

UCLA

UCLA Electronic Theses and Dissertations

Title

Communicative Learning: A Unified Learning Formalism

Permalink

<https://escholarship.org/uc/item/8k76f3vc>

Author

Yuan, Luyao

Publication Date

2022

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Communicative Learning: A Unified Learning Formalism

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Luyao Yuan

2022

© Copyright by

Luyao Yuan

2022

ABSTRACT OF THE DISSERTATION

Communicative Learning: A Unified Learning Formalism

by

Luyao Yuan

Doctor of Philosophy in Computer Science

University of California, Los Angeles, 2022

Professor Song-Chun Zhu, Chair

The uniqueness and superiority of the cognitive infrastructure for human cooperative communication have been widely acknowledged and received systematical analysis in cognitive science in the past decade. Pedagogy and learning, as two of the most common types of human communication, also benefit from the sophistication of this infrastructure. Recently, the efficiency of human learning has engaged researchers in combining cooperative pedagogy with the booming field of machine learning. The pedagogical insight facilitates the adoption of alternative data sources, besides random sampling, in machine learning, *e.g.* intentional messages given by a helpful teacher. In this dissertation, we propose a communicative learning (CL) formalism rooted in human cooperative communication to unify existing machine learning paradigms, *e.g.* passive learning, active learning, algorithmic teaching *etc.*, and enlighten new ones. In this formalism, a teacher and a student communicate with each other in the process of teaching and learning certain knowledge. Each agent has a mind, including the agent's knowledge, utility, and mental dynamics. To communicate with each other effectively, each agent must also have an estimation of its partner's mind. We argue this modeling is necessary for the development of general human-like intelligence and justify the necessity

with a prototypical human-machine collaboration task. We rigorously give the CL formalism and use it to survey existing learning algorithms through a unifying lens and show them as special cases of this formalism. We proved the theoretical guarantee of the CL formalism over non-CL algorithms and illustrate how efficient learning protocols can emerge between agents conducting CL. We also verified the practicability and scalability of this learning formalism with a generic human-robot interaction task. Last but not the least, we illustrate that CL allows learning protocol to go beyond Shannon’s communication limit and put forth the halting problem of learning to discuss the implications of CL.

The dissertation of Luyao Yuan is approved.

Demetri Terzopoulos

Quanquan Gu

Ying Nian Wu

Song-Chun Zhu, Committee Chair

University of California, Los Angeles

2022

To my parents, Yi Yang and many people who ever helped me go through difficulties.

TABLE OF CONTENTS

1	Introduction: Inspect Learning in A Communication Framework	1
1.1	Objective: A Unifying Formalism	1
1.1.1	Cognitive Infrastructure for CL	2
1.1.2	Unifying Framework for Machine Learning	4
1.1.3	Integrating Pedagogical Reasoning	5
1.2	Inspect Learning in A Communication Framework	8
1.2.1	Connection between Communication and Learning	9
1.2.2	Go beyond Shannon and Valiant	11
2	Formalism of Communicative Learning	16
2.1	A Full-blown Teacher-student Mental Representation	16
2.2	A Unifying Framework of Learning	21
2.3	Formalism of Communicative Learning	24
2.3.1	Overall Setting	25
2.3.2	Teacher Setting	27
2.3.3	Student Setting	28
2.4	Communicative Learning Instantiation of Prior Learning Paradigms	31
2.4.1	Level-0 Paradigms	32
2.4.2	Level-1 Paradigms	43
2.4.3	Level-2 Paradigms	53
3	Theoretical Guarantee of Communicative Learning	65

3.1	Machine Parameter Teaching	65
3.2	Teacher-Aware Learning	68
3.2.1	Cooperative Teacher	68
3.2.2	Teacher-Aware Learner	69
3.3	Results	78
3.3.1	Machine Teacher	78
3.3.2	Human Teacher	86
3.4	Discussion and Conclusion	93
4	Emergence of Learning Protocols	95
4.1	Background: Referential Game	95
4.2	Adaptive Emergence of Pragmatic Learning Protocol	97
4.3	Results	100
4.3.1	Referential Game with Symbolic and Pixel Input	101
4.3.2	Connection with RTD	102
4.3.3	Stability of the Protocol	105
4.3.4	Global Mapping and Local Selection	106
4.4	Discussion and Conclusion	107
5	Application of Communicative Learning in Human-machine Interaction 110	
5.1	Communication Paradigm for Generic Human-robot Teaming	110
5.2	Collaborative Game Setup	114
5.3	Value Alignment with CL	117
5.3.1	Level-1 ToM	117

5.3.2	Level-2 ToM	119
5.4	Human Experiment Design	123
5.4.1	Experiment Details and Demographics	124
5.5	Results	125
5.6	Discussion and Conclusion	129
6	Implications and Conclusion	132
6.1	When Aumann Meets Grice: from Distributed Knowledge to Common Knowl- edge	132
6.1.1	Beyond Shannon’s Limits: Better Learning Protocol Enabled by CL .	137
6.2	Halting Problem of Learning	140
6.2.1	Three Hierarchies in Learning	141
6.2.2	Halting Criteria	143
6.3	Conclusion	145
	References	147

LIST OF FIGURES

1.1	Key representations of CL includes <i>six minds</i> that evolve over t time steps or messages. 1) G : oracle in God's mind, usually not time-variant; 2) P_t : mind of the teacher A ; 3) Q_t : mind of the student B ; 4) $\hat{P}_t$: B reflects what A knows; 5) $\widehat{Q}_t$: A thinks what B knows; and 6) C_t : common mind that both A and B know that each other know.	3
1.2	Comparison among different learning/teaching paradigms. Blue characters represent teachers and red characters are students. The bubble means modeling one's partner.	7
1.3	Shannon's diagram of a communication system. The shared codebook fulfills the purpose of a common ground and the norm of communication in the cooperative communication model.	9
1.4	Student's belief update processes when learning from random sampling (blue trajectory) and a cooperative teacher (red trajectory). It is possible to transmit the target set $\{4, 5\}$ with a single message because the recursive teaching dimension (RTD) [ZLH11] of this concept space is 1, lower bounding the regular teaching dimension (TD). Concretely, the student knows the teacher will send 3 for concept A , and thus 1 or 2 indicates D . Similarly, 6 can only mean B , making 5 a unique identifier for C . The prerequisite of such a recursive protocol is that the teacher is pedagogical, and the student is aware of her helpfulness. Here, IG is short for information gain.	13
2.1	State $w = pg$ unfolding over time	17

2.2	A zoomed-in view of CL representations: unifying all existing learning protocols and beyond. Each mind contains four spaces: i) pentagon for hypothesis/model space Θ ; ii) ellipse for the situate $s = pg$ in the state space S where the belief is represented by a cloud; iii) diamond for policy π ; and iv) square for utility u . The arrows illustrate the dynamics: observation, intervention, and messages. The shaded shapes represent belief over belief.	18
2.3	CL formalism. Fig. 2.3a shows the mental state representation of the teacher and the student. The blue bubbles are for the teacher and the red ones are for the student. Beliefs are drawn as ellipses, with the time-variant ones shaded. Diamonds hold the functions that agents used to maintain their beliefs about their partner's behavior and knowledge. We call them the belief-update functions . The value functions for the teacher and the student are in the blue and red rectangles. Fig. 2.3b expands the transition process at the first time step. Numbers in it group the arrows, and the same number indicates operations happen in one function. Arrow 1s correspond to Eq. (2.2). Arrow 2s correspond to Eq. (2.5). Arrow 3s are for Eq. (2.4). Arrow 4s are for Eq. (2.6). Eq. (2.1) and Eq. (2.7) are represented by arrows 5s and 6s respectively. The temporal order of these belief updates is the same as the numerical order of the arrows, with 2s and 3s happen concurrently and 5s and 6s happen concurrently.	25

3.1	Fig. 3.1a-Fig. 3.1f: Cooperative teacher results. Our method always gives a substantial improvement over iterative machine teaching (IMT), showing the effect of teacher-awareness. Within 2000 steps, iterative teacher-aware learning (ITAL) already show convergence, while a naive learner only learns to a limited extent in most tasks. Fig. 3.1g: In the top plot, the height of each bar represents the decrease of the L2-distance between the learner’s reward parameter and the ground-truth parameter. In the bottom, the height represents the accumulated reward. Paired t-tests were conducted between Human (Machine) ITAL & IMT respectively.	81
3.2	inverse reinforcement learning (IRL) map examples. Each map has 8×8 grids. Every grid contains a reward. Maps in the first row plots the ground truth rewards in each grid. Red bars represent positive rewards and blue bars represent negative rewards. The learner tries to learn a policy to walk in the map and collect the most accumulative rewards. The arrows below indicate the most probable action taken by the learner after he learned the reward function. The red grids are targets of all their neighbors.	83
3.3	Total variance between the learner’s policy and the teacher’s policy and the actual gain of the learner during the learning process. The gray horizontal dash line represents teacher’s expected accumulative reward.	83
3.4	Learning results of sparse reward maps. All grids in the 8×8 map have 0 reward but 3 grids with reward 1. In every experiment, we randomly selected the 3 grids. Curves drawn with results using 20 different random seeds. The gray horizontal dash line represents teacher’s expected accumulative reward.	84
3.5	Adversarial teacher results. With an adversarial teacher, a naive learner can no longer learn effectively. ITAL still learns efficiently. SGD and batch learning are included for comparison.	89

3.6	Map configurations	89
3.7	Actual interface used in the human study. The reward spectrum will be shown to the subjects at the experiment introduction. In this example, the subject chose the green arrow as the example at the 1-st iteration. Then, as we annotated with the orange T-shaped boxes, the estimated reward of the target tile of the green arrow increased, while rewards of the arrow source and the surrounding neighbors decreased. The orange boxes were not included in the human study.	90
3.8	Human study results. All the p-values are calculated with paired t-test.	91
3.9	Learning curves for each map.	92
4.1	(a) An example referential game. There are three objects, a blue sphere, a red sphere, and a blue cone. If a student hears “blue” from the teacher, he should be able to identify the blue sphere instead of the blue cone. (b) theory of mind (ToM) Agents Interaction Pipeline. First, the teacher chooses a message according to the context and her prediction of the student’s reaction (blue arrows). After a message is sent, the student updates his belief and the teacher updates her estimation of the student’s belief (purple and orange arrows). Then, the student either waits or selects a candidate (red arrows). Only in the training phase, the actual student belief will be returned to the teacher (gray arrow). Bold arrows stand for the whole message space being passed. Notice ψ_A is part of the Q_A . O and b_{BinA}^{t-1} are passed to ψ_A twice, for message selection and teacher’s new belief estimation. Empty boxes are game and time variants; shadowed boxes are agents’ mental structures. Notations are introduced in Section 4.2 with θ omitted from subscripts.	97

4.2	4 distractors referential game example. Number set on the left (candidates listed in the title with the target in bold fonts) and 3D objects on the right. Due to the space limit, we only show the message distribution for the target and student’s new belief after receiving the most probable message. As for the teacher’s message distribution for distractors, all probability weights concentrate on the unique identifiers after the first phase of training. Student’s belief illustrates that teacher’s most probable message, though consistent with multiple candidates, can successfully indicate the target with more confidence as training goes. In general, both agents’ behavior becomes more certain, and the certainty coordinates.	101
4.3	Finishing one phase of adaptation, agents can correctly identify level 0 targets. Namely, the teacher always sends a unique identifier as long as there is one. After phase 2, the student knows how the teacher will teach level 0 targets. Thus, he can prune out all level 0 candidates if he doesn’t receive their unique identifiers, leaving level 1 candidates the “simplest”. The decrease of level 2 accuracy among 4 candidates might be caused by the lack of level 3 targets. Only 3% of the number set and 0.02% 3D Objects games have level 2 targets, so agents have no motivation to explore for a higher level of reasoning.	104
4.4	Covariance between messages and distractor attributes given the same target but different distractors. In both experiments, our algorithm has a more clear pattern. The blue diagonals show that if an attribute appears among distractors the teacher tends to avoid the corresponding message. The brighter sub-squares illustrate that messages are significantly influenced by distractors’ attributes in the same category. Namely, color messages have stronger covariance with color attributes than with shapes, sizes or positions. Notice that size messages are seldom used in 7-distractor game, thus their covariance with distractors is close to 0. Better be viewed in high resolution, <i>e.g.</i> using Adobe Acrobat Reader. . .	106

5.1	Overview of bidirectional human-robot value alignment.	Pie charts represent the values, <i>i.e.</i> , the importance of different goals in a collaboration task, such as simultaneously considering safety, gaining money, saving time, and reserving resources. t in the superscript represents the time step. A and B in the subscript represent “user” and “machine”, respectively. U_A is the user’s true value, U_{AinB} is the robot’s estimation of the user’s value, and U_{BinA} is the user’s estimation of the robot’s current value. δ denotes the distance between values in the task value space. In every round of interaction, the machine first receives signals from the physical environment and processes its observations to form an abstract state of the environment. Next, the machine presents the processed map together with movement proposals and explanations to human users, who will provide feedback to the system accepting/rejecting the proposals according to human values and current map state. Given the user’s feedback, the machine then updates its estimation of human values and takes actions w.r.t. the new values. Cooperative human-robot communication with appropriate explanation aligns the team values in two directions by diminishing the distance between U_{AinB} and U_A , as well as U_{BinA} and U_{AinB} , resulting in final convergence to the true value U_A	111
-----	---	---	-----

5.2	User interface for the scout exploration game.	(A) From left to right: Legend panel in the first column explains the meaning of various icons used in the game. The value function panel in the second column shows the true values indicating the relative importance of various goals; the values are unknown to the robot scouts and cannot be modified by the user. The four right panels change dynamically over the course of the game. The central panel in the third column shows the current status of the map in the game. The score panel at the bottom shows the current scores for achieving individual goals. The overall score is the sum of the scores for individual goals, weighted by the values known to human users in the value function panel. The Status panel provides a text summary of the current status of the robot system. (B) The Proposal panel shows the robot scouts' current proposals; human users can accept or reject proposals of individual scouts. In the proposal-only group, participants only see a descriptive sentence for each proposal (B.0), whereas, in the brief-explanation and full-explanation groups, participants are presented with a brief explanation about the proposal's purpose (B.1). (C) The Explanation panel shows detailed explanations provided by the scouts, only displayed to the full-explanation group. (D) The bottom table summarizes key components of the game display included in each group.	115
5.3	Study design of the Scout Exploration Game.	Timeline (A) denotes events happening in a single round of the game, starting from scouts receiving environment signals and ending with their next move. Proposals and explanations are presented differently to users depending on their experimental group. The value estimation asks users to infer scouts' value at current time. Answers to these questions will not be used by the scouts during the game, but only for inspecting users' mental model after the game completes. Timelines (B) and (C) depict mental dynamics of the robots and the user, respectively.	116

5.4 Results of value estimation for scouts and humans in three groups.

The legends: proposal, brief, and full refer to the proposal-only group, the brief-explanation group, and the full-explanation group, respectively. Horizontal axis indicates the progress of the game for human participants; vertical axis indicates Kendall’s rank correlation coefficient between estimated values by scouts and humans; higher correlation indicates better value alignment. Top panel **A**: correlation between scouts’ value estimate and the true values that are known to human users as a function of game progress (*i.e.*, scout’s accuracy in estimating human values). Before the game starts, the scouts’ value estimate is initialized as uniform across all goals. Bottom panel **B**: correlation between the human estimate of the scouts’ values and scouts’ estimate of the true values as a function of game progress (*i.e.*, humans’ accuracy in estimating scouts’ values). Asterisks in the plot indicate significant group differences. The error bars indicate the observation minimum and maximum. The solid lines and red dashed lines in the bars respectively indicate the median and mean. 131

6.1 Common and distributed knowledge for inferring a state ω (star) in 1D space between Alice and Bob. Every segment represents a cell in the partition. States fell into the same segment cannot be differentiated. Alice’s and Bob’s perception partition become finer after communication because partners’ messages enable further differentiation of worlds. 134

6.2 An example of partition in 2D space and inference of a state using the pragmatic protocol. **Left**: The partition of the 2D space. States within the same patch trigger identical neural responses. **Right**: Neurons fired for two cells π_a (red) and π_b (green). 138

6.3	CL includes 3 nested loops: i) The reflection loop in black for deliberation of inferential messages to achieve common ground; ii) The learning loop in blue is to achieve a common model, utility, policy <i>etc.</i> ; and iii) The characterization loop in red is to achieve better group understanding of each agent’s characterization, such as value and belief-update functions defined in Section 2.3. The norm of communication forms when group members’ estimations of their partners’ characterizations stabilize.	140
-----	--	-----

LIST OF TABLES

2.1	Learning paradigms in prior work and their recursive level.	32
2.2	Instantiation of CL components for different learning paradigms: Teacher	62
2.3	Instantiation of CL components for different learning paradigms: Student	63
3.1	Selection of β s in the adversarial teacher experiments. For cooperative teachers, the absolute values of the β s are the same, only the signs are flipped.	86
4.1	The first three rows are the total accuracy, while the second three rows mea- sure the accuracy for difficult games (numbers in percentage). We define game difficulty with average cosine similarity between the target and the distractors. The larger the cosine similarity, the harder the game is. We report the accuracy for top 10% hard games. L.A. stands for Lazaridou et al. To verify the general- ization of our algorithm, we form training sets with 70% of all instances and test using the rest 30% unseen instances. The performance doesn't exhibit noticeable degradation: 99.1 $\pm$ 0.1 % and 93.0 $\pm$ 1.7 % for 4 and 7 candidates number set respectively (comparable with the No. Set results in the third line). Mean and std calculated using 3 different random splits, 2 experiments per split.	109
4.2	All models are pretrained with Bayesian belief update. The three rows above are tested with agents seeing different but equivalent candidates, while the three rows below are tested with agents sees identical candidates. The larger the performance difference is, the more the protocol diverges from the initial grounding.	109
4.3	A message is valid if it corresponds to one of the attributes of a novel target. Std calculated with 3 different random splits.	109

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my families, friends, advisers, collaborators, and labmates; I could not make this far without the support from you. In particular, thank you to:

My parents and my girlfriend for their unconditional support in my difficult times and understanding of my pursuing of the Ph.D.

Professor Song-Chun Zhu for the trust and support, and encouraging everyone in VCLA to aim high, work hard and pursue long-term challenges in AI.

Professor Ying Nian Wu for being a supportive and inspiring mentor to me. I really benefit a lot from his thoughtful conversation and academic insights.

Professor Quanquan Gu for the insightful discussions and several last-minute save of our paper as well as the rebuttal.

Professor Hongjing Lu for teaching me the methodologies of cognitive science and patient refurbishment of our manuscript.

Professor Demetri Terzopoulos for offering me the opportunity to be a teaching assistant in his class and for his time and support on my committee.

Professor Yixin Zhu for his help in the lab and frequent academic advice.

Arjun Akula, Sam Baugh, Mark Edmonds, Lifeng Fan, Zipeng Fu, Feng Gao, Xiaofeng Gao, Ran Gong, Baoxiong Jia, Qing Li, Changsong Liu, Jonathan Mitchell, Siyuan Qi, Jingyue Shen, Junhong Shen, Tianmin Shu, Hao Wang, Sirui Xie, Xu Xie, Lu Xu, Yizhou Zhao, Zilong Zheng, Dongruo Zhou, Yaxuan Zhu and everyone in VCLA for their discussion, collaboration and friendship over the years.

The dissertation text is a combination of multiple published and unpublished papers. Chapter 1, 2 and 6 belong to an adaptation of an unpublished manuscript to be submitted as a position journal article in the coming months. Chapter 3 is a version of [YZS21], published

in NeurIPS in 2021. Chapter 4 is a version of [YFS19], included in the Emergent Communication Workshop of NeurIPS in 2019. Chapter 5 is a preprint manuscript submitted to Science Robotics. The author contributions are the following: Luyao Yuan contributed to ToM modeling, experimental design, and human subject data collection; Mark Edmonds contributed to simulator design, experimental design, and human subject data collection; Xiaofeng Gao contributed to simulator design and experimental design; Zilong Zheng contributed to explanation generation and simulator design; Yixin Zhu contributed to simulator design and experimental design; Hongjing Lu contributed to experimental design; Song-Chun Zhu contributed to research direction and served as the PI.

These works were supported by DARPA XAI project N66001-17-2-4029, ONR MURI project N00014-16-1-2007 and NSF DMS-2015577. I am grateful for the collaboration, inspiration, and funding support these grants provided.

VITA

2017–2022	Graduate Research Assistant, UCLA, Dr. Song-Chun Zhu.
2021 Summer	Machine Learning Engineer, Facebook.
2018–2020	Teaching Assistant, UCLA.
2019	Ph.D. Candidate (Computer Science), UCLA.
2017	B.S.(Computer Science) with Highest Distinction, University of Michigan.
2016 Summer	Undergraduate Research Opportunity Program (UROP), University of Michigan, Dr. Honglak Lee.

PUBLICATIONS

Show Me What You Can Do: Capability Calibration on Reachable Workspace for Human-Robot Collaboration. X. Gao, **L. Yuan**, T. Shu, H. Lu, S.C. Zhu, *IEEE Robotics and Automation Letters (RA-L)*, 2022.

