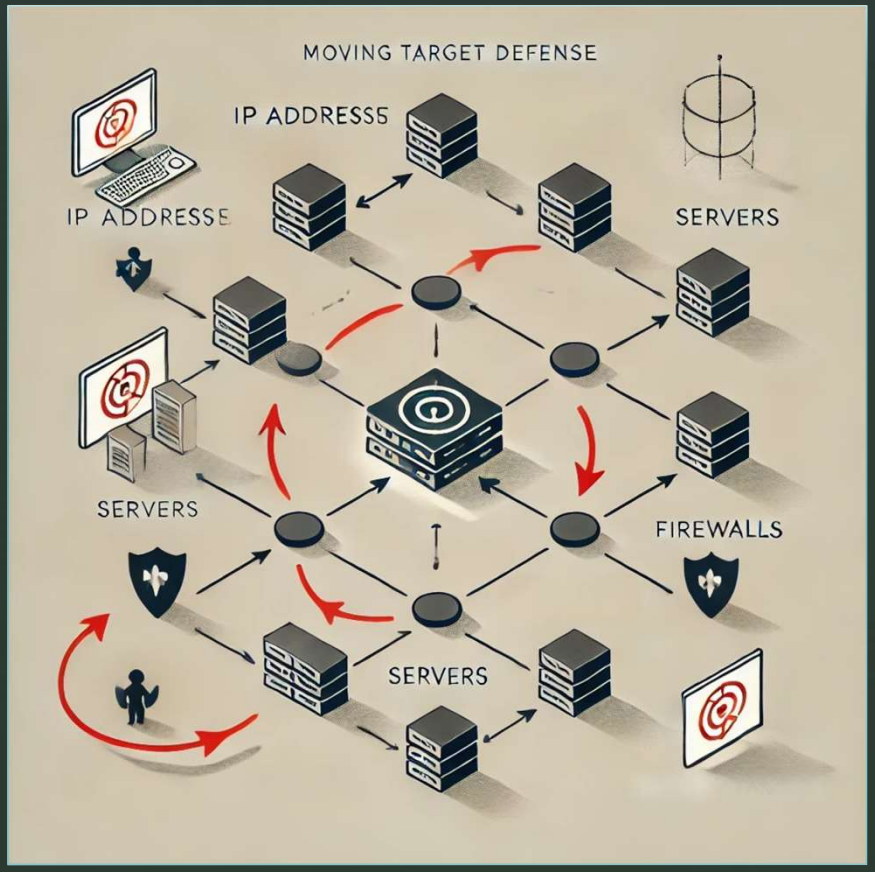
Using Artificial Intelligence to Automate Deployment of MTD operations

Joo Kai Tay (22489437)

Supervisor: Dr Jin Hong

Introduction

- Moving Target Defence (MTD)
  - A dynamic cybersecurity strategy that aims to protect computer systems, networks and data by constantly changing their attack surface.
  - Traditional MTD are time-based or event-based.



Project Background

- Current work in the field focuses on optimizing a specific MTD technique or to combat a specific type of attack.

Attacker Focused

Employing predictive MTD to impede adversaries ability to scope out the network [1]

Strategic selection defence against adversarial ML attacks [2]

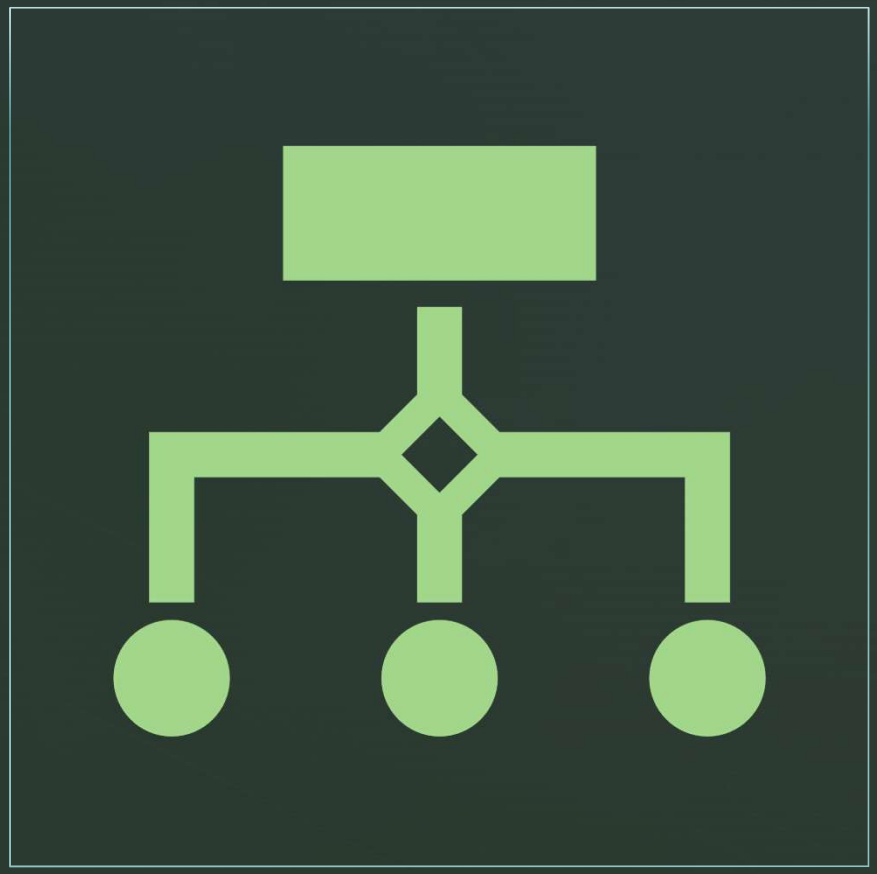
Technique Focused

MTD technique that randomizes HTML elements for web services. [3]

MTF-based approach for embedded deep visual sensing systems to detect and mitigate adversarial examples through deep learning models [4]

MTD Techniques

- **Shuffle based MTD**: Randomises or rearranges system components
- **Diversity based MTD**: Deploys system components with different configurations
- **Redundancy based MTD**: Deploys multiple replicas of the system components



Project Motivation

- **Single MTD** techniques are no sufficient to thwart all attacker types.
- Deploying **multiple MTD**s simultaneously is **costly** for system resources.
- Limited research has been done on **dynamic selection** of MTD techniques in response to different attacker types.

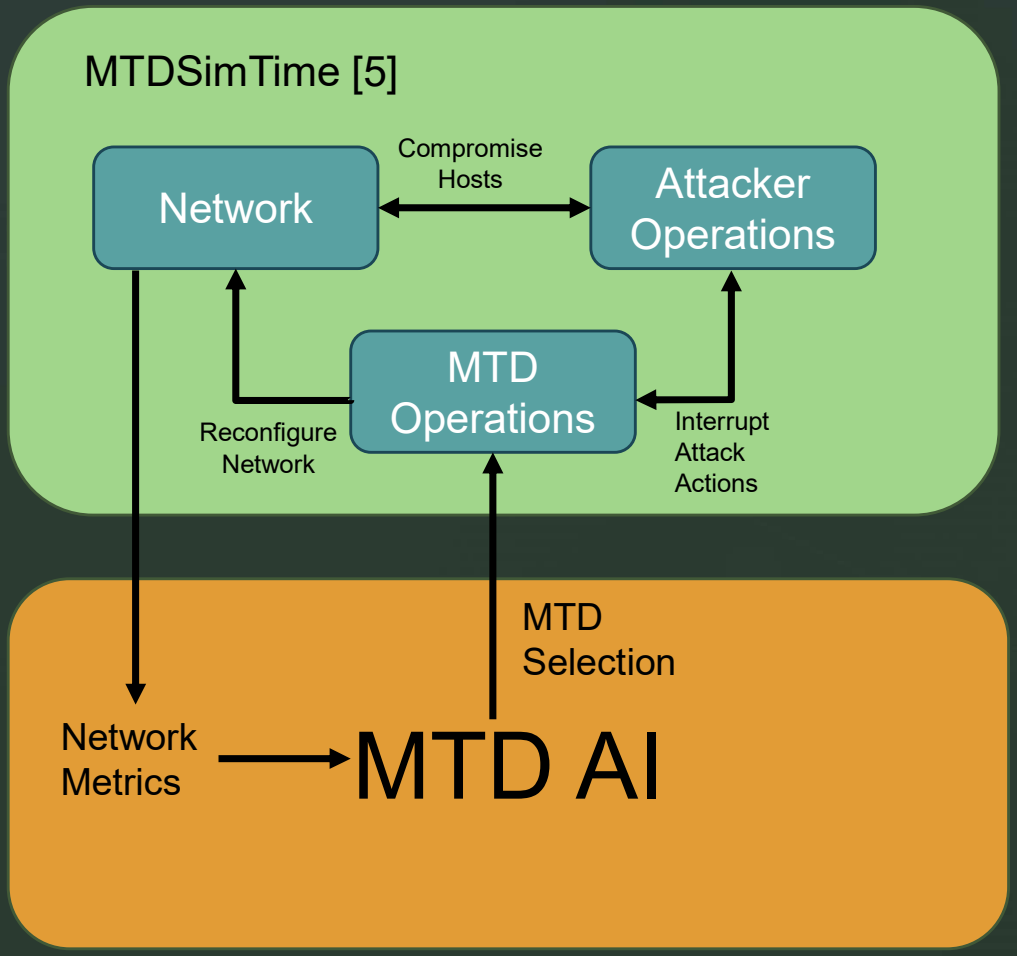
Research Questions

- What are the most important **features** for a machine learning (ML) model designed to execute MTD techniques?
- What is the most suitable **architecture** for a ML model designed to execute MTD techniques?
- What **factors** affect the effectiveness of the ML model when deployed in the network?

