Reinforcement Learning in StarCraft in Real-Time using Deep Learning

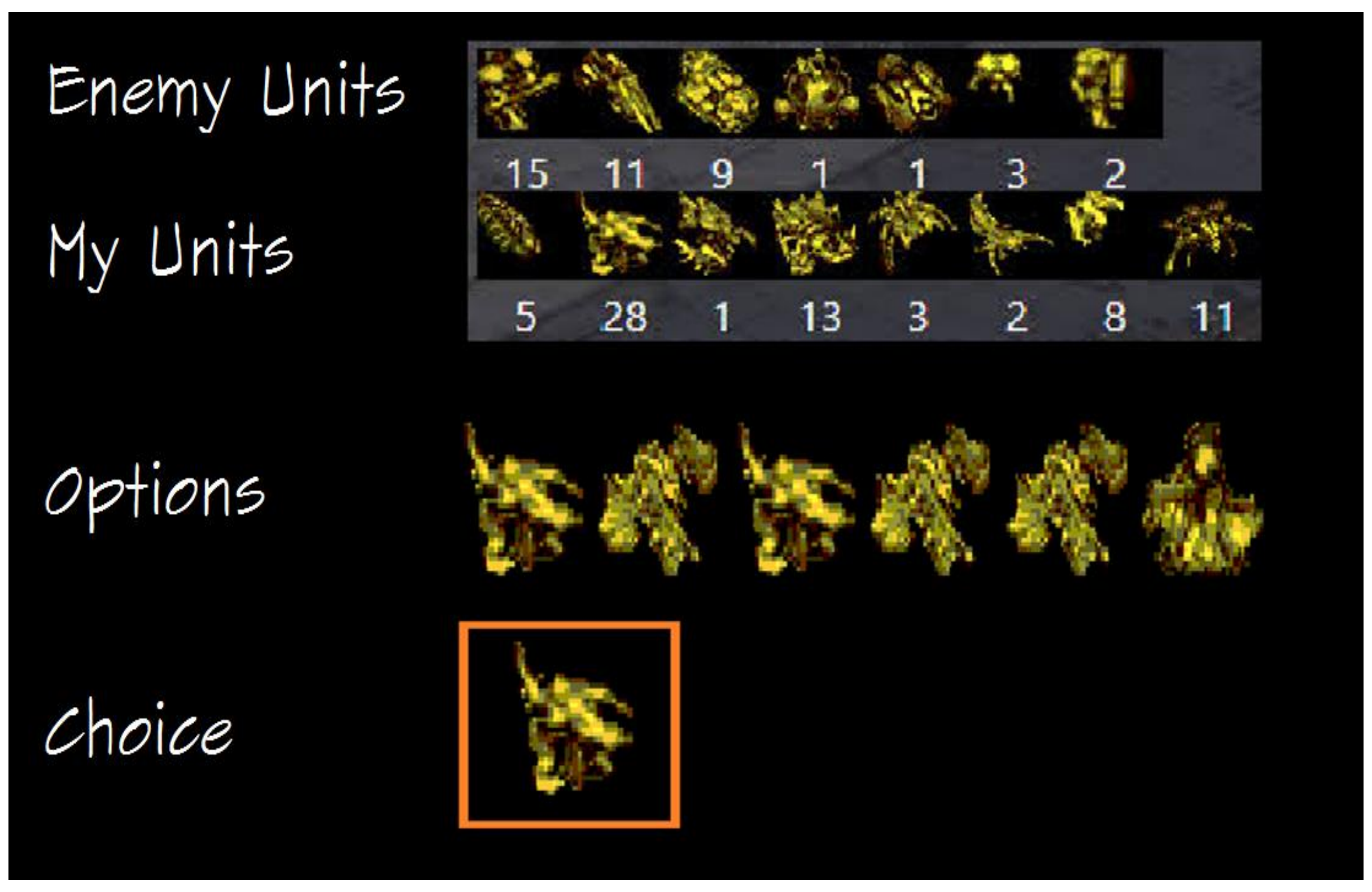
Sungguk Cha

School of Electrical & Computer Engineering, UNIST

Introduction

The Real-Time Strategy game *StarCraft* has proven to be a challenging environment for artificial intelligence techniques, and as a result, current state-of-the-art solutions consist of numerous hand-crafted modules. One of the challenges is ***decision-making***. It focuses on deciding build order, i.e., the order that the player produces materials such as units, buildings, and upgrades, which is a huge part of strategical decision over the game.

