Training Hangman Agent Using Deep Q Learning

Ivan T. Calangan
College of Computing and Information Technologies
National University
Manila, Philippines
calangit@students.national-u.edu.ph

Justin Neo R. Flores
College of Computing and Information Technologies
National University
Manila, Philippines
floresjr1@national-u.edu.ph

Jessa Mae M. Labasan
College of Computing and Information Technologies
National University
Manila, Philippines
labasanjm1@national-u.edu.ph

***Abstract*—In this study, we examine the potential of deep Q-learning to develop an agent that plays the Atari 2600 version of Hangman. Hangman is a word-guessing game that requires the agent to visually perceive and make a series of decisions. Unlike typical Atari games that require quick reflexes and spatial reasoning skills, Hangman is a language-based logic problem given sparse rewards. A Deep Q-Network (DQN) model was developed by us which included reward shaping, convolutional visual processing, and exploration strategies. Our execution, therefore, signals a preliminary capability to devise strategies for letter choice with some recognition of patterns even though training episodes were few and resources were limited. The study pinpoints the main issues of using a standard DQN in language-based scenarios and at the same time, it serves as a reproducible baseline for further progress.**

***Index Terms*—Reinforcement learning, Hangman, Deep Q-Network, Atari 2600, Word-guessing game.**

I. Introduction

Artificial intelligence (AI) has gone a long way in accomplishing the same thing humans do in visual and control-centric games via deep reinforcement learning (DRL) strategies such as Deep Q-Networks (DQN). Still, there is a large gap of unexplored terrain concerning games that combine language-based reasoning with visual comprehension. Through blending perception, decision-making, and somewhat linguistic inference, the Atari 2600 Hangman setting presents a puzzle that crosses disciplines. Here, a letter is either guessed by the agent or given in the concealed words, and it is only when the word is completed that the agent obtains the complete feedback. The existence of an extremely limited information technique causes significant problems for reinforcement learning, e.g., sparse rewards, large state spaces, and complicated perception–action loops.

This research aims to move deep reinforcement learning (DRL) beyond the scope of traditional control environments and into cognitive, language-oriented problem solving. This has a parallel with Hangman as noted in the manuscript in which natural language processing (NLP) applications that must reason under uncertainty can be exemplified by predictive typing or automated decoding. While text-based systems that directly manipulate symbols would be the easiest, here the Hangman agent has to figure out the symbolic meaning of the raw pixels, thus combining vision and sequential reasoning. The main problem the system faces is that it is an environment with sparse rewards where, when given, meaningful feedback is always delayed until success or failure. To make this happen, reward shaping and intrinsic motivation mechanisms are used, which provide intermediate feedback and encourage exploration. The agents receive continuous learning signals from this method, which ultimately leads to a smoother merge of the limited underlying reward structures.

Deep Q-Learning is still at the core of these kinds of environments an idea that it is capable of discerning visual patterns in game frames and linking them to action values. Among others, the improvements in the replay mechanisms and the stability adjustments are the most significant reasons behind the recent refinements to the algorithms, thereby impacting the DQN models' reliability positively. Nevertheless, the systems are still incapable of sufficiently handling situations requiring reasoning rather than reflexive control.

To solve this problem, the Hangman agent adopts hybrid learning methods: reward shaping for handling sparse feedback and curriculum-based progression for gradually increasing task complexity. The curriculum assignment starts off using simple words and moves to more complicated vocabulary, thus allowing the model to get accustomed to new linguistic patterns step-by-step. Moreover, transfer learning concepts are there to give further support to this gradual progression by taking the earlier, simpler tasks as knowledge that can be reused.

In a nutshell, this project seeks to explore the possibilities of RL agents in terms of merging perceptual processing with symbolic reasoning, thus closing the gap between the two. Despite the limited performance so far, this work lays the groundwork for forthcoming models that will combine visual understanding, cognitive reasoning, and adaptive learning under uncertainty.