Model Architecture

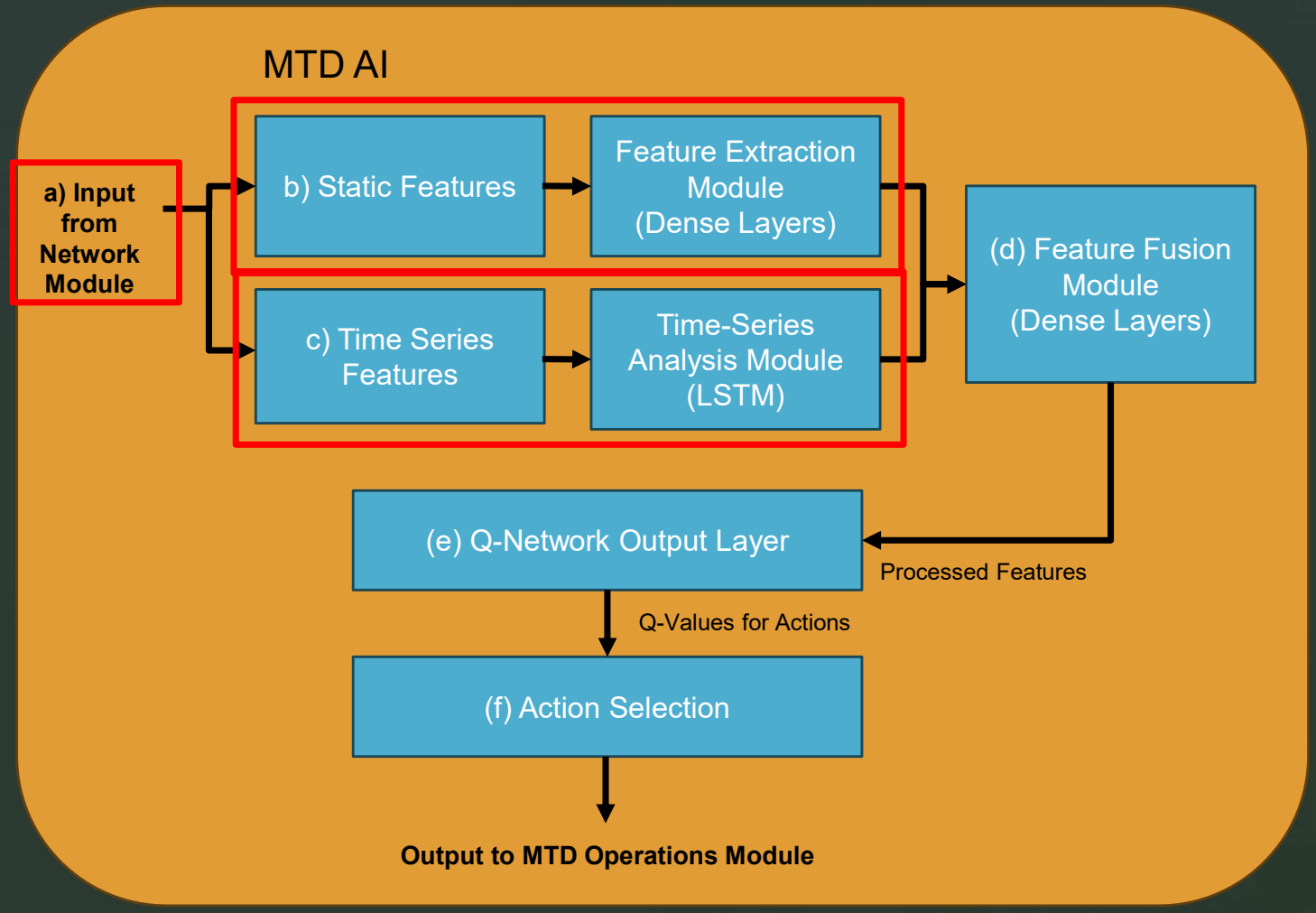
What is the most suitable **architecture** for a ML model designed to execute MTD techniques?

System Overview



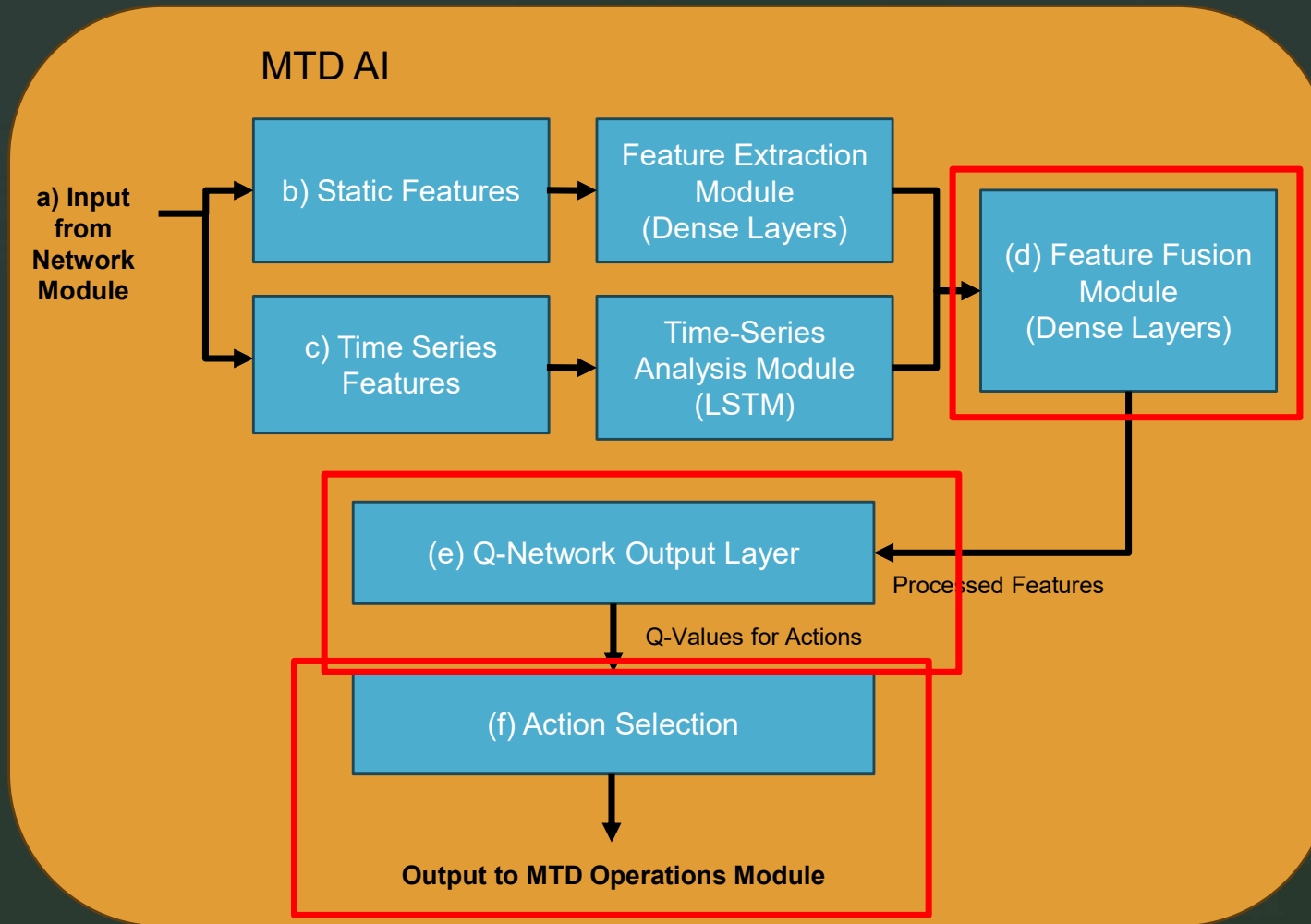
- MTD Operations module deploys MTD into the network on fixed intervals
- The reinforcement learning (RL) model receives a stream of data from the network and configures the deployment of MTD techniques in real-time.