Goal



- **Find best option in real-time**

  at which given backgrounds such as enemy units, and my units. It implies modeling opponent, resource management, and decision making under uncertainty.

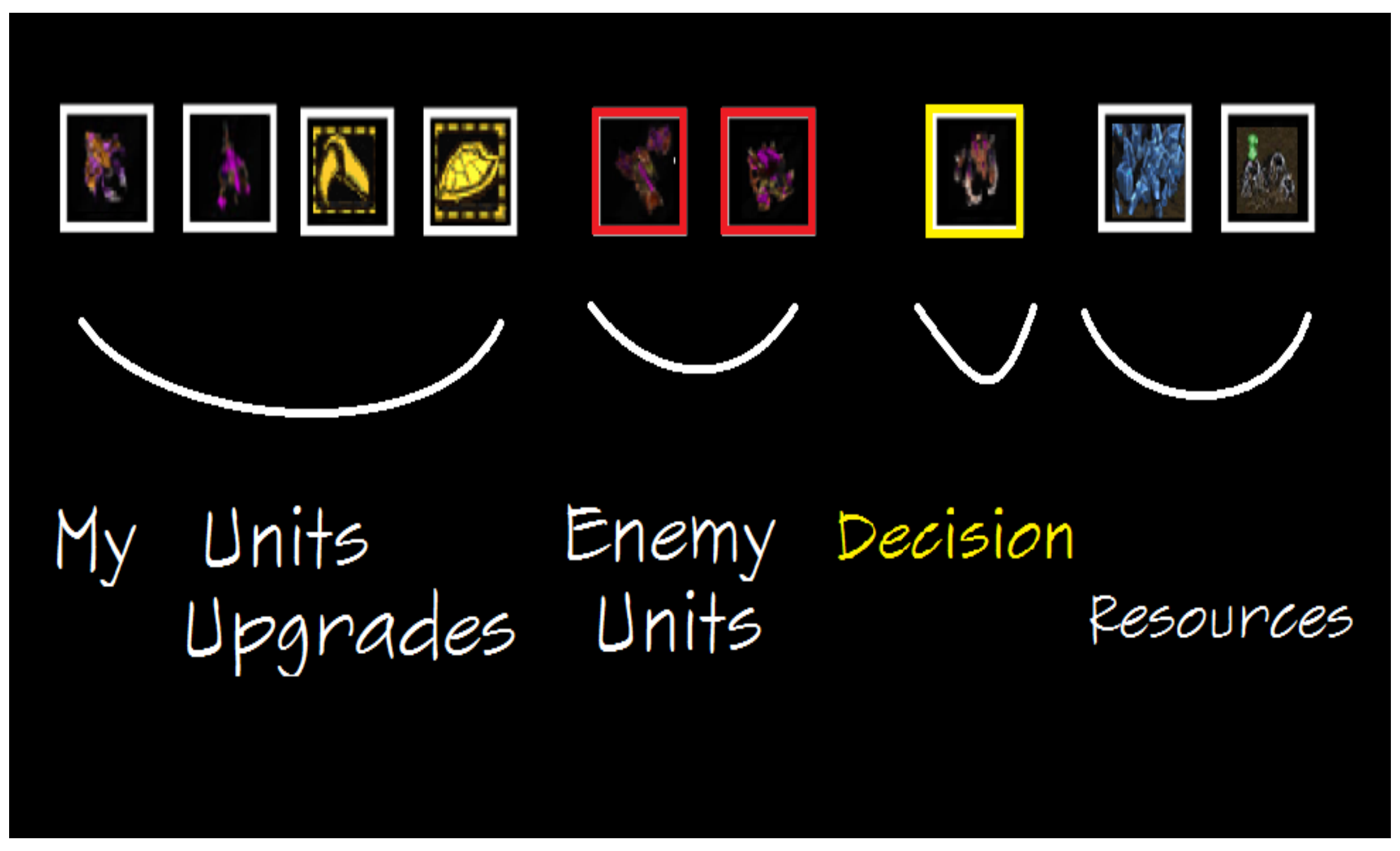
- **Build its strategy**

  In the end, starting from nothing, it learns how to win, and find its build order as a strategy by itself with reinforcement learning.

