

# **Deep Reinforcement Learning Conditioned on the Natural Language**

by  
Yunqiu Xu

Thesis submitted in fulfilment of the requirements for the degree of  
Doctor of Philosophy  
under the supervision of Ling Chen and Chengqi Zhang

Australian Artificial Intelligence Institute  
Faculty of Engineering and Information Technology  
**University of Technology Sydney**

May 2022



# **Certification of Original Authorship**

I, Yunqiu Xu, declare that this thesis is submitted in fulfilment of the requirements for the award of degree of Doctor of Philosophy, in the Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution. This research is supported by the Australian Government Research Training Program.

Signed:

Production Note:

Signature removed prior to publication.

Date: 13/01/2022



# *Abstract*

Language-conditional reinforcement learning refers to the reinforcement learning task where the language information serves as essential components in the problem formulation. In recent years, the advances of deep reinforcement learning and language representation learning lead to increasing research interest in this cross-domain topic, which brings benefits to the studies in both language learning and reinforcement learning. However, challenges arise along with the premises, hindering the language-conditional reinforcement learning from being applied in the real world. In this research, we aim at designing language-conditional RL agent that is capable of handling the major challenges.

We first address the challenges in state representation learning under partial observability. Motivated by the premises of the transformer architecture in natural language processing, we design an adaptable transformer-based state representation generator featured with reordered layer normalization, weight sharing and block-wise aggregation. We empirically validate our method on both synthetic and man-made text-based games with different settings. The proposed method show higher sample efficiency in solving single synthetic games, better generalizability in solving unseen synthetic games, and better performance in solving complex man-made games.

Secondly, we study the reasoning process in language-conditional reinforcement learning. The reasoning ability enables the agent to generate the actions with the support of an explainable inference procedure. To achieve this ability, we propose an agent featured with the stacked hierarchical attention mechanism. Through exploiting the structure of the knowledge graph, this agent is able to explicitly model the reasoning process. Our agent demonstrates effectiveness on a range of man-made text-based games.

Thirdly, we study the generalization problem in language-conditional RL. We consider the knowledge graph-based observation, and address this challenge by designing a two-level hierarchical RL agent. In the high level, we use a meta-policy for task decomposition and subtask selection. Then, in the low level, we use a sub-policy for subtask-conditioned action selection. In a series of 8

game sets with different generalization types and game difficulty levels, our proposed agent enjoys generalizability and yields favorable performance.

Finally, we provide solutions to the challenges of low sample efficiency and large action space. We introduce the world-perceiving modules, which automatically decompose tasks and prune actions by answering questions about the environment. We then propose a two-phase training framework to decouple language learning from reinforcement learning, which further improves the sample efficiency. We empirically demonstrate that the proposed method not only achieves improved performance with high sample efficiency, but also exhibits robustness against compound error and limited pre-training data.

# *Acknowledgements*

First and foremost, I would like to express my sincere gratitude to my supervisors, Prof. Ling Chen and Prof. Chengqi Zhang for their continuous and unconditional assistance throughout the journey in pursuing this degree. They supported me at every stage of my PhD studies, such as leading me to the area of language-conditional reinforcement learning, encouraging me to develop not only the in-depth knowledge in my research topic but also broad research interests in machine learning, helping me to establish connections with academic and industrial bodies, and providing me with valuable career & life advice. Without your patience, encouragement and persistent help, I will never grow into a professional researcher.

Besides my supervisors, I would like to offer my special thanks to Dr. Meng Fang, for providing me with hand in hand guidance to deliver high quality research. Thank you for aiding me to formulate a project management-like research schedule. I benefit from the close and inspiring discussions, which cover almost every aspect of details about how to conduct a research project, such as the problem formulation, model development, experiment design, and the manuscript preparation. Without your help, I probably waste a lot of time on detours.

I would like to acknowledge Dr. Yali Du, Dr. Gangyan Xu, Dr. Joey Tianyi Zhou, Dr. Yang Wang, Dr. Binbin Huang, for providing helpful suggestions throughout our collaboration. I would also like to mention those I met at UTS, specifically Dr. Wei Wu, Dr. Hong Yang, Dr. Yaqiong Li, Dr. Jiamiao Wang, Dr. Jun Li, Mr. Yu Liu, Mr. Shaoshen Wang, Mr. Yayong Li, Ms. Yang Zhang, and Mr. Zihan Zhang. It's my pleasure to work with all of you. In particular, I would like to thank Prof. Yi Yang, for introducing me to Prof. Ling Chen.

