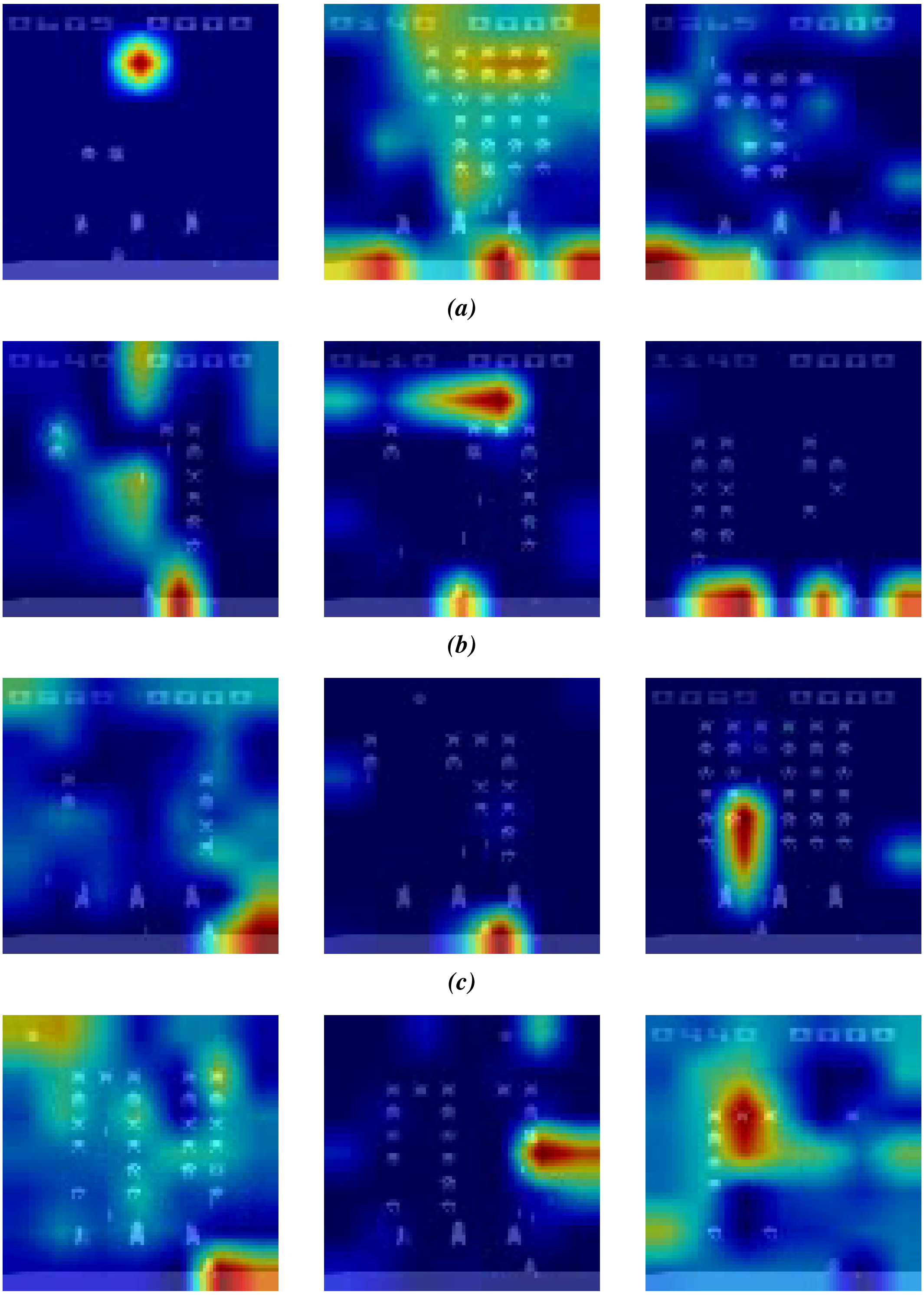


Figure 4.2: *The learning curves of pure SNGAL and SNGAL with the three attention mechanisms: non-local block, CBAM, and dual attention. The curves show the returns averaged on three different runs.*

4.2.3 Visualizing the GradCAMs of Visual Attention Mechanisms

In order to help understand how attention mechanisms affect guided reinforcement learning models, the saliency maps of pure SNGAL and SNGAL with the three attention mechanisms (non-local block, CBAM, and dual attention) were visualized and analyzed. In this visual analysis, three state-action pairs were randomly sampled from each model's trajectory and then their GradCAMs were calculated. Figure 4.3 shows these GradCAMs.