Model Architecture



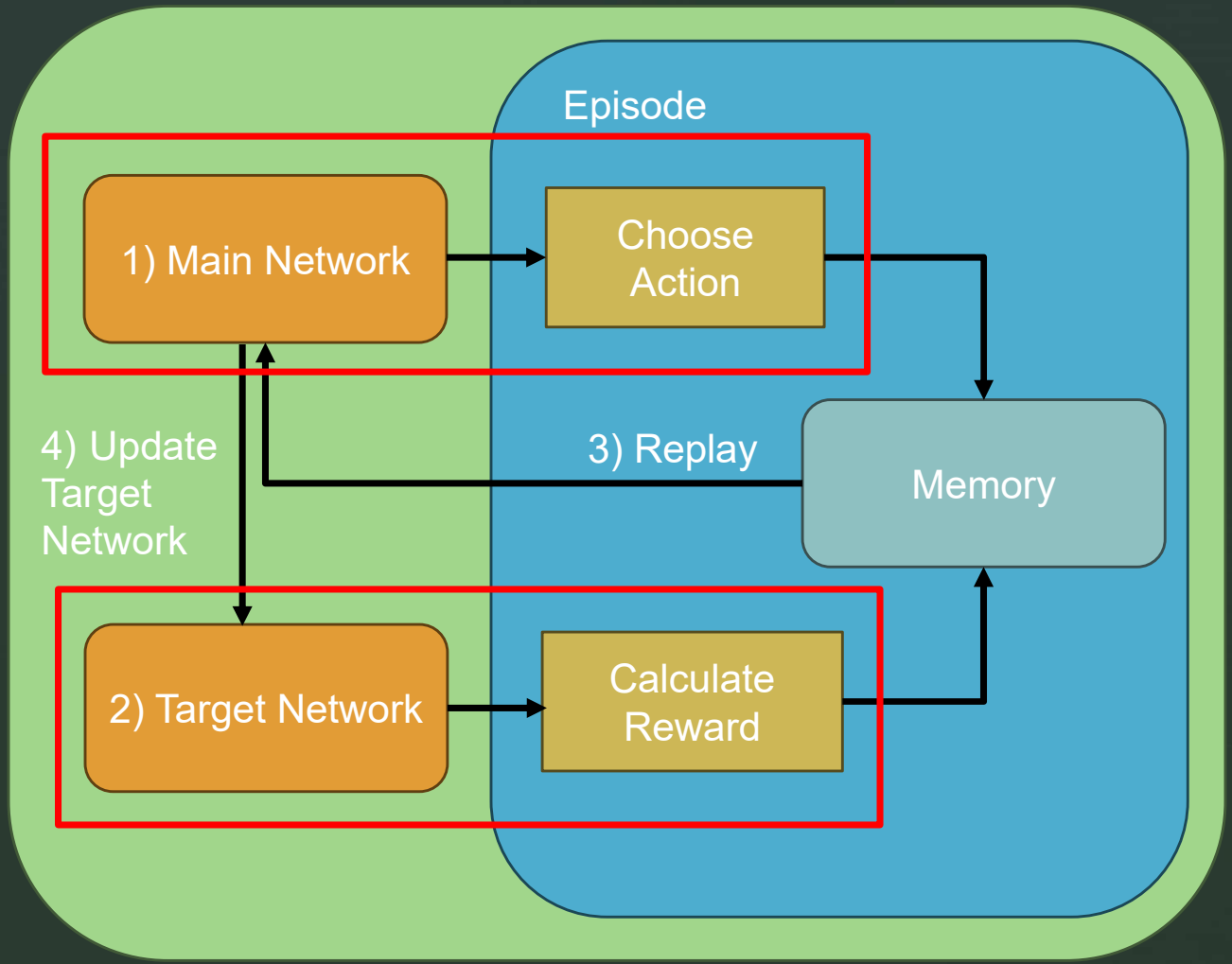
- Static features such as Attack Path Exposure, Host Compromise Ratio and Exposed Vulnerabilities.
- Feature extraction module: Dense Layers -> ReLu -> Dropout
- Temporal features: MTCC, Network Downtime and Time Since last MTD
- LSTM: Captures temporal dependencies in the data

Model Architecture



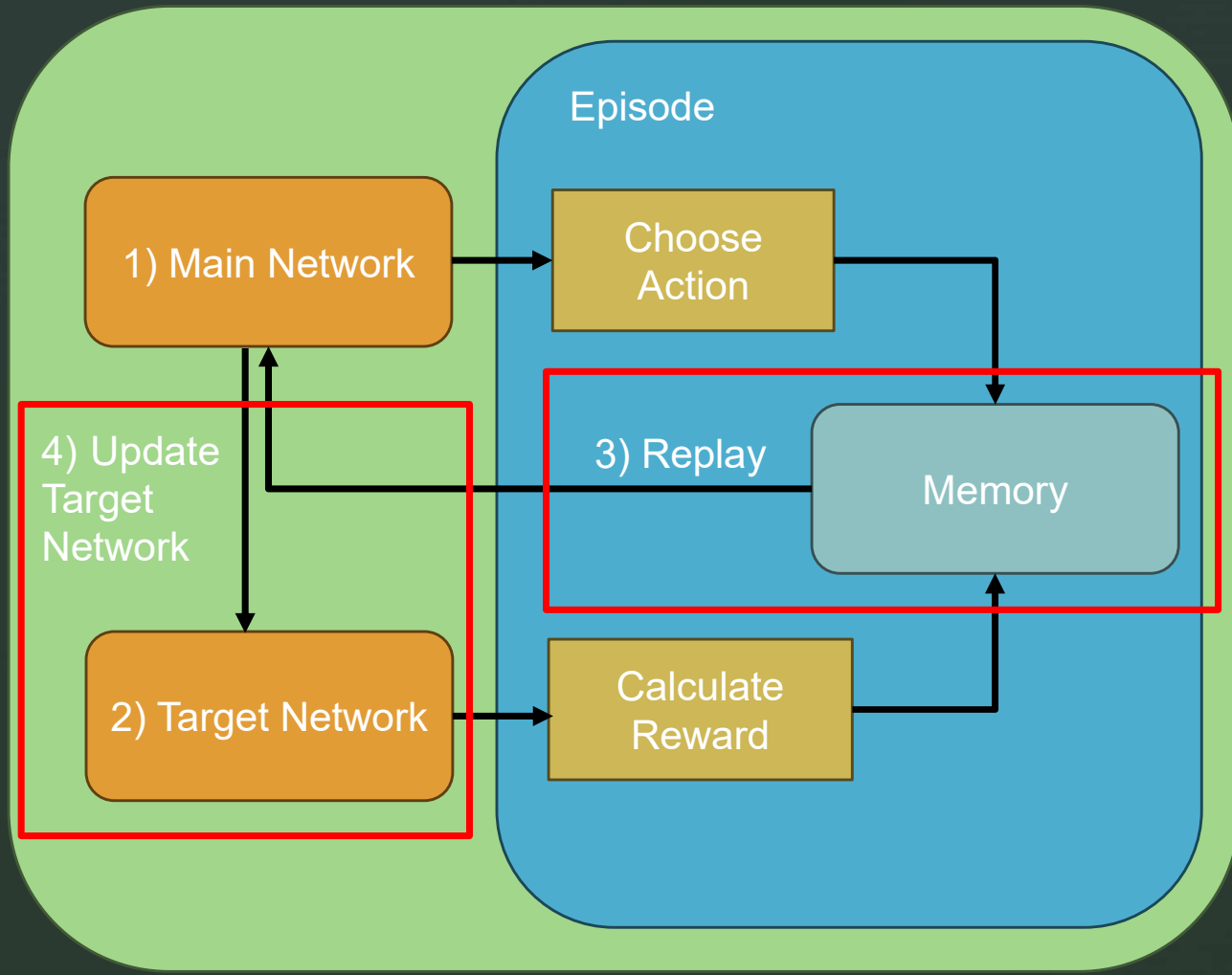
- Feature fusion: Dense layers to integrate the data from the previous 2 modules
- Q-Network Output: Vector of Q-Values
- Action selection tells MTD Module what to do

