Evolutionary transfer learning for complex multi-agent reinforcement learning systems

Hou, Yaqing

2017

Hou, Y. (2017). Evolutionary transfer learning for complex multi-agent reinforcement learning systems. Doctoral thesis, Nanyang Technological University, Singapore.

<http://hdl.handle.net/10356/72997>

<https://doi.org/10.32657/10356/72997>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

**EVOLUTIONARY TRANSFER LEARNING FOR
COMPLEX MULTI-AGENT REINFORCEMENT
LEARNING SYSTEMS**

HOU YAQING

INTERDISCIPLINARY GRADUATE SCHOOL

2017

**EVOLUTIONARY TRANSFER LEARNING FOR
COMPLEX MULTI-AGENT REINFORCEMENT
LEARNING SYSTEMS**

HOU YAQING

Interdisciplinary Graduate School

A thesis submitted to Nanyang Technological University
in fulfillment of the requirements of degree of
Doctor of Philosophy

2017

Abstract

Multi-agent systems (MAS) are computerized systems composing of multiple interacting and autonomous agents within a common environment of interest for problem-solving. Since the behavioral strategies of agents in conventional MAS are usually manually defined in advance, the development of intelligent agents that are capable of adapting to the dynamic environment has attracted increasing attentions in the past decades.

In the past decades, reinforcement learning (RL) has been introduced into MAS as the learning paradigm of individual agents through trial-and-error interactions in a dynamic environment. Due to its generality and simplicity of use, the study of RL is rapidly expanding and a wide variety of approaches has already been proposed to exploit its benefits and applicabilities. A more recent machine learning paradigm of transfer learning (TL) has been proposed as an approach of leveraging valuable knowledge from related and well studied problem domains to enhance problem-solving in the target domains of interest. TL has been successfully used for enhancing RL tasks and many TL methodologies, such as instance or feature transfer, have been explored. Recently, the research on TL has been considered for enhancing multi-agent RL methods.

In the context of computational intelligence, the science of memetics, especially memetic computation, has become an increasing topic of research. Many of the existing work in memetic computation has been established as an extension of the classical evolutionary algorithms where a meme is perceived as a form of individual learning procedure or local search operator in population based search algorithms. However, recent research shows memetic computation can become more meme-centric wherein memes transpire as units of domain information or knowledge building blocks useful for problem-solving. The intrinsic parallelism of natural evolution in such meme-centric computing may derive its strength from the simultaneous explorations of differing regions of a common problem domain and exploitative social interactions between the multiple agent learners. This dissertation hence seeks for the new study on

a meme centric evolutionary knowledge transfer paradigm for problem-solving of multi-agent reinforcement learning systems.

Specifically, this thesis presents an evolutionary transfer learning framework (eTL) which comprises a series of meme-inspired evolutionary knowledge representation and transfer mechanisms. The proposed framework serves to enhance the learning capabilities while addresses the limitations (e.g. blind reliance) of existing knowledge transfer frameworks. Subsequently, a novel transfer learning framework with predictive capabilities (eTL-P) is proposed to cope with the challenges arising in complex multi-agent systems where agents have differing or even competitive objectives. eTL-P endows agents with abilities to interact with competitive opponents, modeling their opponents, and hence predicting their behaviors accordingly. Further, to reduce the complexity of the opponent candidate models, a Top- K model selection method is proposed for selecting a smaller yet remarkably representative candidate model set from the entire model space. Last but not least, a summary of the future study on the evolutionary knowledge transfer paradigm is presented.

Acknowledgments

Firstly, I would like to express my deepest gratitude to my supervisor **Prof. Ong Yew-Soon**. His sincere encouragement during my indecisive moments and inspirational insights into the many aspects of my research direction have faithfully guided me throughout my research work. Without Prof. Ong's guidance, the fruitful achievement of this project will never be possible.

Moreover, I would like to extend my thanks to my seniors, Dr. Liang Feng, Dr. Yiteng Zhai, Dr. Abhishek Gupta and Dr. Siwei Jiang for their kind help along the way ever since the beginning of my PhD program. Working and living with them inspires my research and makes the Ph.D. life lively and interesting. My sincere appreciation also goes to the colleagues and staffs in Multi-plAtform Game Innovation Centre (MAGIC) for their technical and financial supports.

Last but not least, special thanks to my family and girlfriend for their deepest love and everlasting encouragement throughout my PhD candidature thus far.

Contents

Abstract	ii
Acknowledgments	iv
List of Figures	viii
List of Tables	x
1 Introduction	1
1.1 Research Background	1
1.2 Research Objectives	3
1.3 Research Contributions	4
1.4 Organization of the Thesis	6
2 Literature Review	9
2.1 Multi-Agent System	9
2.2 Reinforcement Learning for Individual Learning in MAS	11
2.3 Transfer Learning in Multi-agent Reinforcement Learning Systems	13
2.3.1 Interactive Advice Exchange Mechanism	16
2.3.2 Parallel Transfer Learning	17
2.3.3 Memetic Multi-Agent System	18
3 An Evolutionary Transfer Reinforcement Learning Framework	21
3.1 Evolutionary Memetic Multi-Agent Transfer Reinforcement Learning System	23
3.1.1 Memetic Science	23
3.1.2 Proposed eTL Model	24
3.2 Realization of eTL with Learning Agents	26
3.2.1 FALCON Learning Agent	26

3.2.2	BP Learning Agent	30
3.2.3	Meme Internal Evolution	32
3.2.4	Meme External Evolution	34
3.2.5	eTL with FALCON and BP	39
3.3	Empirical Study	42
3.3.1	Experimental Platform - Minefield Navigation Task	42
3.3.2	Experimental Configuration	43
3.3.3	Performance of eTL using FALCON or BP as Learning Agents.	44
3.3.4	Performance of Proposed eTL using Heterogeneous Agents	49
3.3.5	Performance of Proposed eTL in Unreal Tournament 2004	53
3.4	Summary	58
4	Evolutionary Transfer Learning with Behavior Prediction	59
4.1	Proposed eTL-P Framework	61
4.1.1	Objectives of Multiple Agents in Competitive MAS	61
4.1.2	Model-based Behavior Prediction	62
4.1.3	Multi-agent Transfer Learning	65
4.1.4	Complexity Analysis	67
4.2	Empirical Study and Analyses	68
4.2.1	Experimental Platform and Configuration	68
4.2.2	Effectiveness of eTL	70
4.2.3	Results of eTL-P with True Predator Model	71
4.2.4	Results of eTL-P with Predictive Predator Models of FALCON Dynamics	74
4.2.5	Results of eTL-P with Predictive Candidates of Heterogeneous RL Structures	78
4.3	Summary	79
5	Top-K Model Selection Method in eTL-P	81
5.1	Top- K Model Selection	82
5.1.1	Coverage	85
5.1.2	Diversity	87
5.2	Complexity Analysis	88

5.3	Empirical Study	89
5.4	Summary	93
6	Conclusions and Future Work	94
6.1	Summary of Contributions	94
6.2	Future Work	97
6.2.1	Extensive Study for the Evolutionary Transfer Learning Framework . .	97
6.2.2	Evolutionary Transfer Learning across Problem Domains	98
6.2.3	Evolutionary Knowledge Transfer for Deep Reinforcement Learners . .	99
	References	101

List of Figures

2.1	An illustration of the multi-agent system. (Agents may have differing goals, action space or domain knowledge, which are denoted by different fonts. And the arrow between the agents indicates the possible communication between agents.)	10
2.2	Illustration of a standard reinforcement learning process.	12
2.3	The architecture of Memetic Multi-Agent System.	19
3.1	Illustration of the evolutionary multi-agent Transfer reinforcement Learning framework.	25
3.2	Illustration of <i>meme representation</i> taking the form of FALCON and BP.	27
3.3	The fully general multi-agent scenario with the proposed eTL paradigms. Agents employ FALCON or BP as the learning machine in their mind universes and learn the domain knowledge for performing appropriate actions given environmental states. They may also interact directly as denoted by the arrows connected between agents.	41
3.4	The illustration of MNT.	43
3.5	SRs of agents under AE models with discount factor 0.1, 0.5 and 0.9 on completing the missions in MNTs.	46
3.6	SRs of FALCON or BP agents under eTL, PTL, AE-AVG and Conv. M on completing the missions in MNT.	47

3.7	Snapshots of the FALCON agents' navigation routes on completing 1, 500, 1000 and 2000 learning missions in Minefield Navigation Task. The objective of each FALCON agent (denoted as a tank in the figure) in a learning mission is to arrive at the target successfully by navigating across the minefield safely and within the allocated time span. Thus a learning mission completes when all tanks reach the target, hit a mine or collide with another tank, or exceed the given time steps. All tanks and mines are randomly generated for each learning mission hence different learning missions have unique navigational routes. . . .	48
3.8	SRs of FALCON and BP agents under eTL, PTL, AE-AVG and Conv. M on completing the missions in the heterogeneous MAS.	51
3.9	SRs of FALCON and BP agents under AE models with discount factor 0.1, 0.5 and 0.9 on completing the missions in the heterogeneous MAS.	52
3.10	Pogamut architecture.	54
3.11	A snapshot of the "Unreal Tournament 2004".	55
3.12	KR of FALCON robots under the Conv. M and the proposed eTL fighting against the rule-based <i>Hunters</i>	57
4.1	An illustration of the proposed eTL-P.	62
4.2	Illustration of the adapted MNT (PO-MNT).	69
4.3	SRs of navigators under eTL, PTL, AE and Conv-MAS on completing the missions in classical MNTs.	71
4.4	SRs of eTL, Mod-P, eTL-P and Conv-MAS on completing the missions in PO-MNTs.	73
4.5	SRs of Mod-P, eTL-P and Mod-All on completing the missions in PO-MNTs. .	76
4.6	SRs of Mod-FALCON, Mod-MLP, Mod-No and Mod-Avg on completing the missions in PO-MNTs.	79
5.1	An illustration of eTL-P with the proposed Top- K model selection method. . .	83
5.2	SRs of navigators with 99 predictive candidate models on completing the missions in PO-MNTs.	91
5.3	SRs of MS-C and MS-CD over computational cost ([MT+PT]) on completing the missions in PO-MNTs.	93

List of Tables

3.1	Summary of the parameter configuration in the proposed eTL	44
3.2	Performance comparison among eTL, existing TL approaches and Conv. M (SR: success rate, GM: generated memotypes, KN: knowledge transfer numbers)	46
3.3	The effectiveness of generated memotypes in FALCON agents under the eTL and other existing TL approaches	49
3.4	Comparison among eTL, existing TL approaches and Conv. M in heteroge- neous MAS (SR: success rate, GM: generated memotypes, KN: knowledge transfer numbers).	50
3.5	Sample Knowledge of the Autonomous Robots	56
4.1	Parameter setting in the proposed eTL-P.	70
4.2	Performance comparison among Mod-P, eTL-P and Mod-All in terms of SR. . .	75
4.3	Computational cost taken by eTL-P and Mod-All (MT: candidate models train- ing time; PN: behavior prediction numbers; PT: behavior predicting time). . . .	77
5.1	Performance comparison among Mod-P, eTL-P and Mod-All with Top- K mod- el selection scheme in terms of SR (SR: success rate).	90
5.2	Performance comparison among Mod-P with MS-C and MS-CD (MT: can- didate models training time; PN: behavior prediction numbers; PT: behavior predicting time).	92

Chapter 1

Introduction

1.1 Research Background

Multi-agent system (MAS) comprises multiple interacting and autonomous agents that share a common environment of interest for problem-solving [1][2]. In the last decades, MAS has achieved significant success across a wide variety of real-world problem domains stretching from robotics [3][4][5], finance [6] to economics [7][8], and has attracted significant attention. Conventionally, the actions of each agent are usually pre-defined for the states that would be encountered in the environment. Nevertheless, this approach becomes unreliable since the space of environmental states is enormous and hence manually defining these action-state mappings is infeasible. Moreover, in a dynamic environment that changes over time, a hardwired behavioral strategy may become inappropriate. Thus it is necessary to endow the agents with intelligence capable of adapting to the dynamic environment.

In the literature, reinforcement learning (RL) has been proposed as the learning mechanism of individual agents through trial-and-error interactions in a dynamic environment [9][10]. The multi-agent RL field is rapidly expanding and a wide variety of approaches to exploit its benefits and applicabilities have been developed due to its simplicity and generality of use over the last decades [11][12]. These approaches integrate developments from research fields such as game theory [13][14], single-agent RL [15] and direct policy optimization [16]. A more recent machine learning paradigm of transfer learning (TL) has also been proposed as an approach

of leveraging valuable knowledge from related and well studied problem domains to enhance problem-solving in the target domains of interest [17]. TL has been successfully used for enhancing RL tasks and many TL methodologies, such as instance transfer [18], rule transfer [19] or feature transfer [20], have been explored. Recently, research on TL has also been considered for solving multi-agent RL tasks such as Traffic Management [21], Minefield Navigation task [22] and Smart-Grid problem [23].

In the context of computational intelligence, the meme-inspired computing methodology, or more concisely memetic computation, has attracted increasing research attention recently [24]. Memetic computation is a paradigm wherein the notion of meme(s) is used as the basic information unit encoded in computational representations for the purpose of problem solving. Many of the existing works in memetic computation have been established as an extension of the classical evolutionary algorithms, taking the form of hybrid [25], adaptive hybrid [26][27] or Memetic Algorithm (MA) [28], where a meme is perceived as a form of individual learning procedure or local search operator to improve the capability of population based search algorithms. However, falling back on the fundamental definition of a meme by Dawkins and Blackmore [29][30] (i.e., as the fundamental building blocks of culture evolution), the potential merits or true nature of memes remains yet to be fully exploited. Beyond the simple and adaptive hybrids of memetic algorithms, the current research of memetic computation further culminates into a meme-centric “memetic automaton” [24].

More specifically, meme-centric memetic automaton has been proposed as an adaptive entity or a software agent that is self-contained where memes are defined as units of information or building blocks of knowledge. This software agent is capable of interacting with its surroundings and adapting to complex environment by evolving the memes. In contrast to existing methods on knowledge transfer in the multi-agent RL systems, memetic automaton involves the additional aspects of cultural evolution taking inspiration from Darwin’s theory of natural selection and Darwin’s Universal Darwinism. Specifically, the essential backbone of a memetic automaton comprises a series of meme-inspired cultural evolution mechanisms including

memetic transmission, selection, replication, or variation. Due to the physiological limits of agents' ability to perceive differences, small amount of errors will be introduced. These errors could generate the "growth" and/or "variation" of agents' knowledge that learnt from the domain [31], hence exhibit higher adaptivity capabilities on solving complex problems. Further, memes are naturally defined as building blocks of agents' knowledge that have transpired in the form of recurring real-world patterns and represented by the neuron [32], tree [33] or graph [34] structures. These meme patterns provide high-level knowledge representations of common problems, hence enable reuse across problem domains and accommodate the heterogeneity of agents of differing learning architectures.

Therefore, beyond the concept of memetic automaton and taking memes as the focal point of interest in the context of MAS, this dissertation aims to explore the new manifestation of memes and further introduce a potentially rich meme-inspired multi-agent learning framework for more effective problem solving. In the design, memes are formalized as ideas or knowledge captured as memory items that fill mind universe of an agent and are capable of being transmitted among multiple agents. Beyond the typical notion of TL and multi-agent RL, the particular emphasis in this study is placed on leveraging social interactions across multiple agents through the principles of cultural evolution, such as memetic transmission, selection, replication, imitation, or variation.

1.2 Research Objectives

In order to enhance the learning capabilities of the existing TL methods in multi-agent RL systems, this research explores the concept of memetics in the context of computational intelligence. Specifically, this dissertation focuses on the study of an evolutionary transfer learning (eTL) framework which comprises a series of meme-inspired evolutionary knowledge transfer mechanisms. To showcase the generality of the proposed framework, both cooperative and

competitive MAS with eTL are studied and presented in this thesis. The core research objectives are detailed in what follows.

- Despite the success of TL in RL tasks, existing approaches rarely focus on knowledge transfer among multiple agents as the learning progresses on-line. Only a few recent studies including Parallel Transfer Learning [23] and Advice Exchange [21] have been considered to date. Even so, the current knowledge transfer process of existing frameworks tends to suffer from issues such as blind reliance. Thus, one objective of this research is to study and explore novel evolutionary transfer learning frameworks for MAS so as to benefit from the nature inspired meme-centric computing paradigm.
- In MAS, multiple agents may have differing domain knowledge and goals for the problem of interest. However, existing studies on multi-agent TL approaches have focused on simple multi-agent scenarios where all agents have the same or share similar action spaces, and pursue a common objective. In multi-agent cases where each agent has unique objectives, the learning process of multiple agents becomes more complex and current TL approaches do not cope well, since agents are expected to be aware of the thoughts of their opponents so that efficient strategies can be developed to complete the mission successfully. Thus, another focus of this research is on the study of evolutionary transfer learning framework that can cope with complex multi-agent system scenarios comprising agents with differing or even competitive objectives.

1.3 Research Contributions

The core contributions of this dissertation are two-fold. First of all, an evolutionary transfer learning (eTL) framework, one that is inspired by the theory of natural selection, Universal Darwinism and the notion of a meme in the context of computational intelligence, is proposed for solving multi-agent reinforcement learning tasks. Subsequently, a novel multi-agent transfer

learning framework with prediction capabilities on opponents' behavior is proposed. Further, a Top-K model selection method is also included with the aim of reducing the complexity of opponent behavior prediction given limited model space. Detailed contributions of the proposed methods presented in this thesis are listed in what follows:

- **An Evolutionary Transfer Reinforcement Learning Framework**

In this study, an evolutionary transfer reinforcement learning framework (eTL) is proposed for developing intelligent agents capable of adapting to the dynamic environment of multi-agent systems (MAS). Specifically, this study takes inspiration from Darwin's theory of natural selection and Universal Darwinism as the principal driving forces that govern the evolutionary knowledge transfer process. The core aspects of eTL comprise several meme-inspired evolutionary mechanisms, namely *meme representation*, *meme expression*, *meme assimilation*, *meme internal evolution* and *meme external evolution*. Further, eTL constructs social selection mechanisms that are modeled after the principles of human learning (such as "like attracts like" and "learn from elitist" [35][36][37]) to identify appropriate interacting partners. To verify the efficacy of the proposed paradigm, comprehensive investigations of eTL against existing state-of-the-art TL methods in MAS, are conducted on the "Minefield Navigation Tasks" (MNT) platform and the "Unreal Tournament 2004" first person shooter computer game, in which both homogeneous and heterogeneous learning machines are considered.

- **An Evolutionary Transfer Learning Framework with Behavior Prediction**

The study of eTL has led to the development of a computational framework in which a series of evolutionary mechanisms have been designed for implementing multiple interacting agents. This work further endeavors to address the specific challenges that arise in more complex multi-agent settings where agents share a common environment with other agents that have different and even competitive objectives. Particularly, this study

presents an enhanced evolutionary multi-agent transfer learning framework (eTL-P) with improvement over existing eTL where agents have means of predicting the behavioral strategies of other agents that have competitive objectives. Experiments that are based on a partner-opponent minefield navigation task (PO-MNT) have demonstrated that eTL-P can provide agents with the ability to acquire increasing level of learning capability and reduce the candidate model space by sharing knowledge memes with one another.

- **Top- K Model Selection for Solving Combinatorial Optimization Problem**

In eTL-P, agents are capable of predicting the behavior of their opponents. Since the true model of opponent agents is usually unknown, especially in competitive multi-agent scenarios, the behavior prediction is generally achieved by identifying a set of candidate models that are ascribed to opponent agents. Nevertheless, considering that the search space of all possible candidate models tends to be large, the prediction process thus becomes very time-consuming, and often infeasible due to computational and memory limits of individual agents. Hence, the challenge here is to select an appropriate subset of candidate models of the entire model set, given the limited model space. This study formalizes the model selection task as a NP-hard combinatorial optimization problem and constructs a monotone submodular function, which facilitates to select Top- K models from all available candidate models based on their representativeness in terms of behavioral coverage as well as reward diversity. Experiments based on PO-MNT have shown that the Top- K model selection method exhibits notable capability in choosing a much smaller candidate set that contains effective and representative knowledge from the full model space.

1.4 Organization of the Thesis

In what follows, the organization of the rest of this thesis is described.

- Chapter 2 presents an introduction of multi-agent system (MAS) and summarizes the recent works on solving MAS, including reinforcement learning for individual agent design and transfer learning for social communication among agents. This chapter is aimed at providing the reader with the necessary background for better comprehension of the research presented in this thesis.
- Chapter 3 presents a study on meme-centric evolutionary transfer learning frameworks. In particular, a short discussion on the motivation behind this study is first presented. Subsequently, comprehensive design of the evolutionary mechanisms for conducting meme-inspired knowledge transfer is presented. Taking the existing state-of-the-art multi-agent transfer learning methods as the baseline solvers, experimental studies on the meme-centric evolutionary transfer learning framework are conducted on the commonly used minefield navigation task platform. To further validate the effectiveness of the proposed knowledge transfer paradigm, the experimental study is extended to a commercial first person shooter video game, namely “Unreal Tournament 2004”.
- Chapter 4 presents a study on the novel transfer learning framework with Prediction (eTL-P), which endows agents with abilities to interact with their opponents effectively by predicting their behavioral actions. Based on the data instances archived from previous interactions, behavioral prediction is achieved by building and solving a set of predictive candidate opponent models. Empirical study on a partner-opponent minefield navigation task (PO-MNT) is then presented and the benefits of the proposed model-based behavior prediction and knowledge sharing mechanisms in eTL-P are analyzed and discussed.
- Chapter 5 presents the proposed Top- K model selection which has been designed to increase scalability, and hence reduce the predictive candidate model complexity of eTL-P. In this chapter, the Top- K model selection function has been formalized as a NP-hard combinatorial optimization problem. The mathematical derivation of a monotone

submodular function is presented and mathematically proven for selecting the Top- K candidate models from the full model space. To verify the effectiveness and efficiency of the proposed method, experimental studies conducted on the partner-opponent minefield navigation task (PO-MNT) are also presented.

- Chapter 6 concludes this dissertation and sheds light upon some potential future research work directions.