II. Literature Review

Deep Q-Learning introduced a major advancement in reinforcement learning by integrating Q-learning with convolutional neural networks (CNNs) for visual feature extraction. Such an invention made the agents able to handle the high-dimensional visual inputs, thus, they had a raw pixel data mapping into the latent representations which were used for decision-making. By continuous interactions with different environments, the Deep Q-Network (DQN) model was able to show a successful integration of value estimation and deep learning resulting in the achievement of human-level control of Atari benchmarks [1]. Afterwards, the innovations addressed network stability, replay mechanisms, and estimation bias [3], [4]thus, they refined the original ground. All these enhancements together increased the convergence reliability and sample efficiency which made DQN one of the most stable baselines for reinforcement learning research, thus, it has been widely used ever since.

Different experiments on the optimization of DQN have led to the discovery of noise control and replay prioritization as the most beneficial features. Methods which stabilize gradients and introduce a certain amount of randomness in a controlled manner enable agents to properly explore and exploit even in a changing environment or in one which has a high variance [2], [3]. Advanced replay strategies allow that the most informative experiences become the main contributors to learning, thus, eliminating overfitting to the redundant samples. On the whole, these breakthroughs reveal how performance can be enhanced beyond the standard visual control tasks by means of architectural refinements in DQN. Nevertheless, these methods, though effectively operating in games with fast feedback, are challenged with restrictions in scenarios where progress signals are sparse, for instance, linguistic puzzles like Hangman.

As a result, sparse-reward optimization has become a key issue for reinforcement learning. Reward shaping changes the reward system to give intermediate help that is in line with long-term goals, thus upgrading the learning process in the situations with delayed feedback [5]. Intrinsic motivation goes beyond this idea by implementing internal signals that arouse curiosity and promote exploration [6]. These mechanisms equip agents to look for new or uncertain states actively instead of waiting for external rewards. Such strategies have been credited with great success in complicated areas where delayed feedback makes learning slow, thus, strengthening their importance for the Hangman environment, which is a natural way of restricting reward density. Also, recent publications have delved into curiosity-driven tactics that would lead the adaptive exploration in multi-agent and sparse-reward scenarios [7].

After reward modification, curriculum and transfer learning have been mainly used as key strategies to improve generalization and scalability of reinforcement learning. Curriculum learning gradually introduces tasks, thus the agent can first master simpler goals and then move to more complex ones [12], [14]. This organized method is a reflection of human cognitive growth and it also helps to lessen catastrophic forgetting when moving through different training stages. Moreover, transfer learning is able to complement this by reusing the pre-trained features, thus the training time can be shortened significantly and the adaptation to new word sets or difficulty levels can be easily done [13]. The two techniques combined together create a basis for deep policy learning across different tasks which is the method used in the staged vocabulary design of the Hangman agent.

Most of the reinforcement learning (RL) breakthroughs have been focused on perception and motor control. However, there is a considerable amount of interest in the application of RL to language-driven and text-based environments. Such exercises are dependent on the system not only to perceive but also to reason, recall sequentially, and perform symbolic inference. Xu et al. [9] showed the way hierarchical attention mechanisms could empower agents to juggle contextual reasoning and decision-making in text-based games. Their results point to the fact that the use of linguistic structure in reinforcement learning architectures helps to a greater extent transparency and quality of decisions. These techniques are similar to the cognitive reasoning demands of Hangman, in which agents have to derive the symbolic significance of the visuals and at the same time retain the temporal aspect of the previous guesses.

This expansion of robust Atari learning environment research also delivers better benchmarks for generalization and continual learning [10], [11]. Consequently, these innovations keep Atari as an instrument still relevant for exploring the perceptual and cognitive aspects of agent behavior. Future work in this area, coupled with the introduction of language-reasoning tasks such as Hangman (a domain), will allow RL studies to consider the interplay of visual and symbolic understanding under the condition of uncertainty.