I am proud of my parents and grandparents. Thank you for raising me up, providing me with a joyful childhood, and inspiring my curiosity in science and engineering. Thank you for expressing your endless love and support across the ocean. This thesis is dedicated to my grandfather, a veteran passed away in the last year of my PhD study. Thank you, for fostering me a rigorous attitude, and teaching me to be tough to survive hardships.

Last but not the least, I would like to thank my wife, Ms. Yang Liu, for accompanying me through thick and thin.





# Contents

<b>Declaration of Authorship</b>	<b>iii</b>
<b>Abstract</b>	<b>v</b>
<b>Acknowledgements</b>	<b>vii</b>
<b>List of Figures</b>	<b>xiii</b>
<b>List of Tables</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background and Motivations . . . . .	1
1.2 Research Challenges . . . . .	3
1.3 Contributions . . . . .	4
1.4 Thesis Outline . . . . .	5
1.5 Publications . . . . .	6
<b>2 Literature Review</b>	<b>9</b>
2.1 Reinforcement Learning . . . . .	9
2.1.1 Value-based RL . . . . .	11
2.1.2 Policy-based RL . . . . .	12
2.1.3 Model-based RL . . . . .	13
2.2 Language-conditional RL . . . . .	14
2.2.1 Language-based Observation and Action Space . . . . .	14
2.2.2 Language-based Instruction Following and Reward Function . . . . .	17
2.3 DRL’s Real World Applications and Challenges . . . . .	18
<b>3 Preliminaries</b>	<b>21</b>
3.1 Text-based Games . . . . .	21
3.2 Partially Observable Markov Decision Process . . . . .	22
3.3 Knowledge Graph . . . . .	23
<b>4 Transformer-based State Representation Generator</b>	<b>25</b>
4.1 Introduction . . . . .	25

4.2	Related Work . . . . .	27
4.3	Methodology . . . . .	27
4.3.1	Layer Normalization . . . . .	29
4.3.2	Weight Sharing . . . . .	30
4.3.3	Block-wise Gate Layer . . . . .	30
4.4	Experiments . . . . .	30
4.4.1	Experiment Setting . . . . .	30
4.4.2	Baselines . . . . .	31
4.4.3	Implementation Details . . . . .	32
4.4.4	Training Details . . . . .	32
4.4.5	Evaluation Metrics . . . . .	33
4.5	Results and Discussions . . . . .	33
4.5.1	Synthetic Games: Single Game Setting . . . . .	33
4.5.2	Synthetic Games: Multiple Unseen Games Setting . . . . .	35
4.5.3	Jericho-supported Games . . . . .	36
4.6	Conclusion . . . . .	36
<b>5</b>	<b>Stacked Hierarchical Attention Mechanism</b>	<b>39</b>
5.1	Introduction . . . . .	39
5.2	Related Work . . . . .	40
5.2.1	Reinforcement Learning Agents for Solving Text-based Games . . . . .	40
5.2.2	Attention Mechanism . . . . .	41
5.2.3	Reasoning upon the KGs . . . . .	41
5.3	Problem Statement . . . . .	41
5.4	Methodology . . . . .	43
5.4.1	Sub-graph Division . . . . .	43
5.4.2	Stacked Hierarchical Attention . . . . .	43
5.4.3	Action Selection and Model Optimization . . . . .	46
5.5	Experiments . . . . .	46
5.5.1	Baselines . . . . .	47
5.5.2	Experimental Setup . . . . .	47
5.6	Results and Discussions . . . . .	48
5.6.1	Main Results . . . . .	48
5.6.2	Ablation Studies . . . . .	50
5.6.3	Interpretability . . . . .	51
5.7	Conclusion . . . . .	52
<b>6</b>	<b>Hierarchical Knowledge Graph Agent</b>	<b>53</b>
6.1	Introduction . . . . .	53
6.2	Related Work . . . . .	55
6.2.1	Generalization in Text-based Games . . . . .	55
6.2.2	Hierarchical Reinforcement Learning . . . . .	55
6.3	Problem Statement . . . . .	56
6.4	Methodology . . . . .	57