Chapter 2

Literature Review

This chapter begins with an introduction of the multi-agent system, which is followed by brief comments on both reinforcement learning for individual agent design and transfer learning for social communication among multiple agents. Subsequently, several state-of-the-art transfer learning works for multi-agent reinforcement learning systems are reviewed.

2.1 Multi-Agent System

Multi-agent systems (MAS) are composed of multiple interacting agents that have different information and diverging/common interests within an environment [1]. Due to its adaptativity and effectiveness in solving non-linear and complex problems, MAS have been successfully deployed in many real-world scenarios, such as traffic and transportation [38], military applications [39], distributed system [40], etc.

Formally, a MAS can be depicted using Fig. 2.1. In particular, each individual agent serves both a part of the environment while modeled as a separate entity with respective goals, actions and knowledge. The agents may have different learning architectures (indicated by different colors in Fig. 2.1) and different goals or objectives on their problem of interests. Besides the uncertainty that may be inherent in the domain of MAS, an agent needs to take into account the effects caused by other agents' actions, since each agent has the capacity to affect the entire

Figure 2.1: An illustration of the multi-agent system. (Agents may have differing goals, action space or domain knowledge, which are denoted by different fonts. And the arrow between the agents indicates the possible communication between agents.)

environment with their actions and influence the actions of other counterpart agents. Furthermore, due to the interaction among agents, MAS usually has a big state-action space as the global space is an exponential combination of all local spaces. Thus, the complexity of MAS grows rapidly with the number of agents, which increases the difficulties for agents to develop appropriate behaviors in the changing environment. This has attracted increasing research attentions where new effective approaches for identifying the optimal behavioral actions of agents in MAS are sought after.

In the past few decades, the challenges presented by the task of optimizing learning in MAS has attracted increasing attentions. The majority of existing research work can be categorized into two forms. Agents can optimize their own actions with the hope of reaching the global optimum. One of most significant techniques that has been proven to be a popular way for individuals to optimize behaviors is reinforcement learning (RL) [9], wherein agents can only acquire limited environmental feedback rewards, rather than correctly labeled instances, as is

common in other machine learning contexts such as supervised and unsupervised learning. Another trend to enhance the learning performance in complex MAS is to transfer good strategies from a well studied domain as supplementary knowledge to help and instruct the learning process for more informed decisions. A notable knowledge reuse methodology, known as transfer learning (TL) [17], is to leverage knowledge from related tasks for an easier and more effective learning process on newly encountered tasks. Compare to RL studied in MAS, significantly fewer studies on TL in MAS have been studied, which makes it a fertile research field for further investigations.

In the following subsections, the focus is placed on these two representative techniques for solving complex MAS, specifically, reinforcement learning for individual optimization in MAS and transfer learning for effective social interaction across multiple agents.

2.2 Reinforcement Learning for Individual Learning in MAS

Reinforcement learning (RL) is a paradigm for learning sequential decision making tasks that enable individual agents to optimize their behaviors through trial-and-error interactions [9]. In some challenging cases, the behaviors taken by RL agents may not only have influence of immediate reward, but also the sequential delayed reward of followup situations. The characteristics of trial-and-error optimization and delayed reward is thus considered as two most distinguishing features of RL.

Typically, a decision making task is formulated as a Markov Decision Process (MDP) which is composed of a set of states S , a set of actions A , a reward function $R(s, a)$, and a transition function $P(s'|s, a)$. Figure 2.2 illustrates a standard reinforcement model. Specifically, given a state space $S = (s_1, s_2, \dots, s_n)$ and a set of actions $A = (a_1, a_2, \dots, a_n)$ at step t , the RL agent firstly obtains an observation from the environment representing current state s_t which typically includes the reward r_t . The most appropriate action a_t is then predicted by evaluating each of possible action in set A . Upon receiving a reward $R(s, a)$ after performing this action,

Figure 2.2: Illustration of a standard reinforcement learning process.

the agent arrives at a new state s' which is determined by the probability distribution $P(s'|s, a)$. A policy $\pi = P(a|s)$ specifies a distribution for each state on deciding which action an agent takes. The value $Q^*(s, a)$ of a state-action pair estimates the future reward obtained from (s, a) when following policy π . $Q^*(s, a)$ is computed by the Bellman equation as the follows:

$$Q^*(s, a) = R(s, a) + \gamma \sum_{s'} P(s'|s, a) \max_{a'} Q^*(s', a')$$

where $0 < \gamma < 1$ is the discount factor. The aim of a RL agent is hence to search for the policy π which maps states to appropriate actions and maximizes the expected long-term rewards obtained from the environment.

Notably, the trade-off between exploration and exploitation of state-action strategy is of great importance during the behavior optimizing process of RL agents. The RL learner has to exploit what it has learnt in order to maximize rewards at the current states, and it also has to explore new behavioral strategies for making better action selections in the future. In order to tackle this problem of RL learners in real-world environments, various novel algorithms have been proposed. For example, these algorithms include Q-learning [41] combined with Temporal Difference algorithm [9] for problems in which the merit of an action is only known several steps after the action is conducted, the state-action-reward-state-action (SARSA) [42] for solving high dimensional continuous state-spaces, and others.

In the past decades, RL has gained extensive popularities and a plethora of learning methods, such as direct policy search [43], temporal difference [44] and Monte Carlo [45], have been proposed for building independent autonomous agents that are able to handle increasingly complex problems. Such methods have achieved significant success in a wide variety of practical applications including manufacturing [46], inventory management [47], delivery management [48], etc. However, as RL agents start learning from a *tabula rasa* state, significantly large amount of experiential data and extensively high exploration time are often the necessary conditions for successful learning to take place. They can sometimes fall short in meeting with today's competitive needs for high-efficiency problem-solvers and are hence often deemed as too slow for many complex problem domains. Thus, a significant number of current RL studies endeavors to improve the learning efficiency by leveraging domain expertise with various forms of human-provided knowledge. Typical studies in this direction include those that try deconstruct the task into a set of subtasks [49], learning with high-level abstractions of one-step actions [50] and abstracting over the given state space (e.g. value function [9]), so that the RL agent generalizes its learning experience more efficiently. The paradigm of transfer learning is one that has recently gained increasing attention for improving RL tasks in complex domains. Compared to previous studies, experience generalization in transfer learning can occur not only within a single task, but also across differing but related tasks.

In what follows, a general overview of transfer learning, especially transferring among multiple RL agents, is presented for more detailed understanding.

2.3 Transfer Learning in Multi-agent Reinforcement Learning Systems

Transfer learning (TL) has surfaced as an attractive learning approach for enhancing the learning efficacy of a new task by reusing the valuable data from a different, but related task [51]. The idea of TL is not entirely new and has long been studied in the literature of psychology

[52][53]. Due to its flexibility and ease of use, TL has been well-known as a popular problem solver for various machine learning tasks such as classification [54] and clustering [55], and has enjoyed significant success across a wide variety of real-world applications including computer vision [56], natural language processing [57], etc.

Recently, TL has gained increasing attentions for enhancing the classical RL paradigm, which leverages useful state-action strategies from well studied domains as supplementary knowledge to instruct and coordinate the learning process on newly encountered tasks. From a survey of the literature, TL not only improves agents' performance, but also speeds up the learning process [51]. In the past years, a variety of TL methodologies, such as instance transfer [18], action-value transfer [58], feature transfer [20] [59] and rule exchanging [19], have been proposed, and have benefited a wide range of RL tasks. However, it is worth noting that these early approaches focus almost exclusively on leveraging the knowledge or data instances across problem domains [23]. In these works, the mapping of knowledge across differing problems has been commonly hand-coded. TL among multiple agents in MAS has apparent difference from these studies and demands a way for the system to determine how to map information independently and automatically [60].

To be fully autonomous, an agent who conducts knowledge transfer typically needs to perform the following steps: 1) select appropriate source agent for knowledge transfer to the given target agent; 2) investigate the relevance between the source and target agents; 3) perform effective knowledge transfer from source agents to the possible target agents. The mechanism used for these steps is interdependent and closely related to what assumption is made, how the mapping across tasks is defined and what is transferred. For example, transfer methods that assume the source and target agents have differing state or action spaces may use inter-task mapping [61] to develop corresponding connections. In accordance with specific tasks, the transferred knowledge can range from very low-level information (i.e., $\langle s, a, r, s' \rangle$ instances or an action value function, etc.) to general heuristics that have the capacity to guide the whole

learning process (i.e., partial policies, rules or advice, important features, etc.). Typically, low-level information is transferred across similar agents while high-level information can be used across agents that bear lower similarity. The goal of transfer is to reduce the overall time required to learn a complex task while effectively reusing past knowledge in a novel task.

Considering TL mechanisms at a generic level, the main requirement for successful transfer is the common view that the transferred knowledge from source domains is helpful to the target. Taking this cue, MAS has the capacity to benefit from transfer learning methods. In MAS, transfer can occur within each single agent, for example, transfer with state-action tuples containing useful information for better decisions. Another is the transfer among different agents where agents can be affected by their own actions as well as their neighbors. Although the possibility makes MAS an attractive target for TL methods, the scenario of multiple agents and the complexity of MAS brings new challenges. In MAS, all agents (or tasks) learn simultaneously in the same environment. At the beginning of the learning process, no learned knowledge is available. Therefore, any pair of agents in the environment are likely to possess unique experiences and thus both may have information useful that can be transferred to the other.

Some recent research have considered TL for solving multi-agent RL tasks. Boutsoukis *et al.* [62] focused on evaluating the applicability of transfer learning to multi-agent RL and discussed transferring from single-agent system to multi-agent system and from one MAS to another, in an offline manner. Nevertheless, this work focuses on the knowledge transfer from a previously well studied problem domain to enhance problem-solving in a novel domain of interest. In MAS, on the other hand, since agents interact with one another, the potential for multiple agents to learn from one another in the same problem domain simultaneously or online should be appropriately exploited. The challenge here is thus to identify suitable source tasks from multiple agents that will contain useful information to transfer [23].

To address the issues of requiring source information and selecting source tasks, Oliveira *et al.* [21] described a study on an interactive Advice-Exchange (AE) mechanism, wherein

agents with poor performance seek advice only from the elitist for the next action to take. One major problem observed is that the bad advice, or blind reliance, of the elitist, could hinder the learning process, sometimes even beyond recovery. Similarly, Feng *et al.* [63] proposed to select teacher agents which not only have superior performance but also share similar past experiences. However, as the experience similarity measure is model specific, the approach is likely to fail when agents of diverse models exist. More recently, Taylor *et al.* [23] also discussed some key issues such as when and what to transfer, for conducting TL in MAS while learning progresses online, and further proposed a broadcast-based Parallel Transfer learning (PTL) for MAS. In PTL, each agent broadcasts its knowledge to all the other agents while deciding whose knowledge to accept based on the rewards received from other agents versus the expected rewards it predicts. Nevertheless, agents in this approach tend to infer incorrect actions on unseen circumstances. In the following subsections, more detailed discussions on these works of TL in multi-agent RL system are presented.

2.3.1 Interactive Advice Exchange Mechanism

Oliveira *et al.* [21] presented a study on an interactive advice-exchange mechanism where agents share episodes of given states by seeking advice at critical times, or by requesting demonstration of a solution to a specific problem to agents whose performance is currently better than all other agents. As compared to other ways of exchanging environmental states, or the internal parameters and policies, the option of seeking advice has distinct advantage in the following aspects: 1) Advice exchanging has no restriction on the heterogeneity of learning algorithms; 2) Heterogeneous learning algorithms have unique exploration of the same search space hence providing greater solution diversity; 3) The information included in the exchanged advice is more valuable than exchanged environment states; 4) Advice exchanging has less requirement on pre-coded knowledge.

At the beginning of learning algorithms, each agent that employs the heterogeneous learning algorithm (Random Walk, Simulated Annealing or Q-learning) has to decide on whether to

undergo advice-exchange process according to the current average quality of agents. In case of advice-exchange, the request for the advice is sent to agents with best quality, wherein the current state of the advisee is included as parameters. The advisor thus takes the received state transmitted by the advisee into consideration when developing the best guess as the most possible response to the given state. The response from advisor is then used to reinforce the internal knowledge of the advisee using the standard backpropagation rule. The system repeats the aforementioned steps and benefits from mutual interaction during the learning process until the terminal condition is satisfied.

Advice-exchange is a promising approach but there exists a problem, namely blind reliance or bad advice, which suggests that one agent may always execute the advice from the best agent. Blind reliance can hinder the learning process, sometimes even beyond recovery. The study in this dissertation also takes advantage of the advice-exchange mechanism. However, unlike advice-exchange where agents always learn from the elitist, this dissertation introduces an enhanced knowledge transfer approach wherein agents learn from others that not only have higher historical performance but also share the most similar experience. Another difference between the advice-exchange model and the present work is that the reinforcement signal is not only based on the advice given by advisor but is also directly collected from the environment.

2.3.2 Parallel Transfer Learning

Transfer learning has shown significant success in improving the performance of learning agents. However, most of traditional transfer approaches require the source task to be finished before valuable information can be transferred to the target task. The execution of the source task thus extends the learning time of target task. Taking this cue, Taylor *et al.* [23] proposed a simultaneous TL framework, namely parallel transfer learning (PTL) where both source and target tasks can learn in parallel and bi-directional way. In a PTL system, knowledge transfer occurs whenever the source task is deemed to be helpful to the target task. At each time step,

the information of current agent $agt(c)$ can be broadcasted to other agents every time it has distinct changes compared with some pre-defined boundary. Meanwhile, agent $agt(c)$ checks if its own transfer buffer has received any information from other agents in the system. Once confirmed, agent $agt(c)$ determines whether the received information is better than its current based on the frequency that the information(state) has been visited. Finally, agent $agt(c)$ can assimilate the information with higher quality into their own mind. Through repetition of the aforementioned process, agents in the system can explore valuable information of whole search space more faster.

PTL removes the restriction of executing the source task prior to the target task. The valuable information thus can be shared among agents as soon as it becomes available. As a result, PTL improves the final performance and takes less learning time when compared to the sequential TL approach, i.e., if the time of source task is to be counted. Nevertheless, PTL employs an information broadcasting scheme which has the tendency to be time consuming and redundant since other agents might have already explored the broadcast information before. Moreover, deciding the standard of valuable information is closely relevant to the distribution of differing situations. Agents in this approach thus prefer to infer incorrect actions on unseen situations. Differing to the information broadcasting strategy, this report introduces a knowledge transfer scheme which happens mainly via imitation, wherein agents imitate behaviors of partners with better performance or that are reputed to have better skills in the request for a specific situation. As a result, the redundancy and uncertainty in knowledge sharing could be reduced significantly.

2.3.3 Memetic Multi-Agent System

Memetics is a new science that has attracted increasing attentions in the past few decades. Beyond the formalism of memetic automaton which uses memes¹ as building blocks of information, Feng *et al.* [63] proposed a Memetic Multi-Agent System (MeMAS) for modeling

¹A meme is defined as the basic unit of cultural information in [29] stored in an individual's mind universe.

Figure 2.3: The architecture of Memetic Multi-Agent System.

agents' social interactions by sharing beneficial knowledge in terms of memes across multiple agents.

Fig. 3.2 depicts the detailed design of MeMAS which comprises a series of memetic evolutionary learning mechanisms. In MeMAS, each memetic agent employs a FALCON model [32] as the connectionist learning machine in its mind universe where the memes are acquired and stored in the cognitive field F_2 . Specifically, *meme representation* represents the stored memes which is described internally as the stored knowledge in agents' mind universe, and externally as the manifested behavioral actions that can be observed by others. Meme expression is the process for an individual to express their internal knowledge as actions to others. *Meme assimilation* is the way for an individual to capture these actions and update them into their respective mind universe.

Meme internal evolution and *meme external evolution* comprise the main behavioral learning aspects. Particularly, *meme internal evolution* is the process for agents to update their mind

universe by self learning. Specifically, the agent firstly predicts the rewards for conducting each available action via *meme expression*. The reward value is then used to select an action to perform. According to the possible feedback from the environment, a temporal difference formulation and meme assimilation are used to learn the association among multiple steps. *Meme external evolution* serves to model the social interaction among agents via imitation. In case of *meme external evolution*, a *meme selection* process is firstly used to select the appropriate agent to imitate. Unlike advice-exchange wherein agents only follow the advice of the agent with best quality, the similarity of assimilated knowledge among agents is also considered as part of the selection criteria. Once an agent is successfully selected, *meme transmission* occurs as follows: 1) agent $agt(c)$ passes its current state to the teacher agent $agt(s)$ identified by *meme selection*; 2) after receiving the given state, $agt(s)$ expresses its inferred action that can be observed by $agt(c)$; 3) $agt(c)$ imitates the action from $agt(s)$ and updates its mind universe by ways of *meme assimilation*.

Taking advantage of meme-inspired design of individual agents and their interactions, MeMAS leads to greater level of self adaptivity and effective social behavior. Particularly, in MeMAS, agents are encouraged to learn from their possible teacher agents which not only have superior performance but also share similar past experiences. Nevertheless, as the experience similarity measure is model specific, the approach is likely to fail when agents of diverse models exist. Specifically, MeMAS has merely employed FALCON as the connectionist learner inside agents' mind universe. The heterogeneity of learning agents, however, has yet been considered. In practice, modeling the interaction among heterogeneous learning agents based on their internal architectures is non-trivial since the learning principle, theory and mechanism of the different agents are unique and usually highly complex. Therefore, this report endeavors to develop an evolutionary transfer learning approach which facilitates the effective knowledge transfer among heterogeneous learning agents.

Chapter 3

An Evolutionary Transfer Reinforcement Learning Framework

Takes inspiration from Darwin's theory of natural selection and Universal Darwinism [29] [30] as the principal driving forces that govern the evolutionary knowledge transfer process, this chapter presents an evolutionary Transfer reinforcement Learning framework (eTL) for building intelligent social agents with the capability of adapting to dynamic multi-agent systems (MASs). The intrinsic parallelism of natural evolution and the errors which are introduced due to the physiological limits of the agents' ability to perceive differences [64], could generate the "growth" and "variation" of agents' knowledge that learnt from the problem domain [31], thus exhibiting better adaptivity capabilities on solving non-trivial and complex problems. Further, the essential backbone of the proposed framework comprises of memetic automaton¹ [24], which includes meme-inspired² evolutionary mechanisms, namely *meme expression*, *meme assimilation*, *meme representation*, *meme internal* and *external evolution*. Particularly, *meme representation* and *meme evolution* comprise the main aspects of eTL. *Meme representation* pertains to what is a meme. *Meme evolution*, which includes *meme internal evolution* and *meme external evolution*, instructs the behavioral learning process of multiple agents. To be

¹A memetic automaton is defined as an self-contained software or agent which autonomously learns to enhance adaptivity and capability using memes as the building blocks of information during knowledge evolving.

²A meme is regarded as the fundamental building units of cultural information, hold in an individual's mind, which is capable of being transmitted to others [29].

specific, *meme internal evolution* indicates the learning process that updates an individual's internal knowledge via self learning. The *meme external evolution* which is central to the behavioral aspects of imitation, models the social interactions among multiple agents. In contrast to existing works on TL in MAS, eTL addresses the aforementioned limitations (e.g., blind reliance) found in the current knowledge transfer process of existing frameworks. In particular, the proposed approach constructs social selection mechanisms that are modeled after the principles of human-learning [35] to effectively identify appropriate interacting partners, thus bringing about improved agent social learning capabilities in dynamic environments[65].

The core contributions of the proposed work are concluded as follows.

- (i) A novel evolutionary multi-agent Transfer reinforcement Learning framework (eTL) is proposed for modeling intelligent social RL agents. Further, beyond the formalism of memetic automaton, the core aspects of eTL including *meme representation*, *meme expression*, *meme assimilation*, *meme internal evolution* and *meme external evolution* are proposed and discussed comprehensively.
- (ii) To explore the generality of eTL, the Fusion Architecture for Learning and Cognition (FALCON) [32] based on adaptive resonance theory (ART) [66] and the classical multi-layer perceptron (MLP) [67] with gradient descent based back propagation (BP) [68] are investigated as the online learning machines that form the brains of the intelligent agents.
- (iii) To validate the effectiveness of eTL with both homogenous and heterogeneous learning agents, comprehensive empirical studies are conducted on commonly used Mine-field Navigation Tasks (MNT) [69]. Experimental results obtained show that eTL which transfers knowledge in an evolutionary manner has significantly better performance than existing state-of-the-art TL approaches including AE and PTL.

- (iv) Last but not least, a well-known first person shooter game, namely “Unreal Tournament 2004” (UT2004) [70], is used to investigate the performance of the proposed eTL under complex problem solving scenarios.

The rest of this chapter is organized as follows: Section 3.1 begins with an introduction of the proposed eTL. Particularly, details of the evolutionary operators in the eTL framework are presented. Further, study of the eTL with FALCON and MLP-BP are presented in Section 3.2. Section 3.3 presents a comprehensive empirical study of eTL based on the popular MNT and UT2004 as complex problems for verifying the performance of eTL. Finally, this chapter is concluded with some brief remarks in Section 4.3.

3.1 Evolutionary Memetic Multi-Agent Transfer Reinforcement Learning System

This section begins with a brief introduction of memetic science which is followed by an introduction of the proposed evolutionary multi-agent Transfer Learning framework (eTL).

3.1.1 Memetic Science

The term “meme” can be traced back to Dawkins in his book “The selfish Gene”, where he defined it as “a unit of information residing in the brain and is the replicator in human cultural evolution” [29]. While some researchers believe that memes should be formalized as the information or knowledge restricted to the brain, others consider that the concept extends to artifacts and behaviors. In “The Meme Machine” [71], Blackmore reaffirmed meme as knowledge copied from one person to another and discussed on the theory of “memetic selection” as the survival of the fittest among competitive ideas down through generations. Other researchers on the other hand have defined and discussed memes in different ways, from being “contagious information patterns that replicate by parasitically infecting human minds” [72], “constellation of activated neuronal synapses in memory or information encoded in neural structure” [73] to

“ideas, the kind of complex idea that forms itself into a distinct memorable unit” [74], and even “genotype as mental representation, and phenotype as implemented behavior or artifact” [75].