Generally, the different pieces of research over time show a distinct development: reinforcement learning has moved from controlling low-level sensory inputs to doing tasks that require abstraction, planning, and using language for reasoning. The papers published on DQN optimization, sparse-reward engineering, and curriculum progression have a combined effect that they very clearly point to our study. This study, therefore, represents a step forward in figuring out how agents can seamlessly integrate perception, reward adaptation, and cognitive inference in mixed worlds that are a combination of visual and linguistic challenges by combining these different methods into one framework.

A. Conceptual Framework

The conceptual framework of this study involves the integration of three main elements: (1) **visual state processing**, (2) **decision-making via Deep Q-Learning**, and (3) **feedback optimization through reward shaping**. The agent gets the Atari Hangman environment by receiving the game states in pixels, and through joystick-based actions, the agent which stands for letter guesses, picks the letter of the alphabet.

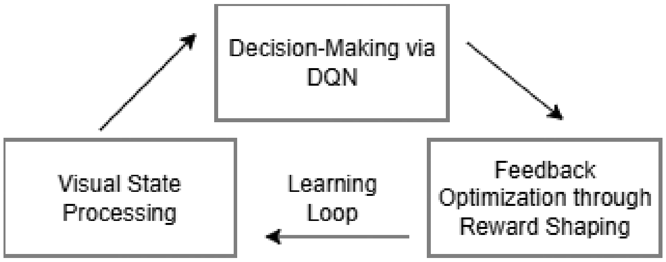

Fig. 1. Conceptual Framework of the Study: Integration of visual state processing, decision-making via Deep Q-Learning, and feedback optimization through reward shaping in a learning loop.

The framework focuses on recursive learning through the Q-learning update rule:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\big[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\big] \quad (1)$$

At first, the process is doing a random exploration and then, it slowly exploits the learned policies. The convolutional layers in DQN identify the visual features such as word blanks and the locations of letters, while the fully connected layers perform the action-value calculations that show the letter chosen. The reward shaping component is there to continue the feedback loop—hence, the system is helped in going the right way by giving it a correct guess and it is punished for an incorrect one—to be able to go beyond the sparsity that is typical of the environment [7].

Concept-wise, this setup is compatible with the principles of curriculum learning according to which the training starts with simple vocabulary levels (Grades 1–3) and later progresses to complex words (High School difficulty). The design is aimed at enhancing generalization and knowledge transfer between the different tasks [8]. The entire process is a depiction of the very cycle that the human mind follows, i.e., perception, evaluation, and decision-making under uncertainty.

III. Methodology

A. Overview

The methodology of this research is a systematic process that leads to the training and evaluation of a Deep Q-Learning (DQN) agent playing Atari Hangman. The detailed procedure has five main stages: (1) environment definition and visual input preprocessing, (2) DQN architecture design, (3) custom reward-shaping integration to solve the sparse reward problem, (4) agent training with an ϵ-greedy exploration policy, and (5) performance evaluation through win rate, convergence behavior, and average episode reward. The conceptual framework outlines the steps starting from pixel-based perception to policy optimization. The environment takes pixel frames as input, which are converted into compact representations by convolutional layers. The Q-network calculates the action-value estimates to select the optimal joystick-based letters, while the feedback from the environment—reward shaping included—learning updates guide the iterative process. Perception, decision-making, and feedback thus form a continuous closed-loop learning cycle through this design.

B. Environment and Problem Formulation

1) Environment: The experiment used a game setting which was the Hangman Atari 2600 environment (ALE/Hangman-v5) made available by the Arcade Learning Environment (ALE) through the Gymnasium library [14]. The environment imitates the traditional word-guessing game in which an agent is figuring out the concealed words by picking letters and at the same time, trying to keep the number of wrong guesses low. In the Atari version, the game is visually presented through screen pixels instead of using symbols, which means that the agent has to learn visual feature extraction and strategic decision-making at the same time.