In Figure 4.3, it is apparent that each model views the environment differently. Figure 4.3a shows that the pure SNGAL model had difficulty focusing on the objects in the environment. This was demonstrated by the model's attention to a large area of aliens and also its attention to the areas which did not contain any objects. Figure 4.3b shows that with the non-local block, SNGAL displayed better focus to the objects in the environment, which included the projectiles, player, and aliens. As for CBAM SNGAL, Figure 4.3c reveals that the model highlights saliency in only a narrow region of the environment close to the player. Lastly, Figure 4.3d demonstrates dual attention's ability to attend to aliens, player, and mothership. However, the model failed to focus on the aliens near the bottom of screen, which would have caused the game to end.



(a)

(b)

(c)

(d)

Figure 4.3: *The GradCAMs of (a) SNGAL, (b) SNGAL with non-local block, (c) SNGAL with CBAM, and (d) SNGAL with dual attention visualized on three randomly sampled states. The states are visualized on the first frames, where the red regions indicate where the model attends to, while the blue regions indicate where the model does not attend to.*

4.3 Discussions

The main objective of this research was to investigate the effects of visual-attention mechanisms on guided reinforcement learning models. To this end, two experiments were conducted, and their findings were presented in the previous sections.

The first experiment aimed to evaluate guided reinforcement learning algorithms in Space Invaders. The results showed that the POfD algorithm failed to learn in the environment, which was demonstrated by the 40% reduction in performance compared to PPO and the high entropy exhibited by the algorithm. This learning instability was likely caused by its use of the minimax GAN, which is known to be unstable during training (Arjovsky & Bottou, 2017; Miyato et al., 2018), as well as the high dimensionality of the environment (Miyato et al., 2018), which can be difficult to learn. As part of this experiment, a novel guided reinforcement learning algorithm called SNGAL was also introduced. SNGAL utilized the Spectral Normalization (Miyato et al., 2018) approach to stabilize GAN training and a reshaped reward function scheduled to perform imitation learning before reinforcement learning. This approach resulted in a significant 94% improvement in performance over POfD and a 17% improvement over PPO, demonstrating the efficacy of spectral normalization in stabilizing GANs in high-dimensional MDP environments.

However, it must be noted that, for most of the training, the learning curves of SNGAL (shown in Figure 4.1a) were almost identical to PPO. SNGAL only significantly outperformed PPO after it had reached the 8000th update. One possible explanation to this sudden increase in returns could be attributed to the models' learning how to destroy the mothership, which would have resulted in a bonus of 200 points. Since the expert trajectories did not demonstrate the destruction of the mothership, this must have occurred due to self-exploration, which was prioritized after the 8000th update, when $\lambda_1 \leq 0.3$.

After obtaining that SNGAL worked best in Space Invaders, the second experiment was then performed to explore the effects of visual attention on the algorithm. In this experiment, three attention mechanisms (non-local block, CBAM, and dual attention) were integrated with SNGAL. The results showed that, when evaluated on final model performance, attention mechanisms did not lead to improved performance compared to pure SNGAL. The best-performing attention mechanism, non-local block, obtained a 37-point reduction in pooled average returns compared to pure SNGAL but still showed a respectable 10% improvement over PPO. CBAM and dual-attention, on the otherhand, performed significantly worse than pure SNGAL and only obtained a marginal improvement over PPO.

Although the attention modules failed to improve SNGAL in final model performance, non-local block SNGAL was exhibited to be performant during training. Figure 4.2 displayed this by showing the models' ability to obtain the highest returns up to the 7000th update before lagging behind in acquired returns to pure SNGAL. There are two possible explanations for this phenomenon. The first possibility is that attention limits the exploration of the agent, while the second possibility is that the pure SNGAL models were initialized with seeds that made the agents explore the destruction of the mothership. The latter option is more likely since previous studies by Manchin et al. (2019) and Tang et al. (2020) had demonstrated that attention does not limit exploration but can improve the performance of reinforcement learning models in some environments. If this is the case, then there is potential for further improvements to the models.

Regarding the perceptual effects of attention, the GradCAMs in Figure 4.3 suggest that visual attention mechanisms can help guided reinforcement learning models better attend to objects in the environment. This was demonstrated by the attention models' ability to focus on salient regions of the images, including the locations of the aliens and projectiles. This result is in line with the claims of Woo et al. (2018) that attention mechanisms help deep learning models focus on the salient objects in an image. Meanwhile, when attention modules were not applied, the models exhibited less structured attention, where they not only focused on a large area of aliens but also on random regions of the images that did not contain any objects. This unstructured attention can be attributed to the feature extraction layers of the non-attention based models utilizing only convolutional layers, which attend to images purely based on the presence of certain features.

