Ravit Amos – 205505951
Kfir Sofer – 212073696

Solving TopSpin problem using DRL

The GitHub repository with the code - https://github.com/kfirs127/TopSpinDRL.git

One of RL's main problems—sparse rewards—makes the TopSpin game(described in assignment 1) challenging for RL.
Reaching the goal state is the only way to receive rewards. It becomes challenging for the model to accomplish a reward when there is a significant distance between the initial and goal states, this requires the model to execute multiple consecutive correct steps.

To address the sparse reward issue, we experimented with four different reward types and two model types to determine the most effective approach for winning the game.

Reward Types:

1. **Basic Step Penalty**: A reward of -1 for each step taken until the state goal is reached, encouraging the model to find the shortest path.
2. **Euclidean Distance**: Rewards are based on the Euclidean distance from the current state to the goal state, and again, the rewards are negative to encourage the model.
3. **BWA***: This approach uses a BWA* path length, with rewards being the path length's negative.
4. **Heuristic-Based Approach**: We trained a heuristic model before the RL training and used its output as the reward function to provide more domain-specific customized recommendations.

These reward strategies aim to progressively introduce more knowledge to the model, from zero knowledge (basic step penalty) to more detailed information about the problem (heuristic-based approach).

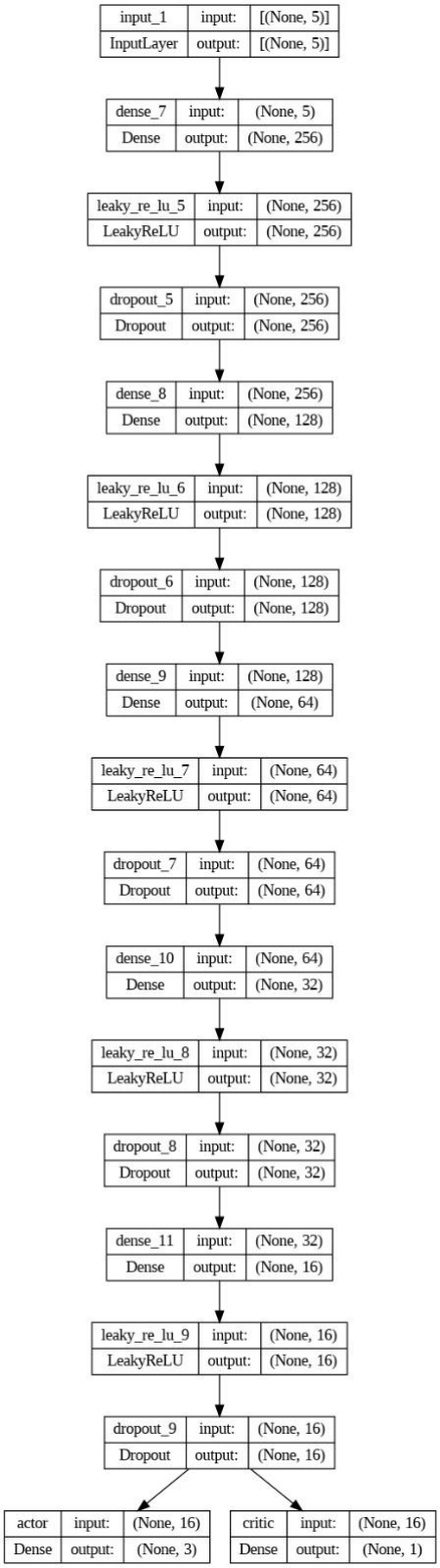
Model Types:

We tested two different models: **Actor-Critic** and **Dueling Actor-Critic**.

1. **Actor-Critic**: The Critic learns to evaluate how close states are to the goal, which ideally helps the Actor to make better decisions.
2. **Dueling Actor-Critic**: Similar to the Dueling-DQN approach, this model separates the estimation of state value and advantage, allowing the agent to learn which actions are valuable, irrespective of the state.

Ravit Amos – 205505951
Kfir Sofer – 212073696

Both models were implemented using the same architecture, differing only in the heads of the networks, as well as the Duling Actor-Critic, which used the same architecture and then separated the heads. The architecture used for these is:



Ravit Amos – 205505951
Kfir Sofer – 212073696

Training Process

During the training process, it was seen that both models showed a quick tendency to overfit. This overfitting is displayed through converging multiple weights to zero, particularly when utilizing the standard ReLU activation function. To address this, we employed the Leaky ReLU activation function, which provides a non-zero gradient for negative inputs. To mitigate the effects of overfitting, the training was limited to 1000 episodes, each comprising 100 steps. This limitation was intended to balance good learning and prevent the models from overly memorizing the training data.