2) State and Action Spaces: The RGB raw image frames taken from the Atari display are of size $210 \times 160 \times 3$ (height $\times$ width $\times$ color channels). First of all, each frame is changed to grayscale, resized to 84×84 pixels, and normalized to the range $[0, 1]$. In order to have the temporal context, they keep a frame stack of the last four frames, so the input tensor has the shape $(4, 84, 84)$. The action space is made up of 18 discrete joystick actions which are the internal mappings of Hangman letter selections and control inputs. A built-in stochastic action repeat of 25% in ALE is the source of the variability in the state transitions.

3) Reward Function: The reward scaling method changes the player's reward coming from the Hangman game along with a set of previously guessed actions from the player and then returns the new scaled reward. This is basically the scaling of the reward to give the agent a feedback which is very informative and training will be facilitated, hence desirable behaviors being encouraged and undesirable ones discouraged.

Positive Reward for Correct Guesses

When the reward is more than zero, the scaled reward becomes 10.0 + reward. This means that the environment provides a basic reward for a correct guess, together with the actual reward.

There is an additional reward of 5.0 if the number of previously guessed letters is less than or equal to 3. This, therefore, motivates the agent to make the right guess at the beginning of the game.

Zero Reward for Wrong Guesses

In the case of a reward being zero, the scaled reward is -1.0, thus it is giving a small penalty for an incorrect guess.

Negative Reward for Game Over

In the case where the reward is negative due to many incorrect guesses, the scaled reward will be -5.0; hence this is assigning a larger penalty for losing the game.

This reward is the one to be used in the Q-Learning Update that is to direct the agent's learning process.

4) Terminal Condition: An episode terminates when: (1) the agent manages to reveal all letters (win), or (2) the maximum number of incorrect guesses is exceeded (loss). Gambling also allows for truncation conditions for maximum episode length, which can be combined with termination through the Boolean expression `done = terminated or truncated`. After termination, the environment changes to a new word and state by reset.

5) Task Type: The task is episodic. In each episode, a new hidden word is given, and the episode terminates upon either a win or a loss. Episode lengths depend on the difficulty of the vocabulary and the strategy and are usually between 140 and 200 steps. The episodic format here makes it possible to easily evaluate performance via win rate and average reward metrics.

C. Algorithm Description

1) Algorithm Used: A Deep Q-Network (DQN) algorithm [11] was employed using a convolutional neural network (ConvDQN) to approximate the Q-function from visual inputs. The method is model-free, value-based, and off-policy, following the original DQN framework.

2) Q-Learning Update Rule: The learning process is governed by the Bellman optimality equation:

$$Q^*(s,a) = \mathbb{E}\big[r + \gamma \max_{a'} Q^*(s',a') \mid s,a\big] \tag{2}$$

where r is the reward, γ the discount factor, and (s',a') the next state-action pair. The network minimizes the mean squared Bellman error (MSBE):

$$L(\theta) = \mathbb{E}\big[(y - Q(s,a;\theta))^2\big] \tag{3}$$

with target values computed via a separate target network:

$$y = r + \gamma \max_{a'} Q(s',a';\theta^-) \tag{4}$$

Parameters θ^- are periodically synchronized with θ to maintain stability.

3) Hyperparameters: The following hyperparameters were used:

- Learning rate (α): 0.00025;
- Discount factor (γ): 0.99;
- Batch size: 32;
- Replay buffer size: 10,000;
- Target network update: every 10 episodes;
- Training episodes: 2000;
- Epsilon decay: 0.995 per episode, minimum $\epsilon = 0.1$.

4) Exploration Strategy: An ϵ-greedy policy was applied to balance exploration and exploitation. Initially, $\epsilon = 1.0$, decaying by 0.995 per episode until reaching 0.1, ensuring persistent exploration and preventing premature convergence.