Model Training



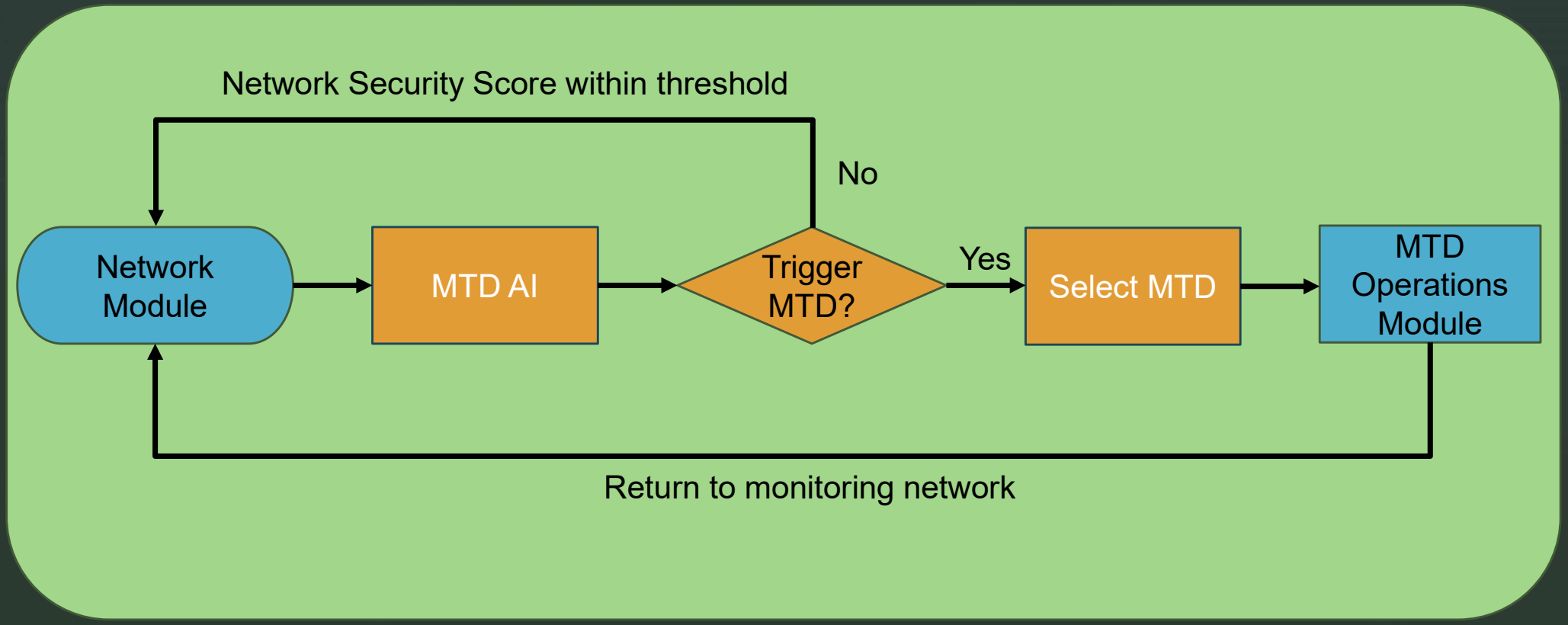
- Main Network: The network that is being actively trained in each episode
- Used to select the best action in each episode
- Target Network: Separate network used to calculate the reward for the actions taken by the main network
- Weights are not updated as frequently as the main network
- Acts as a stable reference to compute Q-values

Model Training



- Replay: agent samples experiences randomly from memory to learn from
- Weights of the main network updated in this step
- Weights of the target network are updated infrequently