For the past few decades, the meme-inspired science of memetics [76][77] has attracted increasing attention and stretched across multi-faceted fields including anthropology, biology, psychology, sociology, etc. In the research field of computational intelligence, memetic computation has also surfaced as an intriguing focus of research. Particularly, one of the most simplest and direct applications of meme-inspired technologies in problem solving has been the memetic algorithm [78][79], which has been established as a key methodology in complex optimization. Beyond the simple and adaptive hybrids of memetic algorithms, the research of memetic computation further culminates into a meme-centric “memetic automaton” [24] that memes transpire as units of domain information or knowledge building blocks useful for problem-solving. Recently, memes have been defined as the transformation matrices that can be transferred across differing problem domains for enhancing the evolutionary search process [80].

As with genes in genetics which serve as “instructions for building proteins”, memes are synonymous to memetics as the instructions for carrying out behavior, constructing artifact for problem solving. In this chapter, neuronal memes are manifested as memory items or portion of an organism’s neurally-stored knowledge inside agents’ mind universe. Further, beyond the formalism of memetic automaton and taking memes as focal point of interest in the field of MAS, this work proposes an evolutionary multi-agent Transfer reinforcement Learning framework (see Fig. 3.1) to develop the evolutionary learning process, particularly on agents’ social interactions, for more efficient problem solving. In what follows, the attention is directed on the representation of memes and evolutionary mechanisms pertaining to eTL.

3.1.2 Proposed eTL Model

In the design of eTL, memes are regarded as the fundamental building blocks of the agents’ mind universe, which are categorized into memotypes and sociotypes. Internally, memotypes

Figure 3.1: Illustration of the evolutionary multi-agent Transfer reinforcement Learning framework.

(depicted by the different LEGO-like block objects that lies in the agents mind universe of Fig. 3.1) are described as the agents' ideas or knowledge captured as memory items or generalized abstractions inside the agents' mind universe. Externally, sociotypes are defined as the manifested behaviors that could be observed by the others. In particular, in the case that differing agents have distinct memotype structures, sociotypes in the proposed eTL offer a channel for agents to transfer knowledge across the whole population.

Meme representation and *meme evolution* form the two core aspects of eTL. It then undergoes *meme expression* and *meme assimilation*. *Meme expression* is defined for an individual to express their stored neuronal memes as behavioral actions, while *meme assimilation* captures new memes by translating corresponding behavioral information into knowledge that blends into the individual's mind universe. That is to say, *meme representation* pertains to what is a meme. *Meme expression/assimilation* on the other hand activates/updates the meme during the

learning process.

The *meme evolution* processes, namely *meme internal/external evolution*, comprise the main behavioral learning aspects of eTL. To be specific, *meme internal evolution* indicates the process for agents to update their mind universe through self learning process or personal grooming. In eTL, all agents undergo *meme internal evolution* by exploring the common environment simultaneously. During self learning process, *meme external evolution* might happen to model the social interaction among agents mainly via imitation, which takes place when memes are transmitted. Specifically, *meme external evolution* happens whenever the current agent identifies an appropriate teacher agent via a *meme selection* process. Once the teacher agent is selected, *meme transmission* occurs to instruct how the agent imitates from others. During this process, *meme variation* then facilitates the innovative characteristics of knowledge transfer among agents. Based on the received feedback from the environment after conducting an action, the agent then proceeds to update its mind universe accordingly.

3.2 Realization of eTL with Learning Agents

This section presents two realizations of learning agents that take the form of neurally-inspired learning structures, namely a Temporal Difference (TD)-FALCON and a BP multi-layer neural network, respectively. To be more specific, FALCON is a natural extension of self-organizing neural models proposed in [32] for real-time reinforcement learning, while BP is a classical multi-layer network that has been widely used in various learning systems. As discussed in Section 3.1, eTL comprises several evolutionary operators including *meme representation*, *meme internal evolution* and *meme external evolution*. In what follows, the detailed realization for each of these operators is presented.

3.2.1 FALCON Learning Agent

The FALCON meme inspired evolutionary mechanisms are discussed in what follows.

3.2.1.1 FALCON Meme Representation

A FALCON learning agent (depicted in Fig. 3.2) employs a 3-channel neural network structure which is composed of a category field F_2 and three input fields, which are known as a sensory field F_1^{c1} for denoting the current input states, a motor field F_1^{c2} for denoting behavioral actions, and a feedback field F_1^{c3} for denoting reward signals.

Figure 3.2: Illustration of *meme representation* taking the form of FALCON and BP.

Input vectors: $\mathbf{I} = (\mathbf{S}, \mathbf{A}, \mathbf{R})$ denotes the input vectors where $\mathbf{S} = (s_1, s_2, \dots, s_n)$ denotes the state vector, and s_n indicates the value of the input sensory n ; $\mathbf{A} = (a_1, a_2, \dots, a_m)$ denotes the action vector, and a_m indicates the m^{th} possible action; $\mathbf{R} = (r, 1 - r)$ denotes the reward vector, and $r \in [0, 1]$.

Activity vectors: $\mathbf{x}^{ck}$ denotes the F_1^{ck} activity vector for $k = 1, 2, 3$ where $\mathbf{x}^{c1}$, $\mathbf{x}^{c2}$, $\mathbf{x}^{c3}$ correspond to the input state $\mathbf{S}$, behavioral action $\mathbf{A}$, and feedback reward $\mathbf{R}$, respectively.

Weight vectors: $\mathbf{W} = (\mathbf{w}_1^{ck}, \mathbf{w}_2^{ck}, \dots, \mathbf{w}_j^{ck})$ denotes the weight vector wherein $\mathbf{w}_j^{ck}$ indicates the weight of the j^{th} neuron in F_2 to learn the input activity vector F_1^{ck} . At the beginning, F_2 only has one uncommitted neuron whose weight is initialized by all 1's. It then becomes committed and another uncommitted neuron will be initialized if the uncommitted neuron is selected to learn the association.

Internally, a memotype is defined as the meme inhabiting inside the cognitive field of F_2 , which is known as the mind universe of a FALCON learning agent. All these memotypes (neurons) constructs the internal knowledge of the agent which models the association of the input states and action, thus providing instruction for selecting the appropriate actions. Externally, the sociotype meme is regarded as its expressed actions or behaviors, which are able to be observed and imitated by others.

3.2.1.2 FALCON Meme Expression

- **Meme activation:** Firstly, a propagating process occurs in a bottom-up manner such that the activity value of the memotype in the F_2 field are calculated. Particularly, based on the activity vectors $\mathbf{x}^{ck}$, the value T_j that represents the similarity of activity vectors with the respective weights for each meme j of F_2 is calculated in what follows:

$$T_j = \sum_{k=1}^3 \gamma^{ck} \frac{|\mathbf{x}^{ck} \wedge \mathbf{w}_j^{ck}|}{\alpha^{ck} + |\mathbf{w}_j^{ck}|}, \quad (\text{Eq. 3.1})$$

where the fuzzy AND operation $\wedge$ is defined by $(\mathbf{p} \wedge \mathbf{q})_i \equiv \min(p_i, q_i)$ for vectors $\mathbf{p}$ and $\mathbf{q}$, and the norm $|\cdot|$ is defined by $|\mathbf{p}| \equiv \sum_i p_i$. Choice parameters α^{ck} and contribution parameters γ^{ck} are predefined, and $\alpha^{ck} > 0, \gamma^{ck} \in [0, 1]$.

- **Meme competition:** *Meme competition* process takes place right after *meme activation*. During this, the F_2 meme with the highest activation value will be identified. Let $\mathbf{y}$ denotes the F_2 activity vector. The winner meme is further labeled with J where

$$T_J = \max\{T_j : \text{for all } F_2 \text{ node } j\}. \quad (\text{Eq. 3.2})$$

Particularly, the following winner-take-all conditions are satisfied: $y_J = 1$ and $y_j = 0$ for all $j \neq J$.

- **Sociotype readout:** After choosing a winning F_2 meme J , the winner performs a readout of its weight vectors into the input field F_1^{ck} :

$$\mathbf{x}^{ck(new)} = \mathbf{x}^{ck(old)} \wedge \mathbf{w}_J^{ck}. \quad (\text{Eq. 3.3})$$

Consequently, the $\mathbf{x}^{ck(new)}$ of F_1^{ck} are the result of fuzzy AND of $\mathbf{x}^{ck(old)}$ and $\mathbf{w}_J^{ck}$.

3.2.1.3 FALCON Meme Assimilation

- **Memotype matching:** After *meme competition*, the selected meme J is estimated by *memotype matching*, which is conducted to access if the weight templates of meme J are sufficiently similar to their respective activity patterns. Particularly, resonance occurs only if the match function m_J^{ck} for each channel k meets the vigilance criterion:

$$m_J^{ck} = \frac{|\mathbf{x}^{ck} \wedge \mathbf{w}_J^{ck}|}{|\mathbf{x}^{ck}|} \geq \rho^{ck}, \quad (\text{Eq. 3.4})$$

where vigilance parameter ρ^{ck} is pre-defined and $\rho^{ck} \in [0, 1]$ for $k = 1, 2, 3$. In case of the any vigilance constraints is violated, mismatch reset is expected to happens. The value T_J will be set to 0 for the duration of the input presentation and another meme is selected for the *memotype matching*.

- **Memotype learning:** If resonance occurs successfully, the weight vector $\mathbf{w}_J^{ck}$ in each channel k is updated as follows:

$$\mathbf{w}_J^{ck(new)} = (1 - \beta^{ck})\mathbf{w}_J^{ck(old)} + \beta^{ck}(\mathbf{x}^{ck} \wedge \mathbf{w}_J^{ck(old)}), \quad (\text{Eq. 3.5})$$

where $\beta^{ck} \in [0, 1]$ denotes the learning rate parameters. The rationale is to learn by encoding the common attribute values of both the input vectors and weight vectors.

3.2.2 BP Learning Agent

Here, this subsection describes the BP meme inspired evolutionary mechanisms.

3.2.2.1 BP Meme Representation

The mind universe of a BP agent has a three-layer architecture (as shown in Fig. 3.2), consisting of an input layer I for representing current states S and available actions A , an output layer O consisting of only one neuron for evaluating the association between selected action and a particular state, a hidden layer H for increasing the expressiveness of the network, and a reward signal R for estimating the errors of output neurons. All hidden and output neurons receive inputs from the initial inputs or the interconnections and produce outputs by transformation using a symmetrical sigmoid function. In contrast to the FALCON agent whose memotypes are generated dynamically during the learning process, the internal memotypes housed in a BP agent are modeled as fixed neurons in the hidden layer.

3.2.2.2 BP Meme Expression

- **Neuronal meme (memotype) forward propagation:** A forward propagation process first takes place in which the reward signals of performing each possible action are computed. Specifically, given the state vector S , for each possible action a_m in A , the outputs of the hidden memotypes are firstly calculated as

$$H_h = \frac{1}{1 + e^{(-Gain \times \sum_{i=1}^p I_i \times W_{ih})}}, h = 1, \dots, q, \quad (\text{Eq. 3.6})$$

where $Gain$ is the gain of sigmoid function, I_i indicates the value of i^{th} output in layer I , W_{ih} indicates the connection weight from i^{th} neuron in layer I to the h^{th} memotype in layer H , p and q are the number of neurons in the layer I and H , respectively. Then, the output reward value $(O_k)_m$ of performing each action a_m from A is further computed as

$$(O_k)_m = \frac{1}{1 + e^{(-Gain \times \sum_{h=1}^q H_h \times W_{hk})}}, k = 1, \quad (\text{Eq. 3.7})$$

where H_h denotes the output of the h^{th} meme in the hidden layer, W_{hk} is the connective weight between h^{th} memotype in the hidden layer and the node in the output layer, q is the total number of memotypes in the hidden layer and $k = 1$ means the index of the only node in the output layer.

- **Neuronal meme competition and sociotype readout:** After *meme forward propagation*, the reward value $(O_k)_m$ is then used to identify the sociotype meme with the highest reward value. The winner of the winner-take-all competitive strategy is indexed as M where

$$O_M = \max\{(O_k)_m : \text{for all actions in } \mathbf{A}\}. \quad (\text{Eq. 3.8})$$

The corresponding action a_M is thus read out as the identified sociotype meme.

3.2.2.3 BP Meme Assimilation

- **Memotype backward error estimation:** After performing the identified sociotype meme a_M , a *memotype backward error estimation* process checks the difference between the actual output O_k and expected output T_k of the network. If neuron b is the node of output layer, the error ε_b is estimated via

$$\varepsilon_b^l = (T_b - O_b) \cdot O_b \cdot (1 - O_b), \quad (\text{Eq. 3.9})$$

where l is the index number of the layer. If neuron b is the memotype of the hidden layer, the error signal ε_b is then back propagated from the output layer:

$$\varepsilon_b^l = \left[\sum_k \varepsilon_k^{l+1} \cdot W_{bk} \right] \cdot H_b \cdot (1 - H_b), \quad (\text{Eq. 3.10})$$

where H_b denotes the output of the b^{th} memotype in the hidden layer.

- **Memotype learning:** With the error signal term, the weight vector of each meme in the hidden layer is updated using the generalized learning rule:

$$W_{ab}^{new} = W_{ab}^{old} + \eta \cdot \varepsilon_b^l \cdot O_a^{l-1} + \tau \cdot (\eta \cdot \varepsilon_b^l \cdot O_a^{l-1})^{old}, \quad (\text{Eq. 3.11})$$

in which $\eta \in [0, 1]$ is the constant learning rate, $\tau \in [0, 1]$ is the pre-defined positive momentum accelerating the convergence rate of error propagation process, O_a^{l-1} denotes the output value of the sublayer related to the connective weight and superscripts *new* and *old* represent the current and most recent training steps, respectively.

3.2.3 Meme Internal Evolution

A general illustration of the *meme internal evolution* and *meme external evolution* processes in the eTL framework is depicted in Fig. 3.1. More specifically, the *meme internal evolution*, as summarized in Algorithm 1, governs the growth of an individual's mind universe via self learning. It is obvious that the internal evolutionary process is made up of a sequence of learning trials, and the trials continue until the ultimate conditions, such as mission numbers and fitness levels, are satisfied (see Line 3). During the *meme internal evolution*, an agent firstly predicts the Q-values by conducting each possible action through a *meme expression* process given current state s and a possible action set $\mathbf{A}$ (see Lines 5-6). Furthermore, the received Q-values are used to select an appropriate action based on an action selection strategy (see Line 7). A TD formula further computes a new estimate of the Q-value for performing the chosen action in the current state upon receiving the environmental feedback after conducting the action. Then, the new Q-value is used to learn the association from the chosen action and the current state to the new Q-value (see Lines 9-19).

In eTL, an ϵ -greedy action selection scheme is employed with the objective to balance exploration and exploitation in the *meme internal evolution* process (see Line 7). Compared to other more complex methods, ϵ -greedy does not require memorization of specific data during exploration and is reported to be usually the first choice method as stated by Sutton [81]. This strategy proceeds to select an action that has the highest $Q(s, a)$ value with probability $1 - \epsilon$ ($0 \leq \epsilon \leq 1$), otherwise takes a random action.

Additionally, in case of the multiple-step prediction problems, the action selection should not only depend on the current states, since the agent can only know the merit of an action

Algorithm 1: Meme Internal Evolution Process

Input: Current state s , a set of possible actions $\mathbf{A} = (a_1, a_2, \dots, a_m)$

1 **Initialization:** Generate the initial agents

2 **Begin:**

3 **while** *stop conditions are not satisfied* **do**

4 **for** *each current agent* $agt(c)$ **do**

5 **for** *each action* $a_m \in \mathbf{A}$ **do**

6 $Q(s, a_m) = Predict(s, a_m)$

7 **Select** an action a_M from $\mathbf{A}$: $a_M = Scheme(Q(s, a))$.

8 **Perform** selected a_M : $\{s', r\} = Perform(a_M)$ where s' is the resultant state and r is the immediate reward from the environment.

9 **Estimate** the reward $Q^{(new)}(s, a_M)$ by:
 $Q^{(new)}(s, a_M) = Q(s, a_M) + \vartheta \delta (1 - Q(s, a_M))$

10 **Do:**

11 **if** $agent \in FALCON$ *agents* **then**

12 *Meme activation* with vector $\{\mathbf{S}, \mathbf{A}, \mathbf{R}\}$

13 *Meme competition* with vector $\{\mathbf{S}, \mathbf{A}, \mathbf{R}\}$

14 *Meme matching* with vector $\{\mathbf{S}, \mathbf{A}, \mathbf{R}\}$

15 *Memotype learning* with vector $\{\mathbf{S}, \mathbf{A}, \mathbf{R}\}$

16 **else**

17 *Memotype forward propagation* with vector $\{\mathbf{S}, \mathbf{A}, \mathbf{R}\}$

18 *Memotype backward error estimation* with vector $\{\mathbf{S}, \mathbf{A}, \mathbf{R}\}$

19 *Memotype learning* with vector $\{\mathbf{S}, \mathbf{A}, \mathbf{R}\}$

20 **End**

beyond several steps in the future. Therefore, the Temporal Difference (TD) method, such as Q -learning, is employed to evaluate the value function of action-state pairs $Q(s, a)$, which denotes the goodness of a learning system to perform an action a given state s (see Line 9). Specifically, the iterative updating rules of Q -values is defined by:

$$Q^{(new)}(s, a) \leftarrow Q(s, a) + \vartheta \delta (1 - Q(s, a)), \quad (\text{Eq. 3.12})$$

where $\vartheta \in [0, 1]$ denotes the learning parameter. δ is the temporal-difference error defined as:

$$\delta = r + \gamma \max_{a'} Q(s', a') - Q(s, a), \quad (\text{Eq. 3.13})$$

where $r \in [0, 1]$ is the immediate reward value, $\gamma \in [0, 1]$ is the discount parameter, a' is the predicted action under the next state s' , and $\max_{a'} Q(s', a')$ is the maximum estimated Q -value of the next state s' . The scale term $(1 - Q(s, a))$ guarantees that the Q -values will always be bounded between within $[0, 1]$, which will be consistent with the input values of the system. Further, the new Q -value that is computed by the TD formula is encoded into the reward for 1) FALCON agents: $\mathbf{R} = (Q^{(new)}(s, a), 1 - Q^{(new)}(s, a))$ and 2) BP agents $\mathbf{R} = (T_k) = (Q^{(new)}(s, a))$.

3.2.4 Meme External Evolution

Meme external evolution, which serves to model the social interaction among the learning agents, is governed by four evolutionary factors, namely *meme selection*, *meme expression*, *meme imitation* and *meme variation*.

3.2.4.1 Meme Selection

Meme selection takes inspiration from Darwin's notion of natural selection, with the goal of identifying the teacher agent with the highest yield predicted. Through this mechanism, agents' internal memes (or knowledge) that are more beneficial for problem solving are duplicated

exponentially in the knowledge universe pool, while the memes that are less helpful or detrimental rarely get replicated. In the past few years, most researchers have concentrated on exploring “how an agent can learn from other agents”.

Among them, uniform random selection is the simplest and most commonly used scheme for choosing a representative agent from the crowd, where each individual has equal chance of being chosen. Nevertheless, more efficient selection approaches are possible if useful information about the individuals is available. A well-known selection strategy is the “imitate-from-elitist”, in which agents learn through a reward mechanism. Based on this elitism selection mechanism, Oliveira *et al.* [21] presented a study on interactive advice-exchange mechanism (AE) where agents share episodes of given states by seeking advice at critical times, or by requesting the demonstration of a solution to a specific problem from agents whose performance is currently better than all the other agents.

However, AE is likely to suffer from blind reliance problem since advisees have no knowledge of the holdings of the reliance and always seek the advice from the teacher agent with the best performance. It was reported that the best teacher agent may not always be the agent with the best performance. This might happen for several reasons. But the most common of them is that each agent is unique both genetically and memetically. Hence, the responding of each agent with a common action for the given states may not suffice. In such cases, an agent also needs to learn which of its partners it can trust and which has the most similar experience in dealing with the given circumstance. To tackle this blind reliance problem, the notion of “like-attracts-like” [37] has been introduced wherein agents learn from others not only based on the updated performance, but also on the similarities of the best matched knowledge under the given circumstance.

In the design of this work, the decision of which peer an agent should learn from depends on both the agents’ superior historical success, namely *elitist* and the agents’ confidence in solving the request for a particular given situation, namely *similarity*. This *meme selection*

process in eTL is thus a fusion of the “imitate-from-elitist” and “like-attracts-like” principles. On one hand, the *elitist* is maintained as a centralized resource to express the judgement of agents’ behaviors in accordance with the past knowledge. It is the simplest and most direct way to provide recommendations since agents do not need to compute the reputation while being able to identify the elitist agent directly. However, when the agents communicate with others, the interaction process might be very labour-intensive and the required information could be inefficient and not always desirable. The reason is that although the elitist has a higher performance than the partners, it might not be the most suitable candidate in solving the given unseen circumstance. On the other hand, *similarity* does not solely focus on the knowledge of the past, but the prediction on future reward under similar situations. This scheme does not impose higher performances in the partners but rather focuses on the sharing of relevant experiences in dealing with similar previously encountered situations. As a consequence, the *similarity* scheme narrows down the relevant *elitist* to learn from. Nevertheless, in the event that agents possess differing prior knowledge (in the form of memotypes) for evaluating a common experience, the *similarity*-based selection scheme would be futile since it is difficult for agents to judge whether the knowledge from the teacher is beneficial or detrimental to itself.