5) Algorithm Modifications: To improve training stability:

- Replay buffer pre-filling with 1,000 random experiences to diversify samples;
- Gradient clipping to a maximum norm of 1.0;
- Target network synchronization every 10 episodes;
- Smooth L1 (Huber) loss to mitigate large TD errors;
- Reward shaping integration to address sparse feedback.

D. Implementation Details

1) Frameworks and Tools: The implementation used `PyTorch` for deep learning, `Gymnasium` with `ale-py` for environment management, `OpenCV` for image preprocessing, and `NumPy` and `Matplotlib` for data analysis and visualization.

2) Training Setup: Training was conducted for 2,000 episodes using mini-batch gradient descent. The Adam optimizer [6] with the learning rate of 0.00025 and $\epsilon = 1 \times 10^{-8}$ was used. After every 2000 steps, the learning rate was decayed by 0.9 by the scheduler. Every 10 episodes, the target-network was updated. Smooth L1 for loss was used and gradient clipping was at 1.0 so that the training would not become unstable. Each episode was cut off once a win or loss was achieved. Average episode reward, total steps, and win rate were used to keep track of the performance.

3) Hardware Specifications: The experiments were done on Google Colab with an NVIDIA Tesla T4 GPU (16 GB VRAM). Training for 2,000 episodes required approximately 1.5 hours, with GPU acceleration significantly reducing computation time compared to CPU-only training.

IV. Discussion

A. Performance Metrics

The performance of the model was gauged by means of average reward, win rate, total steps per episode, and cumulative reward over training intervals. Such instruments of measurement served as a window into the learning and adapting capabilities of the agent as well as the latter's eventual mastering of the game. Win rate was considered the main criterion to success, whereas average reward was used to measure stability and convergence of the activity.

Generally, a steady increase in these figures would be an indication of successful policy learning, although, at times, there were some ups and downs during training.

B. Model Performance and Observations

The DQN agent was only partly successful in the Atari Hangman environment. Average rewards stayed at a low level, and the model was unable to make a significant increase of its performance in a consistent manner throughout the episodes. In fact, very few positive changes in reward could be spotted during the initial training, while later episodes exhibited mixing of returns, often resulting in negative rewards. Such a situation points out that with the current setup and reward shaping, the agent still hasn't achieved stable convergence. Therefore, the low win rate of the agent throughout the experiments supports the conclusion that it was hard for the agent to learn a strategy of sensible letter guessing from pixels.

C. Learning Curve Analysis

Training videos depicted large fluctuations in the episodic reward values. Rewards often went below zero, which represented failing episodes and low accuracy in letter selection. It was even hinted that after the loss function converged, the

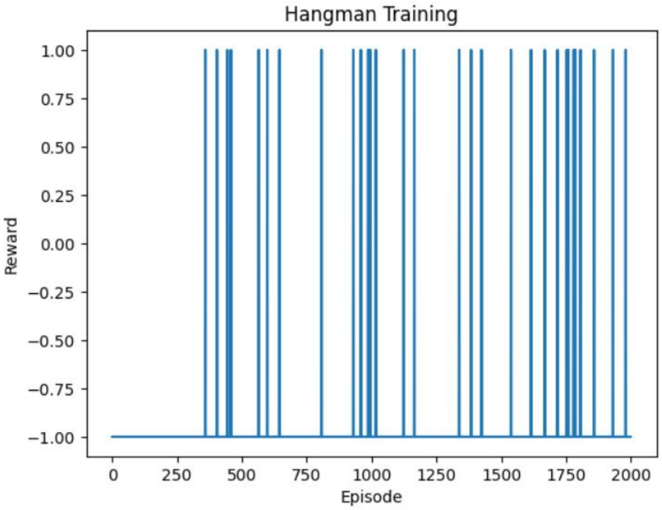

Fig. 2. Model Performance

policy did not become stable, thus implying that loss minimization was not necessarily tightly coupled with significant behavioral upgrade. Such an issue is in line with the known susceptibility of DQNs to sparse-reward and partially observable scenarios [7]. The findings highlight that enhancements in network stability do not necessarily lead to successful policy learning when there is very little intrinsic feedback.

