Review on "StarCraft Micromanagement with Reinforcement Learning and Curriculum Transfer Learning" paper [1]

Tatiana Gaintseva

MIPT DIHT
YSDA

RL reading seminar, 2018

Outline

Plan

Problem features

- Large environment with obstacles
- Many units so interaction needed
- Units' properties are different – different difficulty levels

Attributes	Goliath	Zealot	Zergling	Marine
Race	Terran	Protoss	Zerg	Terran
HitPoint	125	160	35	40
CoolDown	22	22	8	15
Damage Factor	12	16	5	6
Defence Factor	1	1	0	0
Fire Range	5	1	1	4
Sight Range	8	7	5	7

Prior work

- Separate focus: navigation /attack only
- Controlling only one unit
- Successful full combat scenario, but with lack of generalization/ high computational power is needed

State of the art: BicNet [2]: A DRL method, based on the actor-critic architecture

Plan

What we'll bring to the world

Proposed model addresses issues:

- Huge state space
- multi-agent decision making
- Generalization
- Learning algorithm

Key ideas fields:

- State space representation
- Action definition
- Network architecture
- Learning algorithm
- Reward function
- Curricilum transfer learning

Plan

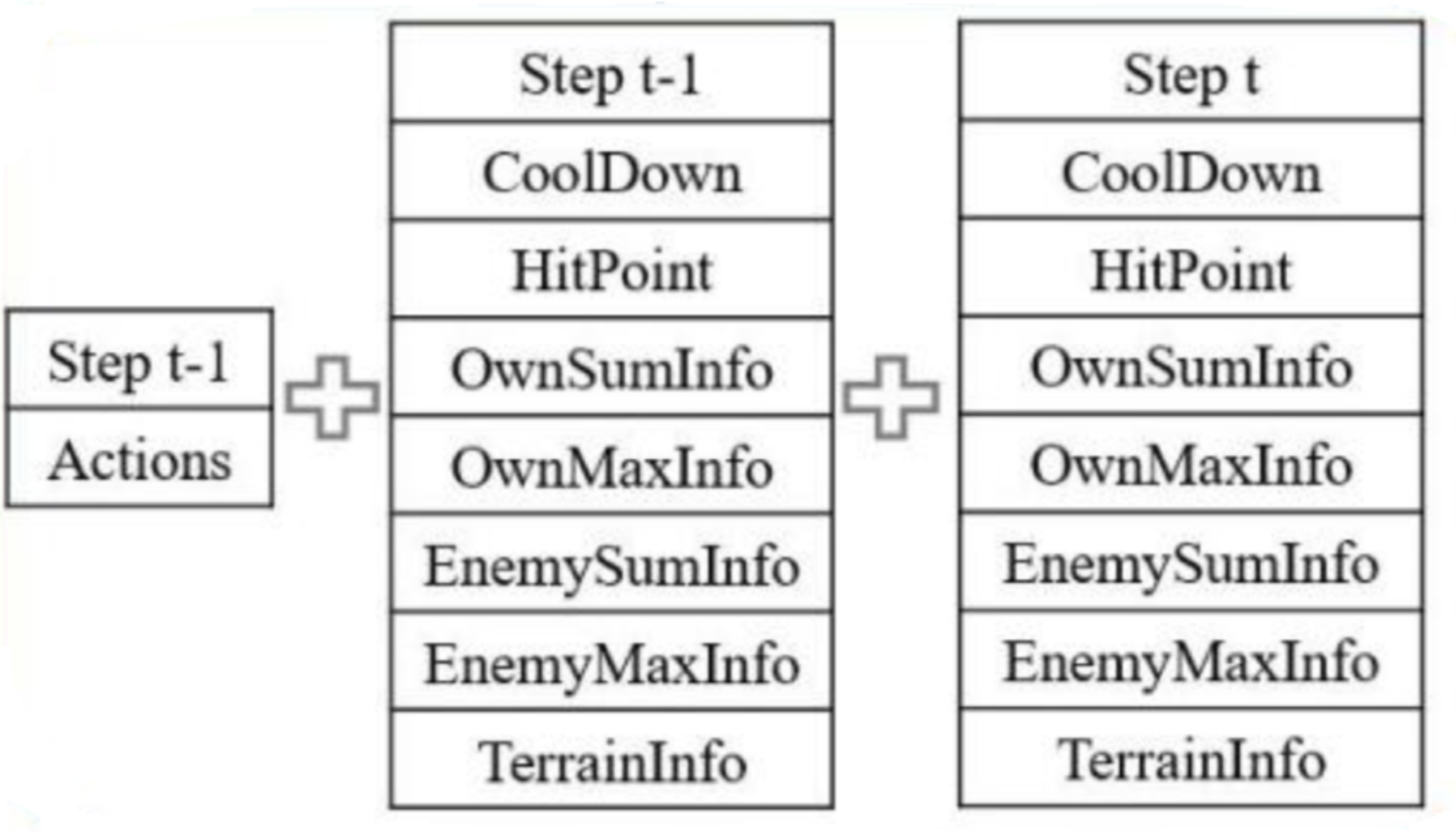
State space representation I

State space dimensions:

- Terrain distance info
- Enemies distance info
- Own team stats
- Last step action

what we take from state

State space representation II



We use last step state info as well [3] [4]
Environment is divided into 8 parts, totalling 93 dimensions

State space representation III

How to calculate distance:

$$dis\_unit(d) = \begin{cases} 0.05, & d > D \\ 1 - 0.95(d/D), & d \leqslant D \end{cases}$$

$$dis\_terrain(d) = \begin{cases} 0, & d > D \\ 1 - d/D, & d \leqslant D \end{cases}$$