Approach

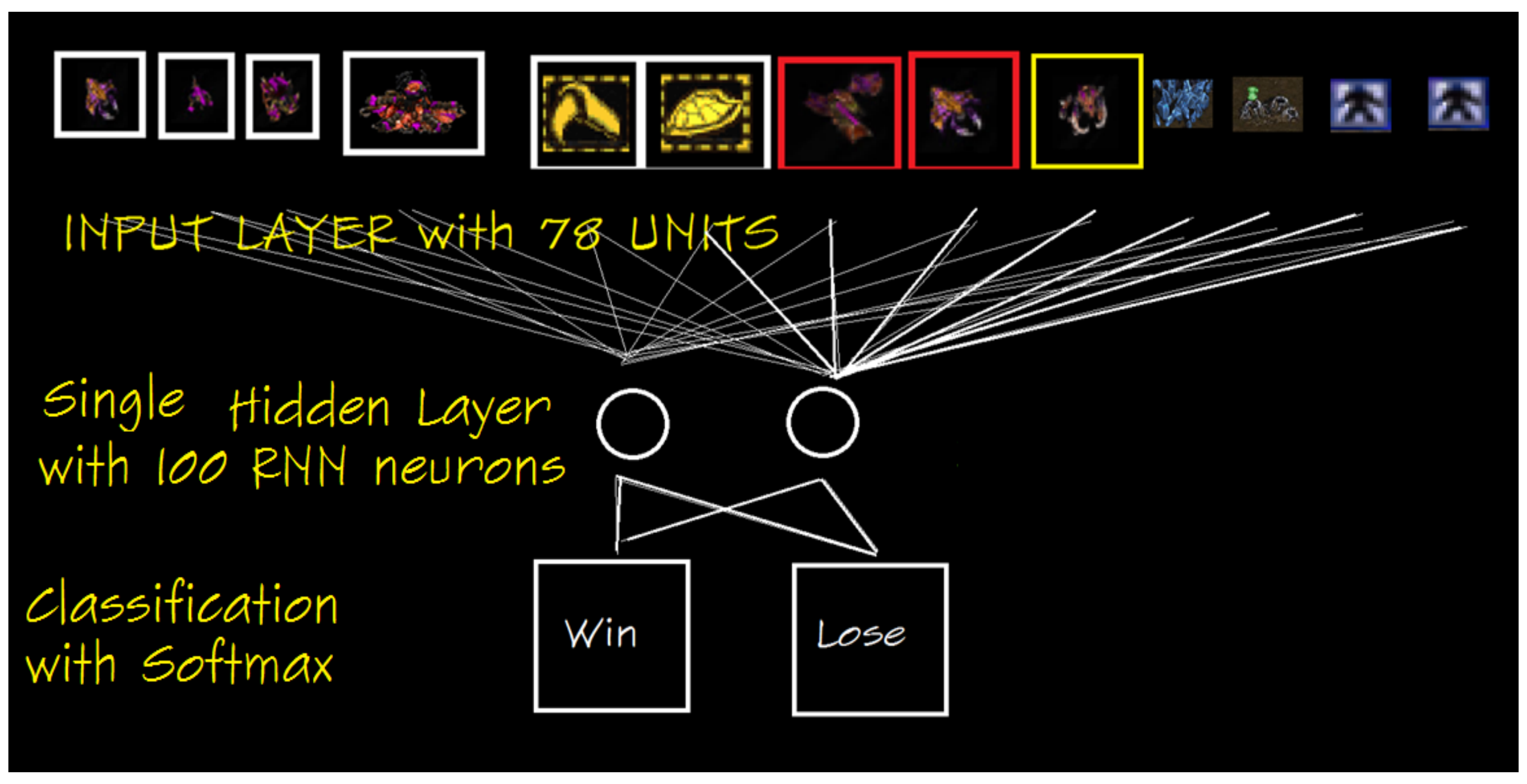
A. Dataset

During the games, all decision-makings regarding build order are recorded with back grounds: the numbers of my units, my upgrades, observed enemy units, and my resources.



B. Neural Network Architecture

Datasets become inputs (input layers). The input layers are connected to a hidden single layer with 100 RNN neurons. The hidden layer is connected to two output layers(win, and lose). The output layers are softmaxed.