CHAPTER V
CONCLUSION AND RECOMMENDATIONS

This chapter will conclude the study by summarizing the key research findings in relation to the research questions and objectives. It will also provide recommendations for future research.

5.1 Conclusion

Based on the research findings, two conclusions can be drawn related to the research questions:

1. The first conclusion is that the implemented visual-attention mechanisms did not lead to improved performance of the guided reinforcement learning models. However, it is worth noting that one of the implemented attention modules (non-local block attention) obtained the highest pooled average returns during the early to middle stages of training, suggesting that there is potential for further improvements to the models.

2. The second conclusion is that visual attention mechanisms helped the guided reinforcement learning models to focus on the objects in the environment, including a better focus on the location of the player, aliens, and projectiles in Space Invaders. In contrast, models without attention modules exhibited less attention to those objects.

5.2 Recommendations

Further research is needed to establish whether guided reinforcement learning with attention performs better than guided reinforcement learning without attention. For this reason, this study provides three recommendations for future research.

Firstly, future experiments in the Atari environments should explore training the models to a higher number of timesteps. A good number to follow would be the 40 million timesteps of Schulman et al. (2017) or the 50 million timesteps of Mnih et al. (2015). In this way, attention models will have more time to learn the importance of each object in the environment.

Secondly, it may be worthwhile to explore the implementation of a different network architecture. One idea could be to use a transformer model for both the agent (W. Li et al., 2023) and discriminator (Jiang et al., 2021) models. Another idea could be to implement pooling layers in the feature extraction layers to provide translation invariance to the model.

Lastly, future research could explore a different reshaped reward function for SNGAL. A different reshaped reward function might result in a different outcome.

REFERENCES

Achiam, J. (2018, November 8). *Key concepts in RL*. OpenAI Spinning Up. Retrieved August 30, 2022, from https://spinningup.openai.com/en/latest/spinningup/rl_intro.html

Albelwi, S., & Mahmood, A. (2017). A framework for designing the architectures of deep convolutional neural networks. *Entropy*, *19*(6), 242. https://doi.org/10.3390/e19060242

Anthony, T., Tian, Z., & Barber, D. (2017). Thinking fast and slow with deep learning and tree search. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, *30*, 5366–5376. https://proceedings.neurips.cc/paper/2017/file/d8e1344e27a5b08cdfd5d027d9b8d6de-Paper.pdf

Arjovsky, M., & Bottou, L. (2017). *Towards principled methods for training generative adversarial networks*. arXiv: 1701.04862 [stat.ML]. https://doi.org/10.48550/arXiv.1701.04862

Arjovsky, M., Chintala, S., & Bottou, L. (2017, August). Wasserstein generative adversarial networks. In D. Precup & Y. W. Teh (Eds.), *Proceedings of the 34th international conference on machine learning* (pp. 214–223, Vol. 70). PMLR. http://proceedings.mlr.press/v70/arjovsky17a/arjovsky17a.pdf

Bellemare, M. G., Naddaf, Y., Veness, J., & Bowling, M. (2013). The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, *47*, 253–279. https://doi.org/10.1613/jair.3912

Bellemare, M., Veness, J., & Bowling, M. (2012). Investigating contingency awareness using atari 2600 games. *Proceedings of the AAAI Conference on Artificial Intelligence*, *26*(1), 864–871. https://doi.org/10.1609/aaai.v26i1.8321

Boutilier, C., Dean, T., & Hanks, S. (1999). Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, *11*(1), 1–94. https://doi.org/10.1613/jair.575

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). *OpenAI Gym*. arXiv: 1606.01540 [cs.LG]. https://doi.org/10.48550/arXiv.1606.01540

Chen, Y., Yang, Y., Wu, T., Wang, S., Feng, X., Jiang, J., Lu, Z., McAleer, S. M., Dong, H., & Zhu, S.-C. (2022). Towards human-level bimanual dexterous manipulation with reinforcement learning. *Thirty-sixth Conference on Neural*

Information Processing Systems Datasets and Benchmarks Track. https://openreview.net/forum?id=D29JbExncTP

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., & Koltun, V. (2017). CARLA: An open urban driving simulator. *Proceedings of the 1st Annual Conference on Robot Learning*, 1–16. https://vladlen.info/papers/carla.pdf