Iterative Teacher-Aware Learning. **L. Yuan**, D. Zhou, J. Shen, J. Gao, J.L. Chen, Q. Gu, Y.N. Wu, S.C. Zhu, *NeurIPS*, 2021.

Emergence of Theory of Mind Collaboration in Multiagent Systems. **L. Yuan**, Z. Fu, L. Zhou, K. Yang, S.C. Zhu, *NeurIPS Workshop on Emergent Communication*, 2019.

Emergence of Recursive Teaching Protocol between Theory of Mind Agents. **L. Yuan**, Z. Fu, J. Shen, L. Xu, J. Shen, S.C. Zhu, *NeurIPS Workshop on Emergent Communication*, 2019.

CHAPTER 1

Introduction: Inspect Learning in A Communication Framework

Better than a thousand days of diligent study is one day with a great teacher.

—Chinese proverb

When I walk along with two others, they may serve me as my teachers.

—Confucius

1.1 Objective: A Unifying Formalism

The recent surge of statistical and machine learning enables artificial intelligence (AI) to achieve impressive performance in many tasks. However, current machine learning paradigms also demonstrate several shortcomings: demand of large training data, uninterpretable and noncommunicable representations, and deficient generality to new tasks. These paradigms belong to a “big-data for small-task” paradigm [ZGF20] – drastically different from human learning, which communicates a wide range of daily tasks effectively using small data, *i.e.* “small-data for big-tasks”, and entails diverse protocols between multiple participants. To fill in the gap between prevailing machine learning paradigms and human learning, in this dissertation, we propose communicative learning (CL). This learning formalism models a multiagent system, where the agents (a teacher and a student) communicate with each other in the process of teaching and learning. In CL, an agent’s mental representation

includes a set of minds: a mind that consists of the agent’s current belief of the knowledge of interest, its utility, and dynamic functions, a mind that estimates its partner’s egocentric mind, a common mind, and a God’s mind. These mental components then jointly drive learning and communication. With CL, we call for a **learning as communication** paradigm and demonstrate its advantage over non-CL methods. Furthermore, we show that this learning formalism encompasses and goes beyond existing learning paradigms. We use the perspective of the teacher and student mutual reasoning to survey prior works on various kinds of machine learning algorithms. We illustrate that despite their diversity, many types of teachers and students proposed thus far can be characterized as instantiations of CL. This offers a unifying lens for the integration of cooperative pedagogy [Shu87] and machine learning, helping readers to better understand and contrast prior methods.

1.1.1 Cognitive Infrastructure for CL

In human societies, communicative behaviors are so prevailing that most of us presume them for granted, without realizing the complexity of the cognitive infrastructure that permits even the simplest communications to occur. Even intentional signals, considered as a rudimentary form of human communication, are extremely rare in the biological world, perhaps confined to primates or even great apes [Tom10, Chapter 2]. Way beyond signaling or informing others deliberately, human communication is a complicated system aiming to establish joint attention and common ground for the completion of shared goals, which is motivated by cooperation norms among people and enabled by communicative conventions and cognitive infrastructure supporting recursive cooperative reasoning [Tom10, Chapter 3]. Human learning, as a lifelong cognitive process of communicating with the physical and social world, also operates in such a cooperative framework. Its sophistication, effectiveness, and complexity give rise to human intelligence: a phenomenon that AI is inspired to replicate.

Decades of studies in cognitive psychology [HHT95, LST15] and anthropology & communications [Tom10] have revealed that human communication and learning is built on many

layers of cognitive infrastructures and protocols. To account for its complexity and sophistication, we formulate the mental representation of CL. Fig. 1.1 shows the key representations between two agents A and B , who can be human or machine, teacher or student in an equal and symmetric setting, *i.e.* they can exchange roles by turns. This representation should endow CL with the following properties:

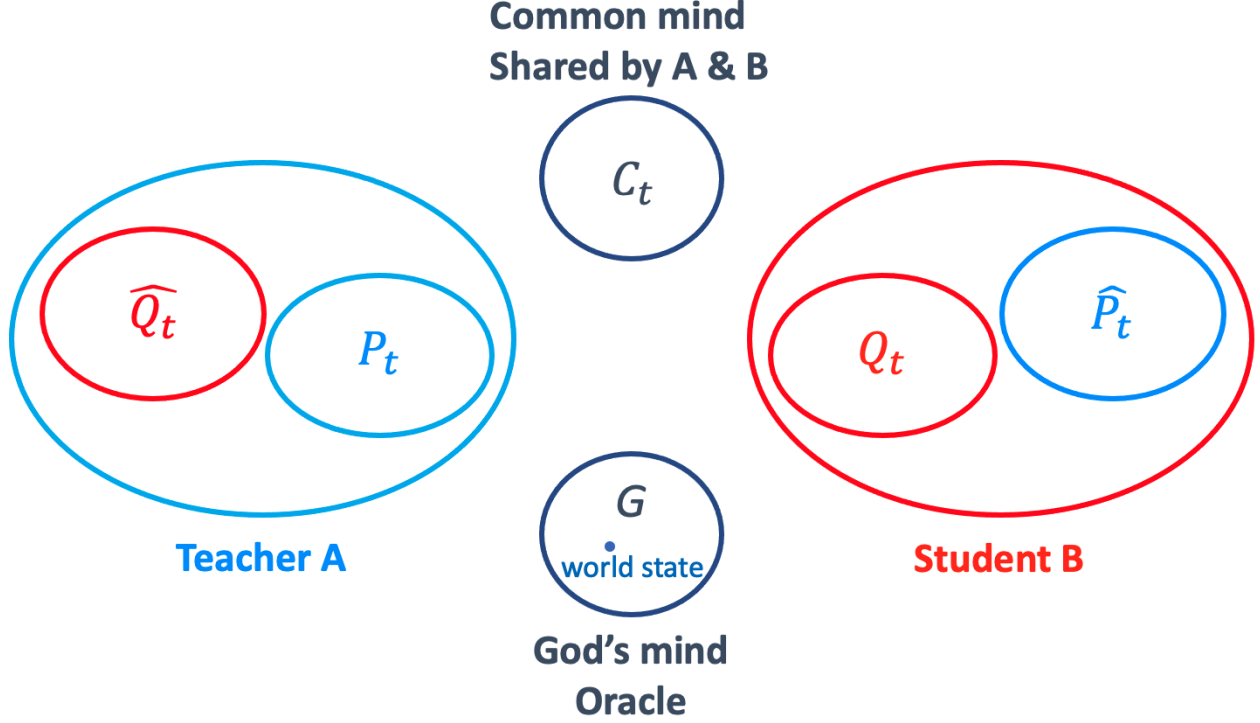


Figure 1.1: **Key representations of CL** includes *six minds* that evolve over t time steps or messages. 1) G : oracle in God's mind, usually not time-variant; 2) P_t : mind of the teacher A ; 3) Q_t : mind of the student B ; 4) $\widehat{P}_t$: B reflects what A knows; 5) $\widehat{Q}_t$: A thinks what B knows; and 6) C_t : common mind that both A and B know that each other know.

- Shared common ground. The two agents have shared commonsense knowledge, shared situation, attention, and intent to ground learning in context for effective communication.
- Theory of mind (ToM) [PW78]¹ representations, which engage six minds between the

¹The ability to attribute others mental states such as beliefs, intents, desires, emotions, and knowledge.

teacher and the student: what the teacher knows; what the student knows; what the teacher thinks the student knows; what the student thinks the teacher knows; common mind; and an oracle. The discrepancy between the six minds drives the communication and learning process through deliberated messages instead of randomly sampled examples.

- Unifying framework for learning, which embraces existing learning methods as its special cases in several axes: supervised vs unsupervised; passive vs. active; observational vs. causal experimentation; and ease the development of new learning protocols in novel conditions.
- Integrating pedagogy into machine learning, which facilitates the learning with the consideration of a helpful teacher and cooperative communication.

Our insight is that, armed with appropriate infrastructure sufficing the first two properties, CL can naturally accomplish both the unification of previous teacher-student teaming in a single formalism and the integration of pedagogy. The critical observation is that the teacher can choose messages for the student according to various criteria [Gri75, LSL00, GS13] given her intended teaching content, and the student should update his belief according to his estimation of the teacher’s message selection mechanism [ESS15, EFG16, HLM16]. With distinctive teaching criteria, we can unify diverse teachers, from passive oracle to cooperative pedagogue. In Chapter 2, we elaborate the CL representation and rigorously define the components of each mind.

1.1.2 Unifying Framework for Machine Learning

CL is a unifying framework where existing machine learning algorithms can be shown as its special cases. Dating back to the 1960s, machine learning was put forth to enable computers to recognize and capture patterns from data [Sam59, Bis06, SB14]. These days, with the availability of extensive data and computational power, complicated models can

be exploited for various tasks, from image classification [DDS09a, KSH12], object detection [GDD14, Gir15, HGD17], sentence generation [DCL18] to exceeding human-level game playing [MKS13, SHM16] and so on. Currently, most of the prevailing machine learning methods focus on the optimization of individual learners, relying purely on unilateral experiences, either passive observations from a Markov decision process (MDP) [MKS13, SHM16], random samples from a data distribution [DDS09a, HZR16], responses of active queries provided by an oracle [Ang88, Set09], or demonstrations from an expert [ACV09]. Lately, the advantage of pedagogical teachers over randomly sampled data or optimal task completion trajectories from experts has been shown in Bayesian concept learning [SGG14, ESS15, MAM17, YYW18, CMS18, CSM18] and in learning from demonstration (LfD) [HRA16, HLM16, HLC18]. Meanwhile, machine teaching algorithms [CL12, Zhu13, Zhu15, LDH17, FTQ18] start to model cooperative teachers giving instructions in continuous parameter space and large datasets. In Fig. 1.2, we compare some typical learning paradigms.

That being said, there have already been lots of machine learning algorithms introduced in the field. Some model learning unilaterally, while others treat learning as a multiagent communicative process. To embrace the appearance of more sophisticated machine learning paradigms in the future, one can only expect a generalized and holistic machine pedagogical formalism, accounting for the multi-layered cognitive infrastructure underneath human and machine learning. With mental representation inspired by human communication [Tom10], CL is able to capture the homogeneity among diverse learning algorithms and unify them in one generic formalism. In Section 2.2, we specify the relationships between CL and common learning paradigms.

1.1.3 Integrating Pedagogical Reasoning

Another merit of CL is enabling the integration of human pedagogy into machine learning. When one pictures **human learning**, either of a toddler at home or a student at school, the scenario often involves two parties: a teacher and a student. The teacher tries to impart her

knowledge to the student with the most helpful teaching material, and the student actively absorbs the information to achieve the learning goal as efficiently as possible. Taking place in a multiagent system, this type of learning is a communicative process, referred to as pedagogy [Shu87]. In **machine learning**, however, the role of the helpful teacher is usually replaced by the training data (in the form of dataset or interaction experience) sampled from some random process [SB14]. Suppose we draw an analogy: Without a cooperative teacher, machine learning algorithms act like scientists, who endeavor to detect, explain and utilize the pattern of the world from their observations of randomly occurring phenomena, such as apples drop, stars shine, and the rain falls. The most common type of learning in human society is not in the form of scientific discovery but cooperative pedagogy, invoked across language, cognitive development, and cultural anthropology, to explain people’s ability to effectively transmit information and accumulate knowledge [FTQ18, WWP20]. There has been sufficient experimental evidence in cognitive science justifying people’s ability and tendency to teach and learn differently in pedagogical situations. From infants and toddlers’ ability to distinguish different sampling process [GTS10] and their awareness of pedagogical behavior [CG06, CG09, XD09] to the application of pedagogical inference in word learning for both children and adults [XT07], these studies illustrate that human learners are sensitive to the distinction between pedagogical teaching material and random examples. Humans, from a very young age, can capture additional information when they are in pedagogical settings [SGG14]. Furthermore, besides being capable learners, children can also learn how to be a good teacher and generate helpful evidence for others when trying to reveal the mechanism of toys [GSS18].

The main difference between a scientist and a student in a pedagogical situation is the existence of a cooperative teacher, who, interacting adaptively with the student, can maximize communication efficiency. Namely, the teacher adjusts her teaching method for different students, and the student, after getting familiar with the teacher’s instruction mechanism, can infer the teacher’s intention and learn faster [Gri75, SW86, CG09, SB14, PCD19]. Re-

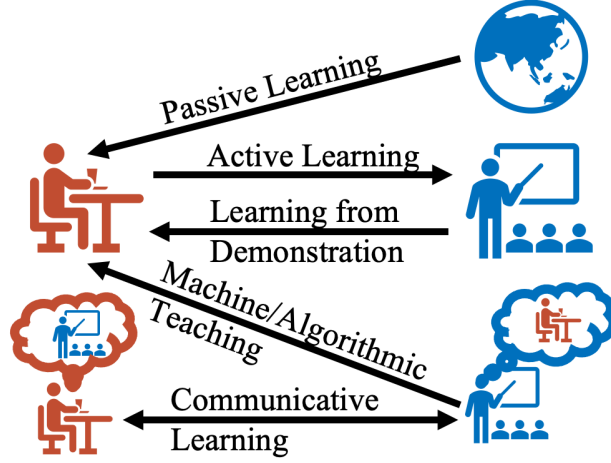


Figure 1.2: Comparison among different learning/teaching paradigms. Blue characters represent teachers and red characters are students. The bubble means modeling one’s partner.

cently, realizing the conundrum of data-hungry and “big data for small tasks” [ZGF20] for conventional machine learning algorithms, especially compared with the efficiency of human learning that can be consummated quickly with very limited examples [SG08, CG09, SGG14], researchers begin to integrate pedagogy into the improvement of machine learning algorithms[CL12, Zhu13, SGG14, ESS15, Zhu15, HRA16, HLM16, LDH17, MAM17, CMS18, CSM18, HLC18, FTQ18, YYW18]. Nonetheless, compared with human pedagogy, these works lack a sophisticated student model that can accommodate the teacher’s cooperation into his learning and acts differently with learning from passive data. Recursive cooperative inference models were proposed in [WWP20, WWS20] with both the teacher and the student having ToM [PW78]. Yet, the analysis of these works is confined to Bayesian concept learning with a finite and relatively small dataset and hypothesis space. In Section 2.3, we illustrate how CL facilitates the combination of human pedagogy and machine learning and the development of more advanced learning algorithms.

The rest of the dissertation is structured as follows: In Section 1.2, we clarify the relationship between learning and communication and motivate a CL paradigm. In Chapter 2, we introduce the CL formalism, starting with the agent modeling and their necessary mental representations in Section 2.1, followed by mathematical definitions of the learning dynamics

in Section 2.3. We also show how existing learning algorithms are special cases of CL and their CL grounding in Section 2.4. Then, we give the theoretical guarantee of the CL formalism in Chapter 3, showing its dominance over non-CL learning approaches. In Chapter 4, we demonstrate how efficient learning protocols can emerge from agents conducting CL using the example of the referential game, a simple but complete communication as well as learning scenario. Afterward, in Chapter 5, we illustrate the applicability and necessity of the CL formalism in a more complicated real-time human-robot collaboration task, a prototypical example for generic human-robot Interaction (HRI). Last but not the least, in Chapter 6, we discuss the implications of CL and how it sheds light on the development of future machine learning algorithms. In particular, we put forth a new representation of learning and give a learning protocol beyond Shannon’s communication limit in Section 6.1. In Section 6.2, we tease out the three hierarchies of machine learning and define the fundamental halting problem of learning.

1.2 Inspect Learning in A Communication Framework

Return to the scientists versus students analogy. It does not take much argument to convince people that learning is a communication process, because most of us once were or still are students and experience learning in academic settings, where we acquire knowledge, skills, and values through communicative interactions with teachers. Within the Western culture, this form of learning is said to originate from Socrates, who exposed the idea of using dialogues between individuals to elicit impart wisdom and transfer knowledge from the teacher’s mind to the student’s. In fact, the one-way communication from nature to a scientist, if we are willing to assume the existence of an omniscient being, *e.g.* God, can also be viewed as transferring knowledge from God’s mind to the scientist’s mind. Just that the messages from God are not as decipherable as those from Socrates (at least to most of the people). More generally, we thus can interpret learning broadly as **having information delivered from**

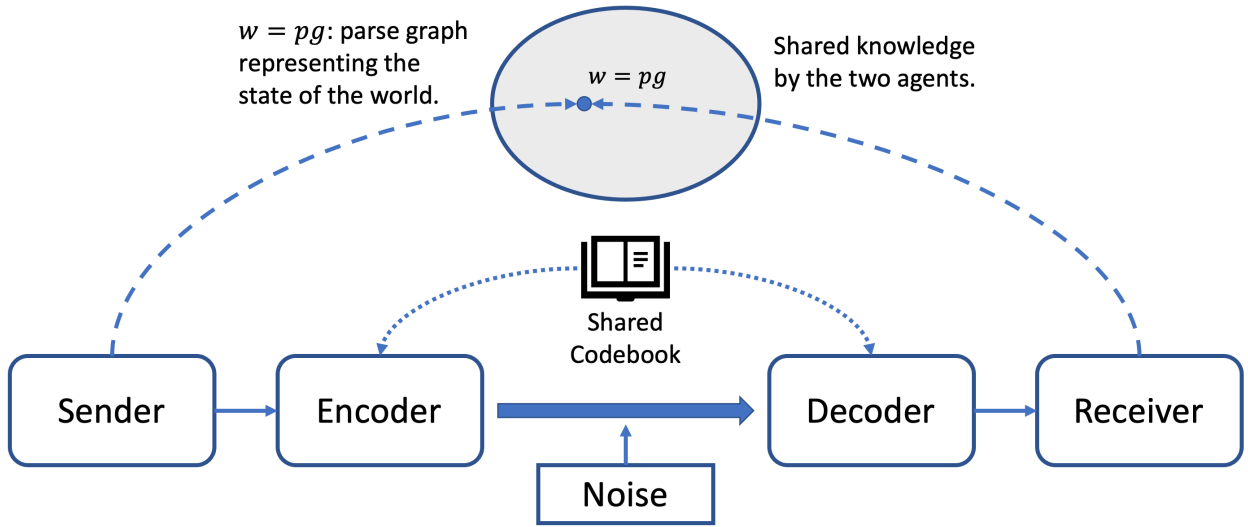


Figure 1.3: Shannon’s diagram of a communication system. The shared codebook fulfills the purpose of a common ground and the norm of communication in the cooperative communication model.

one mind to another. Interestingly, a textbook definition of communication is exactly the act of delivering. Clearly connected, what is the relationship between communication and learning? To answer this question, an obvious way is to look at information theory [Sha48] and statistical learning theory [Fri17], two well-developed disciplines which formally study communication and learning from a mathematical viewpoint. As we will see, they have a lot in common.

1.2.1 Connection between Communication and Learning

Information theory was established by Claude Shannon in 1948 [Sha48] as a framework to account for communication over, say, a telephone wire. As shown in Fig. 1.3, in this framework, the sender and receiver share a codebook, and the messages refer to some world state w , *e.g.* a parse graph of an indoor scene or semantic of a paragraph. Such a communication system is for *reproducing, either exactly or approximately, a message selected at other points*. Loosely speaking, it involves an information source that selects a message from a set of possible messages and then encodes it into a sequence of communication symbols

suitable for transmission. This sequence is subsequently sent over a possibly noisy medium to a receiver that decodes and reconstructs the original message according to a predefined coding scheme. Significantly, Shannon showed that depending on the characteristics of the transmission medium, there is an upper limit on the rate, say bits per second, at which messages can be reliably transmitted through such a system. This limit is termed *channel capacity*. Every time a message is received from the channel, the receiver’s uncertainty about the possible worlds will decrease by an amount bounded by channel capacity. Suppose we denote the set of possible worlds at time t as $\mathcal{W}^t$. Then the information gain brought by message at time $t + 1$ can be represented as

$$\delta^{t+1} = \log \frac{1}{|\mathcal{W}^{t+1}|} - \log \frac{1}{|\mathcal{W}^t|} = \log \frac{|\mathcal{W}^t|}{|\mathcal{W}^{t+1}|}. \quad (1.1)$$

One drawback of Shannon’s theory is that it has intentionally left out the “semantics” or “meanings” of messages. Sender and receiver are assumed to share common ground to make sense of the messages outside this framework. The limits of coding efficiency and channel capacity are based on a protocol that does not model mental states and motives of agents in sending messages. In our new CL framework, messages are selected after **deliberations & reflection** using ToM representations, and carry extra information that is recoverable in a more effective communication protocol, *i.e.* agents are capable of “**reading between lines**”.