- OwnSumInfo: own units' distances are **summed** in each area
- OwnMaxInfo: own units' distances are **maximized** in each area
- EnemySumInfo: enemy units' distances are **summed** in each area
- EnemyMaxInfo: enemy units' distances are **maximized** in each area

State space representation IV

Notice:

- Using last step info as well as current step info [3] [4]
- **Key point:** All the units are part of the environment
- Good generalization

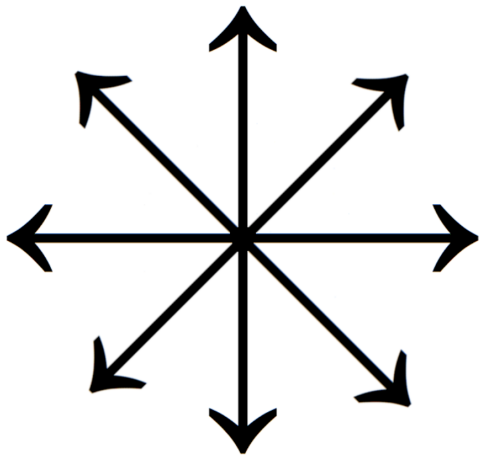
Plan

Actions I

Action choice:

- 8 directions to move (fixed distance)
- Attack the weakest unit

Total dimension is 9

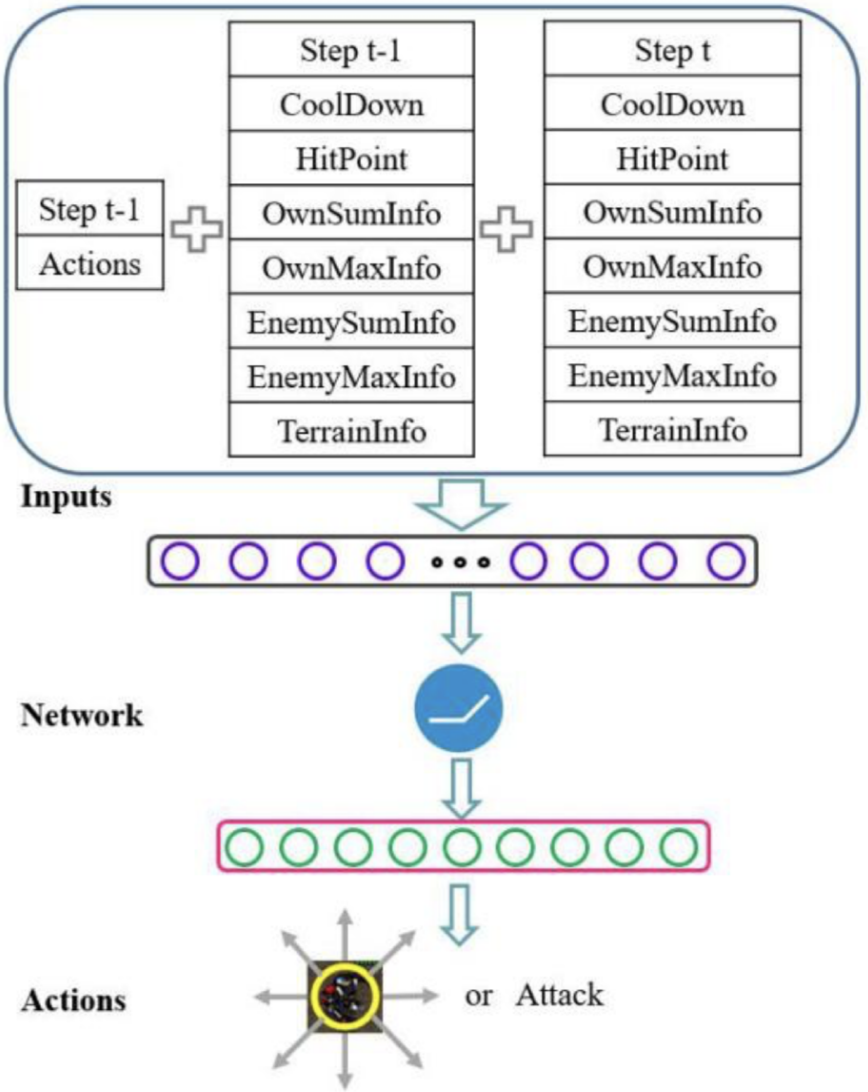


Plan

Model I

- Input – state vector (dim 93)
- 1 hidden layer with 100 neurons
- ReLu activation [5], [6]
- Output – actions probs vector (dim 9)

Key point: Only one net for all the units!



Plan

Learning method I

Idea: Sarsa(λ) extended to multiple units

Gradient descent updates as follows:

$$\delta_t = r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}, \Theta_t) - Q(s_t, a_t, \Theta_t)$$

$$\Theta_{t+1} = \Theta_t + \alpha \delta_t e_t$$

$$e_t\ [7] = \gamma \lambda e_{t-1} + \bigtriangledown_{\Theta_t} Q(s_{t+1}, a_{t+1}, \Theta_{t+1}) \quad (e_0 = 0)$$

ϵ-greedy strategy with exponential decay:

$$action\ a = \begin{cases} randint(N) & random(0,1) < \epsilon \\ argmax_a Q(s,a) & otherwise \end{cases}$$

$$\epsilon = 0.5/\sqrt{1 + episode\_num}$$

Learning method II

Algorithm 1 Parameter Sharing Multi-Agent Gradient-Descent Sarsa(λ)