In *meme selection*, the trustworthiness of performance is estimated in accordance with the positive and (or) negative outcomes produced by the agents in the past. The detailed outline of the *meme selection* process is given in Algorithm 2. Particularly, it defines $El(agt(v))$ as the relative fitness of agent $agt(v)$ and $Si(agt(v))$ as the relative maximum predicted award to perform an action under the given state s :

$$\begin{cases} El(agt(v)) = Fitness(agt(v))/Fitness_{best}, \\ Si(agt(v)) = \max\{Q(s, \mathbf{A})\}/Q_{best}, \end{cases} \quad (\text{Eq. 3.14})$$

where $Fitness(agt(v))$ denotes the fitness value of an agent $agt(v)$, v in $agt(v)$ is the index of the v^{th} agent, $Fitness_{best}$ is the reputation of the elitist or the best performed agent, and $Fitness = Fitness + 1$ if the agent completes a mission successfully; $Q(s, \mathbf{A})$ indicates the

Algorithm 2: Meme Selection Process

Input: Current state s , a set of possible actions $\mathbf{A} = (a_1, a_2, \dots, a_m)$

- 1 **Begin:**
- 2 **Get** current agent $agt(c) \in P$
- 3 **for** each agent $agt(v)$ where $v \neq c$ **do**
- 4 **Get** the current state of agent $agt(c)$ as s
- 5 **Get** the Si value of $agt(v)$ under the given s : $Si(agt(v)) = \max\{Q(s, \mathbf{A})\}/Q_{best}$
- 6 **if** ($El(agt(v)) \geq El(agt(c)) \& Si(agt(v)) \geq Si(agt(c))$) **then**
- 7 **Put** $agt(v)$ into set B
- 8 **for** each agent $agt(v)$ in B **do**
- 9 $Sc(agt(v)) = El(agt(v)) \times Si(agt(v))$
- 10 **Select** the teacher (or source) agent with highest $Sc(agt(v))$
- 11 **End**

maximum predicted Q-value of agent $agt(v)$ for given state s , and the Q_{best} is the maximum historical Q value of agent $agt(v)$. In the learning process, agents that have been found to have both higher *elitist* and *similarity* values than the current agent $agt(c)$ (also known as target agent) are identified to form the set B (see Lines 6-7):

$$B = \{agt(v) \in P | (El(agt(v)) \geq El(agt(c))) \& Si(agt(v)) \geq Si(agt(c))\}, \quad (\text{Eq. 3.15})$$

where P denotes the entire population in the environment. Subsequently, the agent with the highest Sc value shall serve as the teacher agent $agt(s)$ (or source agent) in the selection set for *meme transmission* (see Lines 8-10):

$$Sc(agt(v)) = El(agt(v)) \times Si(agt(v)). \quad (\text{Eq. 3.16})$$

3.2.4.2 Meme Transmission

Once a selection is made, *meme transmission* between agents with unique learning capabilities happens by means of imitation. In the present problems, each agent learns in the same

environment and has the same act abilities. Therefore, imitation could offer the advantage for the imitating agent to behave at approximately the same performance level as its imitator. In addition, if a human player is imitated, the autonomous agents could exhibit more believable and complex human-like behaviors, seamlessly.

Algorithm 3: Meme Transmission Process

Input: Current state s , a set of possible actions $\mathbf{A} = (a_1, a_2, \dots, a_m)$

- 1 **Begin:**
 - 2 **Get** current agent $agt(c) \in P$
 - 3 **Get** current state $s(agt(c))$ of $agt(c)$
 - 4 **Pass** $s(agt(c))$ to the teacher agent $agt(s)$
 - 5 **Get** action $a(agt(s))$ of $agt(s)$ through Steps 5-7 in Algorithm 1
 - 6 **Perform** variation on $a(agt(s))$ with probability ν
 - 7 **Perform** $a(agt(s))$: $\{s', r\} = \text{Perform}(a(agt(s)))$ where s' is the new state and r is the immediate reward.
 - 8 The remaining steps are the same as those in Algorithm 1
 - 9 **End**
-

The *meme transmission* process in eTL is concluded in Algorithm 3. Firstly, current agent $agt(c)$ will pass its state to the chosen teacher agent $agt(s)$ (see Lines 3-4). Further, agent $agt(s)$ expresses its corresponding action $a(agt(s))$ by *meme expression*, which can be observed by agent $agt(c)$ (see Line 5). Then, agent $agt(c)$ assimilates the received action into its mind universe by means of *meme assimilation* (see Lines 7-8). In addition, the detailed variation process which may occur in Algorithm 3 (see Line 6) is described in *meme variation* as follows.

3.2.4.3 Meme Variation

Meme variation serves to give the inherent innovation tendency of selected sociotypes for *meme transmission* in the evolutionary knowledge transfer process. More specifically, for knowledge transmission without variation, agents might believe that the knowledge as represented by sociotypes of an elite agent is always helpful with respect to its particular performance at the

given circumstance. Due to the nonlinearity of the knowledge transfer process, this bias could spread and spiral out of control since the bias infects any other agents which come into contact with [63]. This would suppress the agents' ability to explore the environment. Therefore, *meme variation* is considerably of great importance for retaining diversity in agents' attitude towards the innovative knowledge transfer process.

In the proposed eTL framework, *meme variation* occurs at the expression and imitation stages, in which a probabilistic interference cost is put on to the estimated action's Q-value to allow different actions to be selected for expression:

$$Q^t = \lambda \times Rand + (1 - \lambda) \times Q, \quad (\text{Eq. 3.17})$$

where Q^t is the mutated Q , $\lambda \in [0, 1]$ is the parameter for controlling the degree of randomness, and $Rand \in [0, 1]$ is a random value of uniform distribution. The pre-defined $\nu \in [0, 1]$ is the probability to control the frequency of *meme variation* process.

3.2.5 eTL with FALCON and BP

The proposed eTL is illustrated in Fig. 3.3 where a population of agents learn and evolve in a single MAS environment. Particularly, both the aforementioned FALCON and BP agents denote the two distinct connectionist learners that coexist in the environment simultaneously. In the design of eTL, each individual can interact with the environment or one another by undergoing the *meme internal evolution* process and/or the *meme external evolution* process, where the former is the learning process for updating an individual's internal knowledge via personal grooming. The latter process which is central to the behavioral aspects of imitation, models the interactions among multiple agents.

The basic steps of the eTL are outlined in Algorithm 4. To be specific, a population of N agents is first generated with BP or FALCON architectures. Furthermore, each of the current agents performs *meme internal evolution*. Meanwhile, *meme external evolution* proceeds if

Algorithm 4: Basic eTL Framework

```

1 Begin:
2 Initialization: Initialize  $N$  agents
3 while stop creterias are not satisfied do
4   for each current agent  $agt(c)$  do
5     /*Perform meme internal evolution*/
6     See Algorithm 1, Line 5-7
7     if Identify the teacher agent via meme selection then
8       /*Perform meme external evolution*/
9       Perform meme expression with  $agt(s)$  under the state of  $agt(c)$ 
10      Perform meme variation with probability  $\nu$ 
11      /* $\nu$  is the frequency probability of variation process*/
12      Perform meme transmission for  $agt(c)$  to assimilate  $agt(s)$ 's action
13      /*End meme external evolution*/
14      Evaluate the probability  $C_p$  of each agent
15      /*Perform the action from the meme external evolution with probability  $C_p$ , else
16      perform action from the meme internal evolution*/
17      The remaining steps are the same as those in Algorithm 1.
17      /*End meme internal evolution*/
18 End

```

Figure 3.3: The fully general multi-agent scenario with the proposed eTL paradigms. Agents employ FALCON or BP as the learning machine in their mind universes and learn the domain knowledge for performing appropriate actions given environmental states. They may also interact directly as denoted by the arrows connected between agents.

an agent $agt(c)$ identifies the teacher agent $agt(s)$ via *meme selection*. Then, agent $agt(c)$ shall undergo the action predicted using *meme internal evolution* or *meme external evolution* according to probability C_p :

$$C_p(agt(c)) = 1 - \frac{Fitness(agt(c))}{Fitness(agt(s))} \times \frac{Si(agt(c))}{Si(agt(s))}, \quad (\text{Eq. 3.18})$$

where $Fitness(agt(c))$ is the fitness of agent $agt(c)$. $Fitness(agt(s))$, on the other hand, denotes the fitness of the selected teacher agent, $Si(agt(c))$ is the similarity value of current agent $agt(c)$ for the given states, $Si(agt(s))$ is the similarity value of the teacher agent. Thus, each agent undergoes the action provided by *meme external evolution* with probability C_p . During *meme external evolution*, *meme variation* occurs at a probability of v to maintain the diversity towards agents' selection for *meme transmission*.

3.3 Empirical Study

In order to study the effectiveness and efficiency of the proposed evolutionary transfer learning approach, the experiment firstly validates eTL on a commonly used Minefield Navigation Tasks (MNT) [69]. Subsequently, a well-known commercial first person shooter game, namely “Unreal Tournament 2004” (UT2004) [70], is used to investigate the performance of eTL under complex game scenarios.

3.3.1 Experimental Platform - Minefield Navigation Task

In MNT, the goal of each unmanned tank (or agent) is to navigate across a minefield comprising of randomly positioned mines, other tanks and target within a given stipulated time frame successfully. Fig. 4.2 depicts an overall view of the MNT environment. The unmanned tanks that spawn randomly within the field at the beginning of each mission are equipped with sensors, thence they are able to get access to a set of detections, including mine detection, agent detection and target bearing. Further, each tank in MNT possesses sensors which have a forward view of 180° , including left, left oblique, front, right oblique and right.

According to the input state, each unmanned tank performs one of the five possible actions at each step: $A = \{LEFT, LEFT FRONT, FRONT, RIGHT FRONT, RIGHT\}$. Particularly, an unmanned tank is rewarded with a positive value of 1 when the tank arrives at the target. Or, if the tank hits a mine, collides with other tanks or does not reach the target within the given time frame, the tank will be assigned with zero reward. According to the Q-learning lemma, negative feedback signal is not used in this reward schema, which guarantees the Q value is always within the desired boundary $[0, 1]$. In the experimental study, there is a total of 10 mines and 1 target (known as the red flag) that are randomly generated over missions in a 16×16 minefield. A mission completes when all tanks reach the target (success), hit a mine or collide with another tank (failure), or exceed 30 time steps (out of time), as considered in [32][63].

Table 3.1: Summary of the parameter configuration in the proposed eTL

FALCON Parameters		
Choice Parameters ($\alpha^{c1}, \alpha^{c2}, \alpha^{c3}$)	(0.1, 0.1, 0.1)	
Learning Rates ($\beta^{c1}, \beta^{c2}, \beta^{c3}$)	(1.0, 1.0, 1.0)	
Contribution Parameters ($\gamma^{c1}, \gamma^{c2}, \gamma^{c3}$)	(0.5, 0.5, 0)	
Baseline Vigilance Parameters ($\rho^{c1}, \rho^{c2}, \rho^{c3}$)	(0.2, 0.2, 0.5)	
BP Parameters		
Learning Rate η	0.25	
Momentum Factor τ	0.5	
Gain of Sigmoid Function $Gain$	1.0	
Temporal Difference Learning Parameters		
TD learning rate ϑ	0.5	
Discount Factor γ	0.1	
Initial Q-value	0.5	
ϵ -greedy Action Policy Parameters	FALCON	BP
Initial ϵ value	0.5	0.5
ϵ decay rate	0.0005	0.00025
Transfer Variation Parameters		
Degree of Randomness λ	0.1	
Frequency of Variation ν	0.1	

KN denotes the average number of knowledge transfer events happening between agents. In general, a higher number of knowledge transfer implies higher computational costs incurred to perform transfer learning among agents.

3.3.3 Performance of eTL using FALCON or BP as Learning Agents.

In this subsection, the objective of the present experimental study is to investigate the performance of the proposed eTL with FALCON or BP learning agents on completing the Minefield Navigation Tasks. Specifically, the MAS is composed of six FALCON or BP agents in the 16x16 size map of the Minefield Navigation Task platform where the learning architecture of MAS is discussed in Section II-B. Further, the interaction and implementation of learning agents are presented in Section III. In this study, 60 memotypes are configured in BP to achieve the best success rate while maintaining a low space complexity [32]. To maintain consistency with previous studies [32][63][37], the designed experiments conduct 30 sets of independent

experiments and report the average performances of 1) FALCON agents at 100-mission intervals over a total of 2000 missions and 2) BP agents at 1000-mission intervals over a total of 20000 missions, respectively.

Moreover, two state-of-the-art TL approaches for MAS, namely Advice Exchange (AE) model [21] and Parallel Transfer Learning (PTL) [23], are considered here for the purpose of comparison. In AE, the agents, or advisees, learn from better performed advisors based on the “imitate-from-elitist” selection strategy. More specifically, an agent $agt(c)$ with poor performance will seek advice from the best performing agent $agt(b)$ if $(q_c < dq_b)$, where q_c denotes the average quality of agent $agt(c)$, q_b is the average quality of agent $agt(b)$ and d is a user defined discount parameter in the interval of $[0, 1]$. For fair comparison, this experiment investigates various values of the discount factor, particularly, for i) a small value 0.1 (AE-0.1), ii) a medium value 0.5 (AE-0.5) and iii) a large value 0.9 (AE-0.9), respectively, and reports the average success rates of the agents (AE-AVG). Fig. 3.5 depicts the average success rates of agents under differing AE models. Notably, AE models with different discount factors reported distinct success rates. Further, PTL is a TL method where the current agent leverages from the knowledge broadcasted by the others in the environment. In particular, if an agent has useful information to share, it shall broadcast the information to all other agents. Meanwhile, the current agent will also check its communication buffer to determine if any information has been received and then decides whether to accept the information or discard it.

The complete results pertaining to the success rate (SR), generated memotypes (GM) and interval-unit knowledge transfer numbers (KN) of both BP and FALCON agents under Conv. M, Advice Exchange model (AE), Parallel Transfer Learning (PTL) and the proposed eTL are summarized in Table 3.2 and Figs. 3.6-3.7. The detailed analyses of the obtained results shall be discussed comprehensively in what follows.

Figure 3.5: SRs of agents under AE models with discount factor 0.1, 0.5 and 0.9 on completing the missions in MNTs.

Table 3.2: Performance comparison among eTL, existing TL approaches and Conv. M (SR: success rate, GM: generated memotypes, KN: knowledge transfer numbers)

#	Scenarios	FALCON agents			BP agents		
		SR	KN	GM	SR	KN	GM
1	eTL	71.3	324	188	64.2	1715	60
2	PTL	59.8	131	191	56.4	178	60
3	AE-0.1	63.3	24	152	54.6	59	60
4	AE-0.5	65.4	58	158	60.7	396	60
5	AE-0.9	69.0	428	168	61.4	4959	60
6	AE-AVG	65.8	170	159	58.8	1797	60
7	Conv. M	61.3	0	144	44.5	0	60

3.3.3.1 Comparison of MASs with and without TL

The success rates obtained by the MAS with and without TL approaches are given in the SR columns of Table III. Their corresponding learning trends are also depicted in Fig. 3.6. It is notable that most TL approaches outperform Conv. M. This is attributed to the TL approaches which endow agents with capacities to benefit from the knowledge transferred from the better performing agents, thus accelerating the learning rate of the agents in solving the complex task.

more efficiently and effectively. Agents in Conv. M, on the other hand could only undergo learning via personal grooming. To illustrate the efficacy of agents in the eTL framework in solving the minefield navigation problem, Fig. 3.7 depicts the sample snapshots of the navigated routes made by the FALCON agents. As can be observed, at the early learning stages, such as 1 and 500 learning missions, the FALCON agents have high probability to hit the mines or collide with other agents. While learning progresses, these agents tend to exhibit higher success rates.

Figure 3.6: SRs of FALCON or BP agents under eTL, PTL, AE-AVG and Conv. M on completing the missions in MNT.

3.3.3.2 Comparison of eTL against other state-of-the-art TL approaches

When compared to the state-of-the-art TL approaches, the proposed eTL is shown to achieve superior performance in terms of success rate throughout the learning process. In particular, FALCON and BP agents with the proposed eTL reported approximately 11.5% and 7.8% improvements in success rate, respectively, over PTL at the end of missions (see Table III, column SRs). This is due to the reason that, when deciding whether to accept the information broad-

Figure 3.7: Snapshots of the FALCON agents’ navigation routes on completing 1, 500, 1000 and 2000 learning missions in Minefield Navigation Task. The objective of each FALCON agent (denoted as a tank in the figure) in a learning mission is to arrive at the target successfully by navigating across the minefield safely and within the allocated time span. Thus a learning mission completes when all tanks reach the target, hit a mine or collide with another tank, or exceed the given time steps. All tanks and mines are randomly generated for each learning mission hence different learning missions have unique navigational routes.

casted by the others, agents in PTL tend to make incorrect predictions on previously unseen circumstances.

Further, the proposed eTL also demonstrated superiority in attaining higher success rates than all AE models. As discussed in Section 3.2.4, this can be attributed to the *meme selection* operator of the proposed eTL, which considers a fusion of the “imitate-from-elitist” and “like-

attracts-like” principles so as to give agents the option of choosing more reliable teacher agents over the AE model. Moreover, it is worthy noting that the AE model with a discount factor of 0.9 (labeled as AE-0.9) attained a success rate that is the closest to that of the proposed eTL. However, AE-0.9 incurred a higher knowledge transfer numbers (i.e., computational efforts) during the learning process (see Table III, column KNs).

Moreover, in order to quantitatively evaluate how effective the memotypes are generated in FALCON agents under different TL approaches, the *effectiveness ratio* is defined as follows:

$$Ratio(TL) = GM / (SR(TL) - SR(Conv.M)), \quad (\text{Eq. 3.19})$$

where $(SR(TL) - SR(Conv.M))$ denotes the improvements on success rate of the learning agents exhibited by the TL approaches against Conv. M. In general, a smaller positive *Ratio* value is preferable, which implies a higher knowledge generalization performance.

Table 3.3: The effectiveness of generated memotypes in FALCON agents under the eTL and other existing TL approaches

#	Metrics	FALCON agents		
		eTL	PTL	AE-AVG
1	Generated Memotypes	188	191	159
2	Success Rate Improvement	10.0	-1.5(1)	4.5
3	Memotype Effectiveness	18.8	191	35.3

Table 3.3 summarizes the effectiveness ratio of the memotypes generated in FALCON agents under different TL approaches at the end of the learning missions. Notably, the proposed eTL reported the most effective memotypes in FALCON agents since it attained the smallest effectiveness ratio among all the three TL approaches.

3.3.4 Performance of Proposed eTL using Heterogeneous Agents

In the previous subsection, a study on MAS with a homogenous setting has been considered, i.e., all agents in the system are assumed to be uniform and bear the same learning machine.

This subsection further considers the complex realistic scenario involving a diversity of heterogeneous agents in a MAS. In particular, a mixture of three FALCON agents and three BP agents are employed in the heterogeneous MAS for solving the Minefield Navigation Tasks. Notably, the architecture of MAS and the interaction among heterogeneous learning agents are discussed in Section III.

In the heterogeneous MAS, knowledge transfer among agents is more complex than that in the homogenous MAS. This is because FALCON and BP agents possess unique learning structures, hence they have different learning capabilities, including specialized action prediction rules, learning speed, etc. As such, the knowledge identified by TL approaches working well in homogenous MAS might become futile and even detrimental to the agents with dissimilar learning structures. For instance, in the heterogeneous MAS where both FALCON and BP agents are configured using the PTL as the TL approach, BP agents will broadcast the useful knowledge to all other agents. Considering that the FALCON agents learn much faster than BP agents (as discussed in Section 3.3.3), these knowledge broadcasted from BP agents could always lag behind that of FALCON agents, and hence deteriorate the performance of FALCON agents as depicted in Fig. 3.8(a). Thus, in this subsection, experiments are further conducted to investigate the performance of the proposed eTL in the heterogeneous MAS setting.

Table 3.4: Comparison among eTL, existing TL approaches and Conv. M in heterogeneous MAS (SR: success rate, GM: generated memotypes, KN: knowledge transfer numbers).

#	Scenarios	FALCON agents			BP agents		
		SR	KN	GM	SR	KN	GM
1	eTL	69.7	310	168	68.2	1189	60
2	PTL	59.7	93	167	63.1	267	60
3	AE-0.1	65.5	27	144	60.1	33	60
4	AE-0.5	65.5	62	154	66.7	266	60
5	AE-0.9	67.3	423	158	68.2	4700	60
6	AE-AVG	66.1	171	152	65.0	1667	60
7	Conv. M	65.2	0	145	49.1	0	60

The success rate (SR), knowledge transfer numbers (KN) and generated memotypes (GM) of both FALCON and BP agents obtained under Conv. M and TL approaches are presented in

Table 3.4. Moreover, the learning trends in terms of success rate are depicted in Fig. 3.8.

Notably, FALCON agents under AE and PTL learning approaches often succumbed to poor success rate in the long run. According to the average success rate attained by AE models with discount factor of 0.1, 0.5 and 0.9 (see Figure 3.9), competitive success rates of FALCON agents to that of Conv. M is reported. On the other hand, under the PTL scheme, a significant drop of 5.5% in success rate as compared to Conv. M is observed at the end of the learning process (see Table 3.4, column SRs). As aforementioned, this is a result of the blind reliance on the knowledge transferred from the BP agents that misled the FALCON agents into performing erroneous acts under the heterogeneous MAS environment.

Figure 3.8: SRs of FALCON and BP agents under eTL, PTL, AE-AVG and Conv. M on completing the missions in the heterogeneous MAS.

On the other hand, the proposed eTL significantly outperforms the counterparts, for both FALCON and BP agents, throughout the entire learning process. Specifically, at the end of the learning process, it attained the highest success rate of 69.7% and 68.2% for FALCON and BP agents, respectively (see Table 3.4, column SRs). Particularly, according to the result in Fig. 3.8, FALCON agents are noted to learn much faster than BP agents. Therefore, the

Figure 3.9: SRs of FALCON and BP agents under AE models with discount factor 0.1, 0.5 and 0.9 on completing the missions in the heterogeneous MAS.

knowledge transferred from FALCON agents is helpful for complementing the learning ability of BP agents, thus enhancing the SRs of BP agents significantly on the MNT problem. The result thus further highlights the efficacy of the *meme selection* strategies in the eTL in the choice of reliable and efficient information while reducing blind reliance in transfer learning, specifically in the context of the heterogeneous MAS.

Moreover, the experiments are investigated on KNs and GMs of FALCON and BP agents in both homogenous and heterogeneous MNTs. As can be observed in Tables 3.2 and 3.4, PTL reports the least KNs, significantly less than those of eTL and AE-AVG. However, although PTL generates far less KNs, it generates almost the most number of memotypes in FALCON agents among all TL approaches. This indicates that the transferred knowledge in PTL tends to generate too much redundant and possibly detrimental memotypes, hence leading to the poor success rates of FALCON agents. On the other hand, AE-AVG reports the smallest number of memotypes among all TL approaches for FALCON agents. Particularly, according to Table 3.4, the AE approach with a discount factor of 0.9 (labeled as AE-0.9) generates around

158 memotypes and incurs a higher knowledge transfer number (423). eTL, on the contrary, generates lower KNs, while still obtaining both higher numbers of memotypes and success rates in FALCON agents. The obtained result thus highlights the efficiency of eTL in making use of the transferred knowledge, while generating useful memotypes in FALCON agents.