Compared to information theory, statistical learning theory, though relatively young, went one step further. As per the most well-known Probably Approximately Correct (PAC) model proposed by Leslie Valiant [Val84], learning seeks to accurately acquire a concept selected by the nature by looking at some training data. The PAC-learning theory bounds the number of training data, $n(\epsilon, \delta)$, that are needed to learn the concept with error $\leq \epsilon$ and confidence $> 1 - \delta$. Therefore, by equating a concept with a message and training data with communication symbols, we find a strong parallel between Shannon’s communication model

and PAC learning. To continue the previous metaphor, learning describes the communication of concepts from one mind to another. Specifically,

- acting like an information source, nature selects a concept from a class according to a certain probability. It then uses a sequence of instances to encode the concept with binary labels to form consistent training examples;
- like signals transported over a transmission medium, labels may then be subject to noise and flip;
- upon seeing possibly noisy training examples, a machine runs a learning algorithm to decode and reconstruct the target concept [Riv91];
- analogous to channel capacity, in learning, we have an upper bound on how fast concepts can be transmitted with training examples. It is measured by *sample complexity*, that is, the minimum number of training examples a machine needs to see before it can reliably identify any intended concept.

In summary, unlike communication, statistical learning puts more emphasis on decoding. Note also that because of these well-established theories, the very words communication and learning have been overloaded with meanings.

1.2.2 Go beyond Shannon and Valiant

Now that we reviewed the connection and foundations of information theory and statistical, especially PAC-learning, eagle-eyed readers must have realized that Valiant's learning model, despite its theoretical appeal, has rather limited explanatory power outside the scientist's learning scenario. It makes an inefficient assumption that examples in learning are **random samples**, whereas, in most real-life cases, learning is a communication process where examples are **deliberate messages** uttered by reflecting mental states of the student and

the teacher. In addition, the lack of pedagogy at the teacher’s end deprives the possibility of having a student other than a completely passive learner, whose model is much more rudimentary than our learning experience as human beings. In possession of the agency and guided by purposes, we not only engage in interpretation of received information, but also actively ask questions to dispel doubts. Furthermore, assuming that other people are similarly endowed and motivated, we often question why others behave in a certain way and take into account their possible reactions to our moves before we act. These crucial elements are missing from Valiant’s model.

Here, we can show a toy example to illustrate the advantage of deliberate messages over random sampling. Suppose there are four sets of numbers: $\{1, 2, 3\}$, $\{4, 5, 6\}$, $\{4, 5\}$, $\{1, 2, 4, 6\}$ and the teacher can inform the student which one of the four is the target set by sending numbers belongs to the target set. We can consider the case when the third set $\{4, 5\}$ is the target. For a student learning from random sampling, it takes many repetitive numbers to shape his belief and eliminate the second set, which happens to be the superset of the target. However, if learning from a cooperative teacher, more efficient protocols can be formed between the agents, some can even accomplish deterministic identification of the target with only one message. See Fig. 1.4 for an example. These protocols are only possible between cooperative and pedagogical agents.

Therefore, a model for Socratic learning by dialogue goes beyond just PAC-learning, *i.e.* one-way communication of concepts. It involves meaning and intention. This observation is not new. As Shannon himself acknowledged [Sha48], *frequently the messages have meaning; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem.* In a follow-up work [Wea49], Warren Weaver also pointed out that a broader understanding of communication should include all the procedures by which one mind may affect another. Specifically, he suggested that general communication problems be considered at three levels

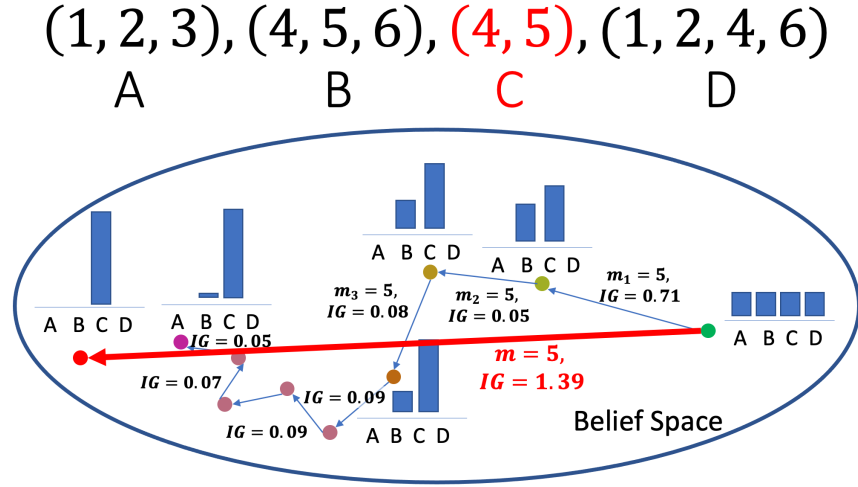


Figure 1.4: Student's belief update processes when learning from random sampling (blue trajectory) and a cooperative teacher (red trajectory). It is possible to transmit the target set $\{4, 5\}$ with a single message because the RTD [ZLH11] of this concept space is 1, lower bounding the regular TD. Concretely, the student knows the teacher will send 3 for concept A, and thus 1 or 2 indicates D. Similarly, 6 can only mean B, making 5 a unique identifier for C. The prerequisite of such a recursive protocol is that the teacher is pedagogical, and the student is aware of her helpfulness. Here, *IG* is short for information gain.

1. How accurately can the symbols of communication be transmitted? (The technical problem.)
2. How precisely do the transmitted symbols convey the desired meaning? (The semantic problem.)
3. How effectively does the received meaning affect conduct in the desired way? (The effectiveness problem.)

Though lying at the lowest level, the technical problem of transmitting symbols, or concepts for that matter, leads to insight in that it helps identify several key ingredients to enable successful communication, at all levels. However, symbol transmission only provides the necessary postal infrastructure for participants of a general communication process to convey *meaning* and express *desire*. Indeed, human communication is way more complex. As an illustration of how meaning construction (level 2) can be decoupled from transmitted

symbols (level 1), it suffices to imagine the different reactions to the same TV images of a football match by the fans of opposing sides. Furthermore, even if there is only one possible interpretation of a particular message, in human communication, the receiver need not simply accept, but may alternatively ignore or oppose a message (level 3). The heated political discourses in the United States on climate change is a prime example of this phenomenon.

Since our goal is to model learning through communication in the broad sense, in light of the prior work, we thus have to account for the two higher levels, primarily the semantic problem level, of communication. To understand how level 2 communication (in semantics) is conducted among human beings, we refer to an account by Michael Tomasello from an evolutionary and cognitive perspective [Tom10]. According to Tomasello, the key to the success of human communication is a **common ground shared by its participants**, which includes joint attention, shared experience, and common cultural knowledge. A common ground provides the critical situational, social, political, cultural, and historical context for people to construct meaning based on their received communication symbols.

To illustrate this point, consider an example given in [Tom10]. Suppose that you and I are walking to the library, and out of the blue I point for you in the direction of some bicycles leaning against the library wall. Your reaction will very likely be “Huh?”, as you have no idea which aspect of the situation I am indicating or why I am doing so, since, by itself, pointing means nothing. But if some days earlier you broke up with your boyfriend in a particularly nasty way, and we both know this mutually, and one of the bicycles is his, which we also both know mutually, then the same pointing gesture in the same physical situation might mean something very complex like “Your boyfriend’s already at the library (so perhaps we should skip it).” On the other hand, if one of the bicycles is the one that we both know mutually was stolen from you recently, then the exact same pointing gesture will mean something completely different. Or perhaps we have been wondering together if the library is open at this late hour, and I am indicating the presence of many bicycles outside as a sign that it is.

It is important to point out that the common ground is not a new notion because it has already existed in Shannon’s communication model and Valiant’s PAC learning model. In traditional communication, it is represented by the common codebook shared by the sender and receiver, whereas in PAC learning, it is the common class of concepts. However, what appears novel in human communication is that common ground is jointly created and selected by the participants, rather than by a third person who sets up a communication system or a learning algorithm.

These observations highlight the connection between learning and communication, and motivate our CL formalism. In the next chapter, we will show the mental representation of the teacher and the student, who communicate with each other to achieve efficient pedagogy.

CHAPTER 2

Formalism of Communicative Learning

In this chapter, we define the CL formalism. We start with the infrastructure of CL and the mental representation of the teacher and the student. As discussed in Section 1.1.1, the infrastructure needs to have sufficient expressiveness to accommodate the common mind and ToM. Then, we present the dynamics of learning in Section 2.3 to illustrate how the learning proceeds with cooperative reasoning and pedagogy integrated.

2.1 A Full-blown Teacher-student Mental Representation

In the previous chapter, we discussed that ToM is the prerequisite of CL without rigorously defining what is a mind. Now we elaborate on the mental representations in the CL framework. There are six minds in the CL framework between two agents: the teacher and the student. The first pair is the egocentric minds of the agents. The second pair are the agents' estimation of their partner's mind. These four minds, together with a common mind between the agents and the mind of an oracle (God)¹, compose six minds in total. Theoretically, the mutual reasoning in multiagent systems is recursive and can be infinite if no convergence exists [FMH03, DG09, AS18, FCA18]. *e.g.*, A knows fact e ; B knows A knows fact e ; A knows B knows A knows fact e , so on and so forth, ad infinitum. To avoid the computational intractability, we only model one level of recursion, which is also corroborated by the cog-

¹This mind is also considered as the objective world or nature, whose dynamic is attributed to two factors. The first is a set of physical rules, either deterministic or stochastic, inherent to the world, and the second is agent actions.

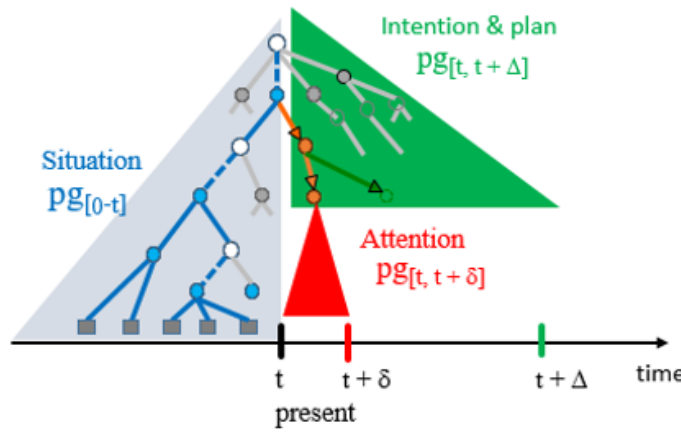


Figure 2.1: State $w = pg$ unfolding over time

nitive ability revealed in multiple human studies [DVV14, DVV15]. Another caveat is that the representation in this section doesn't include the dynamics of the minds, *i.e.* how a mind updates during the interaction between the agents. We defer that part of the discussion to Section 2.3. For now, let's just assume they have a way to evolve given the inputs from the world and the partner. An agent's mind includes the following components:

A state w representing a situation. w stands for world and is often structured as a mental state. *E.g.*, a state $w = pg$ is a parse graph, and in a more general case, w is a spatial, temporal & causal (STC) parse graph, *stc* – *pg* that we have developed for vision [ZM07, QZH18], language [LCS16, LYS16], robotics [SHC17] and commonsense reasoning [EGX17, FZ15, ZHG15, ZZC15]. The CL messages will be exchanged in a so-called **situated communication**, *i.e.* the parse graph represents the composition of a scene, *e.g.* a living room, objects inside the scene, and actions happening together with causal changes of fluent (aka. time-varying states of objects). The situation is unfolding over time, as Fig. 2.1 shows, a parse graph *pg* consists of 3 parts.

- $pg_{[0,t]}$ summarizes current situation (blue);
- $pg_{[t,t+\Delta]}$ predicts intents and plan (green);
- $pg_{[t,t+\delta]}$ is current attention (in red triangle).

As CL is an iterative process, the parse graph will be communicated via messages over time at multiple semantic levels.

CL agents have a common mind, whose state $w_c = pg_c$ (see Fig. 2.2) also contains: i) shared situation $pg_{c[0,t]}$; ii) shared goal/intents $pg_{c[t,t+\Delta]}$; and iii) shared attention $pg_{c[t,t+\delta]}$. Such representation is key to human communication as is suggested by archeology and cognitive studies [Tom10], so that CL agents can “get to the point” rapidly at the right level of details. Thus, in order to deal with complex situations, CL is a **situated learning** and communication process, which is more general than existing learning methods. There has been some prototype CL learning demonstrated in human-robot interaction and teaming [ZZC15, LYS16, SHC17].

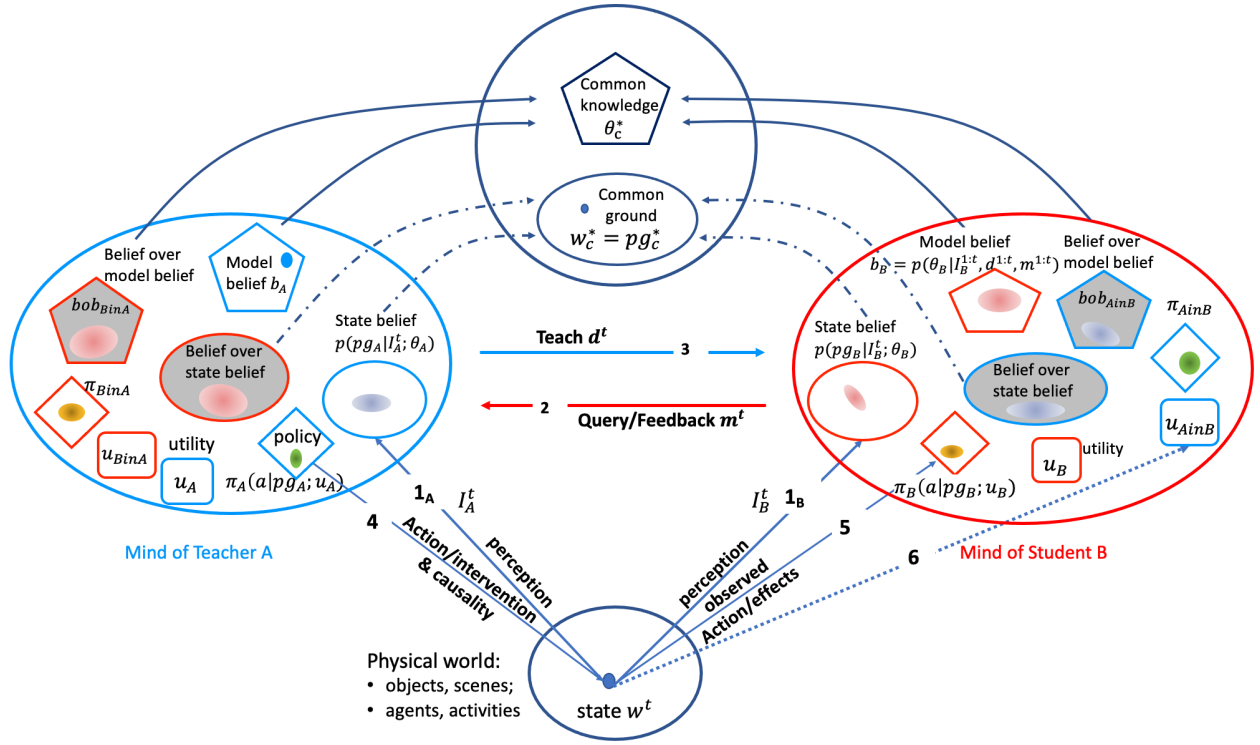


Figure 2.2: A zoomed-in view of CL representations: unifying all existing learning protocols and beyond. Each mind contains four spaces: i) pentagon for hypothesis/model space Θ ; ii) ellipse for the situation $s = pg$ in the state space S where the belief is represented by a cloud; iii) diamond for policy π ; and iv) square for utility u . The arrows illustrate the dynamics: observation, intervention, and messages. The shaded shapes represent belief over belief.

A model $\theta \in \Theta$ is in a hypothesis/model space Θ . In a deterministic setting, like Valiant’s PAC-learning, a model is a concept represented by a set, *e.g.* an object category, which is equivalent to a uniform distribution over this set. In a probabilistic setting, a model defines a probability distribution $p(s; \theta)$ on the state space, usually referring to the parameters of the distribution. It can be the hyperplane of support vector machine (SVM), weights of deep neural network (DNN) or rules of a stochastic grammar. When state $w = pg$ is a structured parse graph with varying configurations, we represent model $p(w; \theta)$ with an And-Or Graph (AOG) [ZM07], whose language is a set of all valid configurations associated with probability. A parse graph pg is an instance and realization of the grammar or AOG. In its full-blown complexity, a model is represented by an STC-AOG integrating three hierarchies:

- Spatial hierarchy: scenes – objects – parts – primitive, for parsing scenes and objects.
- Temporal hierarchy and compositions: events – actions – movements for parsing events.
- Causal hierarchy: actions and fluents for causal reasoning and task planning.

STC – AOG can be viewed as a unified representation for image parsing in computer vision, language parsing in natural language understanding (NLU), task planning in robotics, and cognitive reasoning in commonsense AI.

Belief and belief over belief (BoB) We denote the teacher’s observations and input by I_A , and the student’s input by I_B . I_A and I_B can be an input image or video. We also denote the messages from the teacher at time t as d^t and from the student as m^t . Also, let $a_{A/B}$ be the action of agent A/B . As neither the physical world nor the other agent’s mind is fully observable to an agent, most of the time, agents need to form beliefs over the structure of interest. Denote the history for the student B up to time t as $h_B^t = [I_B^{1:t}, d^{1:t}, m^{1:t}, a_B^{1:t}]$, then $b_{B,w}^t(w) = p(w|h_B^t)$, $b_{B,\theta}^t(\theta) = p(\theta|h_B^t)$. Same for agent A , except that usually we assume $b_{A,\theta}^t(\theta) = b_{A,\theta}^0(\theta)$, that is the teacher has a correct and static model at the beginning. For the ToM structures, uncertainty also exists. Thus, BoB is defined as $bob_{AinB,\theta}^t(b_{A,\theta}) = p(b_{A,\theta}|h_B^t)$

and $bob_{AinB,w}^t(b_{A,s}) = p(b_{A,w}|h_B^t)$, similar to the teacher. As one might notice that the history increases exponentially as the time goes and the model of BoB is theoretically intractable in general cases. In practice, for the history, multiple independence assumptions and inductive biases are indeed required to simplify the computation of the posteriors. For the nested belief, due to the deterministic belief update of Bayesian filters, if the state space is not too big, most of the time approximation techniques like particle filters [DG09] can handle the calculation of BoB. But particle filter can hit its limitation when the state space is large or even uncountably infinite, a usual case for bob_θ . In those scenarios, one can continue the learning by taking the maximum a posteriori (MAP) estimation as the approximation of the distribution [PCD19, HHP20].

A decision policy $\pi : W \mapsto \Delta(\mathcal{A})^2$ mapping current state, w^t , to a distribution of actions $a \in \mathcal{A}$. In more general cases, action can be composed into structured plans, *e.g.* in the form of a $t - pg$. Notice that this action a refers to actions in executing real-world tasks, not about how an agent sends messages in CL. The latter is a CL protocol which we will elaborate late in Section 2.3. π_A and π_B denote the policy of two agents. In practice, to represent such a mapping from states to actions, we usually assume the policy has a certain form and only track the parameterization of it. Also, policies are usually determined by the agent’s value, which is defined as the utility function.

A utility function $u : W \mapsto \mathbb{R}$ represents the value of the agent, *i.e.* what the agent cares about, the loss of mistakes, and the cost of actions. Combining with the model, *i.e.* how the world transit given actions, utility functions can be used for task planning [SHC17]. u_A and u_B denote the utility functions of two agents. The agents A and B in CL must also estimate and learn the utility function of other agents, which we denote by u_{BinA} , u_{AinB} respectively. As we will discuss later, u_{BinA} , u_{AinB} influences the CL protocol and equilibria of the learning process, resulting in different limits of learning.

²We use $\Delta(X)$ here and in the rest of the thesis to represent the space of distributions over X .

2.2 A Unifying Framework of Learning

To summarize the representation discussed in the previous section, we illustrate the CL representations in Fig. 2.2 in a zoomed-in view. There are six minds in total. Each agent has two minds colored differently, one for itself and one for its partner. Every one of these four minds has four components, (w, θ, π, u) , represented by different shapes. We model the uncertainty of w and θ by maintaining beliefs and BoB for them. The other two minds are the common mind and the physical world, *i.e.* God’s mind.

The arrows in Fig. 2.2 show the various dynamics and information flows, including 3 types:

- Observations I_A and I_B from the physical state to perceived state space W ;
- Actions or interventions that cause changes in the physical state;
- Messages between the two agents to exchange information. Depending on the learning modes, these messages are for inference, learning, demonstration, confirmation, etc.

For clarity, we omit arrows for other dynamics: for example, some messages may be generated from a BoB-space to probe what the other agent is thinking, like “I think your state estimate w is ...”. or “what do you know about the state w ?” Some arrows are second-order, for example, the teacher learns the policy π_{BinA} from observing how the student conducts a task, *i.e.* LfD [ACV09, ZZC15], or learning the utility u_{BinA} by watching the student’s decision or choice [SHC17].

In CL, the communication of the teacher and the student converges at three levels (see curved arrows in Fig. 2.2):

- When the inference process converges, they reach a common ground or situation w_c^* .
- When the learning process converges, they reach a common model knowledge θ_c^* ;

- When their policy & utility converges, they reach a common social norm π_c^* and ethics u_c^* .