Experiment Setup:

- N = [5, 7]: These values define the size of the TopSpin game.
- K = 3: This parameter represents the size of the rotating.
- Reward Types = [0, 1, 2, 3]: The four distinct reward strategies implemented, each providing varying levels of feedback to the models, as shown above.
- Model Types = [0, 1]: 0 denotes the Actor-Critic model, and 1 denotes the Dueling Actor-Critic model.
- Distance = [6, 8, 10]: These values specify the number of maximal moves required to transition from any initial state to the goal state in the training, thereby defining the problem's difficulty level.

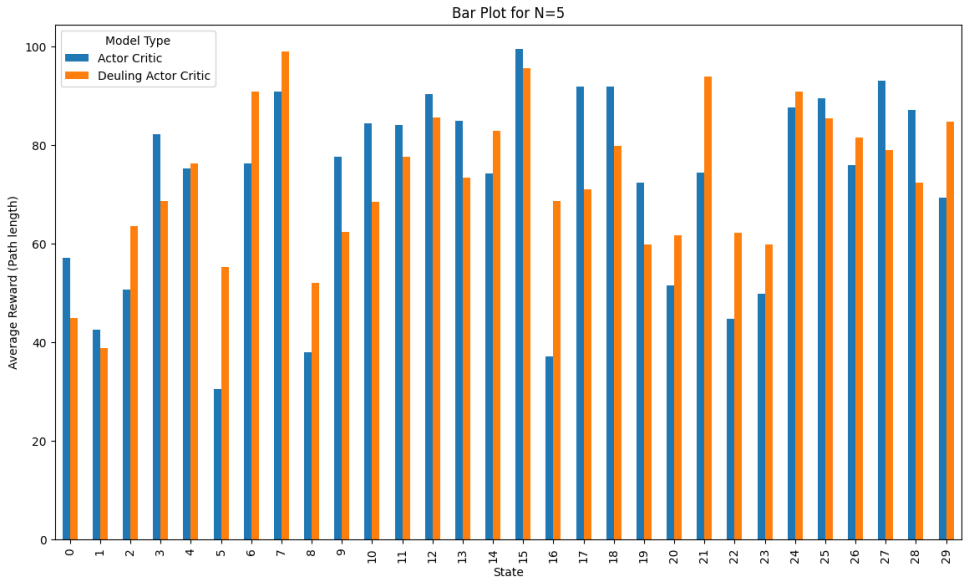
Evaluation

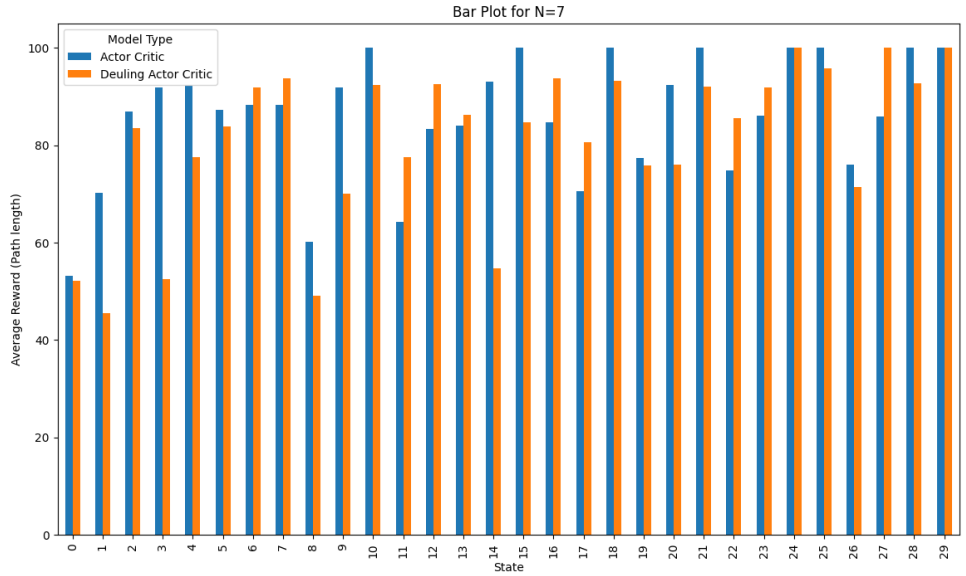
This project evaluates different reward types and model types combinations in the TopSpin game. The goal was to minimize the average steps from the initial state to the goal state. The evaluation includes both model-type comparisons (Actor-Critic and Dueling Actor-Critic) and reward-type comparisons (Trivial, Euclidean, Heuristic, and BWAS) across different state complexities (N=5 and N=7).
To make the evaluation as even as possible, we generated 30 states with different difficulties(e.g., different max distances from the goal state), where the bigger the state index, the more difficult it is.