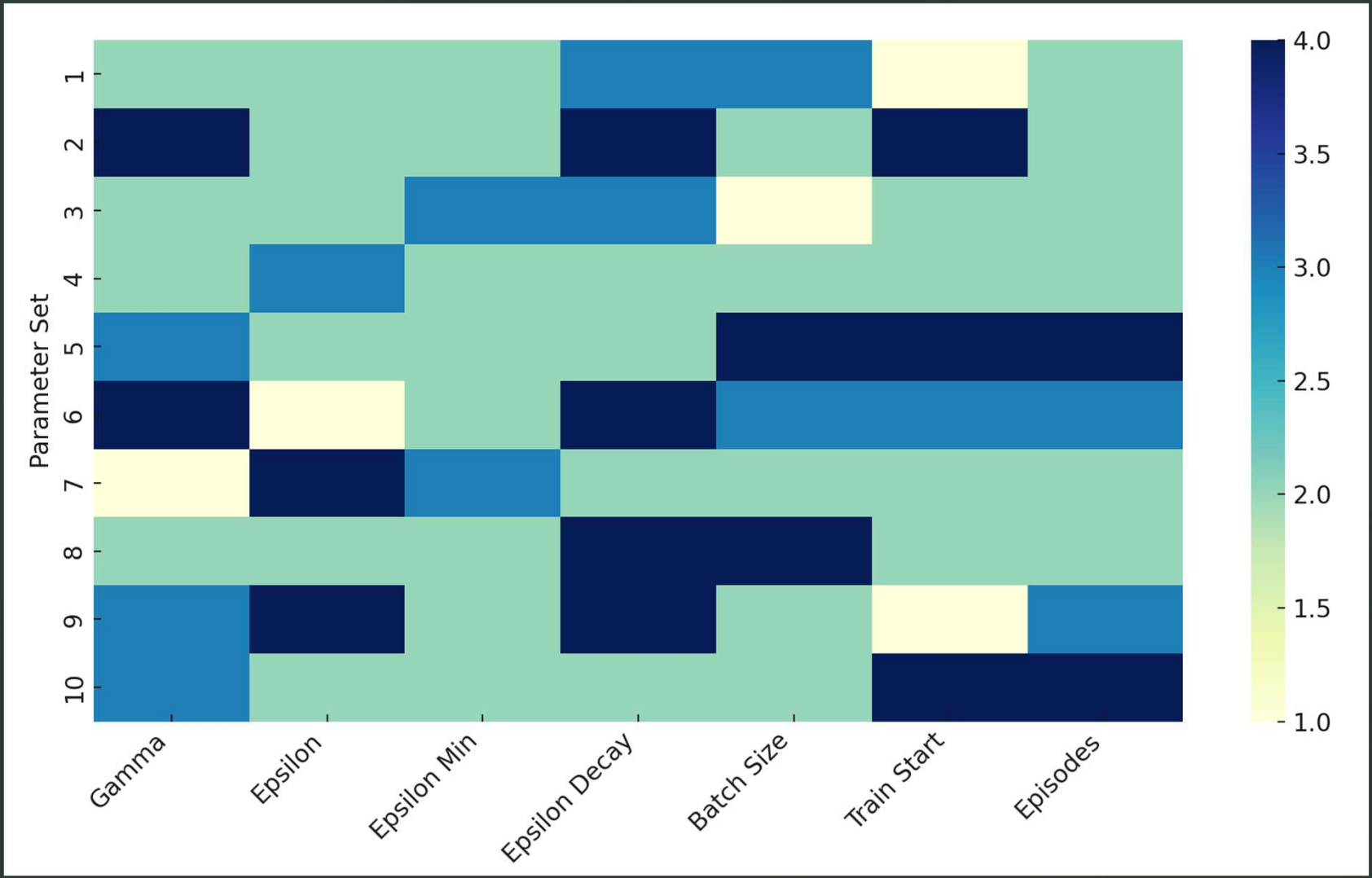
Model Integration



Model Parameters

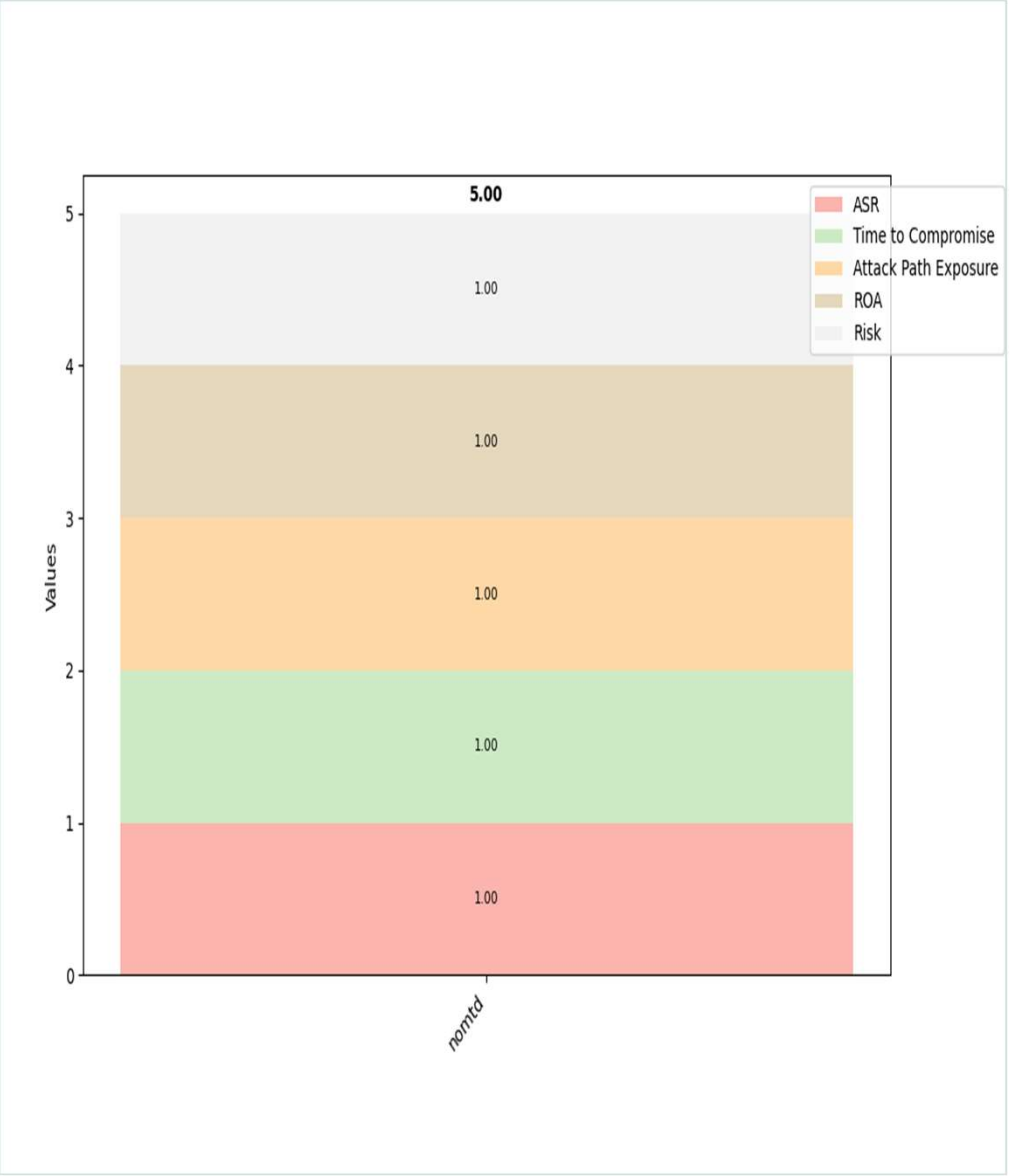
Parameter	Impact on Simulation
Gamma (Discount Factor)	Determines the importance of future rewards versus immediate rewards
Epsilon	Controls the agent's exploration versus exploitation behaviour
Epsilon Min	Ensures that the agent continues exploring occasionally, which helps avoid local optima
Epsilon Decay	The rate at which the agent will switch from exploring to exploiting
Batch Size	Larger batch sizes influence stability of learning
Train Start	The number of experiences that must be stored in the memory buffer before training starts.
Episodes	The total number of episodes (interactions with the environment) the agent will train over.

Heatmap of Parameter Sets



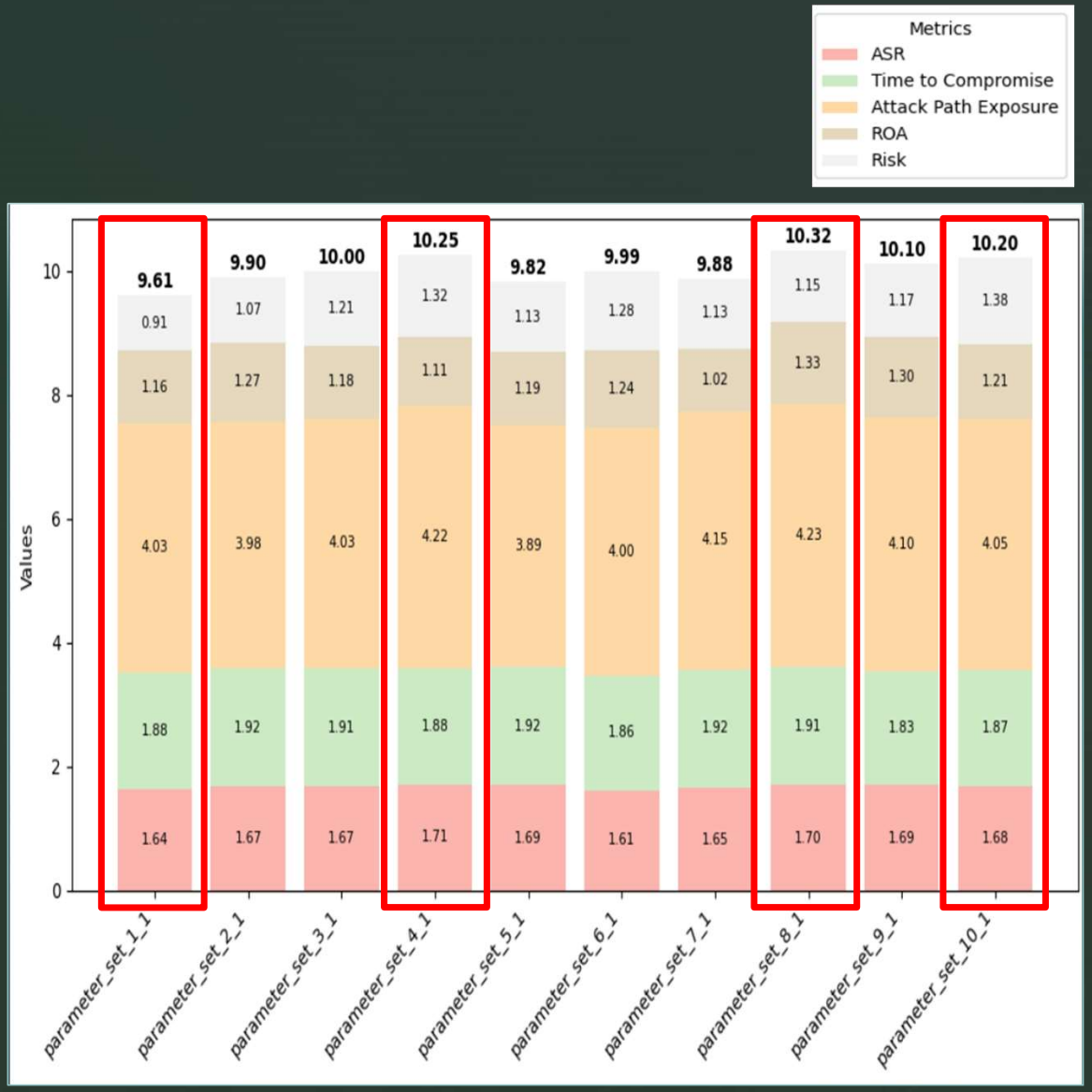
No MTD Scenario

- Base case with no MTD deployed
- All other scenarios will be scaled based on these results