Depending on the learning protocols and characteristics (*i.e.* capacity of generating and interpreting messages) of the agents, the convergences may have different equilibria which decide the limits of learning. In CL, we assume the agents are cooperative and not deceptive, and their utility functions are aligned through learning. In most learning setups, convergence is considered in the second and the third level, and we will use them as the convergence criteria for most of this dissertation. We will discuss multiple levels of convergence in Section 6.1.

The motivation of CL is to integrate the essence of human pedagogy into machine learning and overcome the limitation of existing algorithms. In fact, with current representations, we can see that CL does unify existing learning methods, with the relations as the following:

Shannon’s communication diagram in Fig. 1.3 is a message passing channel between 2 agents. CL extends this communication setting by including the mental states, the BoB-space, utility functions, and a common mind, which all evolve over time. This will allow more sophisticated messages, and enable agents to ”read between lines”.

Valiant/Vapnik’s theory is a passive inductive statistical learning, supervised or unsupervised, from randomly sampled examples. This is shown by the arrow 1_B in Fig. 2.2. In contrast, in CL, messages are deliberated based on reflecting the mental states and utility functions.

Active learning is arrow 2: B can ask A for labeling certain examples selected by B . The example is selected to gain the most information in optimizing B ’s utility/loss function.

Algorithmic teaching [Bal08, GK95] shown as arrow 3, is a protocol complementary to active learning. Teacher A chooses the best examples to teach a student B for efficiency. A must consider what B knows and select critical examples to B , *e.g.* support vectors for classification.

LfD [ACV09, ZZC15, HLM16] is a typical learning protocol in robotics, and is an important component for commonsense acquisition. This learning method is shown by arrows 4 and 5 in Fig. 2.2, agent A teaches a task by a sequence of actions on objects. The student observes the actions and their outcomes directly, and learns the action policy from the teacher.

Causal learning is represented by arrows 1_B and 5, where an agent applies actions to change the fluents of objects and scenes, and learns the causal effects of its action in terms of changed object fluents, including appearance changes (*e.g.* painting a wall, mopping a floor), geometry changes (*e.g.* blow a balloon) and topology changes (*e.g.* cutting a fruit).

The CL can also create new learning methods or protocols which are not well-known. For example:

Perceptual causality learning. In contrast to causal learning [Pea09] where the experiment/intervention requests A to perform actions (arrow 4) and observe the effects of her actions (arrow 1_B and 5). In [FZ15], a new protocol named perceptual causality learning was proposed. Here, the student can learn causality by watching (arrows 1_B and 5) the actions of the teacher (arrow 4). Under the assumption that she is not performing magic (*i.e.* no cheating), the student will infer and mirror the actions of the teacher. This is called “perceived causality”. As shown in [FZ15], this is far more effective for learning causality and opens the door for learning causality from observations, a key aspect of human intelligence.

Utility learning is shown by arrows 4 and 6. The student infers the utility function of A by observing her decisions and choices in actions. Economics theory says that rational agents make decisions and take actions for utility maximization. By observing the actions taken by A , he can infer A ’s utility, denoted by u_{AinB} in CL. For example, like folding T-shirt demonstrated in [SHC17], by watching the teacher folding T-shirts, the student can not only learn the causality and policy π_A , but also learn a utility function u_{AinB} for aesthetics: what states of the T-shirt have a relatively high value to the teacher. B may choose to adopt a similar utility function $u_B \leftarrow u_{AinB}$. In CL, agents will update and align to a common

utility. The HRI usage of CL in Chapter 5 demonstrates another value alignment example in.

Learning by analogy (LbA) is a powerful learning mode used by humans [HHT95], but missing in current popular machine learning methods. It requests shared knowledge (w_c, θ_c) between two agents, and the capabilities of abstraction and projections to transfer knowledge across domains using an abstract graphical representation. Abstraction and projection are key intelligent capabilities in classic Raven’s IQ tests but are missing in current statistical learning. The shared mind will facilitate LbA. As the common mind grows, the two agents will be more and more synced and the student will become as capable as the teacher.

In a nutshell, all having their most suitable using scenarios, learning paradigms above illustrate the usage of various special cases of the CL framework. Nevertheless, deployment of general human-like AI usually involves much more complicated and comprehensive settings, where multiple components of the full CL framework become indispensable. In Chapter 5, we show an example scenario bearing adequate complexity to summon the full structure.

Now that we wrap up the introduction of the CL representation, we can move to the specific formalism that drives the learning and models the mind update dynamics.

2.3 Formalism of Communicative Learning

In Section 2.1 we introduced the representation of the CL framework, with which, integration of human pedagogy and ToM with machine learning becomes possible. In this section, we show the dynamics of the teacher’s and the student’s minds. This completes the CL formalism and enables us to scrutinize prior machine learning algorithms by instantiating this generic formalism.

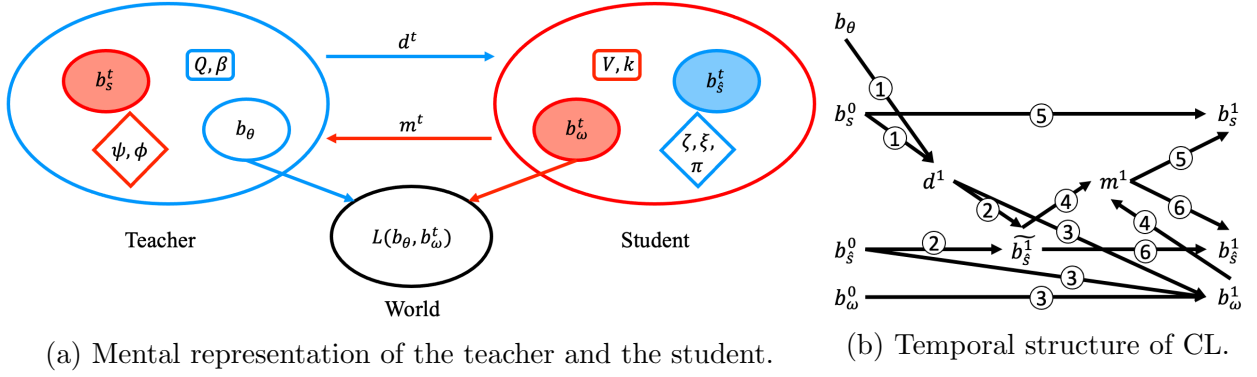


Figure 2.3: CL formalism. Fig. 2.3a shows the mental state representation of the teacher and the student. The blue bubbles are for the teacher and the red ones are for the student. Beliefs are drawn as ellipses, with the time-variant ones shaded. Diamonds hold the functions that agents used to maintain their beliefs about their partner’s behavior and knowledge. We call them the **belief-update functions**. The **value functions** for the teacher and the student are in the blue and red rectangles. Fig. 2.3b expands the transition process at the first time step. Numbers in it group the arrows, and the same number indicates operations happen in one function. Arrow 1s correspond to Eq. (2.2). Arrow 2s correspond to Eq. (2.5). Arrow 3s are for Eq. (2.4). Arrow 4s are for Eq. (2.6). Eq. (2.1) and Eq. (2.7) are represented by arrows 5s and 6s respectively. The temporal order of these belief updates is the same as the numerical order of the arrows, with 2s and 3s happen concurrently and 5s and 6s happen concurrently.

2.3.1 Overall Setting

First, let’s recall that a standard machine learning algorithm is a mapping from training data to a model space [SB14, Zhu15], where a model can be, for instance, a specific hyperplane in SVM, the location of the k centroids in K-means, or the parameters of a neural network. In CL, we generalize learning into a pedagogy process involving two agents, a teacher and a student. Each agent has its model or, in cases where uncertainty needs to be considered, a belief of the model. In Section 2.1, we differentiate an agent’s model with its policy and utility to better specify various types of learning in a situated communication scenario. Nevertheless, as for modeling the learning dynamics, we don’t need to differentiate them explicitly. That is, the *model* that we are referring to here is a representation with broader meanings than the model in Section 2.1. The only purpose of the latter is to interpret the

physical world. In Section 2.4 and Section 2.4, we shall see how concept, policy, value and utility can all be learned as the model with the same CL formalism.

Rigorously, let's denote the teacher's model space as Θ and $b_\theta \in \Delta(\theta)$ as teacher's belief of the model³. We assume b_θ stays constant across the pedagogy process, because the teacher won't receive any new information helping her to refine the model. Moreover, in most of the cases, we assume the teacher knows the true model θ^* . That is, $b_\theta(\theta)$ becomes $\mathbf{1}(\theta = \theta^*)$. Similarly, we have the student's model space Ω and the student's belief of the model as $b_\omega \in \Delta(\Omega)$. Notice that we don't assume that the teacher and the student have the same model space, namely Ω may or may not be the same as Θ . Having two separate model spaces of the agents allows CL to deal with the teacher and student pairs having distinctive world representations. For example, suppose the model is a policy network mapping from the robot's camera inputs to motion commands. Then, robots with different camera configurations are still able to teach and learn from each other. Since the student will update his belief of the model, we use b_ω^t to represent his belief at time t . The same superscript will be applied to other time-variant variables.

The goal of the CL is for the student to have an accurate enough belief of the model so that he can achieve a certain level of performance for a given task compared with the teacher. We can define this metric as minimizing a loss function $L(b_\theta, b_\omega^T)$, where T is the moment where the learning process terminates. L measures the difference between the teacher's performance and the student's performance of a task. The smaller the performance gap is, the smaller L will be. At this moment, to define the framework of CL, we don't specify L in too much detail. In Section 2.4, we'll see how different machine learning paradigms are special cases of CL and further concretize L .

³In the rest of the thesis, the subscripts of beliefs are used as labels to differentiate different beliefs and are not to be confused with parameters, which present inside parentheses, unless explicitly defined.

2.3.2 Teacher Setting

CL is a pedagogy process, in which the student's belief of the model is refined given messages from the teacher. Here we denote teacher's message at time t as $d^t \in \mathcal{D}$, chosen from her message space $\mathcal{D}$. The message selection criteria at each time depend on the student's learning state at that moment, denoted as $s^t \in S$. The learning status is a variable maintained by the teacher to keep track of the student's progress at a particular time. *e.g.*, s^t can be the validation error at the time t . When the exact learning state is not directly available to the teacher, she needs to have a belief, $b_s \in \Delta(S)$, over the learning states. Just like L , we don't further restraint the representation of $\mathcal{D}$ and S until examining specific learning paradigms.

In general, when the teacher has an accurate estimation of the student's learning state, the pedagogy can be more effective. Hence, the teacher has a transition model for the learning states. Namely, how the student will make progress after receiving a message. Assuming Markovian learning states, the transition model can be mathematically defined as, $\psi : S \times \mathcal{D} \mapsto \Delta(S)$. That is, ψ takes in the student's current learning states, teacher's message and returns the distribution of the student's new learning states. In situations like active learning [Set09], the teacher also receives messages (query data) from the student. We can denote student's message at time t as $m^t \in \mathcal{M}$. Without loss of generality, for the rest of the paper, we assume m^t comes after d^t every time t . Messages from the student can also help the teacher to estimate his current learning state. To accommodate them, the teacher has a message model for the student $\phi : S \mapsto \Delta(\mathcal{M})$, mapping learning states to a distribution over the student's messages. With ψ and ϕ , the teacher can update her b_s condition on her outgoing and incoming messages using the Bayesian filter:

$$\begin{aligned} b_s^t(s) &= p(s|d_{1:t}, m_{1:t}) \\ &\propto \phi(m^t|s) \int_{s' \in S} \psi(s|s', d^t) b_s^{t-1}(s') ds', \end{aligned} \tag{2.1}$$

where $d/m_{1:t}$ means $\{d/m^1, d/m^2, \dots, d/m^t\}$. In general, this belief cannot be calculated

exactly, especially when $|S|$ is large or infinite. However, in practice, ψ and ϕ are usually modeled as indicator functions, effectively simplifies the computation. With student's current learning state and her belief of the model b_θ , we can have a teaching policy for the teacher:

$$p(d^t | b_\theta, b_s^{t-1}) = \text{softmax}_\beta \left(Q_\psi(b_\theta, b_s^{t-1}, d^t) \right) \quad (2.2)$$

where $\text{softmax}_\beta(x) = \frac{\exp(\beta x)}{\sum_{x' \in X} \exp(\beta x')}$ is the Boltzmann rationality [BT52, RA07, BST09] and $Q_\psi(b_\theta, b_s^{t-1}, d)$ is a value function takes in teacher's belief of current learning state, teacher's model and a teacher's message. Usually it can be further expanded into integral form with simpler defined $Q_\psi(\theta, s, d)$ weighted by $b_\theta(\theta)$ and $b_s^{t-1}(s)$. As this function can depend on the teacher's transition model of the learning state, it is parameterized by ψ . Q can either be learned from data or defined by experts [GS13, GF16] and in Section 2.4 we'll see examples in both flavors.

2.3.3 Student Setting

In this section, we define the student in CL. Unlike standard machine learning algorithm mapping from data to a model, the student in CL is aware of the cooperative teacher in the pedagogy. That is, the student knows that he receives selected messages instead of random examples from the teacher. To model the teacher-aware student, we start with a teacher-unaware student, whose belief update rule given a new message d^t is simply:

$$b_\omega^t(\omega) = p(\omega | d_{1:t}) \propto b_\omega^{t-1}(\omega) \pi(d^t | \omega),$$

where $\pi : \Omega \mapsto \Delta(\mathcal{D})$ is the teaching likelihood function in the student's mind. In general cases, such estimation of the teaching function doesn't match with the exact pedagogy policy of the teacher defined in Eq. (2.2). However, as we'll see in the coming chapters, most of the time when the student has a reasonable approximation, learning is effective.

Next, for a teacher-aware student, messages are conditioned on more than just the teacher's model. Using all the information available to him, a teacher-aware student has:

$$b_{\omega}^t(\omega) \propto b_{\omega}^{t-1}(\omega)\pi(d^t|\omega, d_{1:t-1}, m_{1:t-1}). \quad (2.3)$$

The downside of using the entire history in π is the exponential growth of the history w.r.t. time. Recall that the teacher relies on two things when teaching; one is her model, θ^* , and the other is her learning state of the student, s . Thus, we define $\hat{s} \in \widehat{S}$ as student's estimation of the learning state in the teacher's mind. Then, the full history, $d_{1:t-1}, m_{1:t-1}$ can be condensed into b_s^{t-1} and Eq. (2.3) becomes

$$b_{\omega}^t(\omega) \propto b_{\omega}^{t-1}(\omega)\pi(d^t|\omega, b_s^{t-1}), \quad (2.4)$$

where $\pi : \Omega \times \Delta(\widehat{S}) \mapsto \Delta(\mathcal{D})$ is the teaching likelihood function accommodating the learning state in teacher's mind. That being said, in every time step, the student maintains two beliefs, one for the model and one for the estimation of the teacher's impression of him. Theoretically, the nested mutual reasoning between the teacher and the student can be infinitely recursive [DG09]. To avoid the intractability and complexity, in CL, we stop at a teacher-aware student and don't go deeper in the recursion. To update $b_{\hat{s}}$, the student needs two more functions to model the transition of $\hat{s}$ after two types of messages. We define $\zeta : \widehat{S} \times \mathcal{D} \mapsto \Delta(\widehat{S})$ and $\xi : \widehat{S} \times \mathcal{M} \mapsto \Delta(\widehat{S})$ as student's transition functions for $\hat{s}$, namely, how teacher's impression about him will change after she **sends** (ζ) and **receives** (ξ) a message. The student's counterpart of ϕ should be a function mapping from $\hat{s} \times \Omega$ to $\Delta(\mathcal{D})$ as his estimation of how the teacher will teach given a model when she has learning state $\hat{s}$. This can be accomplished by π , because any $\hat{s}'$ can be written as a Dirac-delta distribution, $\delta_{\hat{s}'}(\hat{s}) = \mathbf{1}(\hat{s} = \hat{s}')$. Notice that ζ, ξ and π are all approximation of teacher's mental change in the student's mind, so Ω and $\widehat{S}$ are used instead of Θ and S .

Now, we can write down the belief update function for b_s^t . Different from Eq. (2.1), we

want to have an intermediate variable after the student receives d^t and before m^t is sent out, whose purpose is to determine the best m^t to send out. Let's denote

$$\begin{aligned}
\tilde{b}_s^t(\hat{s}^t) &= p(\hat{s}^t | d_{1:t}, m_{1:t-1}) \\
&= \int_{\substack{\omega \in \Omega, \\ \hat{s}^{t-1} \in \hat{S}}} p(\hat{s}^t, \omega, \hat{s}^{t-1} | d_{1:t}, m_{1:t-1}) d\omega d\hat{s}^{t-1} \\
&= \int_{\substack{\omega \in \Omega, \\ \hat{s}^{t-1} \in \hat{S}}} p(\hat{s}^t | \hat{s}^{t-1}, \omega, d_{1:t}, m_{1:t-1}) p(\hat{s}^{t-1}, \omega | d_{1:t}, m_{1:t-1}) d\omega d\hat{s}^{t-1} \\
&\propto \int_{\substack{\omega \in \Omega, \\ \hat{s}^{t-1} \in \hat{S}}} \zeta(\hat{s}^t | \hat{s}^{t-1}, d^t) \pi(d^t | \delta_{\hat{s}^{t-1}}, \omega) p(\hat{s}^{t-1}, \omega | d_{1:t-1}, m_{1:t-1}) d\omega d\hat{s}^{t-1} \\
&\approx \int_{\substack{\omega \in \Omega, \\ \hat{s} \in \hat{S}}} \zeta(\hat{s}^t | \hat{s}^{t-1}, d^t) \pi(d^t | \delta_{\hat{s}^{t-1}}, \omega) b_s^{t-1}(\hat{s}) b_\omega^{t-1}(\omega) d\omega d\hat{s}
\end{aligned} \tag{2.5}$$

as student's belief of the teacher's estimation of his learning state after he receives d^t . The last step is an approximation due to the independent modeling of b_ω and b_s . Using $\tilde{b}_s^t$, the student comes up with the message to send to the teacher with

$$p(m^t | \tilde{b}_s^t, b_\omega^t) = \text{softmax}_k(V_\xi(\tilde{b}_s^t, b_\omega^t, m^t)), \tag{2.6}$$

where $V_\xi(\tilde{b}_s^t, b_\omega^t, m^t)$ is a value function for the student. Here V takes the belief as its argument, because usually the student needs to take attributes of the distribution, such as the entropy, into the consideration for message selection. Just like the value function, Q , to the teacher, V can either be learned from data or defined.

After the student gives his message m^t to the teacher, he finalizes his belief of $\hat{s}$ for time t using

$$b_s^t(\hat{s}) = p(\hat{s} | d_{1:t}, m_{1:t}) \propto \int_{\hat{s}' \in \hat{S}} \xi(\hat{s} | \hat{s}', m^t) \tilde{b}_s^t(\hat{s}') d\hat{s}' \tag{2.7}$$

Eq. (2.7) concludes all the belief updates in CL. As it involves many components and nested inferences between agents, we summarize the CL framework with Fig. 2.3 and specifies it in

Algorithm 1. In the next section, we demonstrate how various learning paradigms can be expressed as special cases of CL.

Algorithm 1: Communicative Learning

Input-Teacher: $\Theta, b_\theta, b_s^0, Q, \psi, \phi, \mathcal{D}$

Input-Student: $\Omega, b_\omega, b_s^0, V, \zeta, \xi, \pi, \mathcal{M}$

Input-World: T, L

Output: b_ω^T

```

1 for  $t = 1 : T$  do
2   Teacher selects  $d^t$  using Eq. (2.2)
3   Student updates  $\tilde{b}_s^t$  and  $b_\omega^t$  using Eq. (2.5) and Eq. (2.4)
4   Student selects  $m^t$  using Eq. (2.6)
5   Teacher updates  $b_s^t$  using Eq. (2.1)
6   Student updates  $b_s^t$  using Eq. (2.7)
7 end
8 return  $b_\omega^T, L(b_\theta, b_\omega^T)$ 

```

2.4 Communicative Learning Instantiation of Prior Learning Paradigms

The formalism proposed in Section 2.3 gives us a unifying lens to summarize existing learning paradigms. That is, we can instantiate the CL formalism with different learning paradigms by constructing their corresponding value and partner estimating functions. Since not all of these paradigms include the full set of recursive mutual reasoning in CL, we can classify them according to their recursive level. If a learning paradigm only has a student and data comes from a random process instead of a cooperative teacher, we call it a level-0 paradigm. For paradigms like active learning [Set09], where the teacher only passively replies to the student’s queries about random data and has no agency, we also classify it as level-0. Suppose there is a cooperative teacher who can choose her messages to the student, but the student is not aware of the existence of this teacher. Such learning paradigms are termed level-1. When the teacher is cooperative in message selection and meanwhile the student is aware of

her helpfulness, we have level-2 paradigms. We summarize the paradigms and their level in Table 2.1. It can be shown that all of these learning paradigms are essentially special cases of CL with one or a few components omitted from their modeling. Within each learning paradigm, there can be various learning algorithms and every algorithm maps to a grounding of the CL components. As the purpose of this dissertation is to propose a unified framework of learning rather than laundry listing all learning algorithms, for each paradigm, we only choose a few exemplar algorithms and specify their CL groundings. For the other algorithms, their CL groundings can be easily migrated from the exemplar algorithms belonging to the same paradigm. We summary and compare groundings of these algorithms in Table 2.2 and Table 2.3.