D. Behavioral Interpretation

The letter choice of the agent's coding did not significantly reflect human strategies, for example, the humans usually prioritize vowels or most frequently occurring consonants. As there were no clear linguistic priors given, the network had to base its decisions mostly on the visual correlations between what changed on the screen and the rewards. Because of this reliance on visuals, the agent was not able to find a consistent pattern since it had to guess the word structures from the pixel changes but it was not successful. The random action repeat (stochastic 25% in ALE by default) thus was one of the reasons for the agent's behavior being not consistent that is to say, there was some variability in the timing of the actions because of it.

E. Training Stability and Sensitivity

The experiment uncovered that the system was very sensitive to hyperparameters such as the learning rate, reward scaling, and ϵ-decay schedule. Very small changes in these parameters had a drastic impact on the performance, with an overly aggressive decay leading to premature exploitation and a slower decay causing exploration to be prolonged without convergence. Reward shaping resulted in a slight improvement and did not completely solve the delayed-feedback problem. Next experiments might opt to implement more sophisticated solutions like prioritized replay buffers [14], dueling architectures, or hybrid intrinsic reward systems [9] to both stabilize and accelerate learning.

F. Environment Complexity

The Atari Hangman environment leads to a set of problems that are different from those of normal visual control tasks. Hangman, a non-movement-based Atari game, needs cognitive reasoning, pattern inference, and sequential memory. Because it is necessary to keep track of the previous guesses and to interpret the symbols in the display coming from the pixel data, this task is more comparable to language reasoning ones. The intricacy of the task points to the limitation of standard DQN structures that were developed for quick decision-making of a reflexive nature and not for semantic inference. The research results show that it may be necessary to integrate memory-based or attention mechanisms in order to capture the temporal and symbolic dependencies that are natural for this environment.

V. Conclusion

A. Key Findings

When Deep Q-Learning was used on the Atari Hangman game, it exposed a significant difficulty in mixing visual understanding with language-based reasoning. After trying various methods such as reward shaping and stability increasing techniques, the agent still could not find a way to make its policy improvement more of a practice than a theory. The very sparse reward system and a state space that was visually encoded made it very difficult for the model to figure out the right thing to do in the long run because it had to link its actions with very distant outcomes. It turned out that 2,000 episodes of training were not enough for the model to really converge, so the need for longer training times and more complex reward schemes was affirmed.

B. Contributions

By conducting one of the few empirical investigations of language-based puzzle environments in the Atari Learning Environment framework, this research adds to the reinforcement learning research area. It sheds light on how DQN architectures can be changed to deal with tasks that mix perception and reasoning. Moreover, the use of a shaped reward system and the thorough performance analysis provide a base for studies that aim to address sparse-reward reinforcement learning and cross-domain generalization.

C. Limitations and Future Work

Gradually, work needs to be extended to training for 10,000-20,000 episodes and also advanced DQN variants such as Rainbow DQN [6] should be incorporated which stabilizes the network by combining prioritized replay, double learning and noisy layers. Curriculum learning methods [8] might be used to improve progressive learning by initially taking simpler sets of words and then moving to more complex vocabularies. Hybrid models that integrate visual CNNs with language priors or frequency-based heuristics may help to close the gap between perceptual and symbolic understanding. Also, transfer learning methods provide a way to speed up the training process by sharing the knowledge from the related tasks.

Moreover, symbolic preprocessing - for instance, extracting the representation of letters from pixels before learning - can make the environment simpler, thus the model can concentrate on reasoning instead of decoding raw visuals.

D. Summary

Essentially, this study highlights the boundaries of just using the conventional DQN setup when dealing with scenarios that require not only perception but also cognition. The results show only a tiny bit of success in a quantitative way; nevertheless, the findings provide substantial qualitative indications of the complexities of training RL agents for pipelining linguistic inference. As a result, future work incorporating these three elements - perceptual input from vision, reward shaping, and learning curriculum - will have the capacity to address the problem of how to link perceptual mechanisms with abstract cognitive processes step by step.