François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., Pineau, J., et al. (2018). An introduction to deep reinforcement learning. *Foundations and Trends in Machine Learning*, *11*(3-4), 219–354. https://doi.org/10.1561/2200000071

Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., & Lu, H. (2019). Dual attention network for scene segmentation. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3146–3154. https://openaccess.thecvf.com/content_CVPR_2019/papers/Fu_Dual_Attention_Network_for_Scene_Segmentation_CVPR_2019_paper.pdf

Goertzel, B. (2014). Artificial general intelligence: Concept, state of the art, and future prospects. *Journal of Artificial General Intelligence*, *5*(1), 1–48. https://doi.org/10.2478/jagi-2014-0001

Goodfellow, I., Bengio, Y., & Courville, A. (2016, November 18). *Deep learning*. MIT Press. https://mitpress.mit.edu/9780262035613/deep-learning

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial networks. https://doi.org/10.48550/arXiv.1406.2661

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved training of wasserstein gans. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 30). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/892c3b1c6dccd52936e27cbd0ff683d6-Paper.pdf

Ha, D., & Schmidhuber, J. (2018). *World models*. arXiv: 1803.10122 [cs.LG]. https://doi.org/10.48550/arXiv.1803.10122

Herman, J., Francis, J., Ganju, S., Chen, B., Koul, A., Gupta, A., Skabelkin, A., Zhukov, I., Kumskoy, M., & Nyberg, E. (2021). Learn-to-race: A multimodal control environment for autonomous racing. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9793–9802. https://openaccess.thecvf.com/content/ICCV2021/papers/Herman_Learn-To-Race_A_Multimodal_Control_Environment_for_Autonomous_Racing_ICCV_2021_paper.pdf

Hester, T., Vecerik, M., Pietquin, O., Lanctot, M., Schaul, T., Piot, B., Horgan, D., Quan, J., Sendonaris, A., Osband, I., et al. (2018). Deep Q-learning from demonstrations. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(1), 3223–3230. https://doi.org/10.1609/aaai.v32i1.11757

Ho, J., & Ermon, S. (2016). Generative adversarial imitation learning. *Advances in Neural Information Processing Systems*, *29*, 4565–4573. https://proceedings.neurips.cc/paper/2016/file/cc7e2b878868cbae992d1fb743995d8f-Paper.pdf

Ioffe, S., & Szegedy, C. (2015, July). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In F. Bach & D. Blei (Eds.), *Proceedings of the 32nd international conference on machine learning* (pp. 448–456, Vol. 37). PMLR. http://proceedings.mlr.press/v37/ioffe15.pdf

Jiang, Y., Chang, S., & Wang, Z. (2021). Transgan: Two pure transformers can make one strong gan, and that can scale up. *Advances in Neural Information Processing Systems*, *34*, 14745–14758. https://proceedings.neurips.cc/paper/2021/file/7c220a2091c26a7f5e9f1cfb099511e3-Paper.pdf

Kang, B., Jie, Z., & Feng, J. (2018). Policy optimization with demonstrations. *Proceedings of the 35th International Conference on Machine Learning*, *80*, 2469–2478. https://proceedings.mlr.press/v80/kang18a.html

Kastaniotis, D., Ntinou, I., Tsourounis, D., Economou, G., & Fotopoulos, S. (2018). Attention-aware generative adversarial networks (ata-gans). *2018 IEEE 13th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*, 1–5. https://doi.org/10.1109/IVMSPW.2018.8448850

Li, C., Xia, F., Martín-Martín, R., Lingelbach, M., Srivastava, S., Shen, B., Vainio, K. E., Gokmen, C., Dharan, G., Jain, T., Kurenkov, A., Liu, K., Gweon, H., Wu, J., Fei-Fei, L., & Savarese, S. (2022, November). Igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. In A. Faust, D. Hsu, & G. Neumann (Eds.), *Proceedings of the 5th conference on robot learning* (pp. 455–465, Vol. 164). PMLR. https://proceedings.mlr.press/v164/li22b.html

Li, W., Luo, H., Lin, Z., Zhang, C., Lu, Z., & Ye, D. (2023). *A survey on transformers in reinforcement learning*. arXiv: 2301.03044 [cs.LG]. https://doi.org/10.48550/arXiv.2301.03044

Lim, J. H., & Ye, J. C. (2017). *Geometric gan*. arXiv: 1705.02894 [stat.ML]. https://doi.org/10.48550/arXiv.1705.02894