Table 2.1: Learning paradigms in prior work and their recursive level.

Paradigm	Teacher	Student	Level
Passive Learning	N/A	Unaware	0
Active Learning	Oracle	Unaware	0
LfD	Expert	Unaware	0
Algorithmic Teaching	Cooperative	Unaware	1
Communicative Learning	Cooperative	Aware	2

2.4.1 Level-0 Paradigms

2.4.1.1 Passive Learning

Passive learning is the most popular learning paradigm whereby the student passively receives data from the outside world and tries to figure out the underlying regularity based exclusively on his input. In passive learning, the teacher doesn't model the student. Namely, she follows the same way of providing data regardless of the student and is usually represented implicitly by a training set due to the lack of agency. Here we revisit the concept learning as an example.

Concept Learning We can represent concept learning [Mit97, SB14] using the notation defined in CL. Suppose we have two concept spaces Θ and Ω , both defined on a domain space, X , and a concept maps an instance $x \in X$ to a label $y \in Y$. Without loss of generality, we have $Y = \{0, 1\}$. There is a ground-truth concept $\theta^* \in \Theta$ only known by the teacher. That is, $b_\theta(\theta) = \delta_{\theta^*}(\theta)$ and the teacher’s message space as $D = X \times Y$. Since the teacher doesn’t model the student, b_s and Eq. (2.1) can be omitted. Eq. (2.2), as she randomly chooses an (x, y) pair that is compatible to θ^* every time, can be instantiated with $Q(\theta, d) = \mathbf{1}(\theta(d_x) = d_y)$ and $\beta \rightarrow \infty$, meaning hard-max with random tie-breaking.

Because the teacher provides nothing but positive sampled [XT07] examples from the θ^* , the student can only check the consistency of models in Ω and prune out models incompatible with his received examples. When the student doesn’t have any prior knowledge or preference over concepts in Ω , this leads to the Candidate-Elimination (CE) algorithm shown in Algorithm 2, whose convergence towards the correct concept has been proved, provided (1) there are no errors in the training examples, and (2) there is some hypothesis in Ω that correctly describes the target model, θ^* [Mit97]. This is equivalent to have

$$\pi(d|\omega) = \mathbf{1}(\omega(d_x) = d_y)/Z_{\mathcal{D}}, \quad (2.8)$$

where $Z_{\mathcal{D}}$ is the partition function to ensure π is a valid distribution over $\mathcal{D}$. As the normalization can always be done in Eq. (2.4), π doesn’t have to be normalized and $Z_{\mathcal{D}}$ can also be omitted. In the rest of the paper, we’ll use $\propto$ instead of $=$ and omit $Z_{\mathcal{D}}$. Notice the belief of the teacher’s estimated learning state can be ignored due to her random selection of data.

Algorithm 2: The CE Algorithm

```

1  $\mathcal{W} \leftarrow \Omega$ 
2 for each training example  $d^t$  do
3   | Remove from  $\mathcal{W}$  any  $\omega$  for which  $\omega(d_x^t) \neq d_y^t$ 
4 end
5 Output  $\mathcal{W}$ .
```

However, as we discussed earlier, there is usually no guarantee that Θ and Ω are the same. Thus, it might be possible that $\{\omega | \forall x \in X, \omega(x) = \theta^*(x)\} = \emptyset$. In this case, suppose the teacher sends a batch of training data all at a time. Then, $T = 1, D = (X \times Y)^N, Q(\theta, d) = \prod_{i=1}^N \mathbf{1}(\theta(d_{ix}) = d_{iy}), \beta \rightarrow \infty$, where N is the batch size of the data. The common learning mechanism for the student in this setting is the empirical error minimization (ERM) [SB14], which returns a concept achieving minimal empirical risk from Ω [SB14]. It can be represented with

$$\pi(d|\omega) \propto \mathbf{1}\left(\text{err}(d, \omega) = \min_{\omega' \in \Omega} \text{err}(d, \omega')\right) \quad (2.9)$$

where $\text{err}(d, \omega) = \frac{1}{N} \sum_{i=1}^N \mathbf{1}(\omega(d_{ix}) \neq d_{iy})$.

It is well-known that if $VCD(|\Omega|) < \infty$, ERM is a PAC-Learning algorithm when $\Theta \subseteq \Omega$ (also known as the realizability assumption) and an agnostic PAC-learning algorithm when $\Theta \not\subseteq \Omega$ [Val84, SB14].

Algorithm 3: ERM

Input: Data batch d
1 $\text{err}^* = \min_{\omega \in \Omega} \frac{1}{N} \sum_{i=1}^N \mathbf{1}(\omega(d_{ix}) \neq d_{iy})$
2 $\mathcal{W} = \{\omega | \frac{1}{N} \sum_{i=1}^N \mathbf{1}(\omega(d_{ix}) \neq d_{iy}) = \text{err}^*\}$
3 return $\mathcal{W}$

We didn't include L in our teacher/student formalization, as neither agent relies on it to select messages or update their beliefs. Only in the evaluation of the learning will L make its occurrence. In concept learning, L can be defined as

$$\mathbb{E}_{\omega \sim b_{\omega}^T} \left[\mathbb{E}_{x \sim X} [\mathbf{1}(\theta^*(x) \neq \omega(x))] \right]. \quad (2.10)$$

Here we assume there is a uniform distribution defined on the instance space X . Usually, this error can be approximated with a testing set mutually exclusive from the training data.

Maximum Likelihood Estimate for Parameter Learning Unlike concept learning, parameter learning usually involves uncountable model space, leading to intractable posterior over Ω unless special distribution families are used to model b_ω [Mur12], which could limit the expressiveness of b_ω . When the full distribution is intractable, the student can use the maximum likelihood estimate (MLE) estimator to approximate $p(d_{1:T}|\omega)$ and thus $p(\omega|d_{1:T})$ given a prior distribution. For a uniform prior, $b_\omega^T = \delta_{\omega_{mle}}$, where $\omega_{mle} = \arg \max p(d_{1:T}|\omega)$ [Mur12]. In most cases, $p(d_{1:T}|\omega)$ is high dimensional and non-convex, with intractable global optimal. Hence, usually gradient-based algorithms are leveraged to find a locally optimal MLE of $p(\omega|d_{1:T})$, a widely adopted technique in modern days' learning of DNN [RHW85, LBD89, MDK11].

We can show that learning neural network parameters with stochastic gradient descent (SGD) [NY78, Saa09] can also be represented using the CL framework. Suppose the student has a DNN $f_\omega : X \mapsto \Delta(Y)$ that takes in an instance $x \in X$ and outputs a distribution of its label. The teacher, like in concept learning, randomly chooses $d \in X \times Y$ from a data set. We can assume there is an oracle DNN parameterized by θ^* that generates all the data. Then we can define $Q(\theta, d) = \mathbf{1}(\arg \max f_\theta(d_x) = d_y), \beta \rightarrow \infty$, meaning y in the selected examples always match with the most probable output of $f_{\theta^*}(x)$. The student follows Bayesian rule to update his belief

$$\pi(d|\omega) \propto f_\omega(d_y|d_x). \quad (2.11)$$

Therefore, suppose the student holds a uniform b_ω^0 , after T rounds, $b_\omega^T(\omega) \propto p(d_{1:T}|\omega)$,

$$\omega_{mle} = \arg \max_{\omega \in \Omega} b_\omega^T = \arg \max_{\omega \in \Omega} \sum_{t=1}^T \log f_\omega(d_y^t|d_x^t). \quad (2.12)$$

As specified in Algorithm 4, SGD optimizes the same objective function as Eq. (2.12) [Kiw01, Saa09], so a student conducting SGD is equivalent to a CL student who uses Eq. (2.11) for belief update and approximates b_ω^T with $\delta_{\omega_{mle}}$.

Algorithm 4: SGD

Input: $\omega, \eta, d_{1:T}$
1 for $t = 1, \dots, T$ **do**
2 | $\omega = \omega + \eta \frac{\partial}{\partial \omega} \log f_{\omega}(d_y^t | d_x^t)$
3 end
4 return ω

2.4.1.2 Active Learning

In certain circumstances, active learning has exponential saving in label queries comparing with passive learning. Here we can define a concept learning task with the notation in the CL formalism and have the following example:

Example 1. Let $X = [0, 1]$ be the instance space. Define a threshold function $h_c : X \mapsto \{0, 1\}$ and $h_c(x) = \mathbf{1}(x \geq c)$. Suppose we have the teacher and the student's model space as $\Theta = \Omega = [0, 1]$, teacher's message space $\mathcal{D} = [0, 1] \times \{0, 1\}$, $\beta \rightarrow \infty$ and $Q(\theta, d) = \mathbf{1}(d_y = h_{\theta}(d_x))$. Namely, the teacher has a threshold θ^* and she randomly sends a number from $[0, 1]$ and whether that number exceeds the threshold to the student.

Consider learning θ^* in the setting of Example 1. Using ERM defined in [SB14] and Section 2.4.1.1, the student can end up with

$$b_{\omega}^T(\omega) = \delta_{h_{x^*}}, \text{ where } x^* = \min_{d: d_y=1} d_x \quad (2.13)$$

The number of random examples the student needs to achieve (ϵ, δ) PAC-learning is $\min \{n \in \mathbb{N}, \frac{1}{n-1} \log \frac{n+1}{\delta} \leq \epsilon\}$ [SB14].

However, thanks to the particular data structure of this problem, the student is able to do so with significantly fewer labels. Specifically, he first queries the instance at the median of the training instances. If its label is 1 (resp. 0), he moves on to query the median of the instances to its left (resp. right). As such, after at most $\log_2 n^*$ label requests, he can determine the exact decision boundary of the n^* training instances, hence their

labels [KMT93].

Next, we take two algorithms as examples to show how the CL framework can be grounded to active learning situations. In the rest of this article, we follow the most common setting of active learning, in which only querying labels of examples from an oracle is allowed [Set09].

CAL There are many active learning algorithms [Han07, Das09, DHM07, BBL09, BDL09]. Here we adopt CAL as an example to analyze its CL instantiation. We give the CAL algorithm in Algorithm 5 with CL notations.

CL can represent CAL similarly as CE plus a student query strategy. That is, the teacher still has $\mathcal{D} = X \times Y, Q(\theta, d) = \mathbf{1}(\theta(d_x) = d_y), \beta \rightarrow \infty$. The student will determine whether to query based on his models' agreement of the latest data. However, there is one nuance in the CL formalization: V doesn't take in d^t directly. Luckily, the student can use $\hat{s}$ to hold the latest unlabeled data and pass $b_{\hat{s}}$ to V . Thus, the student has $\hat{S} = X$ and

$$\zeta(\hat{s}' | \hat{s}, d^t) = \mathbf{1}(\hat{s}' = d_x^t) \quad (2.14)$$

$$\pi(d | \omega, b_{\hat{s}}) \propto \mathbf{1}(\omega(d_x) = d_y). \quad (2.15)$$

ζ updates $\hat{s}$ to the unlabeled data d_x and π is the same as CE to guarantee the consistency between data and models when updating b_{ω} using Eq. (2.4). Let $\mathcal{M} = \{1, 0\}$, indicating query or not. Then, we are ready to define

$$V(b_{\omega}, \tilde{b}_{\hat{s}}, m) = \mathbf{1}\left(\left|\left\{\int_{\hat{s} \in \hat{S}} \omega(\hat{s}) \tilde{b}_{\hat{s}}(\hat{s}) | \forall \omega, b_{\omega}(\omega) > 0\right\}\right| - 1 = m\right). \quad (2.16)$$

As $\tilde{b}_{\hat{s}}$ is a delta function, the integral simply means the output of $\omega(d_x^t)$. The cardinality, either 1 or 2, indicates the number of possible outputs from models that the student still thinks possible. If it equals 1, all the possible models agree on d_x ; otherwise, there is a disagreement, which will trigger a query. As the transition of $\hat{s}$ is not up to the student, ξ doesn't need to be defined. Notice that in this formalization, the teacher always sends

(x, y) , but y is not always used to update b_ω . Although d_y appears in Eq. (2.15), the student doesn't need it until he decides to query, because no belief update will happen if $m^t = 0$ for that round. And d_y is not used to calculate m^t as its value doesn't influence the result of Eq. (2.5) and there is no d_y in Eq. (2.16).

Algorithm 5: CAL

```

1  $\mathcal{W} \leftarrow \Omega$ 
2 Initialize training data  $D_0 = \emptyset$ 
3 for  $t = 1, \dots, T$  do
4     // Disagreement in  $\mathcal{W}$ 
5     if  $\exists \{\omega_0, \omega_1\} \subseteq \mathcal{W}$  s.t.  $\omega_i$  has 0 empirical mean of  $D_{t-1} \cup \{(d_x^t, i)\}$  for  $i = 0, 1$ 
6         then
7             Query  $d_y^t$ 
8             Remove from  $\mathcal{W}$  any  $\omega$  for which  $\omega(d_x^t) \neq d_y^t$ 
9         end
10 end
11 return  $\mathcal{W}$ 

```

Pool-Based Active Learning (PBAL) Another type of active learning involves an unlabeled (or partially labeled) dataset, and the student wants to learn a model to best fit the dataset without knowing all the labels [CAL94, MN98, SC08, ZLG03]. There are many query strategies approaching this setting, such as uncertainty sampling [LG94, CM05, SC08], query by committee [SOS92], expected model change [SCR07, SCF08] etc. Here, we use PBAL [SC08] as an example to illustrate how to ground the CL framework into this situation. Consider a similar setup as the MLE defined in Section 2.4.1.1, in which a student wants to learn a model ω to best explain a partially labeled dataset $\mathcal{L} \cup \mathcal{U}$. The PBAL takes in MLE as a subprocess and embed it in the active learning framework as defined in Algorithm 6. Notice that in Line 3 of Algorithm 6, we use the maximum entropy as our query strategy, but it can be replaced by many other criteria. Details can be found in [SC08].

To ground CL to PBAL, we need to define learning state for the teacher and the student.

Algorithm 6: PBAL

Input: Labeled dataset $\mathcal{L}$, Unlabeled dataset $\mathcal{U}$
1 for $t = 1, \dots, T$ **do**
2 $\omega = \text{MLE}(\mathcal{L})$
 // Choose the most uncertain example
3 $x^* = \arg \max_{x \in \mathcal{U}} \sum_{y' \in Y} -p(y'|x; \omega) \log p(y'|x; \omega)$
4 $\mathcal{L} = \mathcal{L} \cup \{(x^*, \text{label}(x^*))\}$
5 $\mathcal{U} = \mathcal{U} - x^*$
6 end
7 return ω

The learning state for the teacher is very straightforward. As in CL, it is the teacher who sends the message first in each round, we set $s^{t-1} = m^{t-1}$, where m^{t-1} is student's query in the previous step. Thus, the complete teacher can be defined as

$$\phi(m|s) = \mathbf{1}(m = s) \quad (2.17)$$

$$\psi(s'|s, d) = \mathbf{1}(s' = s) \quad (2.18)$$

$$Q(s, \theta, d) = \mathbf{1}(\theta(s) = d) \quad (2.19)$$

$$\mathcal{D} = Y \cup \{\Xi\}, S = X, \beta = 1,$$

where Ξ is special message sending only at the first step to signal the start of the learning process.

The student is more complicated to define than the teacher. The learning state involves the current labeled/unlabeled dataset, $\hat{S} = \mathcal{P}(X) \times \mathcal{P}(X \times Y) \times X$, where $\mathcal{P}$ denotes the power set. The first component of $\hat{s}$ is the unlabeled dataset, the second is the labeled dataset and the third is the queried data. We'll refer to them as $\hat{s}_u$, $\hat{s}_l$ and $\hat{s}_x$. The student can be defined as

$$\zeta(\hat{s}'|\hat{s}, d) \propto \mathbf{1}(\hat{s}'_u = \hat{s}_u - \hat{s}_x \wedge \hat{s}'_l = \hat{s}_l \cup \{(\hat{s}_x, d)\}) \quad (2.20)$$

$$\xi(\hat{s}'|\hat{s}, m) = \mathbf{1}(\hat{s}'_u = \hat{s}_u \wedge \hat{s}'_l = \hat{s}_l \wedge \hat{s}'_x = m) \quad (2.21)$$

$$\pi(d|\omega, b_{\hat{s}}) \propto f_{\omega}(d|\arg \max b_{\hat{s}}(\hat{s})) \quad (2.22)$$

$$V(\tilde{b}_{\hat{s}}, b_{\omega}, m) = H(f_{\omega^*}(m)) \int \mathbf{1}(m \in \hat{s}_u) \tilde{b}_{\hat{s}}(\hat{s}) d\hat{s} \quad (2.23)$$

where $\omega^* = \arg \max b_{\omega}(\omega)$

$$H(f_{\omega}(x)) = \sum_{y \in Y} -f(y|x; \omega) \log f(y|x; \omega).$$

ξ updates the learning state by moving a previously unlabeled data to the labeled set and ζ update the query data for next round. Eq. (2.20) uses $\propto$ as there are no constraints about $\hat{s}'_x$. Using indicator function in Eq. (2.21) guarantees $b_{\hat{s}}$ to be delta distributions. The $\arg \max$ in Eq. (2.22) selects the $\hat{s}$ with non-zero probability and use MLE to update b_{ω} . H in Eq. (2.23) measures the entropy of a given example in the unlabeled set given current MLE estimation of b_{ω} . The integral after H constraints m to be an element in $\mathcal{U}$.

2.4.1.3 Learning from Demonstration

The learning paradigms we discussed so far are mainly focused on supervised learning scenarios, whose signature tasks include concept learning, regression, classification, etc. Another important field of AI is for an agent to be able to act optimally in an environment. Such a problem of learning a mapping between world state and actions lies at the heart of many robotics applications. This mapping, also called a policy, enables a robot to select an action based upon its current world state. LfD provides one approach of policy learning, in which a policy is learned from examples, or demonstrations, provided by a teacher [ACV09]. In this section, we discuss IRL [NR00, ZJ12], a major branch of LfD and see how can the CL framework be grounded to IRL algorithms.

The goal of IRL is to recover the reward function of a MDP such that a set of demonstration state-action trajectories are optimal [NR00, ZMB08]. There are two tasks that IRL accomplishes. The first, reward learning, is estimating the unknown reward function as accurately as possible. It is useful in situations where the reward function is of interest

by itself, for example when constructing models of animal and human learning or modeling opponents in competitive games [AS18]. The second task is apprenticeship learning - using observations of an expert’s actions to decide one’s own behavior. And learning the reward function is generally more succinct, robust, and transferable than directly imitating the expert’s policies [RA07].

Here we follow the same assumption in [ZMB08] and assume a linear reward function. Namely, the reward function in MDP linearly maps the features of each state, $f(x) \in \mathbb{R}^m$, to a state reward value representing the agent’s utility for visiting a state x^4 . The reward value of a trajectory is simply the sum of state rewards, or, equivalently, the reward weight applied to the path feature.

$$\begin{aligned} \text{reward}(\tau) &= \theta^{*T} f(\tau) \\ \text{where } f(\tau) &= \sum_{x \in \tau} f(x) \end{aligned}$$

As proved in [NR00], the necessary and sufficient condition to recover the reward function achieving the same performance as the demonstration is to match the state feature statistics between the demonstration policy and the learner’s. In Algorithm 7, we present maximum entropy inverse reinforcement learning (MEIRL) [ZMB08], which, by employing the principle of maximum entropy [Jay57], tries to match the feature without exhibiting any additional constraints to the trajectory distribution generated by the learner’s policy. The likelihood in Line 2 is for deterministic MDPs. An adjustment term for the MDP transition model will be needed for those with stochastic transition models. Yet, under mild assumptions [ZMB08], MEIRL can also be applied to stochastic MDPs. For deterministic MDPs, the optima in Line 3 can be obtained via gradient-based optimization, which has a closed-form gradient [ZMB08].

⁴Here we use x to denote a state in MDP instead of the conventional s , as s is already occupied in the CL framework.