Initialize policy parameters $\boldsymbol{\theta}$ shared among our units
Repeat (for each episode):
  $\boldsymbol{e}_0 = \mathbf{0}$
  Initialize s_t, a_t
  Repeat (for each step of episode):
    Repeat (for each unit):
      Take action a_t, receive r_{t+1}, next state s_{t+1}
      Choose a_{t+1} from s_{t+1} using ϵ-greedy
      If $random(0, 1) < \epsilon$
        $a_{t+1} = randint(N)$
      else
        $a_{t+1} = \arg\max_a Q(s_{t+1}, a; \boldsymbol{\theta}_t)$
    Repeat (for each unit):
      Update TD error, weights and eligibility traces
      $\delta_t = r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}; \boldsymbol{\theta}_t) - Q(s_t, a_t; \boldsymbol{\theta}_t)$
      $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \delta_t \boldsymbol{e}_t$
      $\boldsymbol{e}_{t+1} = \gamma \lambda \boldsymbol{e}_t + \nabla_{\boldsymbol{\theta}_{t+1}} Q(s_{t+1}, a_{t+1}; \boldsymbol{\theta}_{t+1})$
      $t \leftarrow t + 1$
  until s_t is terminal

Plan

Main reward I

Total reward: $\boldsymbol{r_t = t_t^{main} + r_t^{extra}}$

$$\boldsymbol{r_t^{main}} = [damage\_amount \times damage\_factor - \\ -\rho \times (unit\_hitpoint_{t-1} - unit\_hitpoint_t)] \\ /norm\_constant$$

$\rho = \sum_{i=1}^{H} enemy\_hitpoint_i / \sum_{j=1}^{N} unit\_hitpoint_j$

$H =$ number of enemy units

$N =$ number of our units

$norm\_constant = 10$

Main reward II

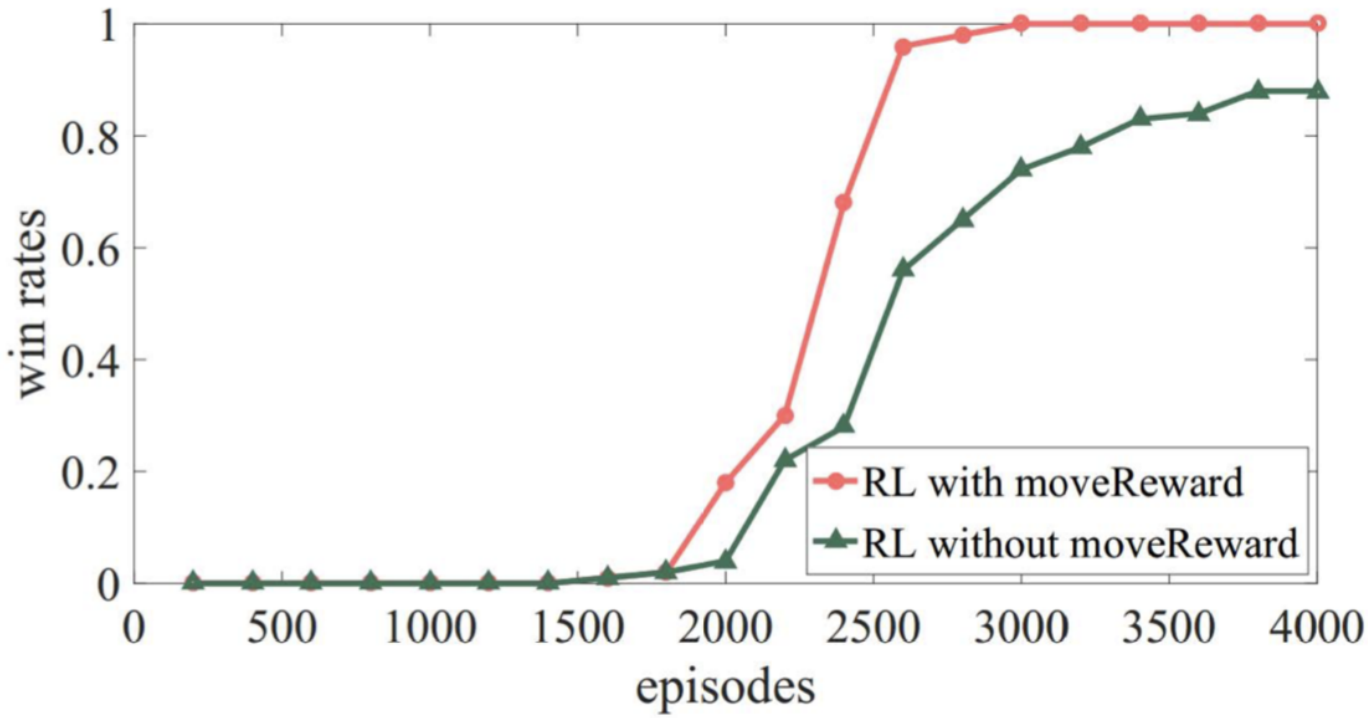
THE COMPARATIVE ATTRIBUTES OF DIFFERENT UNITS IN OUR MICROMANAGEMENT SCENARIOS.

Attributes	Goliath	Zealot	Zergling	Marine
Race	Terran	Protoss	Zerg	Terran
HitPoint	125	160	35	40
CoolDown	22	22	8	15
Damage Factor	12	16	5	6
Defence Factor	1	1	0	0
Fire Range	5	1	1	4
Sight Range	8	7	5	7

Extra rewards I

Extra rewards r_t^{extra}:

- -10 for every destroyed unit
- ***moveReward***: -0.5 for moving to the direction without any units (enemy or ours)



Plan

Experiments setup I

1. Small scale micromanagement
   - 3 Goliaths vs 6 Zealoths
   - 3 Goliaths vs 20 Zerglings
2. Large scale micromanagement
   - X Marines vs Y Zerglings