Ravit Amos – 205505951
Kfir Sofer – 212073696

Model Types Performance

Performance was assessed across two model types: Actor-Critic and Dueling Actor-

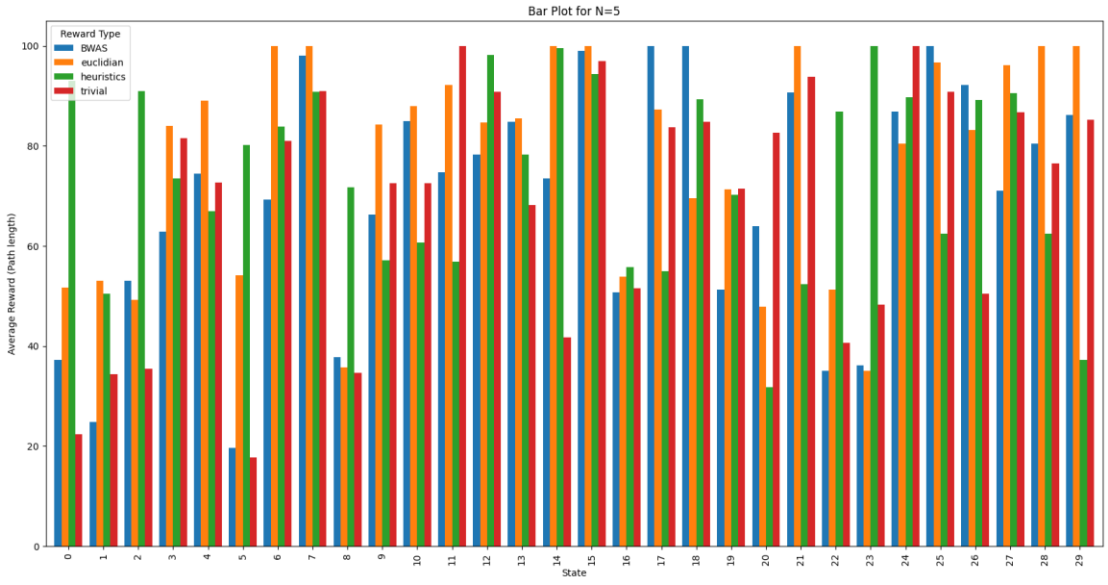


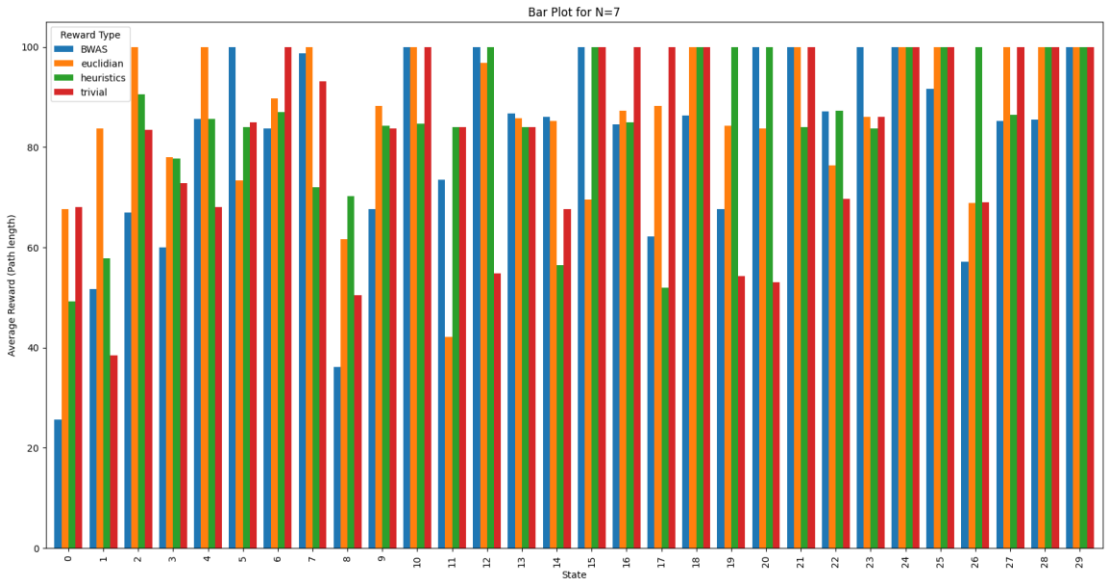


Critic. The following graphs illustrate their average path lengths to the goal, with lower values denoting superior performance.
The Dueling Actor-Critic model outperformed the Actor-Critic model, achieving lower average rewards. The Dueling Actor-Critic model successfully navigated through N=5 and N=7. This is attributed to its architecture, which separately estimates the state value and the advantage of specific actions, providing a more refined approach to decision-making.