6.4.1	Overview . . . . .	57
6.4.2	Meta-policy . . . . .	58
6.4.3	Sub-policy . . . . .	59
6.4.4	Training Strategies . . . . .	60
6.5	Experiments . . . . .	62
6.5.1	Experiment Setting . . . . .	62
6.5.2	Baselines . . . . .	63
6.5.3	Implementation and Training Details . . . . .	64
6.5.4	Evaluation Metrics . . . . .	65
6.6	Results and Discussions . . . . .	67
6.6.1	Main Results . . . . .	67
6.6.2	The Influence of the BeBold Method . . . . .	68
6.6.3	The Influence of the MTL Training Techniques . . . . .	69
6.7	Conclusion . . . . .	70
<b>7</b>	<b>Question-guided World-perceiving Agent</b>	<b>71</b>
7.1	Introduction . . . . .	71
7.2	Related Work . . . . .	73
7.2.1	Hierarchical RL . . . . .	73
7.2.2	Pre-training Methods for RL . . . . .	74
7.3	Problem Statement . . . . .	75
7.4	Methodology . . . . .	76
7.4.1	Framework Overview . . . . .	76
7.4.2	Task Selector . . . . .	77
7.4.3	Action Validator . . . . .	78
7.4.4	Action Selector . . . . .	78
7.5	Experiments . . . . .	79
7.5.1	Experiment Settings . . . . .	79
7.5.2	Baselines . . . . .	80
7.5.3	Implementation Details . . . . .	80
7.5.4	Evaluation Metrics . . . . .	81
7.6	Results and Discussions . . . . .	81
7.6.1	Main Results . . . . .	81
7.6.2	Performance on the Simple Games . . . . .	83
7.6.3	Ablation Study . . . . .	83
7.6.4	Pre-training on the Partial Dataset . . . . .	85
7.7	Conclusion . . . . .	85
<b>8</b>	<b>Conclusions and Future Work</b>	<b>87</b>
8.1	Conclusions . . . . .	87
8.2	Future Work . . . . .	88

<b>A Appendix for Chapter 6</b>	<b>91</b>
<b>B Appendix for Chapter 7</b>	<b>95</b>
B.1 Environment . . . . .	95
B.2 Pre-training Datasets . . . . .	100
B.3 Baseline details . . . . .	102
B.3.1 GATA . . . . .	102
B.3.2 IL . . . . .	103
B.4 More experimental results . . . . .	104
 <b>Bibliography</b>	 <b>109</b>

# List of Figures

2.1	The interaction process in the RL problem . . . . .	10
2.2	The schema of different RL algorithms. . . . .	11
3.1	The interface example of a man-made game “Zork: The Undiscovered Underground”. The observation (black) can be the description of environment status, or the feedback of the previous action. The action (blue) is a textual command. . . . .	22
4.1	Our agent, which contains a representation generator for constructing representation from $s_t$ , and an action scorer for computing the Q values. We concentrate on the representation generator, which is shown in the dashed box. . . . .	28
4.2	(a) Our transformer-based representation generator. (b) The transformer encoder, which consists of $L$ transformer blocks. In each transformer block, we reorder the layer normalization operation by putting it inside the residual connections. We share the learnable weights across all transformer blocks (blue region). We then propose gate modules to aggregate the input flow and output flow of each transformer block. . . . .	28
4.3	The vanilla transformer block (a) and the proposed transformer block (b), where the layer normalization operation is put inside each residual connection. . . . .	29
4.4	The models’ performance in synthetic games under the single game setting. The shaded area indicates standard deviations over 3 independent runs. . . . .	34
4.5	The models’ performance in synthetic games under the multiple unseen games setting. . . . .	35
4.6	The models’ learning curves on the game “zork1”. . . . .	37
5.1	(a) $o_{t,\text{text}}$ in our work, which consists of four parts. (b) The KG-based observation $o_{t,\text{KG}}$ , where those derived from current $o_{t,\text{text}}$ are in yellow. (c) The sub-graphs. . . . .	42
5.2	Overview of the SHA-KG’s encoder. . . . .	44
5.3	The result with respect to the update steps for SHA-KG variants with different encoding methods. . . . .	49
5.4	The result with respect to the update steps for SHA-KG variants with different sub-graphs. . . . .	49
5.5	An example of the reasoning process for game “ztuu”. . . . .	51
6.1	An overview of the proposed H-KGA, where the high level decision making process is in red (goal set generation and goal selection), and the low level decision making process is in blue (action selection). . . . .	57
6.2	The performance of agents on $\mathcal{D}_{\text{test}}^{\text{seen}}$ (“S4”, “Avg Seen”) and $\mathcal{D}_{\text{test}}^{\text{unseen}}$ (“US4”, “Avg Unseen”). . . . .	66
6.3	The performance of agents with or without the BeBold method. . . . .	68