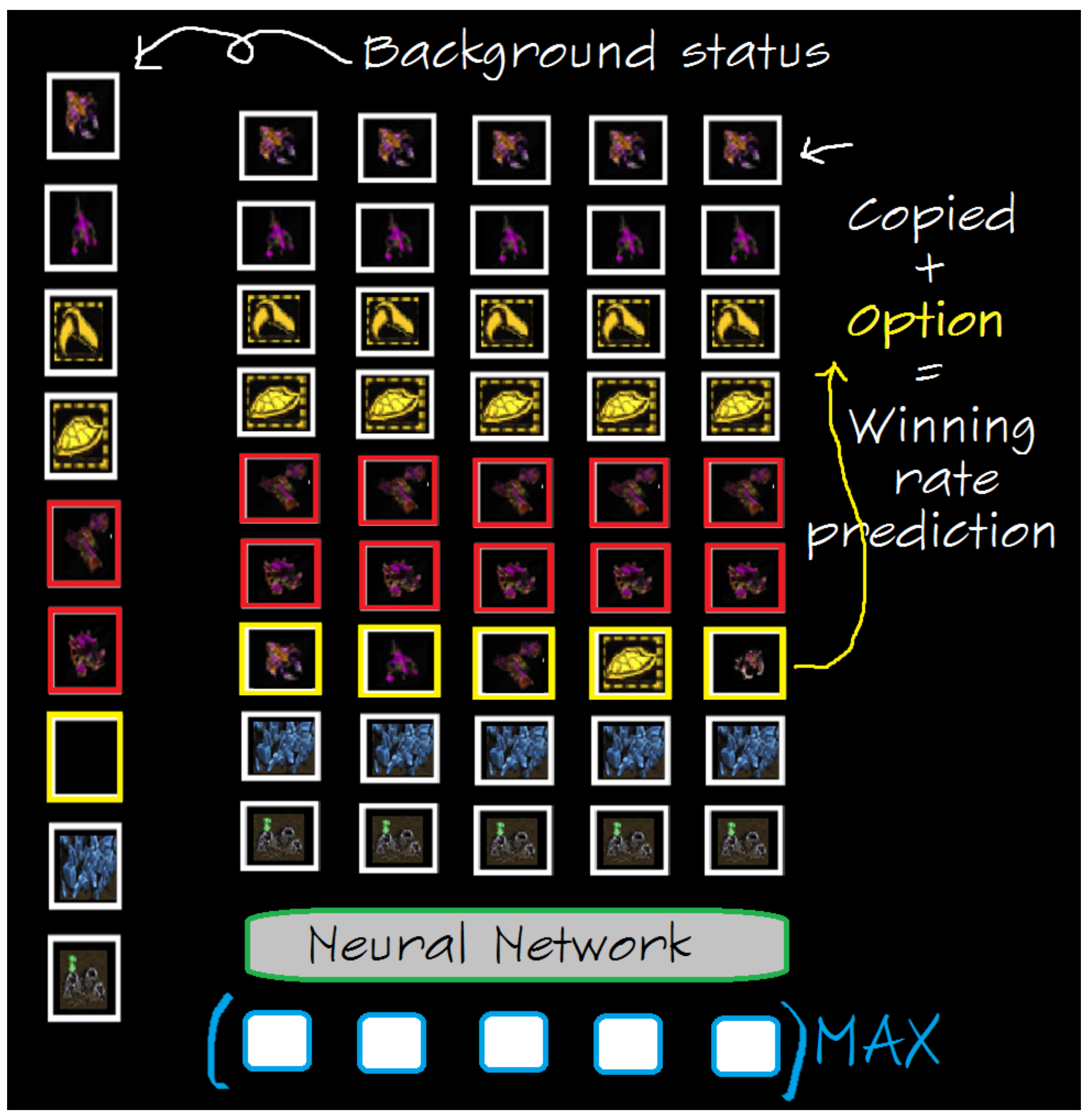

INPUT LAYER with 78 UNITS
Single Hidden Layer with 100 RNN neurons
Classification with Softmax
Win
Lose