Ravit Amos – 205505951
Kfir Sofer – 212073696

Reward Types Performance

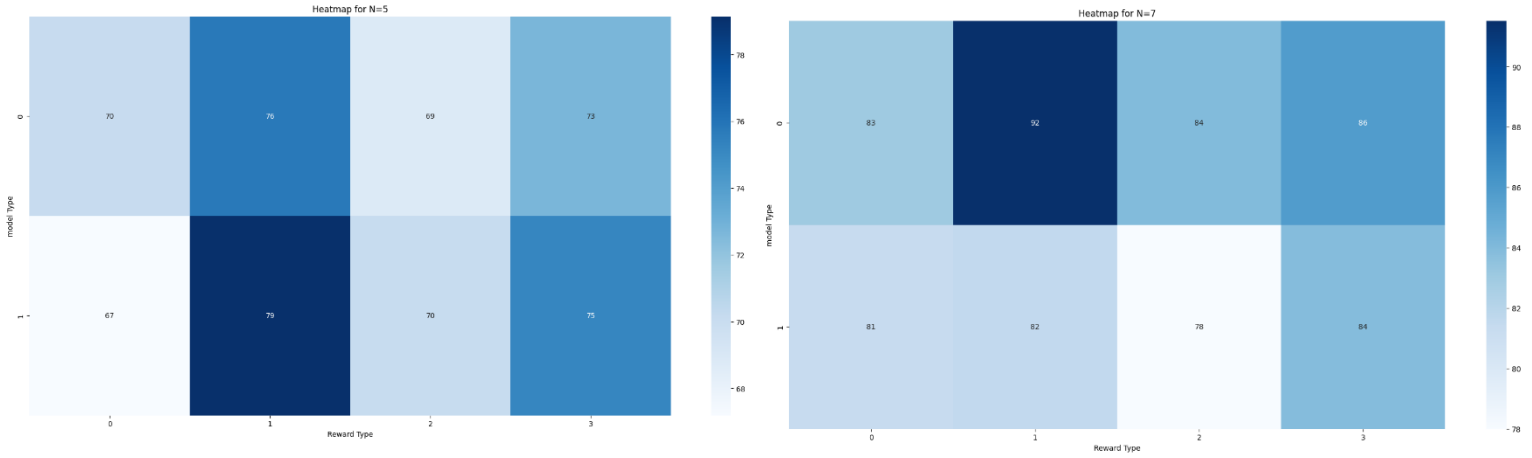




- RewardType 1 (Euclidean): The most effective reward type consistently resulted in the lowest average rewards. Its distance-based feedback mechanism effectively guided the models.
- RewardType 2 (Heuristic): Also performed well, though slightly less effective than RewardType 1. It provided a good balance of exploration and exploitation through heuristic guidance.
- RewardType 3 (BWAS): This reward type underperformed despite its potential. The BWAS method could have provided more effective guidance than expected because we likely chose a big value of W, and the distances of the generated states were too close. Using A* instead of BWAS or increasing the distances of the generated states could have improved its effectiveness.
- RewardType 0 (Trivial): As anticipated, this reward type performed the worst, lacking sophisticated mechanisms to guide the model effectively.

Ravit Amos – 205505951
Kfir Sofer – 212073696

Summary:



The results emphasize the consistent superiority of the Dueling Actor-Critic model, especially when combined with RewardType 1 (Euclidean) or RewardType 2 (Heuristic).

Key Insights:

1. Optimal Configurations: Combining the Dueling Actor-Critic model with RewardType 1 (Euclidean) or RewardType 2 (Heuristic) consistently achieved the best results, indicating the most efficient and effective learning.
2. Impact of Reward Structures: The choice of reward structure plays a critical role in model performance. The Euclidean-based RewardType 1 provided clear and practical guidance, making it the most beneficial reward type in this study.
3. Challenges with BWAS (RewardType 3): The expected benefits of BWAS were not realized in this context. This highlights the need to explore the implementation details further and the potential use of alternative algorithms like A*.

Future Work

If we had more time, we would have explored more model architectures. In particular, the PPO architecture is recognized for its versatility and robustness across diverse reinforcement learning challenges.
We also intended to use the potential of techniques like transfer learning and curriculum learning, as these approaches could improve the efficiency of the learning process, we would have tested larger N values and compared the results to models trained from scratch to ones that use more trans learning techniques on what we already learned, aiming to demonstrate that transfer learning can offer a more efficient and effective learning process.
Also as mentioned, we would have explore more the possibility to lower W in BWAS and using A* instead of BWAS