6.4	The performance of agents with or without the scheduled task sampling strategy (Sch) / level-aware replay buffer (LR). . . . .	69
7.1	(a) An example of the observation, which can be textual, KG-based, or hybrid. (b) The decision making process. Through question answering, the agent is guided to first decompose the task as subtasks, then reduce the action space conditioned on the subtask. . . . .	73
7.2	Subtasks for solving (a) 3 simple games and (b) 1 complex game. . . . .	75
7.3	The overview of QWA. The blue modules will be trained in the pre-training phase, while the red module will be trained in the RL phase. . . . .	76
7.4	The RL testing performance w.r.t. training episodes. The red dashed line denotes the IL agent without fine-tuning. . . . .	82
7.5	The performance of our model and the variant without time-awareness. . . . .	84
7.6	The performance of our model and the variants with expert modules. . . . .	84
7.7	The performance of our model with varying amounts of pre-training data. . . . .	85
A.1	The initial observation of four games, where “S1 Game1” and “S1 Game2” belong to level “S1”, “S2 Game1” and “S2 Game2” belong to level “S2”. . . . .	92
A.2	The initial observation of two games belonging to level “S3”. . . . .	93
A.3	The initial observation of one game belonging to level “S4”. . . . .	94
B.1	The construction process of the subtask set $\mathcal{T}$ , and the pre-training dataset for task decomposition. . . . .	101
B.2	The construction process of the pre-training dataset for action pruning. . . . .	101
B.3	The architecture of GATA baseline. . . . .	102
B.4	The architecture of GATA for action prediction. . . . .	102
B.5	The architecture of IL baseline. . . . .	103
B.6	The pre-training performance of QWA’s task selector. The results are averaged by 3 random seeds, we omit the standard deviation as the performance is relatively stable. . . . .	105
B.7	The pre-training performance of QWA’s action validator. . . . .	105
B.8	The pre-training performance of IL’s task selector and action selector. . . . .	105
B.9	The RL performance of our GATA baseline and the original GATA without AP initialization. . . . .	106
B.10	The RL performance w.r.t. the training episodes (the full result of Fig. 7.4). . . . .	106
B.11	The RL performance of our agent and the variant without time-awareness (the full result of Fig. 7.5). . . . .	107
B.12	The performance of our agent and the variants with expert modules (the full result of Fig. 7.6). . . . .	107
B.13	The performance of our agent with varying amounts of pre-training data (the full result of Fig. 7.7). . . . .	108

# List of Tables

1.1	Thesis structure . . . . .	6
2.1	The DRL agents for text-based games. . . . .	15
4.1	The details of the synthetic games in the CoinCollector domain. . . . .	31
4.2	The number of epochs required to solve the single games. . . . .	34
4.3	The performance of models with different modifications. . . . .	34
4.4	The maximum rewards obtained in synthetic games under the multiple unseen games setting. . . . .	35
4.5	The performance of models on man-made games. . . . .	36
5.1	The main result in 20 games. . . . .	48
6.1	The game statistics. . . . .	63
6.2	The testing result at the end of the training process. . . . .	66
7.1	Game statistics. We use the simple games to provide human labeled data during pre-training, and use the medium & hard games during reinforcement learning. . . . .	79
7.2	The testing performance at 20% / 100% of the reinforcement learning phase. . . . .	81
7.3	The RL testing performance on simple games. . . . .	83
B.1	The observations $o_t$ , subtask candidates $\mathcal{T}$ and action candidates $\mathcal{A}$ of a simple game and a medium game. The underlined subtask candidates denote the available subtask set $\mathcal{T}_t$ . . . . .	96
B.2	The observations $o_t$ , subtask candidates $\mathcal{T}$ and action candidates $\mathcal{A}$ of a hard game. The underlined subtask candidates denote the available subtask set $\mathcal{T}_t$ . The underlined action candidates denote the refined action set $\mathcal{A}_t$ after selecting the subtask “roast carrot”. . . . .	97
B.3	Examples of subtasks. . . . .	98
B.4	Examples of actions. . . . .	99