In summary, the performance of the proposed eTL in the heterogeneous MAS is consistent to that of the homogenous settings. This demonstrates the superiority of the proposed eTL in producing higher success rate against existing TL approaches in both homogenous and heterogeneous MASs.

3.3.5 Performance of Proposed eTL in Unreal Tournament 2004

Further, the performance of the proposed eTL is testified in the popular but complex commercial first-person shooter video game - “Unreal Tournament 2004” (UT2004) which provides an environment for embodying virtual artificial intelligent agents. UT2004 has many built-in game scenarios, and among them the most commonly played ones are known as the DeathMatch, Domination, and Capture The Flag (CTF) scenarios. Over the past decades, first-person shooter computer games have attracted dramatic attentions and garnered a large part of the multibillion dollars computer game industries. Notably, the modeling of autonomous non-player characters is of great importance for the commercial success since it makes the game more playable, challenging and more importantly, it has been deemed to be useful in improving the players degree of satisfaction. Therefore, this section further considers a practical game problem where eTL is applied to model the non-player characters in UT2004.

To customize different game scenarios, UT2004 provides a scripting language namely UnrealScript, which can be coded by users to easily create maps, game models, agents’ algorithms and agents’ actions, etc. Further, to apply the FALCON algorithm for agents in Unreal Tournament 2004, the Pogamut [82] which is an integrated open source platform to facilitate development of behavior for virtual agents embodied in the 3D environment of Unreal Tournament

2004, is used in the present context. The architecture of Pogamut is described in Fig. 3.10. Pogamut is developed as a plugin for NetBeans java development environment. It can communicate with UT2004 by a generic binding called GameBots 2004 (GB2004) [83]. GB2004 provides network TCP/IP text protocol for getting information from the game environment, and allow users to connect to UT2004 and control in-game avatars employing the client-server architecture. Moreover, as GB2004 only exports and imports text messages, a java library – “Gavialib” is adopted for Pogamut to automatically convert text information into java objects and debug the virtual agent remotely through JMX protocol [84] and Pogamut NetBeans Plugin.

Figure 3.10: Pogamut architecture.

Fig. 3.11 depicts a snapshot of the running game taken from the judge view, which showcases the shootout features in UT2004, such as weapons, armour or medical kits, etc. In UT2004, all autonomous robots that spawn randomly within the environment are equipped with a set of sensors. Taking the information captured by sensors as input states, they could learn and utilize the association of their current states, behavior selections and rewards. Table 3.5 illustrates a sample knowledge of FALCON combatants during the training process. For each state, the autonomous robot performs a set of pre-defined combat actions with some rule-based heuristics.

Figure 3.11: A snapshot of the “Unreal Tournament 2004”.

The immediate reward thus helps to update the knowledge in mind universe. In this set experiments, five behavioral actions for TD-FALCON combatants are designed. In particular, the specific actions in the action vector $\mathbf{A} = (EXPLORE, ITEMS, ENGAGE, MEDKIT, ROTATE)$ are respectively described as follows:

- (i) *EXPLORE* action, where a FALCON robot wanders around the map randomly.
- (ii) *ITEMS* action, where a FALCON robot finds and picks up the nearest visible items, such as weapons, armors, ammo, etc.
- (iii) *ENGAGE* action, where a FALCON robot pursues and eliminates the enemies.
- (iv) *MEDKIT* action, where a FALCON robot finds the medical kit to recuperate.
- (v) *ROTATE* action, where a FALCON robot rotates around the spot.

The state vector of an autonomous robot comprises 18 variables encoded by boolean or discretized number within $[0, 1]$. Assigned by groups, these 18 variables are of 9 pairs wherein each pair has a base status and the corresponding complement (i.e., x and $1 - x$ as a pair,

$x \in [0, 1]$). The base status include the health level of an autonomous robot, whether a robot is damaged, whether enemies are spotted, whether a robot has adequate ammo, and another five boolean variables to indicate the current behavior of a combatant by means of archiving previous states and actions.

Table 3.5: Sample Knowledge of the Autonomous Robots

<i>STATE</i>	$S = (1.0, 0.0, 0, 1, 1, 0, 0.4, 0.6, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1)$
IF	health is 1.0, and not being damaged, and opponent is in sight, and has 40 percent of ammo, and currently in collecting item state.
THEN	go into engaging fire state
WITH	reward of 0.768

The aim of an autonomous robot is to win the game by getting higher reward. The reward vector only comprises two variables, namely the reward and its complement. Defining 0.5 as a successful hit and 1 as a successful kill, the reward system is simple for fast learning. Defining a neutral value 0.5 as a successful hit and a high value 1 as a successful kill, the reward system is simple for fast learning.

In the design of this experiment, two groups of six armed robots are firstly initialized in the region of interest. Each armed robot is equipped with a set of sonar sensors and weapons. One group of armed robots are labeled as *Hunters* and each of them has a mind universe that is governed by human defined rules [70]. The second group comprises autonomous armed robots with FALCON as the online learning machine. Under the simple “DeathMatch” scenario, two groups of armed robots fight against each other by making use of the available resources, including weapons, medical kits, armors, etc. When the armed robots are killed by the enemies, a new one will spawn randomly in their corresponding base. The battle repeats until each group reaches a minimum missions of 200, wherein one mission completes only if an armed robot in the group is killed by the enemies. To evaluate the performance of the proposed eTL in UT2004 more quantitatively, the kill rate (KR) of game characters has been defined by the

percentage of enemy kill numbers over agent death numbers [70]:

$$KR = \frac{1}{N_t \times N_c} \times \sum_{i=1}^{N_t} \sum_{m=1}^{N_c} \frac{N(k)_m^i}{N(d)_m^i}$$

where N_t is the simulation numbers, N_c denotes the number of participating combatants in each team, $N(k)$ indicates the number of kills and $N(d)$, on the contrary, is the number of deaths. Thus, K_c is a good measurement of the averaged fighting performance played by different teams. Typically, a higher KR is preferred since it indicates that a game character could kill more enemies while is less killed by the others.

Figure 3.12: KR of FALCON robots under the Conv. M and the proposed eTL fighting against the rule-based *Hunters*.

Fig. 3.12 summarizes the competitive performance of rule-based *Hunters* and FALCON robots under both the Conv. M and proposed eTL, in terms of KR. Notably, at the early learning stage, the rule-based *Hunters* have significant higher KR than FALCON robots since the rules are well-designed for *Hunters* while FALCON robots suppose to randomly explore the

environment at the beginning of the learning process. However, the KR of *Hunters* continues to decrease as the FALCON robots learn during combats. After 200 missions, the KR of the FALCON team under Conv. M is approximately 10% lower than that of the *Hunters*. FALCON robots under the proposed eTL, on the other hand, report significant higher KR than those under Conv. M throughout the learning process. Particularly, FALCON robots with eTL attained KR of 1.0 within 200 missions. The improved performance obtained by the proposed eTL demonstrates its efficacy in improving the combat performance of FALCON robots via the evolutionary knowledge transfer in the first-person shooter computer game - “Unreal Tournament 2004”.

3.4 Summary

This chapter proposes an eTL framework, which is governed by several meme-inspired evolutionary mechanisms, namely *meme representation*, *meme expression*, *meme assimilation*, *meme internal evolution* and *meme external evolution*. In particular, comprehensive designs of the meme-inspired operators and two realizations of the learning agents in eTL that take the form of the (TD)-FALCON and the classical BP multi-layer neural network, respectively, are presented. The performance efficacy of the proposed eTL is investigated via comprehensive empirical studies and benchmarked against both Conv. M without knowledge transfer and two state-of-the-art MAS TL approaches (i.e., AE Model and PTL), on the widely used MNT platform, under homogenous as well as heterogeneous MASs settings. The superior performance achieved by eTL confirms its efficacy in conducting evolutionary knowledge transfer in MAS, hence suggests the basis for greater research in developing evolutionary TL strategies. Also, a well-known first person shooter game, namely UT2004, is used to demonstrate the effectiveness of the proposed eTL under the complex problem solving scenarios.

Chapter 4

Evolutionary Transfer Learning with Behavior Prediction

As discussed in Chapter 2-3, transfer learning (TL) has gained increasing attentions for developing intelligent agents in multi-agent system (MAS) recently. However, existing study has to date focused on simple multi-agent scenarios where all agents have the same or share similar action spaces, and pursue a common objective. In multi-agent cases where each agent has unique objectives, the learning process of multiple agents becomes more complex and current TL approaches do not cope well. This is because agents are expected to be aware of the thoughts of their opponents so that efficient strategy can be developed to complete the mission successfully [85][86]. Searching for an efficient interactive strategy is thus a challenging task since its effectiveness greatly depends on the behavior of the peers or opponents involved. Therefore, endowing agents with capacities to identify the strategies, capabilities or models of their opponents that are present in the environment would benefit problem solving.

Taking this cue, the interest of the current work lies in the development of TL in more complex and realistic MASs where multiple agents have different or even competitive objectives. Specifically, this chapter presents an enhanced multi-agent TL framework where subject agents possess capability to improve their learning performance through predicting the behavior of their opponents. The essential backbone of the proposed approach is the eTL framework [22] which is driven by a series of memetic mechanisms [76] that are inspired by the theory

of Universal Darwinism and Dawkins' notion of meme(s) [87]. Taking advantage from the meme-inspired social knowledge transfer mechanisms, agents are shown to leverage from the useful knowledge transferred from their partners of high payoffs, hence leading to improved agents' performance and more efficient problem-solving .

Further, a well-established temporal difference (TD) - Fusion Architecture for Learning and Cognition (FALCON) [32] is employed as the connectionist reinforcement learning agents in the present study. In particular, the core contributions of this chapter can be summarized as follows:

- (i) A novel evolutionary multi-agent TL framework (eTL-P) is proposed for modeling subject agents in competitive multi-agent settings wherein subject agents are able to predict the behavior/actions of their opponents by building and solving candidate models automatically given the available data instances recorded from the historical interactive activities.
- (ii) eTL-P integrates social selection mechanisms for subject agents to identify their better performing partners in the environment while online. This leads to enhancement in the learning performances and a reduction in the complexity of behavior prediction since subject agents benefit from the useful knowledge which they leverage from their partners' mind universes (i.e., both from the internal connectionist learners and predictive candidate models of opponent agents).
- (iii) To validate the efficacy of the proposed eTL-P, comprehensive empirical studies are conducted on a Partner-Opponent Minefield Navigation Task (PO-MNT) involving both subject agent partners and their competitive opponents. Empirical results show that eTL-P improves the learning effectiveness and efficiency of subject agents in the complex PO-MNT scenarios.

The rest of the chapter is organized as follows. Section 4.1 discusses the comprehensive details of the proposed eTL-P, which is composed of behavior prediction, and multi-agent TL scheme. Subsequently, the empirical study of the proposed eTL-P is investigated on an adapted MNT in Section 4.2. Last but not least, Section 4.3 presents the brief concluding remarks of this chapter.

4.1 Proposed eTL-P Framework

This section presents the proposed eTL-P which endows subject agents with capability to interact with their opponents effectively by predicting their strategies, in addition to the usual ability of competency of learning from friendly teachers agents. The comprehensive details of eTL-P is depicted in Fig. 4.1. Particularly, model-based behavior prediction approach proceeds to predict the behaviors of opponent agents based on the predictive candidate models trained using historical data instances and multi-agent TL enhances the learning capability of subject agents by leveraging beneficial knowledge transferred from their partner subject agents. In what follows, the objectives of multiple agents in a competitive multi-agent environment and the detailed realization of each components in the proposed eTL-P are described.

4.1.1 Objectives of Multiple Agents in Competitive MAS

The reinforcement learning process of a single subject agent in eTL-P is formulated as a sequence of markov decision processes (denoted as MDPs) which continue until the stopping criteria, such as mission numbers or fitness levels, are satisfied. All agents are equipped with a set of sensors so they have access to the environment states. Depending on the obtained states and the individual's knowledge, a subject agent learns to select and perform the most appropriate action, and updates their mind universes under the motivation of receiving positive rewards in the future. During the learning process, each subject agent encounters and interacts with other agents who may be potential partners (agents share a common objective), or hostile

Figure 4.1: An illustration of the proposed eTL-P.

opponent agents (agents have competitive objectives and possess mind universes that are unknown to none other in the environment including the subject agents). In each time step, all subject and opponent agents operate in parallel within a common environment.

4.1.2 Model-based Behavior Prediction

In a competitive multi-agent environment, a subject agent requires an efficient strategy to counter opponent agents in order to complete the missions successfully. However, since a subject agent has no prior knowledge about the opponents, the design of suitable strategies to complete the missions can be non-trivial. The task of interest here is thus to endow subject agents with capabilities to predict the expected behaviors of the opponents.

In the literature, behavior prediction approaches can be broadly divided into two categories

as model-based and model-free. Model-free approaches predict agents' behavior directly from available training data [37], [88]. Typically, such approaches are computationally efficient and elegant, but usually require a lot of experiential data to learn and function comprehensively. On the other hand, model-based approaches conduct the prediction by working with a set of candidate models [89], hence they are often deemed as more computationally intensive than model-free approaches, but have the advantage of requiring significantly less data to function well.

This work considers the model-based approach for predicting the behaviors of opponent agents due to the scarcity of data availability in the environment under study. Particularly, a subject agent maintains a number of potential candidate prediction models to choose or generalize since the true model is not always approximated with high certainty. Further, each candidate model is attributed with some level of confidence on its prediction accuracy. The model confidence level is then updated along with the learning process, according to how well the candidate models performed on predicting the behaviors of the opponent agents. This ensures that subject agents can adapt accordingly whenever the opponent agents change their strategies on-line.

Algorithm 5 summarizes the details of model-based behavior prediction process of subject agents. To begin, all of the candidate models $M = \{m_1, m_2, \dots, m_l\}$ are expected to be trained using the historical data instances of opponent agents (lines 2-5). Notably, the instance-based model training method is chosen since it has been widely used for training RL models in the literature [90], [18]. Typically, data instances are formulated as a sequence of $\langle S, A, R \rangle$ tuples. Given each tuple, a candidate model learns to associate the state, action and reward and assimilates them into their internal learning structures. Once the candidate model training process completes, each subject agent is attributed with a set of candidate models and then starts to undergo self learning independently.

The behavior prediction of opponent agents occurs whenever a subject agent identifies an opponent agent $agt(o)$ nearby during the self learning process of subject agents (line 12). Once

Algorithm 5: Learning with Behavior Prediction

Input: Candidate models $M = \{m_1, m_2, \dots, m_l\}$, Collected data $D = \{D_1, D_2, \dots, D_v\}$,
 where $D_v = \langle S_v, A_v, R_v \rangle$, Action set A
Output: Subject agents $\{agt(c)\}$

```

1 Begin:
2 for all  $M_l \in M$  do
3   for all  $D_v \in D$  do
4     Set input vector  $\langle S_v, A_v, R_v \rangle$  in  $M_l$ 
5     Training with vector  $\langle S_v, A_v, R_v \rangle$ 
6 Generate initial subject agents  $\{agt(c)\}$  and their opponents  $\{agt(o)\}$ 
7 Assign each subject agent with candidate models  $M$ 
8 while the stopping criteria are not satisfied do
9   for each current subject agent  $agt(c)$  do
10    Perform meme internal evolution process
11    /*Behavior Prediction*/
12    if detect an opponent agent  $agt(o)$  then
13      if  $Rand > scheme(\epsilon)$  then
14         $m_s = Random\{m_s : \text{for all model } M\}$ 
15      else
16         $m_s = Max\{conf(m_s) : \text{for all model } M\}$ 
17      for each available action  $a_i \in A$  do
18         $state(agt(c))' = Virtual\_Perform(agt(c), a_i)$ 
19         $a_o = Predict(m_s, state(agt(c))')$ 
20         $state(agt(o))' = Virtual\_Perform(agt(o), a_o)$ 
21        if fail interaction after virtual move then
22          Discard  $a_i$  from action set  $A$ 
23      Get  $a_s$  from  $A$  with highest reward of self learning in  $agt(c)$ 
24      Continue meme internal evolution process
25   Update  $conf(m_s)$ 
26 End

```

an opponent agent is detected, an ϵ -greedy model selection scheme proceeds to instruct how subject agent chooses an appropriate one from all attributed candidate models. This scheme aims to balance the fundamental trade-off between *exploitation*, i.e., sticking to the most believed model, and *exploration*, i.e., trying out other seemingly complementary models. Specifically, subject agent with ϵ -greedy scheme selects a candidate model m_s of the highest confidence level $conf(m_s)$ with probability $1 - \epsilon$ ($0 \leq \epsilon \leq 1$), or otherwise chooses a random model with a probability of ϵ (lines 13-16). The value of ϵ decays gradually with behavior prediction time frames of subject agents.

Upon selecting a candidate model successfully, subject agent $agt(c)$ then predicts its virtual state $state(agt(c))'$ by attempting each available action a_i from the action set A (line 18). Furthermore, the state $state(agt(c))'$ is used to infer the corresponding virtual state $state(agt(o))'$ of opponent agent $agt(o)$. By activating the selected model m_s , subject agent $agt(c)$ is able to predict the action a_o of $agt(o)$ given $state(agt(o))'$ (line 19). Based on the results after virtual performing a_o by $agt(o)$, subject agent then decides whether to discard an action from the available action set A (lines 21 and 22). Subsequently, agent $agt(c)$ proceeds to select an action a_s from the updated A with the highest reward yield (line 23) and continues the self learning process. In addition, the m_s leading to the successful behavior prediction is updated with a higher confidence $conf(m_s)$ of being chosen for the subsequent behavior prediction process (line 25):

$$\begin{cases} conf(m_s) = conf(m_s) + Reward, & \text{if succeeds} \\ conf(m_s) = conf(m_s) - Penalty, & \text{if fails} \end{cases}$$

where reward and penalty are defined as positive integers and $conf(m_s) \in [-100, 100]$.

4.1.3 Multi-agent Transfer Learning

To date, most of the existing works have focused on reducing the candidate model complexity for decision making problems, where a single agent predicts the behavior of another single one, but have yet to study the behavior prediction in a multi-agent setting [91]. Differing

to existing approaches, a multi-agent TL approach is further proposed for reducing subject agents' candidate model complexity with respect to the model space while also enhancing their learning capability.

The TL process between agents with unique learning capabilities is mainly driven by a human-like imitation process. In the multi-agent learning problems, all of the subject agents learn in the same environment and share a common behavioral action space. Therefore, the imitation-driven knowledge transfer process offers the advantage where imitating agents behave at approximately the same level of performance as their target of imitation. Further, when multiple candidate models are available, subject agents with the proposed TL approach are not required to train all the models and they can acquire increasing level of predictive capability from their partners by sharing predicting information with respect to their unique candidate models. TL approach thus reduces candidate model complexity of subject agents by attributing them with a comparatively smaller subset of candidate models.

Algorithm 6 summarizes the multi-agent TL approach in eTL-P. To begin, each subject agent is attributed with a set of candidate models $M^{agt} = \{M_1^{agt}, \dots, M_q^{agt}\}$, where agt indicates the index of subject agent and q is the number of models in M^{agt} (line 2). Further, the current agent $agt(c)$ is expected to check whether there exists a better-performing teacher agent $agt(s)$ that has a higher fitness value $Fit(agt(s))$ (line 4). Once a teacher agent is identified, the current agent $agt(c)$ first passes its state $state(agt(c))$ to the teacher agent $agt(s)$ (line 6). Meanwhile, if agent $agt(c)$ detects an opponent agent $agt(o)$ nearby, it further transfers the inferred state $state(agt(o))$ of opponent agent to the teacher $agt(s)$ (lines 7-8). By simulating the states $state(agt(c))$ and $state(agt(o))$, the teacher agent $agt(s)$ undergoes behavior prediction of agent $agt(o)$ based on its own attributed model M^s and further expresses a predicted action a_s , which is observable and can be transmitted to agent $agt(c)$ (referred to Algorithm. 5). Then, if the transmitted action a_s is available, agent $agt(c)$ is expected to imitate and assimilate the action a_s into its own mind universe (line 13). In addition, a variation process with a probabil-

ity ν is defined to keep the innovative diversity of selected actions therefore preventing agent $agt(c)$ from learning from teacher agents blindly during the multi-agent TL process (line 12).

Algorithm 6: Multi-Agent Transfer Learning

Input: Candidate models $M = \{M^1, M^2, \dots, M^{agt}\}$, where $M^{agt} = \{m_1^{agt}, \dots, m_q^{agt}\}$

- 1 **Begin:**
- 2 **Allocate** each subject agent with candidate model M^{agt}
- 3 **for** each subject agent $agt(c)$ **do**
- 4 **Perform** Algorithm 5, line 10-22
- 5 **if** identify $\{agt(s) | Fit(agt(c)) < Fit(agt(s))\}$ **then**
- 6 **Get** $state(agt(c))$ of current $agt(c)$
- 7 **Pass** $state(agt(c))$ to teacher agent $agt(s)$
- 8 **if** detect an opponent agent $agt(o)$ nearby **then**
- 9 **Pass** $state(agt(o))$ to agent $agt(s)$
- 10 **Perform** Behavior Prediction with M^s
- 11 **Discard** failure actions from action set A
- 12 **Get** a_s from A with highest reward of self learning in $agt(s)$
- 13 **Perform** variation on a_s with probability ν
- 14 **Get** a_s from $agt(s)$, otherwise get a_s from A with highest reward of self learning by $agt(c)$
- 15 **Continue** left steps that are the same as learning process in Algorithm 5
- 16 **End**

4.1.4 Complexity Analysis

As aforementioned, predicting accurate behaviors of opponent agents is computationally intractable due to the complexity of candidate models and thus ways of mitigating the computational intractability are critically needed. Since the complexity is predominantly due to the space of candidate models, this work therefore focuses on reducing the model space of subject agents while avoiding a significant loss in optimality.