Enemy: build-in AI

Platinum-level players win $< 50\%$ games of 100 for each scenario

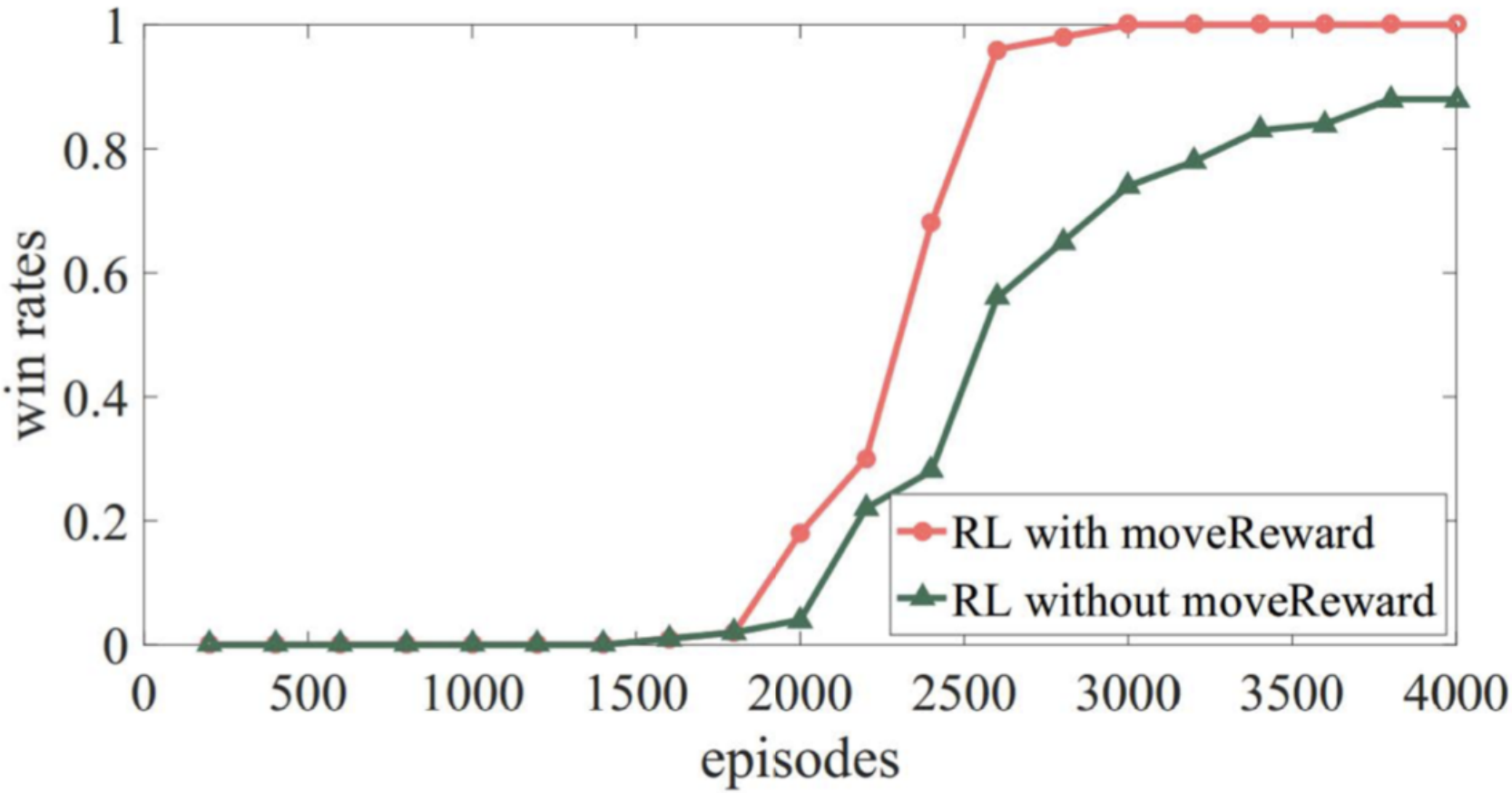
Parameters and hardware I

- $\alpha = 0.001$
- $\lambda = 0.8$
- $\gamma = 0.9$
- max steps per episode = 1000

Hardware: Intel i7-6700 CPU and 16GB of memory.