C. Reinforcement Learning

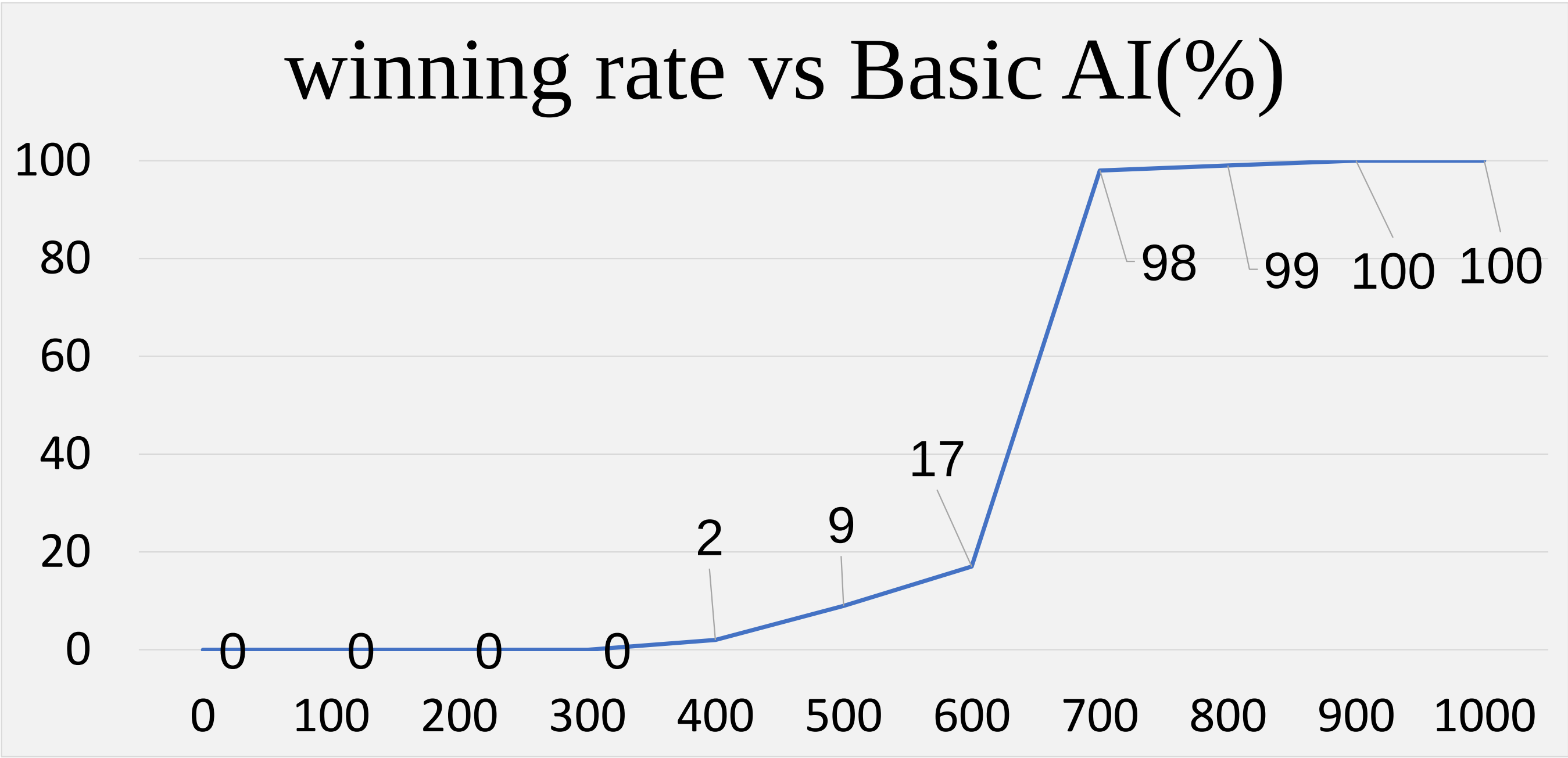
- Two AI modules fight against each other, which have the same structured network, but different initialization.
- After every game ends, it learns with the data of the game for 100~500 times.
- There are stalemates when their build orders converge to the same, even though they are initialized differently. If so, one is loaded with neuron data of 100 games before.

D. Build Order Selection

At every decision making, it checks winning-rate-prediction of the network with possible choices, and choose one with the highest winning-rate(*Greedy*).