Specifically, in the present framework, the entire candidate models M are partitioned into different partial sets according to the number of subject agents: $M = \{m_1^1, m_2^1, m_3^2, \dots, m_l^{agt}\}$ where M^{agt} aggregates the candidate model set labelled by the same subject agent mark. Then,

subject agents will only train candidate models from their corresponding partial sets. According to the proposed demonstration learning process, these non-familial peer subject agents with unique candidate models M^{agt} could communicate socially and takes advantage from each other (see Alg. 6). Hence, the proposed MeMAS-P with multi-agent transfer learning can reduce the candidate model space of each subject agent in the multi-agent system.

4.2 Empirical Study and Analyses

In the literature, Minefield Navigation Task (MNT) has been regarded as a popular platform to testify the capacities of autonomous agents due to its simplicity and efficiency in problem-solving [22][32]. Taking this cue, the effectiveness and efficiency of the proposed eTL-P is investigated under more complex scenarios of the MNT (as depicted in Fig. 4.2) where multiple agents have shared or competitive objectives.

4.2.1 Experimental Platform and Configuration

In the classical MNT, subject agents (denoted by green tanks in Fig. 4.2, namely *navigators* hereafter) share a common objective, which is to navigate across the minefield so as to arrive at the target position (denoted by a red flag and is randomly generated per mission) within a specified time frame, while avoiding collision with any mines nor other agents. In contrast to the classical MNT where only subject agents with a common objective are involved, here the interest is on minefield navigation problems (namely PO-MNT in short) comprising both navigator partners and competitive opponents (denoted by red tanks, namely referred to *predator* hereafter) in the environment. Particularly, the predators are introduced with the objective of capturing navigators before navigators flee to the target place. Further, in the present design, predators are equipped with a thick armor that can never be destroyed by mines.

In addition, both navigators and their opponents share a common action space, which is defined as $A = \{LEFT, LEFT FRONT, FRONT, RIGHT FRONT, RIGHT\}$. They are equipped

Figure 4.2: Illustration of the adapted MNT (PO-MNT).

with a set of sensors and hence have access to observations about the minefield environment, including mine detection, navigator detection, predator detection and target bearing detection. Further, a navigator is rewarded with a positive reward of ‘1’ if it arrives at the target successfully. Otherwise, it is assigned with a ‘0’ reward. The experiment considers a crowd of 6 autonomous agents with 6 mines and 1 target position that are randomly generated across missions within a 12x12 minefield as the navigation environment. For each experiment, 30 independent simulation runs have been conducted and with each run involving a total of 5,000 randomly generated missions. One mission completes only if all navigators arrive at the target, hit the mines, exceed the maximum time steps of 30 step times or when they are apprehended by the predators.

The parameter configurations of (TD)-FALCON and the proposed eTL-P used in the present

Table 4.1: Parameter setting in the proposed eTL-P.

experimental study are summarized in Table 4.1. For the sake of a fair comparison, these configurations are maintained to be consistent with previous studies in [32]. To study the efficacy of the proposed eTL-P in solving the complex PO-MNT, comprehensive empirical results with respect to the following metrics are investigated:

- SR: the average success rate of agents on completing the missions;
- MT: the training time of candidate models that are assigned to navigators;
- PN: the number of prediction during the learning process;
- PT: the computational time cost for behavior prediction during the learning process.

4.2.2 Effectiveness of eTL

The experimental study begins with an investigation on the performance of an eTL with 6 FALCON navigators on completing the classical MNT. This serves as a baseline for comparison. Then, two current state-of-the-art multi-agent TL approaches, namely, the Advice Exchange (AE) mechanisms [21] and Parallel Transfer Learning (PTL) [23] are considered in the experiment for the purpose of comparison. Specifically, AE mechanisms facilitate poor-performing agents to learn from their elite partners by seeking its advice on given circumstances. On the other hand, agents in PTL learn from others by leveraging the knowledge broadcasted by all others.

The resultant success rates (SRs) of the navigators for the different above-mentioned TL approaches on completing the classical MNT are denoted in Fig. 4.3. In particular, the average SRs of all 6 FALCON navigators on completing the classical MNT across the increasing number of missions is depicted in the figure. It can be observed that the SRs of the navigators for

all TL approaches are noted to increase steadily right from the beginning of the navigation mission. Both eTL and AE outperformed the conventional MAS (Conv-MAS) as expected, since the latter has no mechanism for knowledge transfer. Further, eTL obtained superior SR over the other counterparts considered as the navigators are noted to learn better with increasing missions as compared to the existing state-of-the-art multi-agent TL approaches.

Figure 4.3: SRs of navigators under eTL, PTL, AE and Conv-MAS on completing the missions in classical MNTs.

4.2.3 Results of eTL-P with True Predator Model

In what follows, an investigation on the proposed eTL-P is studied in addressing more complex PO-MNT wherein navigators and predators have competitive objectives. Both navigators and predators employ FALCON with $\rho^k = (0.2, 0.2, 0.5)$ as the connectionist reinforcement learning models that form their mind universe. In this study, a mixture of 3 navigators and 3 predators is considered in the PO-MNT. Notably, predators are assumed to be trained before-

hand with the knowledge of apprehending their nearby navigators. Navigators, on the other hand, begins with zero knowledge but learn along with the missions undertaken. This set of experiments we assume all navigators have access to the true model of the predators. In this manner, the navigator agents are able to predict the actions of the predators accurately during their navigation. Such an assumption and experimental study is carried out here to assess and validate the motivation for endowing agents with capacities to infer predators' actions as a means to improve their problem solving capabilities under situations where agents possess differing goals or competitive objectives.

To begin, the performance efficacy of eTL-P is benchmarked against the conventional MAS (Conv-MAS) in which navigators have neither behavior prediction nor any transfer learning capability. Further, to study the effectiveness of each mechanism available in eTL-P, the performances of the agent navigators with only model-based behavior prediction capability (Mod-P), i.e., no TL capability, and agent navigators with only TL capability (eTL, referred to [22]), i.e., no behavior prediction capability, are also investigated, respectively.

Fig. 4.4 summarizes the performance of eTL, Mod-P, eTL-P and Conv-MAS in terms of their respective SR averaged at 100-mission intervals, across a total of 5,000 missions. Particularly, to identify the effect and significance of each mechanism in eTL-P, the performances of eTL (which assumes navigators with only TL capability) and Mod-P (which assumes navigator with only opponent behavior prediction capability) against the Conv-MAS (i.e., no form of TL nor behavior prediction of opponents) from 1,000 missions are firstly studied. Subsequently eTL-P is pitted against eTL from mission 2,000 onwards, where both the capabilities of transfer learning and true model of opponent behavior kick in, in the former.

The SR results obtained showed that the performance of the navigators in Conv-MAS deteriorates significantly in the PO-MNT, i.e., Minefield Navigation Task with friendly partners and offensive opponents as compared to that in the classical MNT where there are no opponents in the latter task (see Section 4.2.2). This is understandable since navigators in PO-MNT now

Figure 4.4: SRs of eTL, Mod-P, eTL-P and Conv-MAS on completing the missions in PO-MNTs.

suffer from bad encounters with predators, i.e., they can be apprehended by their opponents, as compared to the classical MNT. Mod-P, which has the behavior prediction capability of opponents on the other hand, exhibits significantly higher SR over the Conv-MAS. This indicates that, with the availability of behavior model of predators, navigators are shown to survive better in the PO-MNT. Notably, the results showed that navigators in the proposed eTL-P are observed to achieve superior performance in terms of SR at the end of the learning process, i.e., attaining a superior SR of around 63% at the end of 5,000 missions. This result highlights the efficacy of the proposed eTL-P in improving the learning performance of navigators on completing the PO-MNT successfully. Moreover, navigators in both eTL and eTL-P reported superiority in terms of SRs than one without the multi-agent TL scheme. This can clearly be attributed to the transfer learning scheme of eTL and eTL-P which endows navigators with capacities to benefit from the knowledge transferred from their better performing partners, thus

accelerating the learning speed and hence improving the SR of completing the missions.

4.2.4 Results of eTL-P with Predictive Predator Models of FALCON Dynamics

In the previous subsections, a study on eTL-P wherein each navigator is assumed to have access to the true model of the opponent behavior is considered. This subsection considers the more realistic scenario where navigators only have access to predictive candidate models of opponents as opposed to the true model of predators which is typically unknown in practice. Since both navigators and predators employ FALCON as their connectionist agents in the mind universe, here the predictive predator models are formulated as a set of FALCON dynamics with different baseline vigilance values $\rho^k \in [0, 1]$ and further generated with data instances of predators that are collected from the online activities of the missions encountered. This configuration is due to the great importance of baseline vigilance parameter as it controls the knowledge generalization level of FALCON dynamics which affect their performance significantly. After that, navigators shall predict the behavior of predators using proposed model-based behavior prediction approach in Section 4.1.2.

Given a set of predictive models, each navigator is assigned with a differing subset of all candidates. The performance of navigators with attributed candidate is then reported by Mod-P in which navigators only have opponent behavior prediction capability, i.e., no multi-agent TL scheme. Since navigators with unique predictive models are likely to perform differently, the efficacy of eTL-P is therefore validated in enhancing Mod-P by facilitating the knowledge transfer across multiple navigators. In addition, experiments have been also constructed where each navigator is assigned with all the possible predictive models (denoted by Mod-All). The aim in designing such experiments were to assess the efficacy of opponent behavior prediction in choosing an appropriate model from multiple predictive candidates during the navigators' learning process. Based on these experimental settings, this section studies the performance of

all mechanisms in eTL-P, which are comprising of model-based behavior prediction as well as multi-agent TL in what follows.

In this set of experiments, three FALCON models are employed as the predictive candidates, where each is configured with a differing vigilance level ρ^k , known as 1) a low parameter $\rho^k = [0.1, 0.1, 0.1]$, 2) a medium parameter $\rho^k = [0.4, 0.4, 0.4]$ and 3) a high parameter $\rho^k = [0.9, 0.9, 0.9]$, respectively. Hence, each navigators in Mod-P and eTL-P is attributed with a unique FALCON candidate for opponent behavior prediction. The complete results pertaining to the SR of the navigators on the PO-MNT are summarized in Table 4.2 and Fig. 4.5. The analyses of the obtained results shall be discussed comprehensively next.

Table 4.2: Performance comparison among Mod-P, eTL-P and Mod-All in terms of SR.

#	FALCON	SR		
	(ρ^1, ρ^2, ρ^3)	Mod-P	eTL-P	Mod-All
1	(0.1,0.1,0.1)	47.25	55.85	56.54
2	(0.4,0.4,0.4)	50.93	56.15	
3	(0.9,0.9,0.9)	57.56	58.63	

Fig. 4.5(a) depicts the performance of each navigator in Mod-P in terms of success rate. As can be observed, navigators with FALCON candidates of differing ρ^k parameters exhibited distinct SRs throughout the learning process. Among all three candidate models, Mod-P with FALCON of $\rho^k = (0.9, 0.9, 0.9)$ obtained the highest SR (57.56%, see Table 4.2, column 3), significantly higher than the other two navigators (known as 47.25% and 50.93%, see Table 4.2, column 3). This difference among navigators hence highlights the significance of choosing a proper candidate model for navigators to predict the behavior of their opponents. On the other hand, Mod-All (where each navigator is attributed with all predictive FALCON candidates) reported a competitive SR of 56.54% (see Table 4.2, column 5) against the best performing Mod-P with a high vigilance value (57.76%, see Table 4.2, row 3, column 3). This notable performance of Mod-All hence validates the efficacy of the proposed model-based be-

havior prediction method in selecting the appropriate model from all attributed candidates for opponent behavior prediction.

Figure 4.5: SRs of Mod-P, eTL-P and Mod-All on completing the missions in PO-MNTs.

Further, Fig. 4.5(b) depicts the success rate of Mod-P, eTL-P and Mod-All. As a result, both eTL-P and Mod-All obtained higher SRs when compared to Mod-P. In particular, without being attributed with all predictive FALCON candidates, navigators with the knowledge transfer scheme of eTL-P is noted to obtain SR that is slightly higher than Mod-All. It is worth noting that, under the eTL-P framework, navigators with FALCON candidate of $\rho^k = (0.1, 0.1, 0.1)$ achieved a significant SR increase of 8.6% as compared to that under Mod-P at the end of the missions (see Table 4.2, row 1). As discussed, this can be attributed to the navigators with multi-agent TL scheme benefiting from the knowledge by their better performing partners, such as navigators with FALCON candidate of $\rho^k = (0.9, 0.9, 0.9)$ in this case.

Moreover, in order to evaluate the performance of eTL-P on reducing the complexity of candidate models, the computational cost taken by eTL-P and Mod-All are summarized in Table 4.3. Particularly, MT is referred to the training time of candidate models, PN is the

number of predictions and PT is the computational time for behavior prediction.

Table 4.3: Computational cost taken by eTL-P and Mod-All (MT: candidate models training time; PN: behavior prediction numbers; PT: behavior predicting time).

Per Exp.		eTL-P				Mod-All
#	Metrics	Navigator1	Navigator2	Navigator3	Avg	Avg
1	MT (ms)	46	53	325	141	419
2	PN	11688	14037	18787	14837	17558
3	PT (s)	2.55	3.43	51.10	19.02	41.90

When referring to MT, Mod-All tends to be more computationally expensive than eTL-P. According to the result in Table 4.3, eTL-P reported an average MT of 141(ms), which is approximately one third that of Mod-All at 419(ms). That is expected since each navigator in Mod-All employs all candidate models while navigators in eTL-P work only with partial candidates via the multi-agent TL scheme. Further, when considering a rather complex learning system with a larger agent size, multi-agent TL exhibits significantly better scalability. For example, when 10 agents are involved in the system, eTL-P reduces the training cost to a tenth of the required amount for Mod-All.

In addition, eTL-P has been observed to outperform Mod-All by reporting lower computational cost in terms of PT. According to the result attained by eTL-P, a significant cost reduction of 54.61% in average PT as compared to Mod-All is observed per experiment (see Table 4.3, row 3). This is due to the reason that, navigators with multi-agent TL in eTL-P take advantage from their partners that are attributed with more reliable and effective predictive models. For example, with respect to PN, navigators with candidate model of high ρ^k tend to share the predicted information on predators to other navigators and hence obtains the most prediction number (18787, see Table 4.3, row 2), much more than those obtained by navigators with low ρ^k and medium ρ^k (11688 and 14037, see Table 4.3, row 2).

In summary, the proposed eTL-P with behavior prediction and multi-agent TL scheme has demonstrated its superiority in improving the success rate of navigators while also reducing the computational cost efficiency.

4.2.5 Results of eTL-P with Predictive Candidates of Heterogeneous RL Structures

In previous subsections, the performance of eTL-P with homogeneous predictive candidates is investigated, i.e., all candidate models are assumed taking the form of FALCON learning structures. This section further showcase a common scenario involving predictive predator candidates of heterogeneous reinforcement learning structures. Specifically, both FALCON and a classical multi-layer perceptron (MLP) with gradient descent based back propagation [67] are employed as the predictive candidates. For a fairness consideration, the configurations of FALCON and MLP are maintained to be consistent with the study in [32]. Similarly, both FALCON and MLP are trained using the data instances of predators recorded from historical activities of missions encountered.

Firstly, to investigate the performance of heterogeneous RL machines as candidate models in predicting the behavior of predators, three navigators are assigned with FALCON (Mod-FALCON), MLP (Mod-MLP) and without candidate models (Mod-No), respectively. Also, their average results has been reported by Mod-Avg. Then, the learning performance of navigators in terms of SR is depicted in Fig. 4.6(a).

As can be observed, Mod-FALCON shows its superiority in attaining much higher SR than Mod-MLP and Mod-No. This is not surprising as FALCON is the exact learning model of predators in PO-MNT. After training with the collected data instances, it could predict the behaviors of predators in a comparatively accurate manner.

Subsequently, to access the effectiveness of the proposed approach, the success rate of Mod-FALCON, Mod-MLP and Mod-No with eTL-P are depicted in Fig. 4.6(b). It can be observed that the proposed eTL significantly improved the overall learning performance of three navigators in terms of SR. In particular, Mod-MLP and Mod-No achieved approximately 6.0% and 7.3% improvements in SR, respectively. The improved SR hence demonstrate the efficacy of proposed eTL-P in improving the learning performance of navigators using heterogeneous RL techniques as the candidate models in PO-MNT.

Figure 4.6: SRs of Mod-FALCON, Mod-MLP, Mod-No and Mod-Avg on completing the missions in PO-MNTs.

4.3 Summary

This chapter has presented an enhanced evolutionary Transfer Learning framework, eTL-P, for addressing the specific challenges that arise in complex multi-agent systems where agents have competitive objectives. Particularly, by providing a behavior prediction approach, eTL-P endows agents with abilities to predict the behaviors of opponent agents effectively by building the candidate models. In order to reduce the complexity or computational cost of behavior prediction, eTL-P introduces a multi-agent TL scheme to assign each agent with a much smaller subset of candidate models from the entire model space. The performance efficacy of eTL-P is investigated via comprehensive empirical studies in a PO-MNT. Accordingly, eTL-P can significantly reduce the complexity of candidate models while improve the learning capability of multiple agents in an effective manner.

Chapter 5

Top-K Model Selection Method in eTL-P

In Chapter 4, an enhanced evolutionary Transfer Learning framework (eTL-P) is proposed for addressing the challenges arising in complex multi-agent scenarios wherein multiple agents possess competitive objectives. In eTL-P, the particular interests lie in developing subject agents that are capable of interacting with other agents efficiently through predicting their behavioral strategies. However, since the search space of possible candidate models ascribed to other agents tends to be very large. The behavior prediction thereby becomes time-consuming and intractable due to the computational and physical memory limitations. In the light of this consideration, the challenge here is hence to select a proper subset of candidate models from the full model set given the limited model space.

Taking this cue, this chapter proposes a Top- K model selection method for addressing the challenges that arise in reducing the candidate model complexity. Specifically, the proposed method formalizes the task of selecting Top- K models as a NP-hard combinatorial optimization problem. Subsequently, a monotone submodular function is constructed to measure the representativeness of the selected candidate models. Particularly, this function considers two important properties of the selection. One encourages the selected models to be representative of the full model space, and the other positively rewards the diversity. To solve the combinatorial optimization problem, a modified greedy algorithm is employed to iteratively select K models with a constant factor guarantee of optimality [92]. Lastly, from the experimental

results obtained on a Partner-Opponent Minefield Navigation Task (PO-MNT), the proposed Top- K model selection method is shown to significantly reduce the candidate model complexity while improve the learning capability of multiple agents in eTL-P.

The rest of this chapter is organized as follows: Section 5.1 presents the Top- K model selection method for reducing the candidate model complexity of multiple agents in eTL-P. Particularly, the detailed formalization of the proposed method and its mathematical proof are discussed. Further, Section 5.2 presents the complexity analysis of the proposed method. Section 5.3 shows a comprehensive empirical study of the proposed Top- K model selection method. In particular, the performance of eTL-P which integrates the proposed model selection method is also studied. Finally, Section 5.4 concludes this work briefly.

5.1 Top- K Model Selection

As discussed in previous chapters, subject agents in eTL-P are expected to predict the behavior of their opponents in order to interact with others more efficiently. Particularly, the prediction is made by identifying candidate opponents' models that are trained from the available data instances. Comparing with predicting behavior directly from the data, the model-based behavior prediction has better scalability since the knowledge is reusable. Also, if subject agents get access to the true model of the opponent, they can predict its behavior in any environment and conduct more sensible decisions. Since the built models (or strategies) of opponent agents are not always known with certainty, subject agents need to maintain a number of candidate models and assign them confidence of being correct.

Ideally, subject agents with candidate models derived from RL data instances could approximately predict the true behavior of opponent agents if the true model of their opponents is available in the candidate model space. However, to achieve such result becomes computational intractable due to the high complexity of candidate models ascribed to opponent agents and thus ways of mitigating the computational intractability are critically expected. Since the

complexity is predominantly due to the space of or cost on solving candidate models, this work therefore endeavors to provide reasonably effective policies by selecting a proper subset of candidate models given the limited model space while avoiding a significant loss in optimality. Particularly, an overview of eTL-P with the proposed Top- K model selection method is summarized in Fig. 5.1. The task of model selection is hence to select a subset $M^K \subseteq M$ for representing the entire model set M .

Figure 5.1: An illustration of eTL-P with the proposed Top- K model selection method.

In the past few years, researchers have concentrated on compressing the candidate model space of other agents. Among them, some works focus on selecting a subset of models that are expected to have the largest joint coverage of solutions of all candidate models [91]. By taking this coverage into consideration, the behavioral information loss is reduced effectively. The second category, on the other hand, aims to select models that are concise and contain as

few redundant models as possible [93]. Accordingly, models that provide similar information of opponent agents are not expected to be selected concurrently. In practice, the redundancy can be effectively reduced by enforcing the diversity among the selected models. Taking these inspirations, a novel Top- K model selection scheme which considers a fusion of coverage and diversity principles is initiated to select the representative candidate models.

Obviously, the size of M^K shall be much smaller than M and is expected to be restricted within the given model size budget K . Such constraints on M^K can be modeled as *knapsack constraints*: $\sum_{m_i \in M^K} c_{m_i} \leq K$ where c is the non-negative cost of collecting m_i . If a set function $\mathcal{F} : 2^M \rightarrow \mathbb{R}$ is employed for measuring the quality of M^K , the model selection task can be summarized as a combinatorial optimization problem in Eq. 5.1.

$$M^{K*} \in \underset{M^K \subseteq M}{\operatorname{argmax}} \mathcal{F}(M^K) \text{ subject to : } \sum_{m_i \in M^K} c_{m_i} \leq K. \quad (\text{Eq. 5.1})$$

As a generalization of the cardinality constraint where $\forall m_i, c_{m_i} = 1$, the combinatorial optimization problem constitutes a well known NP-hard maximum coverage problem [92, 94]. Particularly, taking both of the abovementioned coverage and diversity into consideration, the specific definition of $\mathcal{F}(M^K)$ is further formulated as the following.

Given : M, K

Objective :

$$\max_{M^K \subseteq M, |M^K|=K} \mathcal{F}(M^K) = \sum_{i=1}^K \sqrt{\sigma(P_i \cap M^K)} \quad (\text{Eq. 5.2})$$

where P_i is a partition of the entire model set M into separate clusters and $\sigma(P_i \cap M^K)$ is the estimated coverage probability of the selected models $P_i \cap M^K$ to the entire model space M . Applying the square root $\sigma(P_i \cap M^K)$ further rewards diversity in that there is higher payoff of selecting a model from a cluster not yet having one of its units already chosen. Notably, the comprehensive discussion on coverage and diversity objectives shall be found in what follows.