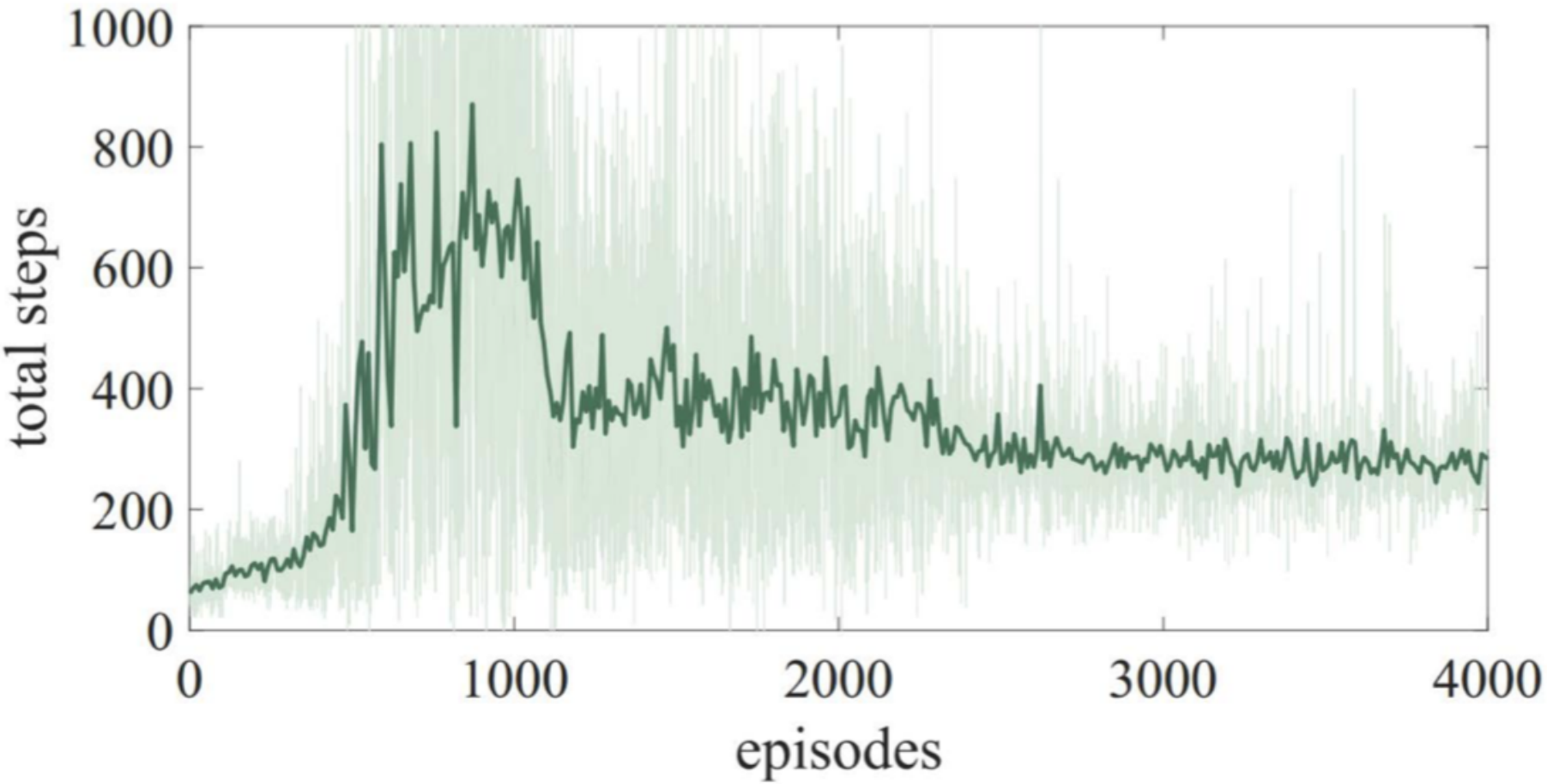
Plan

3 Goliaths vs 6 Zealoths I



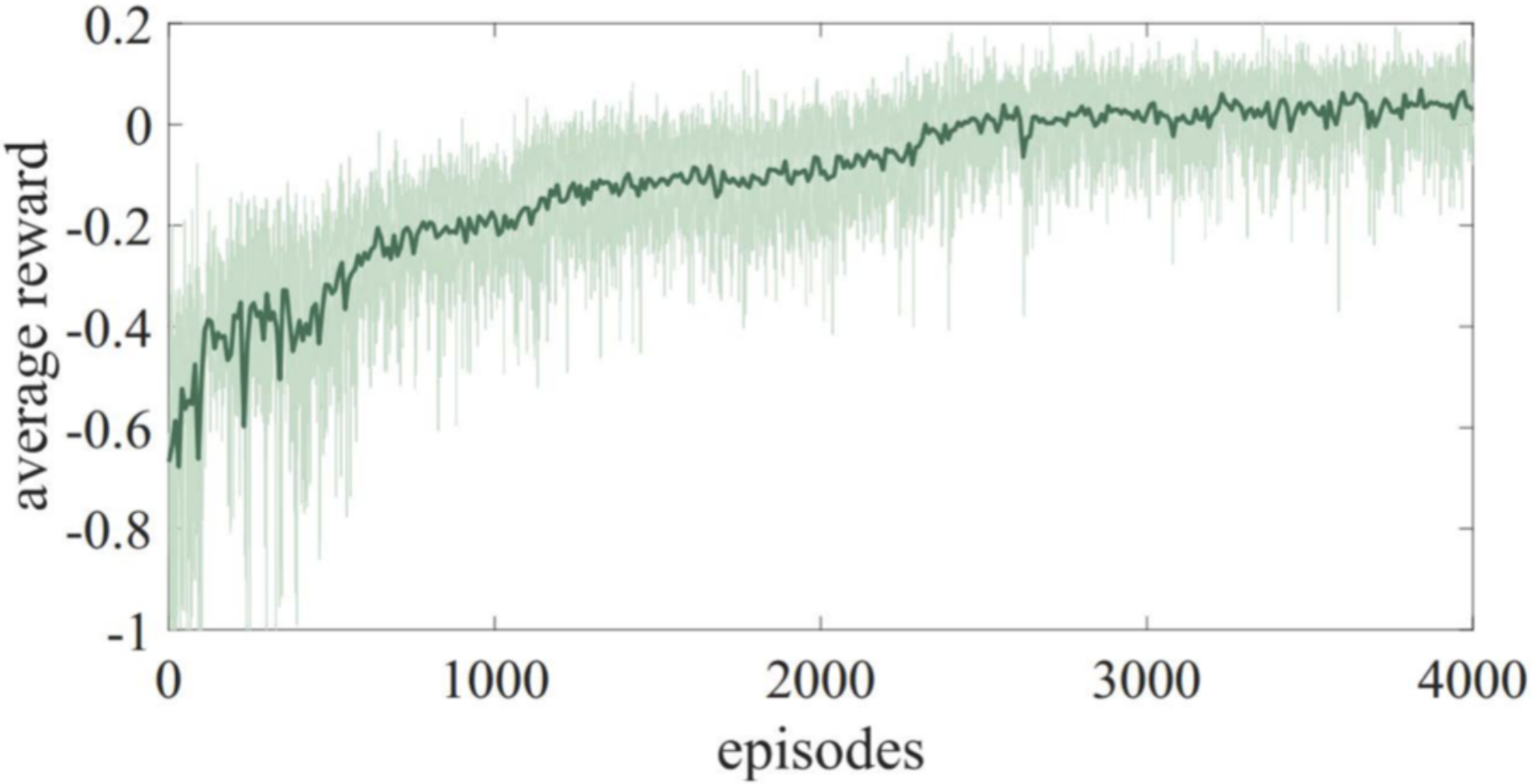
The win rates of our units in 3 Goliaths vs. 6 Zealots micromanage- ment scenario from every 200 episodes' training.