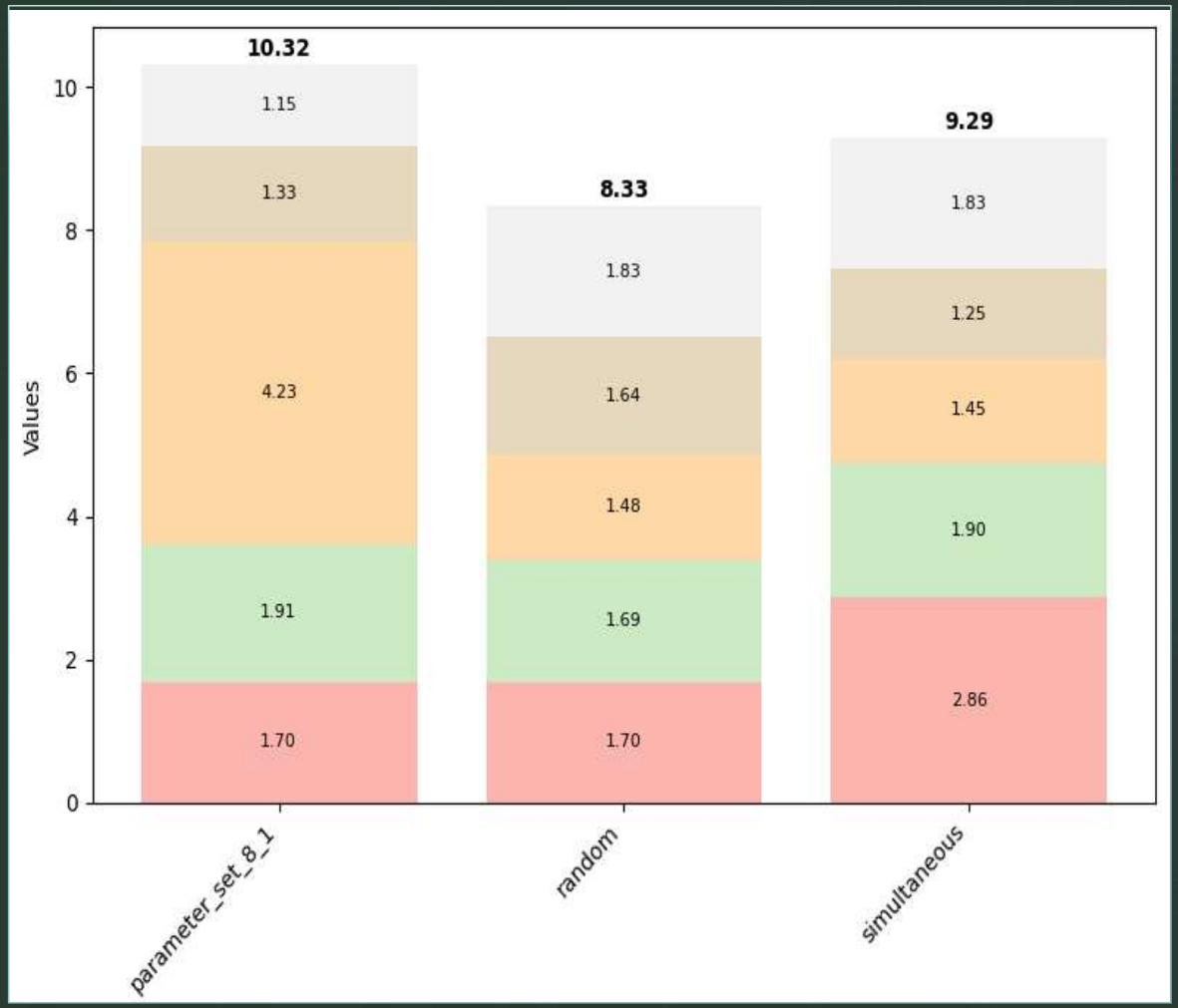
Parameter Sets 1-10

- Parameter set 8: Balances exploration and exploitation with a large memory buffer and slower epsilon decay, promoting more comprehensive learning.
- Parameter set 4: Starts with high exploration, gradually shifting to exploitation, providing a balanced approach to exploration and exploitation over time.
- Parameter set 10: Optimizes long-term rewards and learning stability by delaying training, collecting more experiences, and training over an extended number of episodes.
- Parameter set 1: Focuses on faster learning with a strong emphasis on short-term rewards and quicker convergence to exploitation.



MTD AI vs Other Schemes

- Random: Randomly deploys an MTD technique at fixed intervals
- Simultaneous: Deploys all available MTD techniques at fixed intervals



Environmental Factors

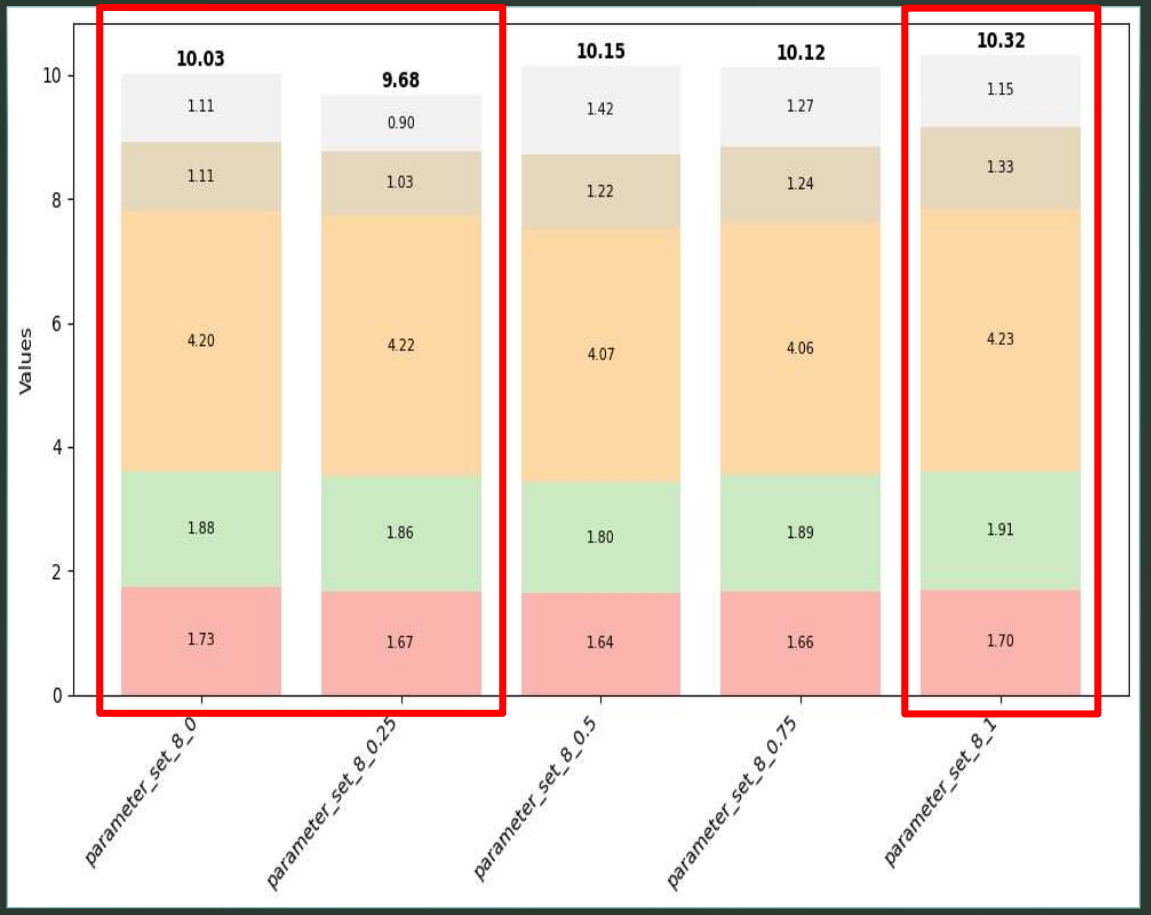
What **factors** affect the effectiveness of the ML model when deployed in the network?

Intrusion Detection Systems

- Intrusion Detection Systems (IDS) are security solutions designed to monitor network or system activities for malicious actions or policy violations.
- They monitor unauthorized attacks on the system.
- Experiments were conducted to determine if the presence of data from the IDS would supplement the ability of the ML model to respond to threats.

Sensitivity Analysis

- Model with full knowledge of attacker performed the best due to being able to learn how to counter each attack type with MTD
- Model which received only limited information performed worse than model with no attacker information



Conclusion

- Using MTD AI outperforms classical methods of MTD
  - Different environments and features can cause the ML model to bias towards optimizing a particular security metric
  - Focus on long term rewards and exploration leads to better results
- Having information from an IDS system is not always better
  - If the IDS system is able to capture attacker actions it can lead to significant benefits
  - If the IDS is providing little or misleading information it can confuse the ML model which leads to worse performance

Future Works

- Investigate which parameters and features can be used to increase model performance in the areas of Attack Success Rate and Risk.
- Investigate other types of reinforcement learning models that can be applied to the simulator such as Policy Gradient (PG)
- Integrate external intrusion datasets into the simulator to test the effectiveness of MTD AI on real-world datasets