In the literature, the optimization of maximum coverage problem is deemed to be approximately solved when $\mathcal{F}$ is monotone submodular using generic greedy forward selection algorithms [92]. Especially, submodularity and monotonicity comprise two necessary ingredients which guarantee such greedy algorithms give near-optimal solutions. Let S be a finite set. A submodular function $\mathcal{F} : S \rightarrow \mathbb{R}$ typically satisfies the property of *diminishing returns*: for any $A \subseteq B \subseteq S$ and $s \in S \setminus A$, $\mathcal{F}(A \cup s) - \mathcal{F}(A) \geq \mathcal{F}(B \cup s) - \mathcal{F}(B)$. In this case, the incremental value of s decreases as the context in which s is considered to grow from a smaller set A to a larger set B . In addition, a submodular function $\mathcal{F}$ is called monotone nondecreasing if $\forall A \subseteq B, \mathcal{F}(A) \leq \mathcal{F}(B)$.

The following subsections provide the detail of $\mathcal{F}(M^K)$ and prove it to be monotone and submodular.

5.1.1 Coverage

Firstly, suppose $M' = P_i \cap M^K$, $\sigma(M')$ is integrated for representing the coverage of M' by M . Existing study on the possible $\sigma(M')$ has reported several ways for representing the coverage of M' . For instance, $\sigma(M')$ could be facility location function [95], $\sigma(M') = \sum_{m_j \in M} \max_{m_i \in M'} \sigma(m_i, m_j)$ or the graph cut function [92], $\sigma(M') = \sum_{m_j \in M \setminus M'} \sum_{m_i \in M'} \sigma(m_i, m_j)$, where $\sigma(m_i, m_j)$ represents the similarity between m_i and m_j . In the proposed approach, $\sigma(M')$ is defined in a rather simple and common-used manner as the following:

$$\sigma(M') = \sum_{m_j \in M} \sigma(M', m_j) \quad (\text{Eq. 5.3})$$

where $\sigma(M', m_j)$ is computed as the proportional degree that m_j is covered by the model set M' in Eq. 5.4.

$$\sigma(M', m_j) = 1 - \prod_{m_i \in M'} [1 - \sigma(m_i, m_j)] \quad (\text{Eq. 5.4})$$

where $\sigma(m_i, m_j)$ indicates the degree to which m_j is represented by m_i from M' , or in detail is calculated by how similar the behavior predicted by m_i is to that of m_j in Eq. 5.5.

$$\sigma(m_i, m_j) = \frac{1}{N} \sum_{t_{m_i} \in T_{m_i}, t_{m_j} \in T_{m_j}} \lambda(t_{m_i}, t_{m_j}) \quad (\text{Eq. 5.5})$$

where t_{m_i} is the state-action tuple from a set of predicted instances T_{m_i} , N is the size of T_{m_i} and $\lambda(t_{m_i}, t_{m_j})$ counts the number of identical action predictions given same states for T_{m_i} and T_{m_j} . In what follows, the monotone submodularity of $\sigma(M')$ is proved.

Proof. 1. The coverage function $\sigma(M')$ is monotone and submodular.

Let $M'_1 \subseteq M'_2 \subseteq M'$ and $m'_i \in M' \setminus M'_1$, $\sigma(M')$ hence satisfies the following condition:

$$\begin{aligned} & \sigma(M'_1 \cup m'_i) - \sigma(M'_1) \\ &= \prod_{m_i \in M'_1} [1 - \sigma(m_i, m_j)] - \prod_{m_i \in M'_1 \cup m'_i} [1 - \sigma(m_i, m_j)] \\ &= \sigma(m'_i, m_j) \prod_{m_i \in M'_1} [1 - \sigma(m_i, m_j)] \end{aligned} \quad (\text{Eq. 5.6})$$

As both $\sigma(m'_i, m_j)$ and $\sigma(m_i, m_j)$ are within the range of $[0,1]$, $\sigma(M'_1 \cup m'_i) - \sigma(M'_1) \geq 0$. Therefore, $\sigma(M')$ is monotone. Further, the proof on the submodularity of $\sigma(M')$ is given by Eq. 5.7.

$$\begin{aligned} & \sigma(M'_1 \cup m'_i) - \sigma(M'_1) - (\sigma(M'_2 \cup m'_i) - \sigma(M'_2)) \\ &= \sigma(m'_i, m_j) \left(\prod_{m_i \in M'_1} [1 - \sigma(m_i, m_j)] - \prod_{m_i \in M'_2} [1 - \sigma(m_i, m_j)] \right) \\ &= \sigma(m'_i, m_j) \prod_{m_i \in M'_1} [1 - \sigma(m_i, m_j)] \left(1 - \prod_{m'_i \in M'_2 \setminus M'_1} [1 - \sigma(m_i, m_j)] \right) \end{aligned} \quad (\text{Eq. 5.7})$$

Obviously, $\sigma(M'_1 \cup m'_i) - \sigma(M'_1) \geq \sigma(M'_2 \cup m'_i) - \sigma(M'_2)$ Hence $\sigma(M')$ is submodular.

Intuitively, $\sigma(M')$ is monotone since the model coverage always increases with a larger model set. On the other hand, $\sigma(M')$ is submodular: with two model sets A and B where $|A| < |B|$, the increment when adding a new model to A shall be larger since the information exhibited by the new model might have already been covered by models that are in the larger model set B yet not in A . This is known as the exact property of *diminishing returns*.

5.1.2 Diversity

As separate model clusters P_i could represent distinct behaviors expected from candidate models, $\mathcal{F}(M^K)$ rewards diversity of M^K by encouraging to select models from different disjoint P_i of the entire model set M ($\cup_i P_i = M$). In order to generate P , k -medoids algorithm is employed to cluster M . Particularly, k -medoids is a clustering algorithm similar to k -means, but chooses a candidate model $m_j \in P_i$ with highest similarity value $\sum_{m'_j \in P_i} \sigma(m_j, m'_j)$ as the center of each cluster P_i . If an element is selected from a cluster, other elements from the same cluster tend to having diminishing gain because of the square root function. For example, suppose $m_1, m_2 \in P_1$, $m_3 \in P_2$ and $\sigma(m_1) = 3$, $\sigma(m_1 \cup m_2) - \sigma(m_1) = 2.5$ and $\sigma(m_1 \cup m_3) - \sigma(m_1) = 2$. If m_1 is already in M^K , greedily selecting the next shall be m_3 rather than m_2 since $\sqrt{3+2.5} < \sqrt{3} + \sqrt{2}$.

Proof. 2. The function $\mathcal{F}(M^K)$ is monotone and submodular.

As the submodular functions possess properties in common with concave and convex functions, including the applicability and generality under a series of common operators such as mixtures, truncation, complementation, or certain convolutions [96]. Therefore, if f is non-decreasing concave and $\mathcal{F}$ is nondecreasing submodular, the composition function $\mathcal{F}'(A) = f(\mathcal{F}(A))$ is nondecreasing submodular. In this case, the square root is a monotone concave function. Inside each square root exists a monotone submodular function $\sigma(M')$. Therefore, applying the summing square root to non-negative $\sigma(M')$ again yields a submodular function.

Summing up, the model selection function $\mathcal{F}(M^K)$ is both monotone and submodular. In particular, $\mathcal{F}(M^K)$ considers a fusion of coverage and diversity principles; hence it facilitates the selection of models that are fine representatives of each cluster while also being positively diversified across the multiple clusters. In addition, since $\mathcal{F}(M^K)$ is a complex optimization problem, its NP-hard characteristic also can be proved by simply converting $\sigma(M')$ in $\mathcal{F}(M^K)$ into the *budgeted maximum coverage problem (BMCP)* [94] by setting a budget of $L = K$, a collection of sets $S = M$ and associated unit cost of $c = 1$. The goal of BMCP hence is to

find a collection $S' \subseteq S$ such that the total weight covered by S' is maximized given a budget L and a collection of sets S defined over a domain of weighted elements X . Since BMCP is NP-hard, $\mathcal{F}(M^K)$ is NP-hard as well.

To recap, if $\mathcal{F}$ is monotone submodular and NP-hard, it has been established that greedy algorithms can solve the combinatorial optimization problem in Eq. 5.1 at near-optimally with a $(1 - 1/e)$ -approximation of the optimal solution. Therefore, this work further propose Algorithm 7, which occurs prior to Algorithm 5 and employs a direct greedy method to optimize the model selection process. Particularly, the algorithm starts with $M^K = \emptyset$ and computes the $\mathcal{F}(M^K)$ for the selected models (line 2-5). Then, it iteratively adds the model that yields the greatest increment of $\mathcal{F}(M^K)$ until the model size of $|M^K| = K$ (line 6) is reached.

Algorithm 7: Top- K Model Selection

```

1 function: ModelSelection( $M, K$ )
2  $M^K = \emptyset$ 
3 for  $Ite = 1$  to  $K$  do
4   for all  $m_i \in M$  do
5      $m_i \leftarrow \operatorname{argmax}_{m_i} [\mathcal{F}(M^K \cup m_i) - \mathcal{F}(M^K)]$ 
6      $M^K \leftarrow M^K \cup m_i$ 
7 return  $M^K$ 
    
```

5.2 Complexity Analysis

As abovementioned, predicting the accurate behaviors of opponent agents is difficult and sometimes infeasible due to the large complexity of candidate models. Thus, a Top- K model selection scheme and the multi-agent TL approach are proposed for reducing the candidate model complexity.

Specifically, Top- K model selection scheme proceeds to select a proper subset of models M^K that have the largest behavioral representativeness of the entire model space M . The complexity for learning the candidate models hence reduces from $|M|$ to $|M^K|$ ($|M^K| \ll |M|$).

Further, according to the multi-agent TL scheme in eTL-P, M^K is partitioned into different partial sets and each set is labeled with the unique agent index: $M^K = \{m_1^1, m_2^1, \dots, m_q^1, \dots, m_K^{agt}\}$ where M^{agt} aggregates the candidate model set with the same subject agent superscript. In this way, subject agents merely need to train candidate models from their corresponding partial sets. In this manner, the proposed eTL-P offers significantly reduction in the candidate model space of each subject agent in the competitive multi-agent setting.

5.3 Empirical Study

This section validates the performance of eTL-P in a complex scenario wherein navigators are confronted with greater number of predictive candidate choices. Since the search space of candidate models become extremely large, this work endeavors to mitigate the model complexity of navigators' behavior prediction process based on the proposed Top- K model selection scheme described in Section 5.1.

In this set of experiments, 99 FALCON models are considered as the predictive candidates which have differing vigilance values ρ^k at 0.01 intervals from $\rho^k = (0.01, 0.01, 0.01)$ to $\rho^k = (0.99, 0.99, 0.99)$. For comparison consideration, all 99 candidate models are firstly classified into categories of 1) a low parameter set $\rho^k \in [0.01, 0.33]$, 2) a medium parameter set $\rho^k \in [0.34, 0.66]$ and 3) a high parameter set $\rho^k \in [0.67, 0.99]$, respectively. In this case, each navigator in eTL-P is attributed with a differing category of 33 predictive models. Further, to reduce the model complexity, this study facilitates to select K models out of all 99 candidates using the proposed Top- K model selection method wherein the monotone submodular function considers a fusion of coverage and diversity principles. Particularly, a direct K -medoids clustering algorithm, where $K = \lfloor 0.1N \rfloor$ and N is the size of all candidates, is employed to enforce diversity principle as discussed in Section 5.1. Hence, Top- K model selection method chooses 9 models with the largest representativeness from the entire model space. Each navigator in eTL-P with Top- K models is merely assigned with 3 out of 99 candidate models.

Table 5.1: Performance comparison among Mod-P, eTL-P and Mod-All with Top- K model selection scheme in terms of SR (SR: success rate).

FALCON			SR (99 candidate models)			SR (Top- K models)		
#	Model Set	ρ^k	Mod-P	eTL-P	Mod-All	Mod-P	eTL-P	Mod-All
1	Low	[0.01, 0.33]	53.81	58.56		55.18	59.90	
2	Medium	[0.34, 0.66]	54.06	59.33	55.23	54.25	59.68	55.58
3	High	[0.67, 0.99]	56.13	60.28		55.02	60.55	

Table 5.1 summarizes the complete results obtained pertaining to the success rates of Mod-P, eTL-P and Mod-All with all 99 candidate models or the selected Top- K models, at the end of 5,000 missions. Their corresponding learning performance are depicted in Fig. 5.2. It can be observed that overall, eTL-P with Top- K models outperformed most of its counterparts throughout the learning process. At the end of the 5,000 missions, the highest average SR of around 60% (see Table 5.1, column 8) has been attained, which is significantly higher than that of Mod-P and Mod-All.

Notably, navigators with Top- K models achieved competitive performance in terms of SR to navigators that consider all 99 candidate models. In particular, when attributed with a differing subset of Top- K candidate models, all navigators in Mod-P exhibit a consistently and high SR throughout the missions (see Fig. 5.2(a) and Fig. 5.2(c)). As discussed, this can be attributed to the proposed monotone submodular function in Top- K model selection wherein both coverage and diversity principles are taken into consideration.

Moreover, to validate and assess the efficacy of coverage and diversity principles, here this experiment investigates differing monotone submodular functions, namely MS-C which only considers coverage principle ($\mathcal{F}(M^K) = \sigma(M^K)$) and MS-CD which further enforces diversity principle beyond the selections of MS-C ($\mathcal{F}(M^K) = \sum_{i=1}^K \sqrt{\sigma(P_i \cap M^K)}$), respectively. In order to evaluate MS-C and MS-CD quantitatively, an efficiency ratio is defined as follows:

$$\text{Ratio}(\text{Mod-P}) = \frac{\text{SR}(\text{Mod-P}) - \text{SR}(\text{Conv-MAS})}{\text{MT} + \text{PT}} \quad (\text{Eq. 5.8})$$

where $[\text{SR}(\text{Mod-P}) - \text{SR}(\text{Conv-MAS})]$ indicates the SR improvement of navigators exhibited by behavior prediction over Conv-MAS, and $[\text{MT} + \text{PT}]$ is the corresponding computing cost.

(a) Navigator1,2&3 in Mod-P without Top- K model selection.

(b) Mod-P, eTL-P and Mod-All without top- K model selection.

(c) Navigator1,2&3 in Mod-P with Top- K model selection.

(d) Mod-P, eTL-P and Mod-All with Top- K model selection.

Figure 5.2: SRs of navigators with 99 predictive candidate models on completing the missions in PO-MNTs.

Typically, a higher Ratio value is preferred since it denotes that the behavior prediction is more efficient by attaining higher SR improvement at a lower computational cost.

Table 5.2: Performance comparison among Mod-P with MS-C and MS-CD (MT: candidate models training time; PN: behavior prediction numbers; PT: behavior predicting time).

#	Mod-P $\mathcal{F}(M^K)$	Metrics (per experiment)			
		SR(%)	MT(ms)	PT(s)	Ratio(Mod-P)
1	MS-C	55.76	729.23	35.65	0.37
2	MS-CD	55.36	489.55	25.31	0.51

The complete results of MS-C and MS-CD are summarized in Table 5.2. As can be observed, although MS-C achieved a competitive SR of 55.76% to MS-CD of 55.36%, it incurred a significantly higher computational cost (i.e., PT and MT) during the learning process (see Table 5.2, column 4 and 5). This is because MS-C is expected to select models that have a larger joint-coverage of all solutions. It prefers the candidate models with higher vigilance values which tend to contain an unnecessarily large number of redundant rules since knowledge generated becomes over-specific.

On the other hand, MS-CD reduces the redundancy of the similar or overlapping information by enforcing the diversity principle in model selection. According to Table 5.2, MS-CD reported a Ratio(Mod-P) of 0.51, which is significantly higher than 0.37 of MS-C (see Table 5.2, column 6). Moreover, Fig. 5.3 denotes the learning performance of MS-C and MS-CD in terms of averaged success rate over computational cost ($[MT+PT]$). As observed, MS-CD tends to achieve higher SR against MS-C under the same computing cost. This result thus highlights the efficacy of the proposed monotone submodular function, which considers a fusion of coverage and diversity objectives, so as to provide navigators the option of selecting more representative Top- K candidates.

Figure 5.3: SRs of MS-C and MS-CD over computational cost ([MT+PT]) on completing the missions in PO-MNTs.

5.4 Summary

In this chapter, a Top- K model selection method is proposed to reduce the candidate model complexity of multiple agents in eTL-P. Particularly, the model selection task has been formalized as a NP-hard combinatorial optimization problem, wherein a monotone submodular function is constructed to evaluate the representativeness of the selected models with respect to two important properties, namely behavioral coverage as well as diversity. To solve the formalized complex optimization problem, a modified greedy algorithm is further proposed to iteratively select Top- K models with high representativeness from the full model space. Based on the experimental results on a PO-MNT, the proposed Top- K model selection approach is shown to reduce candidate model complexity greatly while still obtain significant performance for problem solving in eTL-P.

Chapter 6

Conclusions and Future Work

This chapter concludes the research presented in this thesis with a summary of the contributions made on evolutionary transfer learning for multi-agent reinforcement learning systems, wherein multiple agents may have either common or differing objectives. Hereafter, several potential future research directions are identified.

6.1 Summary of Contributions

The focus of this research is, on one hand, to develop meme-centric evolutionary transfer learning frameworks for enhancing the performance of multi-agent system, and on the other hand, to solve the specific challenges arising in rather complex multi-agent settings where agents tend to have different and even competitive objectives. In this subsection, the research contribution of this dissertation are re-emphasized as follows.

- The first contribution of this dissertation work is to propose an evolutionary Transfer reinforcement Learning framework (eTL) for developing intelligent agents capable of adapting to the dynamic environment of multi-agent systems. The essential backbone of this work comprises of memetic automaton [24] in which memes compete and cooperate in an evolutionary process by conducting memetic transmission, selection, replication and variation. Beyond the formalism of memetic automaton and taking memes as the

building blocks of information in the context of MAS, eTL develops the evolutionary learning process, particularly on agents social interactions, for more efficient problem solving. Specifically, in the design of eTL, memes are defined and categorized into 1) memotypes as agents' knowledge or memory items inside their mind universe and 2) sociotypes for manifesting the behaviors of agents that could be observed by the others. The evolutionary learning process is governed by a series of memetic mechanisms, including *meme representation* which defines what is a meme, *meme expression/assimilation* which activates/updates the meme, *meme internal evolution* for individual learning and *meme external evolution* for social interaction among agents. In particular, *meme external evolution* comprises four evolutionary factors, namely *meme selection*, *meme expression*, *meme imitation* and *meme variation*. The detailed design of evolutionary learning mechanisms and the realizations of two kinds of learning agents that take the form of neurally inspired learning structures, are presented. By leveraging the proposed culture-inspired evolutionary transfer learning operators into multi-agent reinforcement systems, useful knowledge in the form of memes learned from better-performing agents can be leveraged to enhance the learning performance of their partners in the same environment. To verify the effectiveness of this work, eTL is benchmarked against both a conventional MAS without knowledge transfer and two state-of-the-art MAS transfer learning approaches (i.e., AE Model and PTL) that are proposed in the literature previously. The significant performance obtained confirmed the efficacy of eTL in conducting evolutionary knowledge transfer in MAS. Also, the investigation on a well-known first person shooter game also demonstrated its effectiveness in solving complex problems.

- In the second contribution of this dissertation, a novel evolutionary transfer learning framework with opponent prediction (eTL-P) is proposed for addressing the complex multi-agent systems. To date, little or no investigation of multi-agent transfer learning approaches has been conducted for multi-agent cases where each agent has unique

objectives. In this regard, one of the greatest challenges for agents is to search for an efficient strategy to interact or respond to their opponents. As the behavioral actions of opponent agents is usually unknown, such strategies could be hard to define. Thus, this work endeavors to extend the agents' learning ability to recognize the strategies of other agents present in the environment. Given the instance data recorded from previous interaction activities, this work introduces a model-based behavior prediction method which is achieved by building and solving a set of candidate models that are ascribed to opponents. By leveraging from social knowledge sharing schemes, this work also provides the ability for agents to enhance their learning performance while reduce the complexity of behavior prediction. With comprehensive studies conducted on a partner-opponent minefield navigation task (PO-MNT), the effectiveness and efficiency of this work is confirmed by reporting an increasing level of learning capability and decreasing candidate model complexity by knowledge sharing with respect to agents' representative knowledge and unique predictive candidate models.

- In the third contribution of this dissertation, a Top- K model selection method that aims to mitigate the candidate model complexity of agents in eTL-P is proposed. As discussed in the second contribution of this thesis, agents in eTL-P are expected to search an efficient strategy to predict the behaviors of their opponents by building predictive candidate models. However, the search space could be extremely large since the true model of opponent agents is usually unknown. Hence, this works endeavors to solve the challenges of selecting a much smaller subset of candidate models that contain highly representative knowledge of the entire model set given the limited model space. Specifically, in this work, the model selection process is achieved by a monotone submodular function which selects Top- K models from all available candidates based on their representativeness in terms of behavioral coverage as well as reward diversity. The specific mathematical formulations of the monotone submodular function have been derived in

the context of an NP-hard combinatorial optimization problem. To verify the effectiveness of the proposed model selection method, experiments are conducted on PO-MNT and the results show that the Top- K model selection method exhibits superior learning capability in choosing a proper subset of models from the entire model space.

6.2 Future Work

The methodologies and frameworks presented in this dissertation have provided some ground-works pertaining to the development of meme-inspired evolutionary transfer learning methods for solving complex multi-agent reinforcement learning tasks. Future work will continue to focus on the challenges posed by multi-agent reinforcement learning problems with uncertain and dynamic environment, and further extend the applicability and practicality of the proposed transfer learning paradigm. In what follows, a summary of some potential future works are outlined and described.

6.2.1 Extensive Study for the Evolutionary Transfer Learning Framework

As stated in Chapter 3, memes encoded in computational representations form the underlying building blocks (knowledge, belief, emotion, etc.) of the mind-universe of an individual. For future work, the generality and adaptivity of the proposed framework are expected to be studied, by focusing on novel representations of memes as well as the corresponding realizations of the evolutionary mechanisms in eTL, i.e., *meme external evolution*, namely *meme selection*, *meme transmission*, *meme imitation* and *meme variation*.