References

[1] A. P. Badia, A. Piot, S. Kapturowski, P. Sprechmann, A. Vitvitskyi, Z. Dabney, A. Paduraru, D. Budden, D. Horgan, and C. Blundell, "Agent57: Outperforming the Atari Human Benchmark," *Proc. Int. Conf. Mach. Learn. (ICML)*, PMLR 119, pp. 507–517, 2020.

[2] T. Clark, M. Towers, C. Evers, and J. Hare, "Beyond the Rainbow: High Performance Deep Reinforcement Learning on a Desktop PC," *arXiv preprint arXiv:2411.03820*, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2411.03820

[3] W. Wang, J. Huang, X. Zhang, Z. Liu, and J. Yuan, "NROWAN-DQN: A Stable Noisy Network with Noise Reduction and Online Weight Adjustment for Exploration," *Expert Syst. Appl.*, vol. 203, p. 117343, 2022, doi:10.1016/j.eswa.2022.117343.

[4] P. M. Panahi, A. Patterson, M. White, and A. White, "Investigating the Interplay of Prioritized Replay and Generalization," *arXiv preprint arXiv:2407.09702*, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2407.09702

[5] S. Ibrahim, T. Haque, R. Singh, and F. Anwar, "Comprehensive Overview of Reward Engineering and Shaping in Advancing Reinforcement Learning Applications," *IEEE Access*, 2024, doi:10.1109/ACCESS.2024.0429000.

[6] A. Kayal, E. Pignatelli, and L. Toni, "The Impact of Intrinsic Rewards on Exploration in Reinforcement Learning," *Neural Comput. Appl.*, vol. 37, pp. 16269–16303, 2025, doi:10.1007/s00521-025-11340-0.

[7] J. Li and P. Gajane, "Curiosity-driven Exploration in Sparse-reward Multi-agent Reinforcement Learning," *arXiv preprint arXiv:2302.10825*, 2023, doi:10.48550/arXiv.2302.10825.

[8] Y. Wu, Z. Zhang, H. Liu, J. Yuan, and X. Tang, "Read and Reap the Rewards: Learning to Play Atari with the Help of Instruction Manuals," *arXiv preprint arXiv:2302.04449*, 2023, doi:10.48550/arXiv.2302.04449.

[9] Y. Xu, M. Feng, R. Yan, and X. Zhao, "Deep Reinforcement Learning with Stacked Hierarchical Attention for Text-based Games," *arXiv preprint arXiv:2010.11655*, 2020, doi:10.48550/arXiv.2010.11655.

[10] Q. Delfosse, L. Esch, D. Zhang, A. Torrado, and M. Bellemare, "HackAtari: Atari Learning Environments for Robust and Continual Reinforcement Learning," *arXiv preprint arXiv:2406.03997*, 2024.

[11] J. Fan, "A Review for Deep Reinforcement Learning in Atari: Benchmarks, Challenges, and Solutions," *arXiv preprint arXiv:2112.04145*, 2021.

[12] P. Huang, C. Fan, M. Zhao, and K. Tang, "Curriculum Reinforcement Learning Using Optimal Transport via Gradual Domain Adaptation," *Proc. Neural Information Processing Systems (NeurIPS)*, 2022.

[13] Z. Zhu, K. Lin, A. K. Jain, and J. Zhou, "Transfer Learning in Deep Reinforcement Learning: A Survey," *arXiv preprint arXiv:2009.07888*, 2020, doi:10.48550/arXiv.2009.07888.

[14] P. Soviany, R. T. Ionescu, T. Rasche, and M. Leordeanu, "Curriculum Learning: A Survey," *Int. J. Comput. Vis.*, 2022. [Online]. Available: https://arxiv.org/abs/2101.10382