Questions

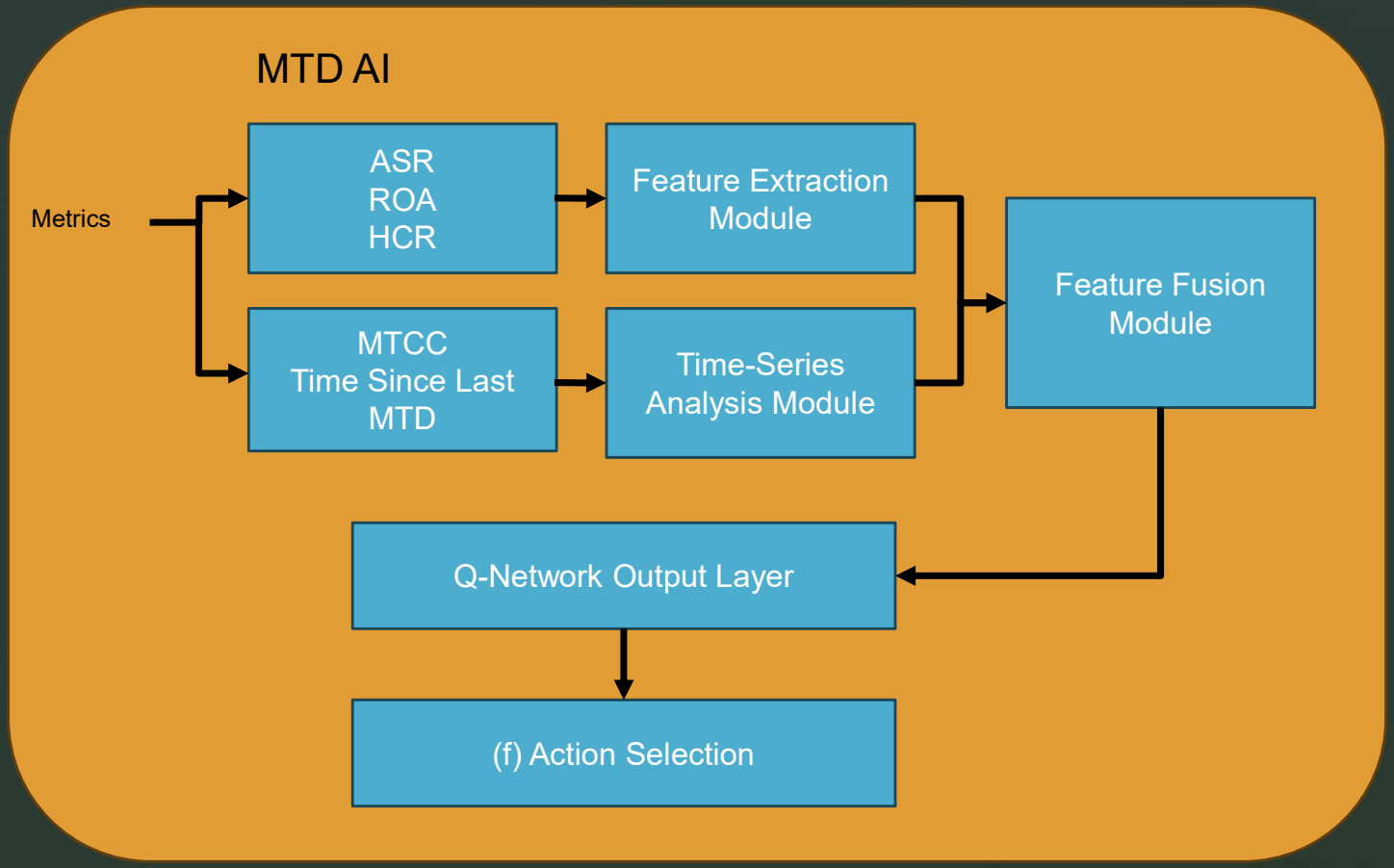
References

- [1] R. Colbaugh and K. Glass, “Predictability-oriented defense againstadaptive adversaries,” in Proc. IEEE Int. Conf. Syst. Man Cybern.(SMC), Oct. 2012, pp. 2721–2727.
- [2] E. Farchi, O. Shehory, and G. Barash, “Defending via strategic MLselection,” arXiv preprint arXiv:1904.00737, 2019.
- [3] S. Vikram, C. Yang, and G. Gu, “NOMAD: Towards non-intrusivemoving-target defense against Web BOTs,” in Proc. IEEE Conf.Commun. Netw. Security (CNS), 2013, pp. 55–63.
- [4] Q. Song, Z. Yan, and R. Tan, “Moving target defense for deep visualsensing against adversarial examples,” in Proc. Conf. Embedded Netw.Sensor Syst. (SenSys), 2019, pp. 124–137.
- [5] W. Zhang “MTDSimTime”, Github 2021. [Online]. Available: https://github.com/MoeBuTa/MTDSimTime