3 Goliaths vs 6 Zealoths II



The episode steps of our units in 3 Goliaths vs. 6 Zealots micromanagement scenario during training.

3 Goliaths vs 6 Zealoths III



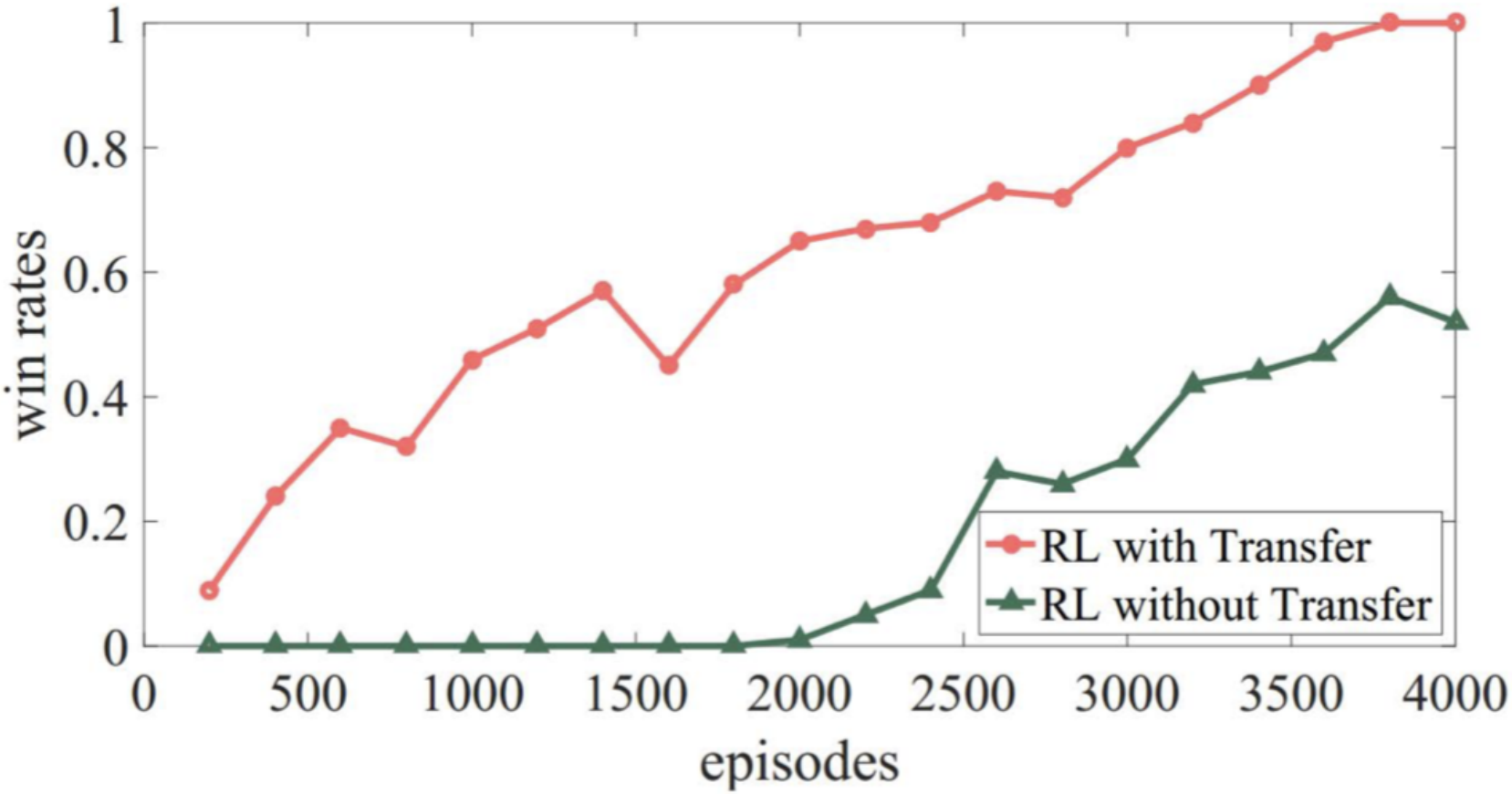
The average reward of our units in 3 Goliaths vs. 6 Zealots micromanagement scenario during training.

3 Goliaths vs 6 Zealoths IV

What we see:

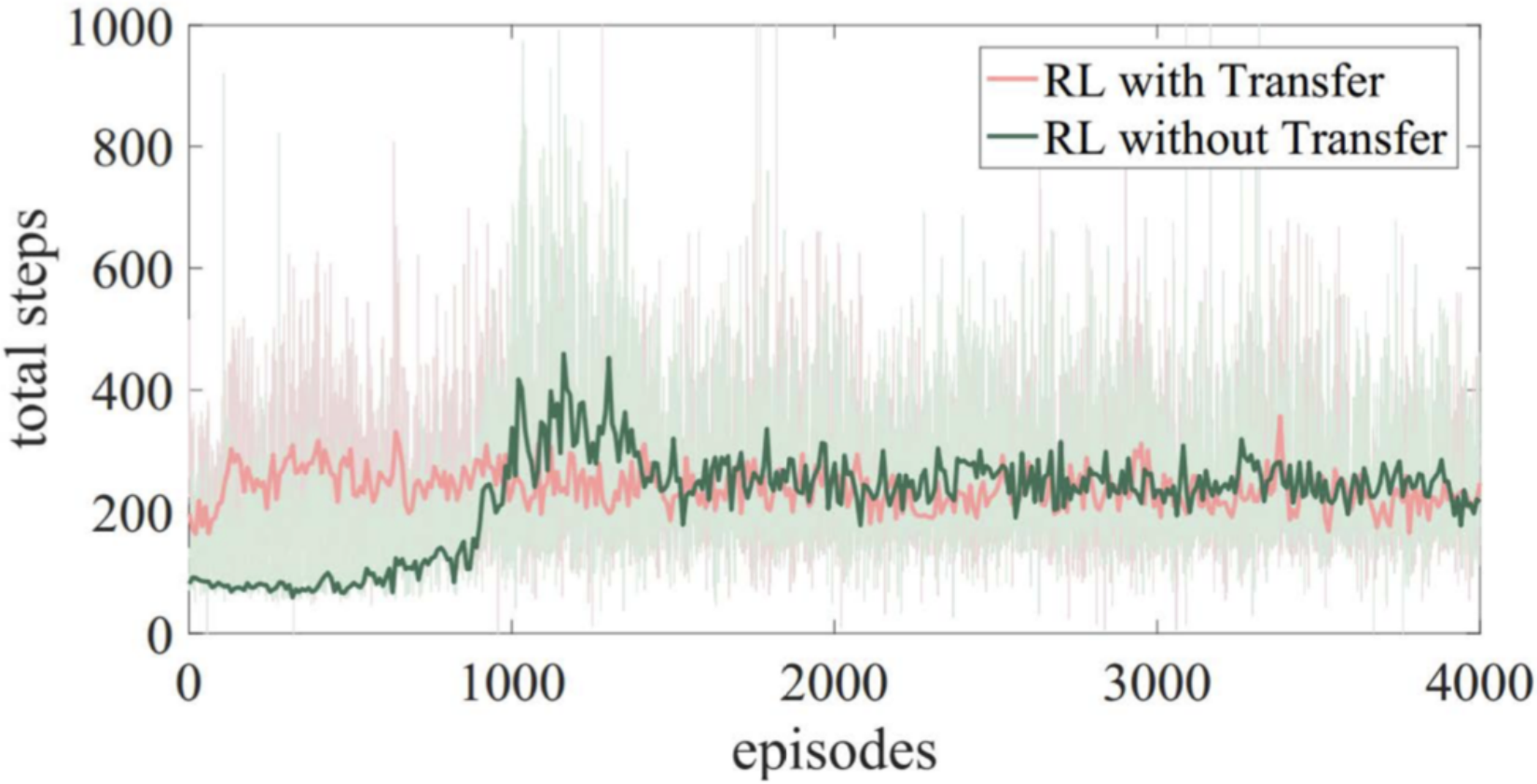
- moveReward is so helpful!
- we reach 100% winning games
- steps amount and reward curves seem natural

3 Goliaths vs 20 Zerglings I



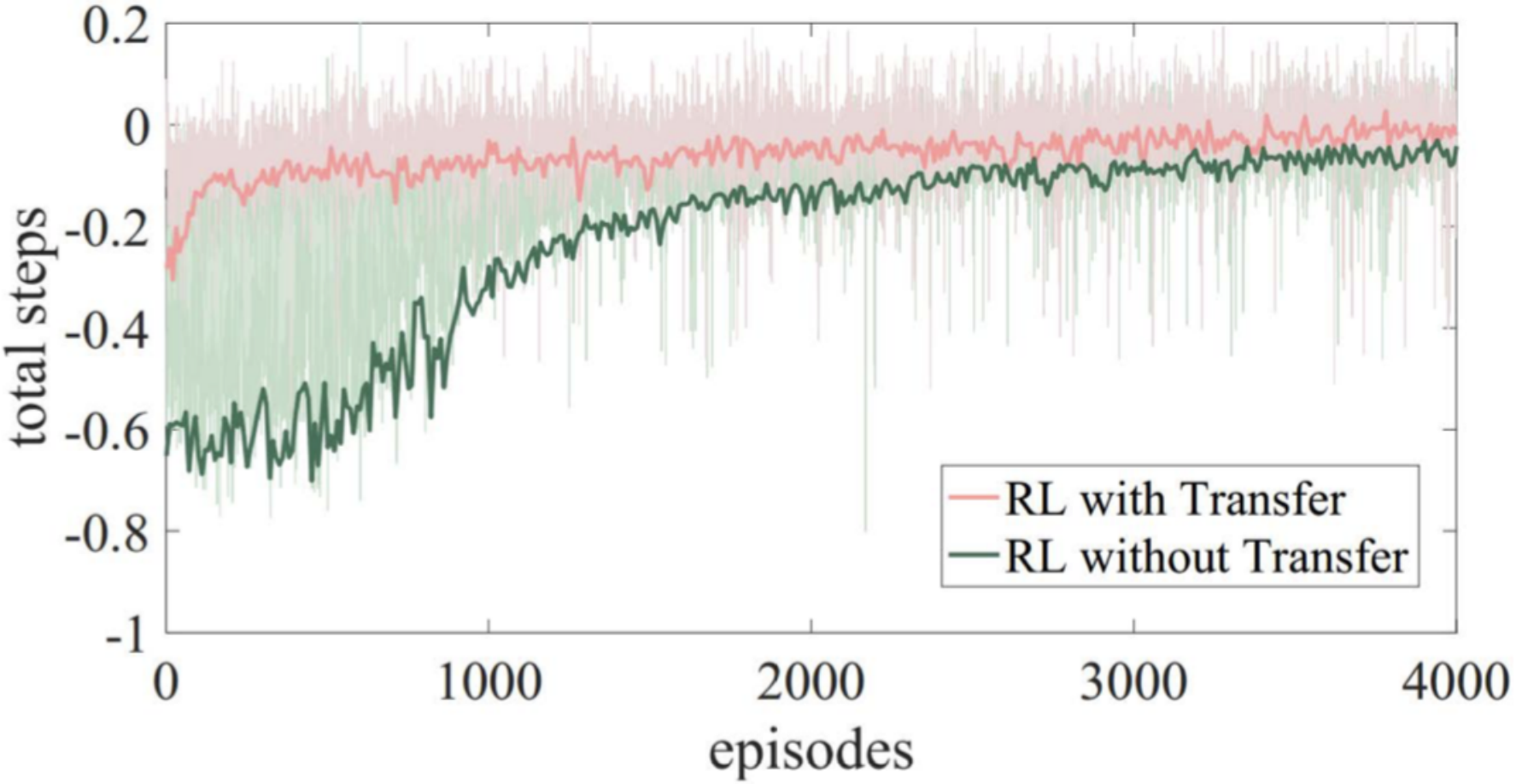
The win rates of our units in 3 Goliaths vs. 20 Zerglings micromanagement scenario from every 200 episodes' training.