Algorithm 7: MEIRL

Input: Trajectories $\mathcal{D} = \{\tau^1, \dots, \tau^T\}$
1 $\tilde{f} = \frac{1}{T} \sum_{i=1}^T f(\tau^i)$
2 Let $p(\tau|\omega) \propto \exp(\omega^T f(\tau))$
3 $\omega^* = \arg \max_{\omega \in \Omega} \sum_{i=1}^T \log p(\tau^i|\omega)$
4 **return** ω^*

Modeling IRL with the CL framework is natural. The fundamental assumption of IRL is the accessibility of (approximately) optimal expert’s demonstrations [NR00, RA07, ZMB08, ZJ12], which resembles the assumption of a helpful teacher in CL. The student in IRL usually follows similar parameter learning procedures as MLE defined in Section 2.4.1.1. Suppose both the teacher and the student know the feature function f^5 . Denote the action space of the MDP as $\mathcal{A}$. Then, we can define the message space for the teacher as $\mathcal{D} = (X \times \mathcal{A})^*$, where $*$ is the Kleene star [EFT13] representing trajectories that are sequences of state-action pairs. As there is no reasoning about the student or messages from the student, the teacher doesn’t need to keep track of learning states. Thus, we can omit s in Q and have $Q(\theta, d) = \theta^T f(d)$. Here we assume that all the trajectories in $\mathcal{D}$ are consistent with the transition model of the underlying MDP and all of them have finite length. There are also IRL algorithms directly treat state-action pairs as demonstration and try to learn a reward function optimizing the Q -value [SB98] of the demonstration pairs [RA07]. The student follows the same procedure as MLE having

$$\pi(d|\omega) \propto \exp(\omega^T f(d)). \quad (2.24)$$

Then, the final belief will be $b_\omega^T(\omega) \propto \prod_{t=1}^T p(d^t|\omega)$, which can be approximated as $\delta_{\omega^*}(\omega)$ with ω^* defined in Line 3 in Algorithm 7.

Notice that, unlike algorithms in Section 2.4.2, the teacher, though helpful in the sense of

⁵In fact, there is no restriction for the teacher and the student to have different f s, here for the sake of notation simplicity we assume they have the same feature function.

acting optimally, doesn't model how the student learns and won't change her pedagogy given different students, so LfD is still under the umbrella of level-0 paradigms. In the next section, we are going to see how the teacher can accommodate the student's learning behavior into her pedagogy and execute level-1 learning paradigms, in which an improved version of IRL named cooperative inverse reinforcement learning (CIRL) will be introduced.

2.4.2 Level-1 Paradigms

In level-0 learning paradigms, the teacher doesn't bear more agency than randomly sampling consistent data, passively providing answers to queries, or showing optimal demonstrations without knowing the student. Despite their solid theoretical guarantees, both passive and active students require a rather large dataset to make up for not knowing the underlying data distribution to form a reliable belief about the target model. Suppose the teacher can intentionally customize teaching materials for the student, the learning process is usually accelerated significantly. To see that teaching can result in far fewer training examples, we consider Example 1 again. Aware of the target threshold θ^* and that the student learns with Eq. (2.13), a teacher simply gives one labeled instance *i.e.* $(\theta^*, 1)$ to the student to acquire exactly the same hypothesis. Hence, when optimally taught, the student can do as well as his teacher at little computational effort, whereas a student by himself has to process at least $O(1/\epsilon)$ instances and queries at least $O(\log 1/\epsilon)$ labels to guarantee a similar outcome with high probability [SB14].

Acknowledging the benefits of modeling the teacher, the next question is how to teach. In general, teaching can be posed as an inverse problem to learning [Zhu15] and its goal is to find the smallest set of examples based on which an intended student can output the desired belief over models. This idea was first formalized in [SM91, GK95], where they consider teaching an arbitrary consistent student *i.e.* ERM. In this section, we first go over the teaching of a consistent student, then move on to more advanced teaching of complicated students.

TD Given a concept space Θ , for any teaching set $d \subseteq X \times Y$, we can define a consistent set for d as the set of all concepts consistent with d

$$Cons(d, \Theta) = \{\theta \in \Theta | \forall (x, y) \in d, \theta(x) = y\} \quad (2.25)$$

Then, we can define the TD for any concept $\theta \in \Theta$ as the cardinality of the minimum teaching set allowing the consistent student to uniquely identify θ in Θ , the worst case of which defines the TD of the entire concept space. Strictly,

$$TD(\theta; \Theta) = \min \left\{ |d| \mid Cons(d, \Theta) = \{\theta\} \right\} \quad (2.26)$$

$$TD(\Theta) = \max_{\theta \in \Theta} TD(\theta; \Theta). \quad (2.27)$$

Armed with the definition of TD, we can instantiate CL components for teaching a consistent student. In this setting, we assume $\Theta = \Omega$ and the teacher knows the learning algorithm of the consistent student, so no message from the student is necessary. For the teacher, we only need to define

$$Q(\theta, d) = \mathbf{1}(|d| = TD(\theta, \Theta)) \quad (2.28)$$

and $\beta \rightarrow \infty$. Meanwhile, the student only needs

$$\pi(d|\omega) \propto \mathbf{1}(\omega \in Cons(d, \Omega)). \quad (2.29)$$

Here, we directly use a set of paired data as teacher's message and have $T = 1$. It is easy to use single data pairs as messages and conduct sequential pedagogy by using the learning state of the teacher to keep track of data already sent and avoid repetitive examples.

In Section 2.4.1.1, we briefly mention another complexity measure for a concept space, the Vapnik-Chervonenkis (VC)-dimension [Val84]. It has been shown with examples that

for arbitrary concept classes, neither TD nor *VCD* can dominate the other [GK95]. There are many extensions and generalizations of TD [ZLH11, LZO16]. We’ll cover some variation of TD in cooperative teaching and learning [ZLH11] in more detail when we discuss level-2 paradigms in Section 2.4.3.

IMT To produce teaching materials suited to her audience, the teacher has to possess some knowledge about how the student learns. When dealing with problems involving uncountable model spaces, such as parameter learning, a consistent student may no longer be intractable. In those cases, gradient-based optimization learning algorithms, such as SGD studied in Section 2.4.1.1 come in handy. We now see an example of teaching such type of students.

Consider the supervised learning setting, where we have instance space X and Y . We assume a linear model mapping X to Y , *i.e.* $y = h(x; \theta^*) = h(\langle \theta^*, x \rangle)$. h can be the identity function for regression or the soft max function for classification. Here, we don’t assume the teacher and the student share the same representation of a same instance; *e.g.*, they can be robots with different camera configurations⁶. We use x and $\tilde{x}$ to denote the instance representation of the teacher and the student. While x can be different from $\tilde{x}$, we assume the agents always have the same y . The student, as mentioned in the previous paragraph, follows a gradient-based optimization given a data point $d = (x, y)$

$$\omega^t = \omega^{t-1} - \eta \frac{\partial l(h(\tilde{x}; \omega^{t-1}), y)}{\partial \omega^{t-1}}, \quad (2.30)$$

where l is the loss function shared by the teacher and the student and η is the learning rate. When l is the negative log-likelihood (NLL), Eq. (2.30) is indeed the update function in Line 2 in Algorithm 4 and the student becomes the MLE student in Section 2.4.1.1. Here we assume l to be a convex function.

Although the student is the same as the SGD learner, the teacher can execute more

⁶This is one of the important reasons that the teacher cannot directly pass her model to the student.

efficient teaching than a random data stream by employing the student's learning pattern. Suppose they have the same data representation and the teacher is aware of the student's current parameter θ^t . Then, the optimal data point she should pick at time t is

$$\begin{aligned} d^t &= \arg \min_{d \in \mathcal{D}} \left\| \theta^{t-1} - \eta \frac{\partial l(h(d_x; \theta^{t-1}), d_y)}{\partial \theta^{t-1}} - \theta^* \right\|_2^2 \\ &= \arg \max_{d \in \mathcal{D}} \left(-\eta^2 \left\| \frac{\partial l(h(d_x; \theta^{t-1}), d_y)}{\partial \theta^{t-1}} \right\|_2^2 + 2\eta \left\langle \theta^{t-1} - \theta^*, \frac{\partial l(h(d_x; \theta^{t-1}), d_y)}{\partial \theta^{t-1}} \right\rangle \right). \end{aligned} \quad (2.31)$$

The expression after $\arg \max$ in Eq. (2.31) is defined as the teaching volume $TV_{\theta^*}(d|\theta^t)$, which represents the student's progress in this iteration. It is a trade-off between the difficulty and the usefulness of an example [LDH17]. Nonetheless, the teacher doesn't know student's current parameter, ω^{t-1} . Therefore, she needs to find an approximation of the teaching volume. Leveraging the convexity of the loss function l and the linearity of h , she has

$$\begin{aligned} \left\langle \theta^{t-1} - \theta^*, \frac{\partial l(h(d_x; \theta^{t-1}), d_y)}{\partial \theta^{t-1}} \right\rangle &\geq l(h(\langle \theta^{t-1}, d_x \rangle), d_y) - l(h(\langle \theta^*, d_x \rangle), d_y), \\ d^t &= \arg \max_{d \in \mathcal{D}} -\eta^2 \left\| \frac{\partial l(h(\alpha_{d_x}), d_y)}{\partial \theta^{t-1}} \right\|_2^2 + 2\eta \left(l(h(\alpha_{d_x}), d_y) - l(h(\langle \theta^*, d_x \rangle), d_y) \right), \end{aligned} \quad (2.32)$$

where α_x is the inner product of data x and student's current model. We denote Eq. (2.32) as $\widehat{TV}_{\theta^*}(d|\alpha)$. Although the teacher doesn't need to know the exact parameter of the student using $\widehat{TV}$, she still needs to know the student's inner products of the data. The advantage of the inner product is this approximation can be directly applied even when they use different data representations [LDH17]. The complete algorithm can be found in Algorithm 8. Notice that IMT proposes a greedy teaching algorithm. It has been proved in [LZZ19] that for an omniscient teacher, teaching greedily is suboptimal. Yet, the alternative teaching schema originating from optimal control cannot be directly applied to more practical teaching scenarios, where the teacher doesn't know the student's current parameter and different representations are used between them.

Now we show that IMT can be represented with the CL framework. As the student is

Algorithm 8: IMT

Input: $\omega^0, \theta^*, \mathcal{D}, l$
1 for $t = 1, \dots, T$ **do**
2 Student reports inner products to the teacher, $\alpha_x = \langle \tilde{x}, \omega^{t-1} \rangle$
3 $d^t = \arg \max_{d \in \mathcal{D}} \widehat{TV}_{\theta^*}(d|\alpha)$
4 $\omega^t = \omega^{t-1} - \eta \frac{\partial l(h(\tilde{d}_x; \omega^{t-1}), d_y)}{\partial \omega^{t-1}}$
5 end

essentially the same as the MLE, he will have

$$\pi(d|\omega) \propto \exp(-l(h(d_x; \omega), d_y)), \quad (2.33)$$

which is the same as Eq. (2.11) when l is NLL. Yet, compared with MLE, in IMT the student needs to provide inner product as feedback to the teacher. Hence, we also need to define $\mathcal{M} = \mathbb{R}^{|\mathcal{D}|}, k \rightarrow \infty$ and

$$V(b_\omega, m) = \prod_{i=1}^{|\mathcal{D}|} \mathbf{1}(m_i = \langle \arg \max b_\omega(\omega), \tilde{\mathcal{D}}_{ix} \rangle), \quad (2.34)$$

where $\mathcal{D}_i$ is the i -th data point in the dataset. One nuance of this setting is the size of $\mathcal{D}$ need to be finite for the student to give feedback. In practice, the teacher uses a mini-batch as $D^t \subset \mathcal{D}$ in every time step, whose size can be fairly small.

The teacher will use her learning state to keep track of student's inner products and greedily selects the teaching message. Thus, we have $S = \mathbb{R}^{|\mathcal{D}|}$ and

$$\phi(m|s) = \mathbf{1}(m = s) \quad (2.35)$$

$$Q(s, \theta, d) = \widehat{TV}_\theta(d|s), \beta \rightarrow \infty. \quad (2.36)$$

CIRL As theoretically and empirically illustrated in [Zhu13, Zhu15, LDH17, LDL18], when the teacher intentionally customizes data for the student, the learning can be significantly

sped up. Besides these works in the supervised learning setting, [HLM16, HRA16, HLC18, BN19] also showed benefits of introducing a pedagogical teacher into IRL. We covered the standard IRL in Section 2.4.1.3, in which the student tries to recover the reward function of the MDP given the optimal behavior trajectories from the teacher. Nevertheless, the trajectory optimal for task completion may not necessarily also be optimal for pedagogy [HLM16, HRA16, HLC18]. Take the example from [HLM16]: Consider cooking a chicken. To cook one for dinner, you would do it in the most efficient way possible while avoiding contaminating other foods. But, what if you wanted to teach a completely naïve observer how to prepare poultry? In that case, you might take pains to emphasize certain aspects of the process. For example, by ensuring the observer sees you wash your hands thoroughly after handling the uncooked chicken, you signal that it is undesirable (and perhaps even dangerous) for other ingredients to come in contact with raw meat. [HRA16] also proved that in a human-robot teaming setting, the best-response robot policy triggered by optimal human demonstrations is no longer optimal to collaborate with the demonstrations.

CIRL proposed in [HRA16] provided an improved teaching mechanism for the student learning with MEIRL. As the student is trying to match the feature statistics of trajectories motivated by his reward function with the demonstrations, the teacher can select trajectories that trade-off between the sum of rewards $\theta^{*T} f(\tau)$ and the feature dissimilarity $\|f(\tau) - f_{\theta^*}\|_2$, where f_{θ^*} is the expectation of feature statistics induced by the ground truth reward function θ^* . Rigorously, the CL framework definition has $T = 1$ and the teacher uses

$$Q(\theta, d) = \theta^T f(d) - \eta \|f(d) - f_{\theta}\|^2, \beta \rightarrow \infty, \quad (2.37)$$

where η is the coefficient to control the relative importance between the optimality and the informativeness of the trajectory. The student’s CL components are the same as those defined in Section 2.4.1.3.

By adding the additional term in Eq. (2.37), the teacher witnesses significant improve-

Algorithm 9: CIRL

Input: $\mathcal{D} = \{\tau^1, \dots, \tau^{|\mathcal{D}|}\}$
1 $d = \arg \max_{d \in \mathcal{D}} \theta^{*T} f(d) - \eta \|f(d) - f_{\theta^*}\|^2$
2 **return** MEIRL($\{d\}$)

ment in terms of the convergence of the reward parameters and the human-robot collaboration performance [HRA16]. Nevertheless, there are two limitations of CIRL. The first is that learning is usually a sequential instead of a one-shot process. In [HHP20], a sequential teacher was proposed, who also takes in student’s nonverbal feedback into consideration to better choose the demonstration. The second limitation, suffered by both CIRL and [HHP20], is the heuristic design of the message selection criteria, which is task-specific and requires expertise to come up with. Next, we’ll go over approaches that learn how to choose optimal messages without expert knowledge.

Learning to teach (L2T) So far, we’ve introduced several level-1 learning paradigms, in which the teacher models how the student learns and personalizes pedagogy materials to facilitate the learning. Yet, the teaching algorithms in previous works are all based on heuristic designs, such as greedy choice of optimal data [LDH17, LDL18] and a trade-off between optimality and informativeness [HRA16]. Despite the performance gain brought by these approaches, heuristic teaching cannot be easily generalized to students other than their target learners. Hence, designing a teacher who can adapt to different students becomes a booming research problem. The goal is to have a teacher that can learn how to teach a certain student while interacting with him. Ideally, such a teacher can also transfer her knowledge of teaching to other students about other tasks. *E.g.* a teacher model learns how to teach from the interaction with an multilayer perceptron (MLP) student by selecting data from the MNIST [LBB98] dataset, and then the learned teacher can be directly applied to teach a convolutional neural network (CNN) student model on the CIFAR-10 [KH09] dataset [FTQ18].

The intrinsic complexity of the student’s model space is large, usually uncountable, *e.g.* parameters of a neural network. And the learning status of the student needs to accommodate the belief over that model space, or at least a lower dimension approximation of it, like MLE and MAP. Hence, finding a closed-form optimal teaching schema is often intractable, especially for sequential students, whose complexity is exponential in terms of the number of iterations [LZZ19]. One tentative approach is to model the teaching problem as an MDP, whose environment is the student’s learning state and actions are teacher’s pedagogical actions, such as data, loss functions, actions suggestions and so on [WM17, FTQ18, WTX18, OKL19]. Then define the performance of the student, *e.g.* evaluation loss as the reward and solve the teaching problem as finding the optimal policy of the MDP. Since the simulation of the student’s future performance can be difficult, it is common to use model-free reinforcement learning (RL) [WD92, Wil92, LHP16] to solve for a good teaching policy in these problems.

Here we take L2T [FTQ18] as an example to illustrate how the CL framework accommodates this school of works. Moreover, we can see the REINFORCE [Wil92] algorithm used for training the teacher model in L2T can also be adopted to train the teacher and student in CL given proper defined rewards and features. In L2T, the learning state in the teacher’s mind has three components, represented as three feature vectors: data features, student model features, and the combination of the two. **Data features** contain information for the teaching material, such as its label category, (for texts) the length of sentence, linguistic features for text segments [TFL16], or (for images) gradients histogram features. **Student model features** include the signals reflecting how well the current neural network is trained. Several simple features are passed mini-batch number (*i.e.*, iteration), the average historical training loss and historical validation accuracy. **Combination features** target to represent how important the arrived training data is for the current learner. In the classification tasks in [FTQ18], three features were utilized: 1) the predicted probabilities of each class; 2) the loss value on that data, which appears frequently in self-paced learning [KPK10]; 3) the

margin value. Given the learning state and her correct model, the teacher selects a teaching action, a subset of the data batch in our case, to teach the student. To be more specific, suppose the entire training set is G , every time a data batch D^t is sampled randomly from G . The teacher's action is $a^t = \{0, 1\}^{|D^t|}$, with $a_i^t = 1$ means the i -th data instance in D^t is selected. Then, the student updates his model with the selected data and is tested with the evaluation data. His performance is used as the reward to train the teacher with CL. The algorithm is sketched in Algorithm 10. In the execution phase, the last two lines can be skipped.

Algorithm 10: L2T (Training)

Input: Training Dataset G

```

1 for  $t = 1, \dots, T$  do
2   Teacher gets a random mini-batch  $D^t \subset G$ 
3   Teacher queries student about  $D^t$  for combination features
4   Teacher selects data  $d^t \subset D^t$  for the student to learn given the state feature and her  $Q$ 
5   Student updates model and return reward  $r$ 
6   Teacher updates  $Q$  with  $r$  using REINFORCE
7 end
```

As the teacher needs to query the student to have the combination features, she needs to interact with the student twice before he can update his b_ω . In order to ground the CL framework, which only allows one message from the teacher every time b_ω changes, to L2T, we need to let the teacher send both the selected data for current step and the unselected data for next step so that the student uses the selected data to update his model and gives his feedback about unselected data with the new model via his message to the teacher in this round. We can define the learning state space $S = \mathcal{P}(G)^2 \times Y^B \times \mathbb{R}^2$ and message space $\mathcal{D} = \mathcal{P}(G)^2$, where B is the batch size for $D^t, \forall t$ and $\mathcal{P}(G)$ means the power set of G . That is, every learning state has four components: current D^t , D^{t+1} for next step, student's label about D^t and student model features including number of iterations and averaged training loss, while every message has the data selected from D^t and the complete D^{t+1} . Let's denote

the four components of a learning state as $s_{1:4}$ and the two components of a message as $d_{1:2}$. The message space of the student only needs to include the feedbacks for D^{t+1} , *i.e.* $\mathcal{M} = Y^B$. With the definitions above, we have

$$\psi(s'|s, d) \propto \mathbf{1}(s'_1 = s_2) \quad (2.38)$$

$$\phi(m|s) = \mathbf{1}(s_3 = m \wedge s_4 = l(m, s_1)) \quad (2.39)$$

$$Q(s, \theta, d) = Q((s_1, s_3, s_4), \theta, d), \quad (2.40)$$

where Eq. (2.38) means after the teacher sends out the message, her learning state replaces current dataset with next dataset, Eq. (2.39) means the student's feedback is updated for next step and the student model feature has been updated with the loss function l . Eq. (2.40) is learned with REINFORCE, so it doesn't have a closed form. (s_1, s_3, s_4) correspond to the three features mentioned in [FTQ18]. The student need to use $\hat{s}$ to capture teacher's query so $\hat{S} = \mathcal{P}(G)$ and

$$\pi(d|\omega) \propto f_\omega(d_1) \quad (2.41)$$

$$\zeta(\hat{s}'|\hat{s}, d) = \mathbf{1}(\hat{s}' = d_2) \quad (2.42)$$

$$V(b_\omega, b_{\hat{s}}, m) = \mathbf{1}\left(m = f\left(\arg \max b_{\hat{s}}(\hat{s}); \arg \max b_\omega(\omega)\right)\right), \quad (2.43)$$

where Eq. (2.41) is the likelihood of the selected data, same as the MLE student in Section 2.4.1.1, and Eq. (2.42) deterministically updates $\hat{s}$ with data query from the teacher. The first $\arg \max$ in Eq. (2.43) picks the only non-zero $\hat{s}$ while the second one is the MLE approximation of b_ω , making m the output for the teacher's query.

L2T provides a generic schema for a teacher to learn how to teach a specific type of students. Such schema can also be applied in CL to jointly train the teacher and the student as a team [YFS19]. When the agents are trained jointly, they will adapt to each other. Namely, the student will be aware of the teacher and learn faster by taking advantage of the

teacher’s pedagogy. Integrating such teacher-awareness with learning algorithms produces level-2 learning paradigms, in which the teacher and the student reason about each other mutually and sometimes recursively, as we’ll see in the following section.