Results

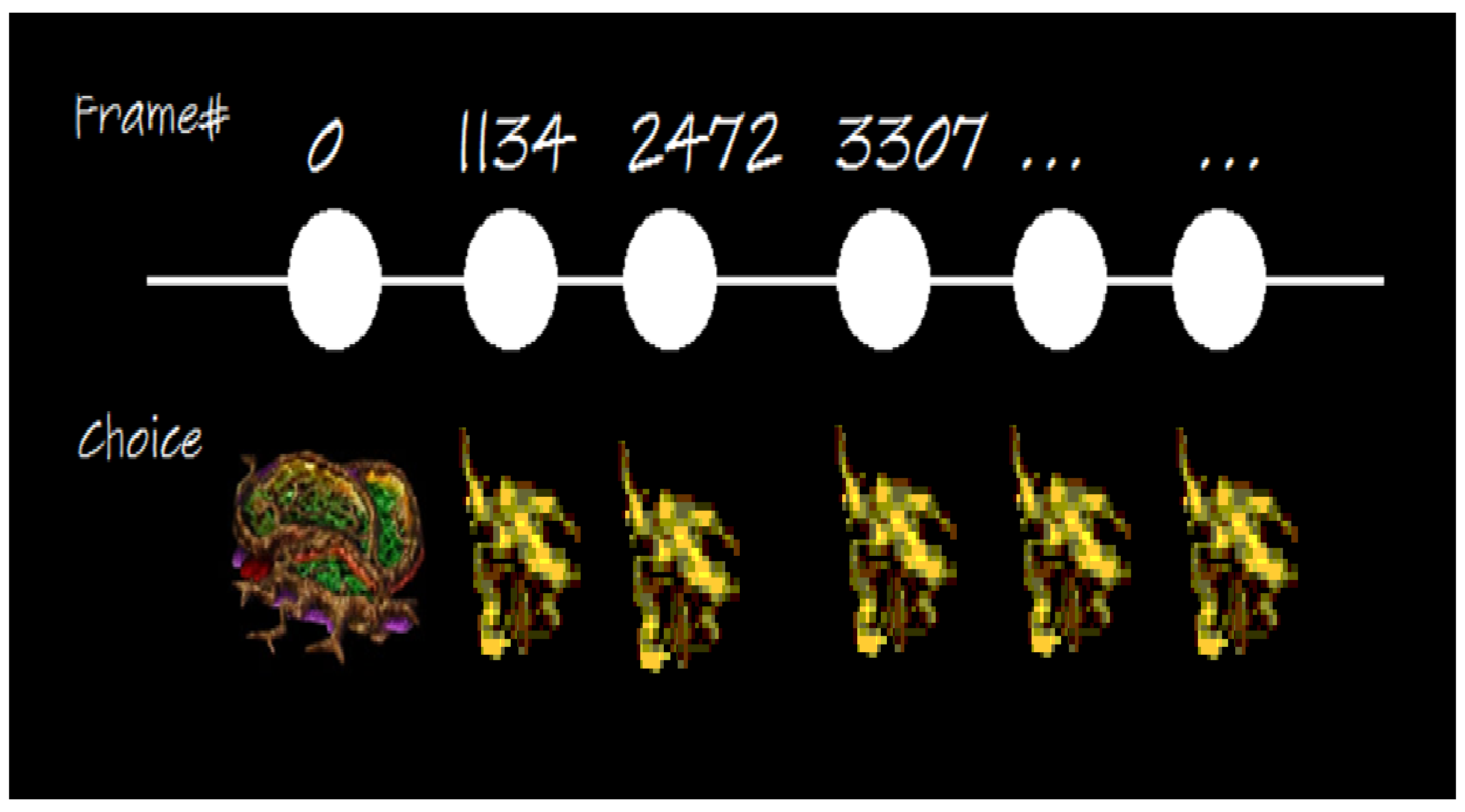


VICTORY!

As a result of choosing the best option in real-time, it achieved increasing winning rate against basic AI, and 100% winning rate in the end. For every 100 games of learning, it played against basic AI for 100 times without learning.

Strategy FOUND!

At the beginning, it behaved very weird in gamer's point of view, in a similar way of having a meal with three spoons. After some number of games, there is a highly likely used early game sharp strategy (a.k.a. 4drone rush).


Frame#
0 1134 2472 3307 … …
Choice