Further, in eTL-P, model-based behavior prediction and multi-agent knowledge transfer comprise the core learning components. In the immediate future, the generality and adaptivity of the proposed eTL-P are expected to be explored in addressing the increasing complexity and diversity of problem solving, by focusing on the novel on-line model predicting approaches,

model-based or model-free behavior approximating methods as well as more suitable representative model selection functions.

In addition, as the contribution of this dissertation lies in the novel meme-centric evolutionary transfer learning framework, “Unreal Tournament 2004” has been considered as a real world game platform to verify the proposed evolutionary knowledge transfer under complex game scenario. As the interaction among agents with competitive objectives becomes rather complex, only an initial study on eTL is investigated in the present work. However, plans to study the generality and extension of the proposed framework involving complex real world video games with diverse platforms is considered in the immediate future.

6.2.2 Evolutionary Transfer Learning across Problem Domains

In the present study, the proposed evolutionary transfer learning framework has exclusively focused on multi-agent reinforcement learning systems, wherein multiple agents share the same state and action spaces in a single problem domain of interest. Particularly, the memetic knowledge transfer process is primarily driven by imitation, which takes place when sociotype memes (defined as the manifested behavioral actions) are transmitted. Since multiple agents possess common state and action spaces, assimilation of sociotype memes transferred from the teacher agent to the imitating agent can be seamlessly attained since both shares a common mind universe.

Due to the unique representations of state and/or action variables in different problem domains, it is often the case that the knowledge acquired from a problem domain cannot be directly used in another. In addition, adaptation of knowledge from unique agents to enable TL across problem domains is a non-trivial task. From a survey of the literature, current research works have attempted to build the mappings for given pairs of differing problem domains. Recent studies have demonstrated that existing transfer methods are able to successfully improve the learning capacity of RL tasks [97][98]. Nevertheless, most of these studies assume that the

mappings are manually pre-defined in accordance with analysis of specific problems, which can be very labour intensive and time consuming [19].

Falling back on the discussion in Chapter Introduction, memes can be naturally materialized as a wide variety of recurring real-world patterns. These meme patterns provide high-level common knowledge representations of problems, and hence enable knowledge reuse across problem domains. Particularly in the field of evolutionary computation, memes have already been defined as the transformation matrices that are transferred across differing problem domains for enhancing the evolutionary search process [80]. On this basis, an immediate future work is thus to extend eTL in working with differing RL problems from multiple related domains.

6.2.3 Evolutionary Knowledge Transfer for Deep Reinforcement Learners

The study on evolutionary knowledge transfer paradigm in deep reinforcement learning (RL) is another potentially intriguing topic of future research. As discussed, a RL agent such as FALCON and BP in the present study typically optimizes its behavioral strategies based on the reward that it obtains while interacting with the environment. However, due to the increasing complexity of real-world RL applications, conventional RL agents are either limited to the solving of simple problems, or assumed to possess human-provided knowledge about the domain dynamics when solving complex problems.

Google DeepMind has recently proposed the deep reinforcement learning which integrates the classical RL methods with the representational power of state-of-the-art deep learning approaches [99][100]. Particularly, deep RL has been demonstrated to provide agents with increasing capabilities of addressing very high-dimensional and complex tasks for which previously conventional RL methods were inadequate. However, most of the current deep RL works have focused exclusively on enhancing the learning performance of individual RL agent [101].

Therefore, the potential benefits of evolutionary knowledge transfer paradigm with multiple deep RL agents look promising for addressing the increasing complexity and dynamic nature of problem-solving.

References

- [1] P. Stone and M. Veloso, “Multiagent systems: A survey from a machine learning perspective,” *Autonomous Robots*, vol. 8, no. 3, pp. 345–383, 2000.
- [2] J. Ferber, *Multi-agent systems: an introduction to distributed artificial intelligence*, vol. 1. Addison-Wesley Reading, 1999.
- [3] G. Dudek, M. R. Jenkin, E. Milios, and D. Wilkes, “A taxonomy for multi-agent robotics,” *Autonomous Robots*, vol. 3, no. 4, pp. 375–397, 1996.
- [4] P. Stone, *Layered learning in multiagent systems: A winning approach to robotic soccer*. MIT Press, 1998.
- [5] T. Balch and R. C. Arkin, “Communication in reactive multiagent robotic systems,” *Autonomous robots*, vol. 1, no. 1, pp. 27–52, 1994.
- [6] T. Lux and M. Marchesi, “Scaling and criticality in a stochastic multi-agent model of a financial market,” *Nature*, vol. 397, no. 6719, pp. 498–500, 1999.
- [7] L. Tesfatsion, “Agent-based computational economics: modeling economies as complex adaptive systems,” *Information Sciences*, vol. 149, no. 4, pp. 262–268, 2003.
- [8] N. Cai, N. T. T. Nga, and J. Mitra, “Economic dispatch in microgrids using multi-agent system,” in *North American Power Symposium (NAPS), 2012*, pp. 1–5, IEEE, 2012.
- [9] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, vol. 1. MIT press Cambridge, 1998.
- [10] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement learning: A survey,” *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.

REFERENCES

- [11] L. Busoniu, R. Babuska, and B. De Schutter, “A comprehensive survey of multiagent reinforcement learning,” *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 38, no. 2, pp. 156–172, 2008.
- [12] M. L. Littman, “Markov games as a framework for multi-agent reinforcement learning,” in *Proceedings of the eleventh international conference on machine learning*, vol. 157, pp. 157–163, 1994.
- [13] M. Bowling and M. Veloso, “An analysis of stochastic game theory for multiagent reinforcement learning,” tech. rep., CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE, 2000.
- [14] J. Hu and M. P. Wellman, “Multiagent reinforcement learning: theoretical framework and an algorithm,” in *ICML*, vol. 98, pp. 242–250, 1998.
- [15] H. M. Schwartz, “Single-agent reinforcement learning,” *Multi-Agent Machine Learning: A Reinforcement Approach*, pp. 12–37.
- [16] C. D. Paternina-Arboleda and T. K. Das, “A multi-agent reinforcement learning approach to obtaining dynamic control policies for stochastic lot scheduling problem,” *Simulation Modelling Practice and Theory*, vol. 13, no. 5, pp. 389–406, 2005.
- [17] S. J. L. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transaction on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [18] M. E. Taylor, N. K. Jong, and P. Stone, “Transferring instances for model-based reinforcement learning,” in *Machine learning and knowledge discovery in databases*, pp. 488–505, Springer, 2008.
- [19] M. E. Taylor and P. Stone, “Cross-domain transfer for reinforcement learning,” in *Proceedings of the 24th international conference on Machine learning*, pp. 879–886, ACM, 2007.
- [20] B. Banerjee and P. Stone, “General game learning using knowledge transfer,” in *IJCAI*, pp. 672–677, 2007.

- [21] E. Oliveira and L. Nunes, “Learning by exchanging advice,” In R.Khosla, N. Ichalkaranje, and L. Jain, editors, *Design of Intelligent Multi-Agent Systems*, chapter 9. Springer, New York, NY, USA, 2005.
- [22] Y. Hou, Y.-S. Ong, L. Feng, and J. M. Zurada, “An evolutionary transfer reinforcement learning framework for multi-agent system,” *IEEE Transactions on Evolutionary Computation*, 2017.
- [23] A. Taylor, I. Dusparic, E. Galván-López, S. Clarke, and V. Cahill, “Transfer learning in multi-agent systems through parallel transfer,” in *Theoretically Grounded Transfer Learning at the 30th International Conference on Machine Learning (ICML)*. Omnipress, 2013.
- [24] X. Chen, Y.-S. Ong, M.-H. Lim, and K. C. Tan, “A multi-facet survey on memetic computation,” *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 5, pp. 591–607, 2011.
- [25] J. C. Spall, *Introduction to stochastic search and optimization: estimation, simulation, and control*, vol. 65. John Wiley & Sons, 2005.
- [26] Y.-S. Ong, M.-H. Lim, N. Zhu, and K.-W. Wong, “Classification of adaptive memetic algorithms: a comparative study,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 1, pp. 141–152, 2006.
- [27] E. Özcan, B. Bilgin, and E. E. Korkmaz, “A comprehensive analysis of hyper-heuristics,” *Intelligent Data Analysis*, vol. 12, no. 1, pp. 3–23, 2008.
- [28] P. Moscato and C. Cotta, “Memetic algorithms,” *Handbook of Applied Optimization*, pp. 157–167, 2002.
- [29] R. Dawkins, “The selfish gene,” *Oxford: Oxford University Press*, 1976.
- [30] T. Bäck, U. Hammel, and H.-P. Schwefel, “Evolutionary computation: Comments on the history and current state,” *Evolutionary computation, IEEE Transactions on*, vol. 1, no. 1, pp. 3–17, 1997.

REFERENCES

- [31] M. A. Runco and S. R. Pritzker, *Encyclopedia of creativity*, vol. 2. Elsevier, 1999.
- [32] A.-H. Tan, N. Lu, and D. Xiao, “Integrating temporal difference methods and self-organizing neural networks for reinforcement learning with delayed evaluative feedback,” *Neural Networks, IEEE Transactions on*, vol. 19, no. 2, pp. 230–244, 2008.
- [33] Y. Kameya, J. Kumagai, and Y. Kurata, “Accelerating genetic programming by frequent subtree mining,” in *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pp. 1203–1210, ACM, 2008.
- [34] S.-S. Choi, K. Jung, and B.-R. Moon, “Lower and upper bounds for linkage discovery,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 201–216, 2009.
- [35] T. L. Huston and G. Levinger, “Interpersonal attraction and relationships,” *Annual review of psychology*, vol. 29, no. 1, pp. 115–156, 1978.
- [36] L. Siwik and S. Natanek, “Elitist evolutionary multi-agent system in solving noisy multi-objective optimization problems,” in *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, p-p. 3319–3326, IEEE, 2008.
- [37] X. Chen, Y. Zeng, Y.-S. Ong, C. S. Ho, and Y. Xiang, “A study on like-attracts-like versus elitist selection criterion for human-like social behavior of memetic multitagent systems,” in *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pp. 1635–1642, IEEE, 2013.
- [38] B. Burmeister, A. Haddadi, and G. Matylis, “Application of multi-agent systems in traffic and transportation,” in *Software Engineering. IEE Proceedings-[see also Software, IEE Proceedings]*, vol. 144, pp. 51–60, IET, 1997.
- [39] D. L. Hancock and G. B. Lamont, “Multi agent systems on military networks,” in *Computational Intelligence in Cyber Security (CICS), 2011 IEEE Symposium on*, pp. 100–107, IEEE, 2011.
- [40] M. Pipattanasomporn, H. Feroze, and S. Rahman, “Multi-agent systems in a distributed smart grid: Design and implementation,” in *Power Systems Conference and Exposition, 2009. PSCE'09. IEEE/PES*, pp. 1–8, IEEE, 2009.

- [41] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.
- [42] G. A. Rummery and M. Niranjan, *On-line Q-learning using connectionist systems*. University of Cambridge, Department of Engineering, 1994.
- [43] M. P. Deisenroth, G. Neumann, J. Peters, *et al.*, "A survey on policy search for robotics.," *Foundations and Trends in Robotics*, vol. 2, no. 1-2, pp. 1–142, 2013.
- [44] S. P. Singh and R. S. Sutton, "Reinforcement learning with replacing eligibility traces," *Machine learning*, vol. 22, no. 1-3, pp. 123–158, 1996.
- [45] A. Lazaric, M. Restelli, and A. Bonarini, "Reinforcement learning in continuous action spaces through sequential monte carlo methods," in *Advances in neural information processing systems*, pp. 833–840, 2007.
- [46] K. Ueda, I. Hatono, N. Fujii, and J. Vaario, "Reinforcement learning approaches to biological manufacturing systems," *CIRP Annals-Manufacturing Technology*, vol. 49, no. 1, pp. 343–346, 2000.
- [47] I. Giannoccaro and P. Pontrandolfo, "Inventory management in supply chains: a reinforcement learning approach," *International Journal of Production Economics*, vol. 78, no. 2, pp. 153–161, 2002.
- [48] A. E. Gaweda, M. K. Muezzinoglu, A. A. Jacobs, G. R. Aronoff, and M. E. Brier, "Model predictive control with reinforcement learning for drug delivery in renal anemia management," in *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*, pp. 5177–5180, IEEE, 2006.
- [49] T. G. Dietterich, "Hierarchical reinforcement learning with the maxq value function decomposition," *J. Artif. Intell. Res.(JAIR)*, vol. 13, pp. 227–303, 2000.
- [50] R. S. Sutton, D. Precup, and S. Singh, "Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning," *Artificial intelligence*, vol. 112, no. 1-2, pp. 181–211, 1999.

REFERENCES

- [51] M. E. Taylor and P. Stone, “Transfer learning for reinforcement learning domains: A survey,” *Journal of Machine Learning Research*, vol. 10, pp. 1633–1685, 2009.
- [52] R. S. Woodworth and E. Thorndike, “The influence of improvement in one mental function upon the efficiency of other functions.(i).,” *Psychological review*, vol. 8, no. 3, p. 247, 1901.
- [53] B. F. Skinner, *Science and human behavior*. Simon and Schuster, 1953.
- [54] G. P. C. Fung, J. X. Yu, H. Lu, and P. S. Yu, “Text classification without negative examples revisit,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 18, no. 1, pp. 6–20, 2006.
- [55] B. Bakker and T. Heskes, “Task clustering and gating for bayesian multitask learning,” *The Journal of Machine Learning Research*, vol. 4, pp. 83–99, 2003.
- [56] C. H. Lampert, H. Nickisch, and S. Harmeling, “Learning to detect unseen object classes by between-class attribute transfer,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 951–958, IEEE, 2009.
- [57] D. Wang and T. F. Zheng, “Transfer learning for speech and language processing,” in *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pp. 1225–1237, IEEE, 2015.
- [58] M. E. Taylor and P. Stone, “Representation transfer for reinforcement learning,” in *AAAI 2007 Fall Symposium on Computational Approaches to Representation Change during Learning and Development*, pp. 1–8, 2007.
- [59] T. J. Walsh, L. Li, and M. L. Littman, “Transferring state abstractions between mdps,” in *ICML Workshop on Structural Knowledge Transfer for Machine Learning*, 2006.
- [60] P. Vrancx, Y.-M. De Hauwere, and A. Nowé, “Transfer learning for multi-agent coordination.,” in *ICAART (2)*, pp. 263–272, 2011.
- [61] M. E. Taylor, S. Whiteson, and P. Stone, “Transfer via inter-task mappings in policy search reinforcement learning,” in *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, p. 37, ACM, 2007.

- [62] G. Boutsioukis, I. Partalas, and I. Vlahavas, “Transfer learning in multi-agent reinforcement learning domains,” in *Recent Advances in Reinforcement Learning*, pp. 249–260, Springer, 2012.
- [63] L. Feng, Y.-S. Ong, A.-H. Tan, and X.-S. Chen, “Towards human-like social multi-agents with memetic automaton,” in *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pp. 1092–1099, IEEE, 2011.
- [64] J. W. Eerkens and C. P. Lipo, “Cultural transmission, copying errors, and the generation of variation in material culture and the archaeological record,” *Journal of Anthropological Archaeology*, vol. 24, no. 4, pp. 316–334, 2005.
- [65] F. Bousquet and C. Le Page, “Multi-agent simulations and ecosystem management: a review,” *Ecological modelling*, vol. 176, no. 3, pp. 313–332, 2004.
- [66] G. A. Carpenter and S. Grossberg, “A massively parallel architecture for a self-organizing neural pattern recognition machine,” *Computer vision, graphics, and image processing*, vol. 37, no. 1, pp. 54–115, 1987.
- [67] C. J. C. H. Watkins, *Learning from Delayed Rewards*. PhD thesis, King’s College, Cambridge, U.K, May 1989.
- [68] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning representations by back-propagating errors*. MIT Press, Cambridge, MA, USA, 1988.
- [69] D. Gordon and D. Subramanian, “A cognitive model of learning to navigate,” in *Proc. 19th Conf. of the Cognitive Science Society*, vol. 25, p. 271, 1997.
- [70] D. Wang and A. Tan, “Creating autonomous adaptive agents in a real-time first-person shooter computer game,” *Computational Intelligence and AI in Games, IEEE Transactions on*, 2014.
- [71] S. Blackmore, *The meme machine*, vol. 25. Oxford University Press, 2000.
- [72] G. Grant, “Memetic lexicon,” *Principia Cybernetica Web*, 1990.
- [73] J. Delius, “Of mind memes and brain bugs; a natural history of culture,” 1989.

REFERENCES

- [74] D. C. Dennett, *Consciousness explained*. Penguin UK, 1993.
- [75] L. Gabora, “The origin and evolution of culture and creativity,” *Journal of Memetics: Evolutionary Models of Information Transmission*, vol. 1, no. 1, pp. 1–28, 1997.
- [76] Y.-S. Ong, M. H. Lim, and X. Chen, “Research frontier-memetic computation - past, present & future,” *IEEE Computational Intelligence Magazine*, vol. 5, no. 2, p. 24, 2010.
- [77] N. Krasnogor and J. Smith, “A tutorial for competent memetic algorithms: model, taxonomy, and design issues,” *Evolutionary Computation, IEEE Transactions on*, vol. 9, no. 5, pp. 474–488, 2005.
- [78] Y. S. Ong and A. J. Keane, “Meta-lamarckian learning in memetic algorithms,” *Evolutionary Computation, IEEE Transactions on*, vol. 8, no. 2, pp. 99–110, 2004.
- [79] Y.-S. Ong, M.-H. Lim, F. Neri, and H. Ishibuchi, “Special issue on emerging trends in soft computing: memetic algorithms,” *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, vol. 13, no. 8, pp. 739–740, 2009.
- [80] L. Feng, Y.-S. Ong, M.-H. Lim, and I. W. Tsang, “Memetic search with interdomain learning: A realization between cvrp and carp,” *Evolutionary Computation, IEEE Transactions on*, vol. 19, no. 5, pp. 644–658, 2015.
- [81] Heidrich-Meisner, “Interview with richard s. sutton,” *Knstliche Intelligenz*, vol. 3, no. 1, pp. 41–43, 2009.
- [82] J. Gemrot, R. Kadlec, M. Bída, O. Burkert, R. Píbil, J. Havlíček, L. Zemčák, J. Šimlovič, R. Vansa, M. Štolba, *et al.*, “Pogamut 3 can assist developers in building ai (not only) for their videogame agents,” in *Agents for Games and Simulations*, pp. 1–15, Springer, 2009.
- [83] R. Adobbati, A. N. Marshall, A. Scholer, S. Tejada, G. A. Kaminka, S. Schaffer, and C. Sollitto, “Gamebots: A 3d virtual world test-bed for multi-agent research,” in *Proceedings of the second international workshop on Infrastructure for Agents, MAS, and Scalable MAS*, vol. 5, Montreal, Canada, 2001.

- [84] J. Lindfors and M. Fleury, *JMX: Managing J2EE with Java Management Extensions*. Sams Publishing, 2002.
- [85] D. Yang, X. Chen, and J. Jiang, “Multi-agent cooperation based on behavior prediction and reinforcement learning,” in *Intelligent Control and Automation, 2004. WCICA 2004. Fifth World Congress on*, vol. 6, pp. 4869–4872, IEEE, 2004.
- [86] Y. Hou, Y. Zeng, and Y.-S. Ong, “A memetic multi-agent demonstration learning approach with behavior prediction,” in *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pp. 539–547, International Foundation for Autonomous Agents and Multiagent Systems, 2016.
- [87] R. Dawkins, “The selfish gene,” *Oxford: Oxford University Press*, 1976.
- [88] Y. Zeng, Y. Chen, and P. Doshi, “Approximating behavioral equivalence of models using top-k policy paths,” in *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pp. 1229–1230, International Foundation for Autonomous Agents and Multiagent Systems, 2011.
- [89] J. Gläscher, N. Daw, P. Dayan, and J. P. O’Doherty, “States versus rewards: dissociable neural prediction error signals underlying model-based and model-free reinforcement learning,” *Neuron*, vol. 66, no. 4, pp. 585–595, 2010.
- [90] D. Ormoneit and S. Sen, “Kernel-based reinforcement learning,” *Machine learning*, vol. 49, no. 2-3, pp. 161–178, 2002.
- [91] R. Conroy, Y. Zeng, and J. Tang, “Approximating value equivalence in interactive dynamic influence diagrams using behavioral coverage,” 2016.
- [92] H. Lin, J. Bilmes, and S. Xie, “Graph-based submodular selection for extractive summarization,” in *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*, pp. 381–386, IEEE, 2009.
- [93] Y. Zeng and P. Doshi, “Exploiting model equivalences for solving interactive dynamic influence diagrams,” *J. Artif. Intell. Res.(JAIR)*, vol. 43, pp. 211–255, 2012.

REFERENCES

- [94] S. Khuller, A. Moss, and J. S. Naor, “The budgeted maximum coverage problem,” *Information Processing Letters*, vol. 70, no. 1, pp. 39–45, 1999.
- [95] G. Cornuejols, M. Fisher, and G. L. Nemhauser, “On the uncapacitated location problem,” *Annals of Discrete Mathematics*, vol. 1, pp. 163–177, 1977.
- [96] L. Lovász, “Submodular functions and convexity,” in *Mathematical Programming The State of the Art*, pp. 235–257, Springer, 1983.
- [97] H. B. Ammar, E. Eaton, P. Ruvolo, and M. E. Taylor, “Unsupervised cross-domain transfer in policy gradient reinforcement learning via manifold alignment,” in *Proc. of AAAI*, 2015.
- [98] Y. Liu and P. Stone, “Value-function-based transfer for reinforcement learning using structure mapping,” in *Proceedings of the National Conference on Artificial Intelligence*, vol. 21, p. 415, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.
- [99] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning,” *arXiv preprint arXiv:1312.5602*, 2013.
- [100] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [101] R. Glatt, F. L. da Silva, and A. H. R. Costa, “Towards knowledge transfer in deep reinforcement learning,” in *Intelligent Systems (BRACIS), 2016 5th Brazilian Conference on*, pp. 91–96, IEEE, 2016.