2.4.3 Level-2 Paradigms

In human pedagogy, teachers and students can interact adaptively to maximize communication efficiency. The teacher adjusts her teaching method for different students, and the student, after getting familiar with the teacher’s instruction mechanism, can infer the teacher’s intention to learn faster. Usually, the mutual reasoning stops at this level without moving any deeper [FCA18], especially in human interactions, where the cognitive burden outweighs the communication efficiency [DVV15] improvements. However, this mutual inference loop can continue infinitely until convergence [ZLH11, YYW18, WPS19] for machine learners. In this section, we first discuss a few level-2 learning paradigms designed for machine learning settings and see how they can be represented by the CL framework. Then, we see some works involving a human as either the teacher or the student.

Balbach TD In Section 2.4.2, we reviewed TD and the classic machine teacher for a consistent student. What if the student is aware of the teacher’s teaching mechanism? Will that benefit the learning? The answer is yes. Suppose the student knows $TD(\omega, \Omega), \forall \omega$, then sometimes concepts could also be eliminated by the mere number of examples presented to the student. For instance, assume the student knows that all but one concept ω have a teaching set of size one and that the teacher will teach using teaching sets. After having seen 2 examples, no matter what they are, the student could eliminate all concepts but ω [Bal08, ZLH11]. If the teacher knew that the student eliminates by consistency and by sample size then she could consequently reduce the size of some teaching sets (*e.g.*, here, if $TD(\omega, \Omega) \leq 3$, a new “teaching set” for ω could be built consisting of only 2 examples). Moreover, if the student knew that the teacher uses such reduced “teaching sets” then he

could adapt his assumption on the size of the samples to be expected for each concept, which could, in turn, result in a further reduction of the “teaching sets” by the teacher and so on [ZLH11]. And this new recursively defined TD, after converge, is referred to as the Balbach TD [Bal08]. Formally, this can be defined as the following. Recall the definition of consistency and TD in Eq. (2.25) and Eq. (2.26). Let $BT D^0(\theta, \Theta) = TD(\theta, \Theta)$, then, for $k \in \mathbb{N}$:

$$Cons_B^k(d, \Theta) = \{\theta \in Cons(d, \Theta) \mid BT D^k(\theta, \Theta) \geq |d|\} \quad (2.44)$$

$$BT D^{k+1}(\theta, \Theta) = \min \{|d| \mid Cons_B^k(d, \Theta) = \{\theta\}\} \quad (2.45)$$

Denote κ as the minimum k s.t. $\forall \theta \in \Theta, BT D^{k+1}(\theta, \Theta) = BT D^k(\theta, \Theta)$, then we can define $BT D$ in the same way as TD:

$$BT D(\theta, \Theta) = BT D^\kappa(\theta, \Theta) \quad (2.46)$$

$$BT D(\Theta) = \max_{\theta \in \Theta} BT D(\theta, \Theta). \quad (2.47)$$

The crux of understanding $BT D$ is in Eq. (2.44). To be considered as a consistent dataset, one not only needs to satisfy the standard consistent relationship, *i.e.* $\theta(x) = y$, but also cannot be dominated by any alternative datasets. The intuition is that if there is another dataset strictly smaller than d that already pinpoints θ , using d to imply θ is obviously suboptimal, making d inconsistent with θ . In linguistics, this phenomenon can be explained with pragmatics, which studies how context can contribute to language meanings [Gri75, LSL00, Mey01, FG12, Jag12]. One good example is the scalar implicature [Gri75]. Suppose there are three boys in a class and teacher A told teacher B that “some boys went to the party”, then teacher B will legitimately infer that one or two of the boys went to the party. Although according to the literal meaning, it could be that all three of the boys went to the party, teacher B is less likely to interpret the utterance that way, because teacher A , a cooperative speaker, would have said “all the boys went to the party” in that case to

prevent ambiguity. As for Balbach TD, the literal meaning of a dataset is if it satisfies $\theta(x) = y, \forall (x, y) \in d$ and the pragmatics meaning is if there are no d' with a smaller size that can exactly indicate θ .

To represent the teacher and the student using *BT**D* with the CL framework is very straightforward, because it has the same representation as the teacher and the student using standard TD, only changing the *TD* to *BT**D* and *Cons* to $Cons_B^\kappa$ in Eq. (2.28) and Eq. (2.29) respectively. The derivation of *BT**D* can be done unilaterally without communication.

It has been showed that Balbach TD lower bounds TD [ZLH11]. Besides Balbach TD there are other cooperative TDs that can lower bounds TD, such as subset TD [ZLH11] and recursive TD [ZLH11, DFS14, YFS19]. In this article, we only show how the CL framework can be used to represent Balbach TD. For the other TDs, finding the CL grounding is similar. Their definitions follow the same recursive pattern as Balbach TD, only leveraging different approaches to further refine and eliminate suboptimal datasets.

Cooperative Bayesian inference (CBI) The definition of previous TDs are based on the setting of concept learning with binary consistency. That is, for $\theta \in \Theta$, an instance x is either in or out, corresponding to $\theta(x) = 1$ or 0. In many scenarios, however, the consistency is soft, and for every pair of concepts and data, there is compatibility. In the previous setting, the student prunes out inconsistent concepts completely and assigns equal weights to the rest. Now, he can never fully eliminate a concept, but only assigns lower weights to concepts that are unlikely given the data⁷. In the basic setting, where the data are i.i.d., this becomes the well-studied Bayesian inference [Mur12, Chapt. 1, 5] discussed in Section 2.4.1.1 and the student learns with a fixed likelihood function, Eq. (2.11). To go beyond the basic setting, the teacher and the student need to infer each other. The inference process, as we already see in Balbach TD, can be recursive and continue until convergence [SGG14, WPS19, WWP20, WWS20]. The CBI model put forth by [SGG14, WPS19] studied the recursive reasoning

⁷Meanwhile, the conventional TD can be substituted by the expected TD [YYW18].

between a pedagogical teacher and student. It has been shown that the mutual reasoning can be modeled as an iterative process, performing which can be achieved by conducting Sinkhorn-Scaling [SK67] on the compatibility matrix. And that operation is guaranteed to converge to a fixed point given mild assumptions [WPS19, WWP20].

Here we show an sequential variation of CBI [WWS20] and present its CL groundings. In CBI, the teacher and the student have the same concept space, *i.e.* $\Theta = \Omega$. Suppose the teacher only sends the message once, then

$$p_T(d|\theta) \propto p_0(d)\hat{p}_L(\theta|d), p_L(\omega|d) \propto b_\omega^0(\omega)\hat{p}_T(d|\omega), \quad (2.48)$$

where $p_0(d)$ is the prior over the message space, for instance, shorter messages are preferred over longer ones. When all the messages are homogeneous like data points, the teacher usually uses a uniform $p_0(d)$. $\hat{p}$ indicates estimation of other's action probability. To have a reasonable estimation of others, the teacher and the student need to have some common knowledge. In CBI, the only knowledge an agent has is the relation between concepts and D , in terms of a positive matrix whose normalization can be treated as the likelihood matrix in a Bayesian sense. The teacher has a default⁸ data selection probability represented as a matrix $\mathbf{T} \in \mathbb{R}^{|D| \times |\Theta|}$, while the student has a default inference matrix $\mathbf{S} \in \mathbb{R}^{|D| \times |\Theta|}$. The common knowledge requires $\mathbf{T} = \mathbf{S}$, meaning both of the agents have the same understanding of the literal meaning of the messages. Then, $\hat{p}$ are the same for the agents when they start to reason about each other recursively. $\hat{p}_L(\theta|d)$ is the normalization of the row, $\mathbf{T}_{<d,>}$ and $\hat{p}_T(d|\omega)$ is the normalization of the column, $\mathbf{S}_{<,\omega>}$. Eq. (2.48) can be solved using Sinkhorn scaling, where $(r; c)$ -Sinkhorn scaling of a matrix M is simply the iterated alternation of row normalization of M with respect to r and column normalization of M with respect to c [YYW18, WWS20]. The limit of such iterations exists if the sums of elements in r and c are the same [Sch89]. After Sinkhorn scaling, the teacher will sample $d \sim \text{Normalize}(\mathbf{T}_{<\theta^*,>})$

⁸These default matrices are up to change after they start mutual reasoning.

and the student will have posterior $b_\omega^T = \mathbf{S}_{<d,>}$. To generalize this teaching process from single round to sequential teaching, the student only needs to update his prior with the last step's posterior. This doesn't introduce misalignment between the teacher and the student, because, in every step, they share the same matrix (up to normalization) from Sinkhorn scaling and the student's posterior is completely predictable to the teacher. In Algorithm 11 we formalize sequential CBI, where $M^{<|D|b_\omega>}$ means $(\mathbf{1}_{|D|}, |D|b_\omega)$ -Sinkhorn scaling. Namely, the row summations of the matrix are all equal to 1, and column summations of the matrix equal to $|D|b_\omega$. The coefficient $|D|$ is to guarantee the convergence.

Algorithm 11: Sequential CBI

```

/* Teacher Part                                     */
Input:  $\mathbf{T} = M$ ,  $b_\theta = \delta_{\theta^*}$ , Uniform  $b_\theta^0$ 
1 for  $t = 1, \dots, T$  do
2    $d^t \sim \text{Normalize}(\mathbf{T}_{<\theta^*>}^{<|D|b_\theta^{t-1}>})$ 
3    $b_\theta^t = \mathbf{T}_{<d^t,>}^{<|D|b_\theta^{t-1}>}$ 
4 end
5 return  $\{d^1, \dots, d^T\}$ 
/* Student Part                                     */
Input:  $\mathbf{S} = M$ , Uniform  $b_\omega^0$ ,  $\{d^1, \dots, d^T\}$ 
6 for  $t = 1, \dots, T$  do
7    $b_\omega^t = \mathbf{S}_{<d^t,>}^{<|D|b_\omega^{t-1}>}$ 
8 end
9 return  $b_\omega^T$ 

```

To represent sequential CBI with the CL framework, the teacher needs to keep track of the student's current belief, so $S = \Delta(\Theta)$. Thus, we have the teacher following

$$Q(s, \theta, d) = \log \left(\text{Normalize}(\mathbf{T}_{<\theta>}^{<|D|s>})_d \right) \quad (2.49)$$

$$\psi(s'|s, d) = \mathbf{1}(s' = \mathbf{T}_{<d,>}^{<|D|s>}). \quad (2.50)$$

On the student's side, the learning state estimation $\hat{S} = \Delta(\Omega)$ and $b_{\hat{s}}$ is always a delta distribution as there is no uncertainty between the teacher and the student. Let $\hat{s}^* =$

$\arg \max b_{\hat{s}}(\hat{s})$ be the learning state estimation with probability 1, then

$$\pi(d|\omega, b_{\hat{s}}) \propto \mathbf{S}_{<d, \omega>}^{<|D|\hat{s}^*>} \quad (2.51)$$

$$\zeta(\hat{s}'|\hat{s}, d) = \mathbf{1}(\hat{s}' = \mathbf{S}_{<d, >}^{<|D|\hat{s}>}). \quad (2.52)$$

Because the format of Eq. (2.50) and Eq. (2.52) is the same and $\mathbf{T} = \mathbf{S}$, $\forall t, \hat{s}^t = s^t$. In fact, the assumption of $\mathbf{T} = \mathbf{S}$ can be relaxed to that $\mathbf{T}$ is a scaled matrix of $\mathbf{S}$, *i.e.*, there exist invertible diagonal matrices E_1 and E_2 such that $\mathbf{T} = E_1 \mathbf{S} E_2$. Given that condition, the Sinkhorn scaling will still converge to the same fixed point.

Although CBI's advantage over regular Bayesian inference has been proved theoretically and empirically [SGG14, YYW18, WPS19, WWP20, WWS20], the Sinkhorn scaling procedure is hard to scale up, for instance, from concept learning to parameter learning, whose model space is uncountably infinite. Even if there is a tractable Sinkhorn convergence point, when the computational power of one agent is too limited to pursue the recursive calculation, *e.g.* one of the agents is a human, CBI cannot be directly applied. In the following part, we'll review approaches that try to mitigate such restrictions and still integrate pedagogy into learning.

Teacher-aware active learning (TAAL) When the model space is infinite or the computational resources restrict the level of recursion, finding the fixed point of a cooperative pedagogy can be intractable. Thus, to scale up, the teacher and the student need to have a fixed level of recursions. So far, as far as we know, in settings where the belief over the models doesn't have a closed-form representation, like matrices, no prior works went beyond level-2 recursion, namely having a pedagogical teacher and a teacher-aware student [WWP20]. This level of recursion also corresponds to most people's cognitive ability. It has been shown that when interacting with more convoluted models, the performance of the human-machine group witnessed very limited improvements [YDF08, DVV14, DVV15, WDV18]. Now, we

introduce a teacher-aware active learning algorithm [PCD19] as an example of level-2 CL framework with more generic belief update functions, which aims to learn the reward parameter in a linear multi-armed bandit setting [ACF02].

The basic setting of TAAL is Bayesian Bernoulli bandits. At each iteration t , the student chooses an arm $m^t \in \{1, \dots, K\}$ and receives a stochastic reward $y^t \in \{0, 1\}$, depending on the chosen arm. Every arm has a feature vector x_k and the reward is sampled from $\text{Bernoulli}(\sigma(\theta^{*T} x_{m^t}))$, where $\sigma(\cdot)$ stands for the logistic sigmoid function. The features for all the arms are common knowledge between the teacher and the student, only θ^* needs to be learned. Suppose there isn't a helpful teacher, then, at time t given $h^t = \{(m^1, y^1), \dots, (m^t, y^t)\}$, the student will form a posterior

$$p(\omega|h^t) \propto p(\omega) \prod_{i=1}^t \text{Bernoulli}(y^i|\sigma(\omega^T x_{m^i})). \quad (2.53)$$

Here we have $\Theta = \Omega$ and they will be used interchangeably to indicate the teacher's and the student's end. Given the posterior at time t , the student needs to choose the next arm to maximize his expected accumulated reward $R^t = \mathbb{E}[\sum_{i=t}^T y^i]$. There are many bandit arm selection strategies to select the next arm to query about. Here, we use a stochastic strategy, Thompson sampling [Tho33]. Namely,

$$p(m^{t+1} = k|p(\omega|h^t)) = \int \mathbf{1}(\arg \max_{i \in \{1, \dots, K\}} \omega^T x_i = k) p(\omega|h^t) d\omega. \quad (2.54)$$

In TAAL, there is a pedagogical teacher, who selects $y^t, \forall t$ intentionally instead of sampling it from the Bernoulli distribution. Finding the optimal way of teaching can be formulated as solving an MDP, where the transition dynamics follow the dynamics of the naive student described in Eq. (2.53) and the responses y^t are the actions. There are multiple ways of defining the reward for the teacher, in TAAL, it is defined the same as the student's reward. Namely, the teacher wants to teach the student so that the accumulated reward

from the pulled arms is maximized. Details can be found in [PCD19, sec. 3.2].

The essence of level-2 paradigms is the teacher-aware student, who can interpret the teacher’s responses, acknowledge the teaching intent and grasp more than just the literal meanings of teacher’s messages. At each iteration t , the student assumes that the teacher chooses the action y^t with probability proportional to the action being optimal in value

$$p(y^t|h^t, \theta) \propto \frac{\exp(\beta Q(h^t, \theta, y^t))}{\sum_{y \in \{0,1\}} \exp(\beta Q(h^t, \theta, y))}, \quad (2.55)$$

where h^t is the message history between the teacher and the student up to time t , Q is Q -function [SB98] of the teacher MDP and β is a teacher optimality parameter (or inverse temperature; for $\beta = 0$, the distribution of y^t is uniform; for $\beta \rightarrow \infty$, the action with the highest value is chosen deterministically). As the calculation of Q for every θ is intractable, in [PCD19] a one-step planning ahead approximation was used, *i.e.* use the expected reward of from the next arm as the Q . Also, in order to use conjugate likelihood and prior to simplify the student’s belief update, in every iteration, the student’s posterior will be approximated as a multivariate Gaussian with Laplace approximations [GCS13, sec. 4.1]. In Algorithm 12 we show the formalized algorithm of TAAL from [PCD19].

Next, we show how can the CL framework represent TAAL. Since this is an active learning setting, where the teacher responds to the student’s query. Thus, in CL, the teacher’s learning state needs to have two components, the first is the query of the student to be answered; the second is the estimated belief of a naive student learning with Eq. (2.53). We refer them as s_1 and s_2 and have $S = \{1, \dots, K\} \times \Delta(\Theta)$. Notice that the teacher doesn’t need h anymore, as it has been accommodated into s_2 . The message space $\mathcal{D} = \{0, 1\}$. Recall that the arm features, X are common knowledge. The teacher follows

$$\phi(m|s) \propto \mathbf{1}(s_1 = m) \quad (2.56)$$

$$\psi(s'|s, d) \propto \mathbf{1}(s'_2 = \tau(s_2, d)) \quad (2.57)$$

Algorithm 12: TAAL

```

/* Teacher-Aware Student */
1 for  $t = 1, \dots, T$  do
2   | Sample  $m^t$  with  $b_\omega^{t-1}$  using Eq. (2.54)
3   | Receive  $y^t$  from the teacher and update history  $h^t$ 
4   | Update  $b_\omega^t$  with Eq. (2.55) as the likelihood function
5 end
/* One-step Planning Teacher */
Input: history  $h$ , query arm  $x$ 
6 for  $y \in \{0, 1\}$  do
7   | Calculate  $p(\theta|h, y)$  with Eq. (2.53)
8   | Sampling next arm pulling distribution  $p_{hy}$  with  $p(\theta|h, y)$  and Eq. (2.54)
9   | Calculate expected reward as  $Q_y = \sum_{i=1}^K \theta^{*T} x_i p_{hy}(i)$ 
10 end
11 return  $y \sim \frac{\exp(\beta Q_y)}{\exp(\beta Q_0) + \exp(\beta Q_1)}$ 

```

$$Q_\psi(s, \theta, d) = \sum_{i=1}^K \theta^T x_i TS(\psi(s, d)_2)_i, \quad (2.58)$$

where $\tau : \Delta(\Theta) \times \{1, \dots, K\} \times \{0, 1\} \mapsto \Delta(\Theta)$ is the belief update function using Bayesian filter with the likelihood being Bernoulli($d|\sigma(\theta^T x_{s_1})$), $TS : \Delta(\Theta) \mapsto \Delta(\{1, \dots, K\})$ is the Thompson sampling in Eq. (2.54) and the subscript i means the probability assigned to the i -th arm. Eq. (2.57) can be designed as a deterministic process because the belief transition for a naive student strictly follows Bayesian rule and is deterministic.

As for the student, he also has two components as his learning state estimation of the teacher, the first is the query waiting for reward; the second is the naive student's belief. *i.e.*, $\hat{S} = \{1, \dots, K\} \times \Delta(\Omega)$. Notice that the second component of $\hat{s}$ is the same as its counterparts in s , because the student also knows the belief update process of a naive learner. If the student's idea of a naive learner and the teacher's idea of a naive learner are misaligned, that estimation will be different. Student's message space is $\mathcal{M} = \{1, \dots, K\}$. Finally, the

Table 2.2: Instantiation of CL components for different learning paradigms: Teacher

Name	Q	s	β	ψ	ϕ
Level 0					
CE	$\mathbf{1}(\theta(d_x) = d_y)$	N/A	∞	N/A	N/A
ERM	$\prod_{i=1}^N \mathbf{1}(\theta(d_{ix}) = d_{iy})$	N/A	∞	N/A	N/A
MLE	$\mathbf{1}(\arg \max_{\theta} f_{\theta}(d_x) = d_y)$	N/A	∞	N/A	N/A
CAL	$\mathbf{1}(\theta(d_x) = d_y)$	N/A	∞	N/A	N/A
PBAL	$\mathbf{1}(\theta(s) = d)$	query instance	∞	$\mathbf{1}(s' = s)$	$\mathbf{1}(m = s)$
LfD	$\theta^T f(d)$	N/A	1	N/A	N/A
Level 1					
TD	$\mathbf{1}(d = TD(\theta; \Theta))$	N/A	∞	N/A	N/A
IMT	$\widehat{TV}_{\theta}(d s)$	$\tilde{X}\omega$	∞	N/A	$\mathbf{1}(m = s)$
CIRL	$\theta^T f(d) - \ f(d) - f_{\theta}\ ^2$	N/A	∞	N/A	N/A
L2T	learned with REINFORCE	3 features + next step data	1	$\mathbf{1}(s'_1 = s_2)$	$\mathbf{1}(s_3 = m \wedge s_4 = l(m, s_1))$
Level 2					
Bal TD	$\mathbf{1}(d = BTD(\theta; \Theta))$	N/A	∞	N/A	N/A
CBI	$\log \left(\text{Normalize}(\mathbf{T}_{<\theta>}^{< D s>})_d \right)$	b_{θ}	1	$\mathbf{1}(s' = \mathbf{T}_{<d,>}^{< D s>})$	N/A
TAAL	$\sum_{i=1}^K \theta^T x_i TS(\psi(s, d))_i$	belief of naive learner	1	$\mathbf{1}(s'_2 = \tau(s, d))$	$\mathbf{1}(s_1 = m)$

student can be defined as

$$\zeta(\hat{s}'|\hat{s}, d) \propto \mathbf{1}(\hat{s}'_2 = \tau(\hat{s}, d)) \quad (2.59)$$

$$\xi(\hat{s}'|\hat{s}, m) \propto \mathbf{1}(\hat{s}'_1 = m) \quad (2.60)$$

$$V(b_{\omega}, m) = \log(TS(b_{\omega})_m) \quad (2.61)$$

$$\pi(d|\omega, b_{\hat{s}}) \propto \exp \left(\sum_{i=1}^K \omega^T TS(\tau(\hat{s}_2^*, d))_i \right) \quad (2.62)$$

where Eq. (2.61) just simulate the Thompson sampling in Eq. (2.54), the subscript m means the probability assigned to the m -th arm and $\hat{s}^*$ refers to the teacher's learning state with non-zero $b_{\hat{s}}(\hat{s})$.