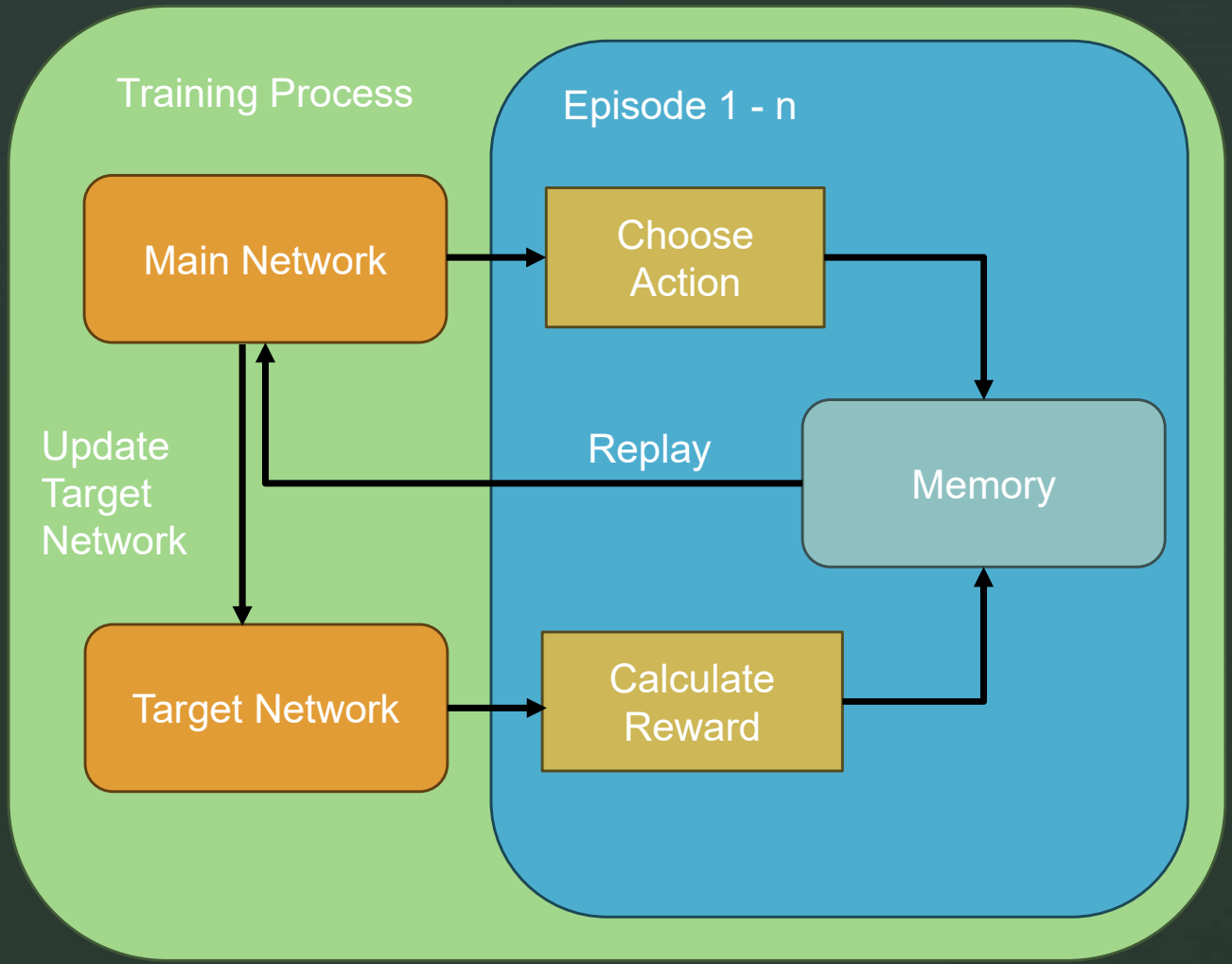
Parameter Sets

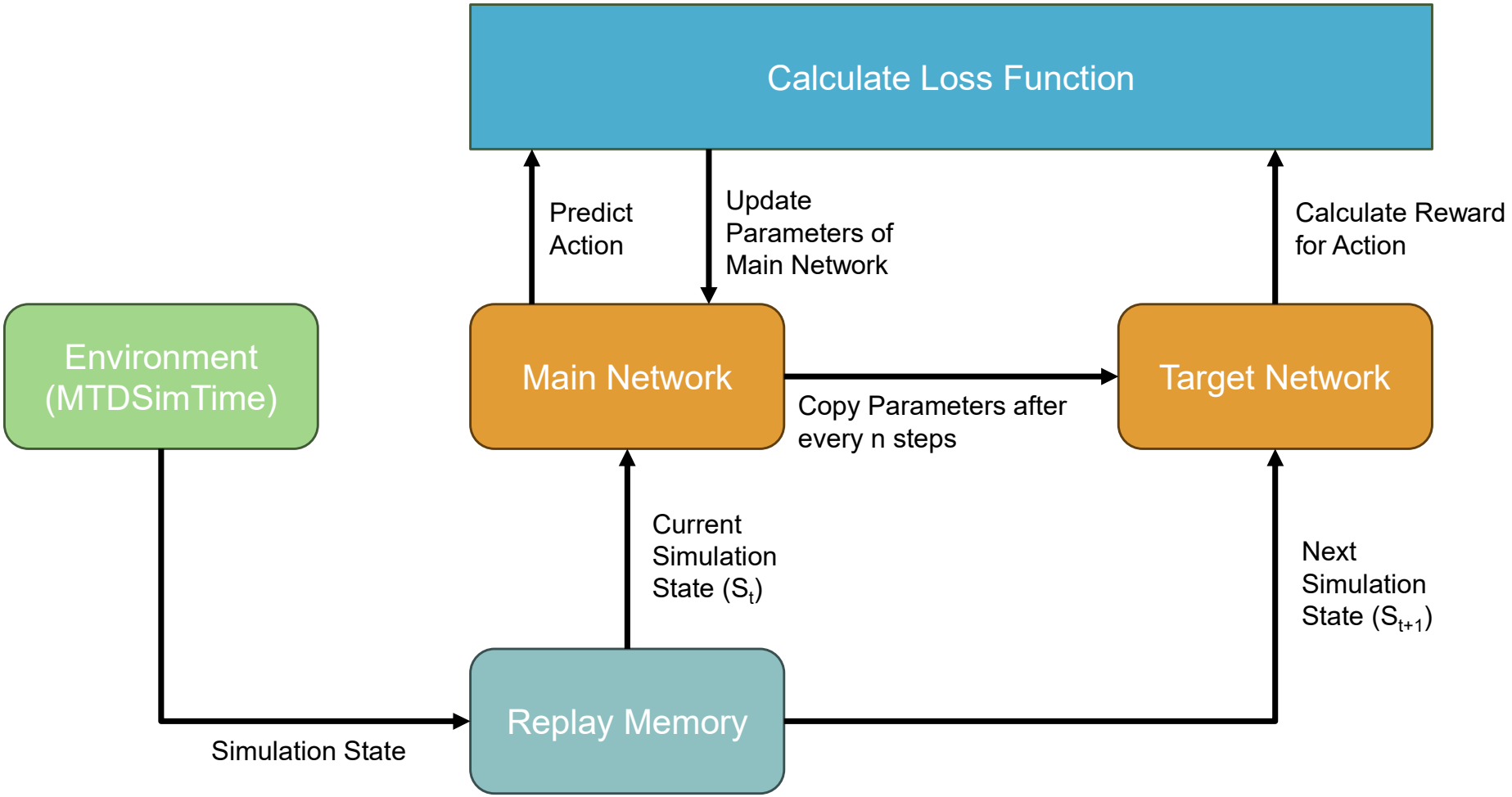
Parameter Set	Gamma	Epsilon	Epsilon Min	Epsilon Decay	Batch Size	Train Start	Episodes
1	Standard	Standard	Standard	High	High	Low	Standard
2	Very High	Standard	Standard	Very High	Standard	Very High	Standard
3	Standard	Standard	High	High	Low	Standard	Standard
4	Standard	High	Standard	Standard	Standard	Standard	Standard
5	High	Standard	Standard	Standard	Very High	Very High	Very High
6	Very High	Low	Standard	Very High	High	High	High
7	Low	Very High	High	High	Standard	Standard	Standard
8	Standard	Standard	Standard	Very High	High	Very High	Standard
9	High	Very High	Standard	Very High	Standard	Low	High
10	High	Standard	Standard	Standard	Standard	Very High	Very High

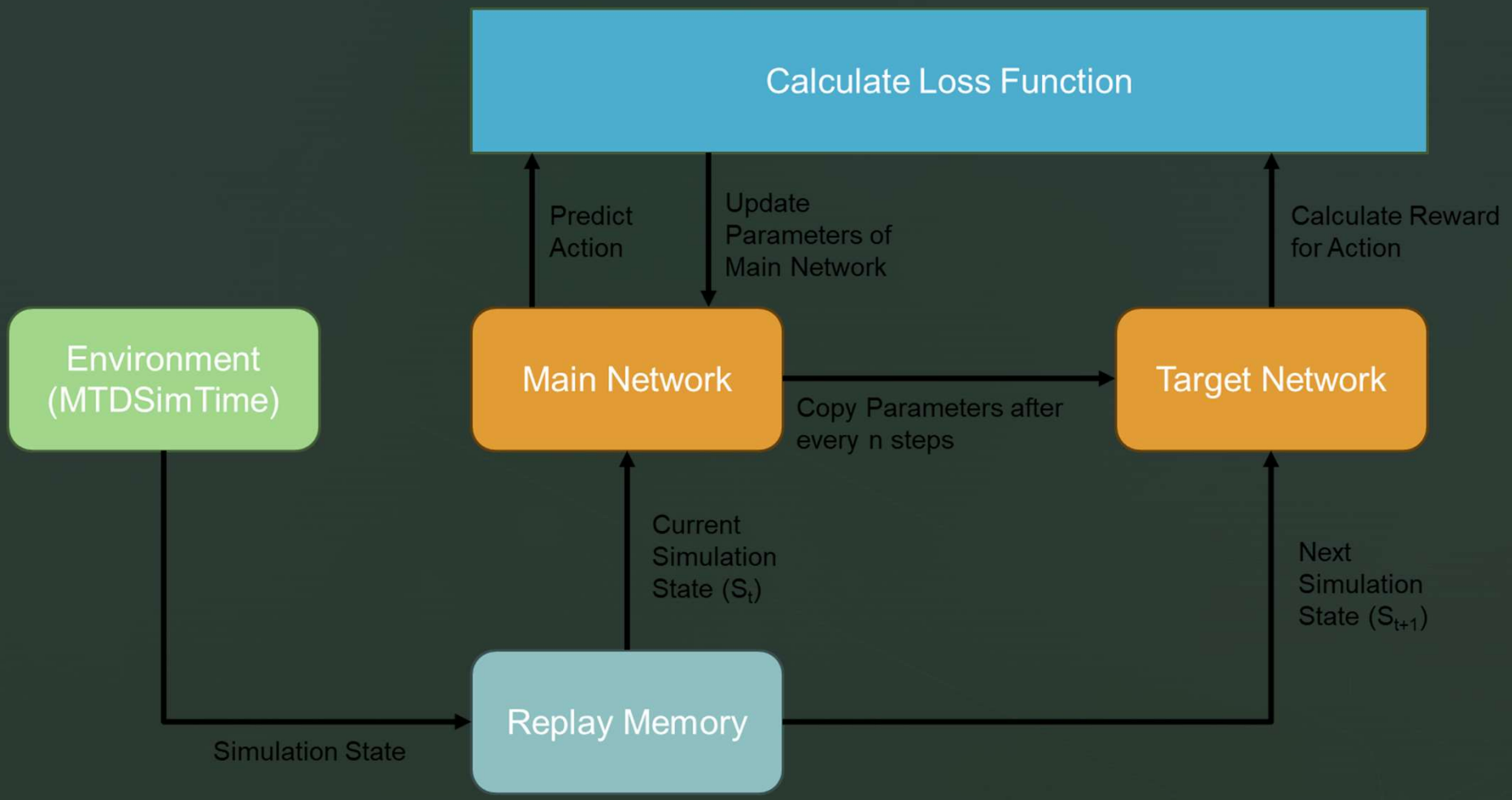
Model Architecture



Model Training




Calculate Loss Function
Predict Action
Update Parameters of Main Network
Calculate Reward for Action
Environment (MTDSimTime)
Main Network
Target Network
Copy Parameters after every n steps
Current Simulation State (S_t)
Next Simulation State (S_{t+1})
Replay Memory
Simulation State


Calculate Loss Function
Predict Action
Update Parameters of Main Network
Calculate Reward for Action
Environment (MTDSimTime)
Main Network
Target Network
Copy Parameters after every n steps
Current Simulation State (S_t)
Next Simulation State (S_{t+1})
Replay Memory
Simulation State

No MTD Scenario

- Base case with no MTD deployed
- All other scenarios will be scaled based on these results