3 Goliaths vs 20 Zerglings II



The average episode steps of our units in 3 Goliaths vs. 20 Zerglings micromanagement scenario during training.

3 Goliaths vs 20 Zerglings III



The average reward of our units in 3 Goliaths vs. 20 Zerglings micromanagement scenario during training.

3 Goliaths vs 20 Zerglings IV

What we see:

- Transfer Learning is so helpful!
- We again reach 100% winning games!
- With Transfer Learning model only needs to adapt to environment

Plan

Marines vs. Zerglings I

Setup:

Using curricilum transfer learning from small-scale scenarios [8] [9] [10]

Training curriculum:

Scenarios	Class 1	Class 2	Class 3
M10 vs. Z13	M5 vs. Z6	M8 vs. Z10	M8 vs. Z12
M20 vs. Z30	M10 vs. Z12	M15 vs. Z20	M20 vs. Z25

Marines vs. Zerglings II

Comparison with some baselines:

Scenarios	Weakest	Closest	GMEZO	BicNet	PS-MAGDS
M10 vs. Z13	23%	41%	57%	64%	**97%**
M20 vs. Z30	0%	87%	88.2%	**100%**	92.4%

- **Weakest**: A rule-based method, attacking weakest in the fire range.
- **Closest**: A rule-based method, attacking closest in the fire range.
- **GMEZO** [11] : A DRL method, based on the zero-order optimization, having impressive results over traditional RL methods
- **BicNet** [2]: A DRL method, based on the actor-critic archi- tecture, having the best performance in most StarCraft micromanagement scenarios

Marines vs. Zerglings III

Look more at generalization:

Well-trained Scenarios	Test Scenarios	Win rates
M10 vs. Z13	M5 vs. Z6	80.5%
	M8 vs. Z10	95%
	M8 vs. Z12	85%
	M10 vs. Z15	81%
M20 vs. Z30	M10 vs. Z12	99.4%
	M15 vs. Z20	98.2%
	M20 vs. Z25	99.8%
	M40 vs. Z60	80.5%

Marines vs. Zerglings IV

What we are **good** in:

- Good generalization
- Act wisely:
  - Disperse Enemies
  - Keep the Team
  - Hit and Run (like a human!)

Oh no, it's going to kill us all!!!

Marines vs. Zerglings V

Wait...we are still **bad** in:

- Units prefer moving as far as possible to avoid enemies
- Goliaths are bad in making teams
- It's just a PC game, not a real world =)

What we see here

- Very good perfomance in all the scenarios with built-in AI as enemy
- Good generalization potential
- Key idea: units' interaction with each other
- Many successful heuristics such as moveReward, state representation
- What if the opponent is human? We still have obvious weak points

Bibliography I


Shao K., Zhu Y., Zhao D. StarCraft "Micromanagement with Reinforcement Learning and Curriculum Transfer Learning"
arXiv preprint arXiv:1804.00810. – 2018


P. Peng, Q. Yuan, Y. Wen, Y. Yang, Z. Tang, H. Long, and J. Wang, "Multiagent bidirectionally-coordinated nets for learning to play StarCraft combat games"
arXiv preprint arXiv:1703.10069, 2017.


J. X. Wang, Z. Kurthnelson, D. Tirumala, H. Soyer, J. Z. Leibo, R. Munos, C. Blundell, D. Kumaran, and M. Botvinick, "Learning to reinforcement learn,"
International Conference on Learning Repre- sentations, 2017

Bibliography II


P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. J. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, and K. Kavukcuoglu, "Learning to navigate in complex environments
International Conference on Learning Repre- sentations, 2017


X. Glorot, A. Bordes, Y. Bengio, X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks
International Conference on Artificial Intelligence and Statistics, 2011, pp. 315–323.


V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines
International Conference on Machine Learn- ing, 2010, pp. 807–814.


S. P. Singh and R. S. Sutton, "Reinforcement learning with replacing eligibility traces
Machine Learning, vol. 22, no. 1, pp. 123–158, 1996.

Bibliography III


A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabskabarwiska, S. G. Colmenarejo, E. Grefenstette, T. Ramalho, and J. Agapiou, "Hybrid computing using a neural network with dynamic external memory.
Nature, vol. 538, no. 7626, p. 471, 2016.


Y. Wu and Y. Tian, "Training agent for first-person shooter game with actor-critic curriculum learning
International Conference on Learning Representations, 2017.


Q. Dong, S. Gong, and X. Zhu, "Multi-task curriculum transfer deep learning of clothing attributes
IEEE Winter Conference on Applica- tions of Computer Vision, 2017, pp. 520–529.

Bibliography IV


N. Usunier, G. Synnaeve, Z. Lin, and S. Chintala, "Episodic exploration for deep deterministic policies: An application to StarCraft micromanagement tasks
International Conference on Learning Representa- tions, 2017.