Discussion of TAAL wraps up our survey of prior learning algorithms. As a level-2

Table 2.3: Instantiation of CL components for different learning paradigms: Student

Name	V	$\hat{s}$	k	ζ	ξ	π
			Level 0			
CE	N/A	N/A	N/A	N/A	N/A	$\mathbf{1}(\omega(d_x) = d_y)$
ERM	N/A	N/A	N/A	N/A	N/A	$\mathbf{1}(err(d, \omega) = err^*)$
MLE	N/A	N/A	N/A	N/A	N/A	$f_\omega(d_y d_x)$
CAL	$\mathbf{1}\left(\left \int_{\hat{s} \in \hat{S}} \omega(\hat{s}) \tilde{b}_{\hat{s}}(\hat{s})\right \forall \omega, b_\omega(\omega) > 0\right\} - 1 = m\right)$	d_x	∞	$\mathbf{1}(\hat{s} = d_x)$	N/A	$\mathbf{1}(\omega(d_x) = d_y) \tilde{b}_{\hat{s}}(d_x)$
PBAL	$\frac{H(f_{\omega^*}(m))}{\int \mathbf{1}(m \in \hat{s}_u) \tilde{b}_{\hat{s}}(\hat{s}) d\hat{s}}$	$\mathcal{U} \times \mathcal{L} \times X$	∞	$\mathbf{1}(\hat{s}'_u = \hat{s}_u - \hat{s}_x \wedge \hat{s}'_l = \hat{s}_l \cup \{\hat{s}_x, d\})$	$\mathbf{1}(\hat{s}'_u = \hat{s}_u \wedge \hat{s}'_l = \hat{s}_l \wedge \hat{s}'_x = m)$	$f_\omega(d \arg \max b_{\hat{s}}(\hat{s}))$
LfD	N/A	N/A	N/A	N/A	N/A	$\exp(\omega^T f(d))$
			Level 1			
TD	N/A	N/A	N/A	N/A	N/A	$\mathbf{1}(\omega \in Cons(d, \Omega))$
IMT	$\prod_{i=1}^{ \mathcal{D} } \mathbf{1}(m_i = \langle \arg \max b_\omega(\omega), \tilde{\mathcal{D}}_{ix} \rangle)$	N/A	∞	N/A	N/A	$\exp(-l(h(d_x; \omega), d_y))$
CIRL	N/A	N/A	N/A	N/A	N/A	$\exp(\omega^T f(d))$
L2T	$\mathbf{1}\left(m = f\left(\arg \max b_{\hat{s}}(\hat{s}); \arg \max b_\omega(\omega)\right)\right)$	query data	∞	$\mathbf{1}(\hat{s}' = d_2)$	N/A	$f(d_1; \omega)$
			Level 2			
Bal TD	N/A	N/A	N/A	N/A	N/A	$\mathbf{1}(\omega \in Cons_B^\kappa(d, \Omega))$
CBI	N/A	b_ω	N/A	$\mathbf{1}\left(\hat{s}' = \mathbf{S}_{<d, >}^{< D \hat{s}>}\right)$	N/A	$\mathbf{S}_{<d, \omega>}^{< D \hat{s}^*>}$
TAAL	$\log(TS(b_\omega)_m)$	(m, b_ω)	1	$\mathbf{1}(\hat{s}'_2 = \tau(\hat{s}, d))$	$\mathbf{1}(\hat{s}'_1 = m)$	$\exp\left(\sum_{i=1}^K \omega^T TS(\tau(\hat{s}_i^*, d))_i\right)$

learning algorithm, it illustrates the advantage of ToM modeling and recursive reasoning. Nonetheless, compared with the task in Chapter 5, TAAL is still primitive in the sense of game setting, and the effectiveness of interacting with the human was only shown with a proof-of-concept toy example.

One follow-up question about the CL formalism is how to acquire the belief-update and value functions. Usually, there are two approaches. The first is to use predefined heuristic functions, specifically designed or engineered for particular tasks. Most learning algorithms in this section and the HRI example in Chapter 5 fall into this category. The second is to learn these functions as an emerged norm of communication. Algorithm 10 [FTQ18] and the referential game example in Chapter 4 belong to this category. That is, after learning and teaching multiple models (every model learning follows Algorithm 1), the teacher and the student start to know each other better and form a tacit norm of communication. The convergence of model and the convergence of communication norms form different hierarchies in learning. In Section 6.2.1, we will come back to this problem in more detail in the context of the halting problem of learning. In the next chapter, we'll scrutinize a learning algorithm accommodating both a helpful teacher and a teacher-aware student. We'll find its performance advantage over non-CL algorithms.

CHAPTER 3

Theoretical Guarantee of Communicative Learning

In this chapter, we examine the theoretical guarantee of CL using machine parameter learning as an example. We compare two types of learning paradigms. The first involves a cooperative machine teacher and a naive learner (level-1 following the definition in Section 2.4); the second involves the same teacher and a teacher-aware learner. The learner estimates the teacher’s data selection process with distribution and corrects his likelihood function with this estimation to accommodate the teacher’s intention. Maximizing the new likelihood enables the learner to utilize both explicit information from the selected data and implicit information suggested by the pedagogical context. We showed that the teacher and student group following the CL formalism can achieve both empirical and theoretical better performance. This work not only justified the practicability of CL, but also provide insights into generic human-machine interactions, such as online education, and machine-machine communications, such as ad-hoc teamwork [BRK17].

3.1 Machine Parameter Teaching

Finding the optimal way of teaching parameters has been a challenging problem because of the continuous state space and long horizon planning. One common framework is machine teaching [Zhu13, Zhu15]. Here, we adopt an iterative variation of machine teaching [LDH17], consisting of three entities: the learner, the teacher and the world. **The world** is defined as a parameter ω^* , fixed and known only by the teacher. Given a model $y = h(x; \omega)$ parameterized by ω , the world is defined as $\omega^* = \arg \min_{\omega \in \Omega} \mathbb{E}_{(x,y) \sim \mathcal{P}(x,y)} [l(h(x; \omega), y)]$, where

$\mathcal{P}(x, y)$ is the data distribution in standard machine learning. Here, l and h can vary across tasks, *e.g.* l can be squared loss for regression, cross-entropy for classification, and negative log-likelihood for IRL [BMS11, ML15]. In this chapter, we assume l to be a convex function and $h(x; \omega) = h(\langle x, \omega \rangle)$. h can be an identity function for linear regression and softmax function for classification. Thus, we can omit h in the loss function and write $l(\langle x, \omega \rangle, y)$ for short. l and h are common knowledge of the teacher and the learner.

Representation: The teacher represents an example as (x, y) while the student represents the same example as $(\tilde{x}, \tilde{y})$ (typically $y = \tilde{y}$ and we use y when there is no ambiguity). The representation $x \in \mathcal{X}$ and $\tilde{x} \in \tilde{\mathcal{X}}$ can be different but deterministically related by an unknown mapping, $\tilde{x} = \mathcal{G}(x)$. Suppose the teacher and the learner use model $h(\langle x, \omega \rangle)$ and $h(\langle \tilde{x}, \nu \rangle)$ respectively, then ω^* and ν^* are very likely in different spaces too. This is a common scenario when the teacher and the learner are a human and a robot, or two robots from different factories. As the representation of examples can be complex, such as features extracted by deep neural networks [MKS13, RDS15, HZR16], using a linear model h doesn't impinge the expressive power of the overall model. In the rest of the chapter, we use ω for the teacher's parameter and ν for the learner's if they are from different spaces. Otherwise, we use ω for both of them. We use x to refer to an example and its teacher representation. We use $\tilde{x}$ for its learner representation. Also, we don't specify the choice of $\mathcal{G}$. Our only assumption about the teacher and the learner's representation will be discussed in Theorem 1.

Teacher: In general, the teacher can only communicate with the learner via examples. This restriction doesn't impinge the generality of the machine teaching framework, as the format of the data can be generic, such as demonstration used in the IRL [ZMB08, BMS11, Vro14, ML15]. In this chapter, data are provided iteratively. We use x^t to denote the example used in the t -th iteration. The teacher aims to provide examples iteratively so that the student parameter ν converges to its optimum ν^* as fast as possible. Since the teacher doesn't know ν^t or ν^* , we let the learner provide some feedback to her in each iteration so

that she can track the pedagogy progress (details in Section 3.2.1).

Learner: The learner has an initial parameter ν^0 before learning. At time t , he has learning rate η_t . The learning algorithms for teacher-unaware learners are often simple. For iterative gradient-based optimization, the learner usually uses stochastic gradient descent [LDH17, LDL18, FTQ18, WTX18]. Suppose the learner receives (x^t, y^t) from the teacher, his iterative update is:

$$\nu^t = \nu^{t-1} - \eta_t \frac{\partial l(\langle \tilde{x}^t, \nu^{t-1} \rangle, y^t)}{\partial \nu^{t-1}}. \quad (3.1)$$

Mutual knowledge: We limit the mutual knowledge between the teacher and the learner, otherwise, the mutual reasoning between the two can theoretically become an infinite recursion. In this chapter, we consider a teacher who assumes a naive learner using Eq. (3.1) to update his model. Meanwhile, the learner knows the teacher selects data deliberately instead of randomly (detailed in the next section). If we define a naive learner as having level-0 recursive reasoning, then the teacher and the teacher-aware learner have level-1 and level-2 recursive reasoning respectively. This level of recursion is very close to human cognitive capability [DVV15, WVV17] and was also adopted by [PCD19].

To summarize, the loss function l , the model h , and the naive learner update function are common knowledge to the teacher and the learner. ω^* and the teaching mechanism are only known by the teacher, while ν^t and the learning mechanism, i.e. how to update ν^t given data, are only known by the learner. He knows the teacher intentionally selects helpful data according to her estimation of himself, and the teacher assumes that the learner learns following Eq. (3.1). For our teacher-aware learner, this assumption is inaccurate, but we'll show how the proposed learner can learn much faster than a naive learner.

3.2 Teacher-Aware Learning

3.2.1 Cooperative Teacher

We first define the teacher whom the learner should be aware of. Let's consider a teacher using the same feature representation as the learner and knowing his parameter in each iteration. [LDH17] termed this kind of teacher as the omniscient teacher, who, in the t -th iteration, greedily chooses example from a data batch $D^t = \{(x_i, y_i) \sim \mathcal{P}(x, y)\}$:

$$\begin{aligned} (x^t, y^t) &= \arg \min_{(x, y) \in D^t} \left\| \omega^{t-1} - \eta_t \frac{\partial l(\langle x, \omega^{t-1} \rangle, y)}{\partial \omega^{t-1}} - \omega^* \right\|_2^2 \\ &= \arg \max_{(x, y) \in D^t} \left(-\eta_t^2 \left\| \frac{\partial l(\langle x, \omega^{t-1} \rangle, y)}{\partial \omega^{t-1}} \right\|_2^2 + 2\eta_t \left\langle \omega^{t-1} - \omega^*, \frac{\partial l(\langle x, \omega^{t-1} \rangle, y)}{\partial \omega^{t-1}} \right\rangle \right). \end{aligned} \quad (3.2)$$

The expression after $\arg \max$ in Eq. (3.2) is defined as the teaching volume $TV_{\omega^*}(x, y|\omega^t)$, which represents the learner's progress in this iteration. It is a trade-off between the difficulty and the usefulness of an example [LDH17, see, sec. 4.1]. Notice that the teacher has no control over D^t , which is sampled from the data distribution $\mathcal{P}$ or from a large dataset. She only selects the best example from D^t . Given D^t with a mild batch size, *e.g.* 20, the $\arg \max$ in Eq. (3.2) can be exactly calculated.

[LZZ19] has proved that, for an omniscient teacher, teaching greedily is suboptimal. Yet, their findings cannot be directly applied to more practical teaching scenarios. Thus, we keep leveraging the greedy heuristic to model our cooperative teacher and generalize it to a non-omniscient teacher who doesn't fully know the learner in every iteration.

Suppose the teacher neither knows the learner's ν^{t-1} nor ν^* and they use different feature representations of the data. To teach cooperatively, she has to imitate the learner's model in her own feature space and use ω^* to guide the teaching. This can be done approximately if, in every round, the learner gives the inner products of ν^{t-1} and the data to the teacher as feedback. Given the convexity of the loss function l , we have: $\left\langle \omega^{t-1} - \right.$

$\omega^*, \frac{\partial l(\langle x, \omega^{t-1} \rangle, y)}{\omega^{t-1}} \rangle \geq l(\langle x, \omega^{t-1} \rangle, y) - l(\langle x, \omega^* \rangle, y)$. Now, Eq. (3.2) can be approximated by inner products between the model parameter and the data [LDH17]. Denote the learner's feedback as $\alpha_x = \langle \tilde{x}, \nu^{t-1} \rangle, \forall (x, y) \in D^t, \tilde{x} = \mathcal{G}(x)$, then the teacher will teach as following:

$$\arg \max_{(x, y) \in D^t} \left(-\eta_t^2 \left\| \frac{\partial l(\alpha_x, y)}{\partial \alpha_x} x \right\|_2^2 + 2\eta_t (l(\alpha_x, y) - l(\langle x, \omega^* \rangle, y)) \right). \quad (3.3)$$

It has been shown that cooperative teachers using Eq. (3.3) can substantially speed up the learning of a standard SGD learner [LDH17]. Nonetheless, only having a cooperative teacher doesn't provide us the most effective interaction between the two agents, as the learner doesn't exploit the fact that the data come from a helpful teacher [SGG14]. In the next section, we introduce a teacher-aware learner.

3.2.2 Teacher-Aware Learner

Now, we propose a learner who integrates the teacher's pedagogy into his parameter updating process. Suppose we have a distribution $p(x, y | \nu^* = \nu) \propto \exp(-l(\langle \tilde{x}, \nu \rangle, y))$, denoted as $p_\nu(x, y)$. Then, applying gradient descent to $l(\langle \tilde{x}, \nu \rangle, y)$ is equivalent to maximizing $\log p_\nu(x, y)$ w.r.t. ν . Hence, a learner updating parameters with Eq. (3.1) can be considered as performing maximum likelihood estimation (MLE) when the data are randomly sampled from $\mathcal{P}(x, y)$.

Nonetheless, in the machine teaching framework, data are no longer randomly sampled from $\mathcal{P}(x, y)$. A teacher-aware learner should rectify his updating rule by considering the teacher's helpfulness. Given the dataset D^t at the time t , the learner can postulate that the teacher is more likely to choose the example she thinks helpful following $p(x, y | \nu^* = \nu, \nu^{t-1}, D^t)$, denoted as $q_\nu(x, y | \nu^{t-1}, D^t)$ for short:

$$q_\nu(x, y | \nu^{t-1}, D^t) = \frac{\exp(\beta_t \widehat{TV}_\nu(\tilde{x}, y | \nu^{t-1}))}{\int_{(x', y') \in D^t} \exp(\beta_t \widehat{TV}_\nu(\tilde{x}', y' | \nu^{t-1}))}, \beta_t \geq 0, \quad (3.4)$$

$$\text{with } \widehat{TV}_\nu(\tilde{x}, y|\nu^{t-1}) = -\eta_t^2 \left\| \frac{\partial l(\langle \tilde{x}, \nu^{t-1} \rangle, y)}{\partial \nu^{t-1}} \right\|_2^2 + 2\eta_t \left(l(\langle \tilde{x}, \nu^{t-1} \rangle, y) - l(\langle \tilde{x}, \nu \rangle, y) \right). \quad (3.5)$$

The Boltzmann noisy rationality model [BT14] indicates that the teacher samples data according to the soft-max of their approximation of the teaching volumes, calculated w.r.t. her ν^* and the inner product feedback from the learner. Although, in practice, this estimation is usually different from the teacher's actual example selection distribution, which is a hard-max, corresponding to $\beta_t \rightarrow \infty$, maximizing it w.r.t. ν can still improve the learning.

The learner now wants to learn a ν , which not only makes y^t more likely to be the correct label of x^t , but also (x^t, y^t) more likely to be chosen from D^t . Intuitively, given all data in D^t are coherent with the true distribution, the teacher gives $(x^t; y^t)$ but not other examples. With what ν can the probability of this selection be maximized? So, at every time t , the student maximizes $p_\nu(x^t, y^t)$ and $q_\nu(x^t, y^t|\nu^{t-1}, D^t)$ w.r.t. ν simultaneously. We can still use gradient descent. Omit y when there is no confusion. Denote $g_x(\gamma) = \frac{\partial l(\langle \tilde{x}, \gamma \rangle, y)}{\partial \gamma}$, then we have:

$$\begin{aligned} \frac{\partial \log q_\nu(\tilde{x}^t, y^t|\nu^{t-1}, D)}{\partial \nu} &= \frac{\partial}{\partial \nu} \left(\beta_t \widehat{TV}_\nu(x^t, y^t|\nu^{t-1}) - \log \int_{(x,y) \in D} \exp(\beta_t \widehat{TV}_\nu(x, y|\nu^{t-1})) \right) \\ &= -2\beta_t \eta_t g_{x^t}(\nu) + \frac{2\beta_t \eta_t \int_{(x,y) \sim D} g_x(\nu) \exp(-\beta_t \widehat{TV}_\nu(x, y|\nu^{t-1}))}{\int_{(x,y) \in D} \exp(-\beta_t \widehat{TV}_\nu(x, y|\nu^{t-1}))} \\ &= -2\beta_t \eta_t (g_{x^t}(\nu) - \mathbb{E}_{x \sim q_\nu}[g_x(\nu)]), \\ \nu^t &= \nu^{t-1} - \eta_t \left(\frac{\partial l(\langle \tilde{x}^t, \nu^{t-1} \rangle, y)}{\partial \nu^{t-1}} - \frac{\partial \log q_\nu(\tilde{x}^t, y^t)}{\partial \nu} \Big|_{\nu=\nu^{t-1}} \right) \\ &= \nu^{t-1} - \eta_t g_{x^t}(\nu^{t-1}) - 2\beta_t \eta_t^2 (g_{x^t}(\nu^{t-1}) - \mathbb{E}_{x \sim q_{\nu^{t-1}}}[g_x(\nu^{t-1})]). \quad (3.6) \end{aligned}$$

Notice that ν^{t-1} is a constant in $q_\nu(x^t, y^t|\nu^{t-1}, D^t)$ and the optimization is w.r.t. ν , which is treated as ν^* in the calculation. The gradient is computed at $\nu = \nu^{t-1}$. This is equivalent to maximizing a new log-likelihood function $\log(p_\nu(x^t, y^t)q_\nu(x^t, y^t|\nu^{t-1}, D^t))$, an approximation of the log probability that (x^t, y^t) being sampled in D^t and then being selected by the teacher

given $\nu^* = \nu$. The product is an approximation of $p((x^t, y^t), D^t | \nu^{t-1}, \nu^* = \nu)$ because the sampling of data in D^t except (x^t, y^t) is regarded as deterministic. When $\beta_t = 0$, i.e. the learner thinks the teacher uniformly samples data, and Eq. (3.6) becomes regular SGD.

An interpretation of the benefits brought by Eq. (3.6) is that the learner not only learns from the literal meaning of the example selected by the teacher (**the second term**), but also compares that example with “also-rans” in D^t (**the third term**), forming a context incorporating additional information. This is a prevailing phenomenon in human communication, as messages often convey both literal meanings and pragmatic (contributed by the context) meanings [VBP13, SGF13, YFS19]. In other words, we can acquire not only explicit information from what others said but also implicit information from what others didn’t say. When the message space is finite and known, exact computations of the implicit information become tractable. Therefore, in scenarios like human-robot interactions, where robots usually provide predefined user interfaces with a fixed choice of instructions, our algorithm can easily conduct counterfactual reasoning by comparing the user’s selected instruction with the others and deliver faster learning than only using the selected one. In Section 3.3.2, our experiment with humans as the teacher illustrates such an advantage.

One nuance is that if we use ν^{t-1} as the ν in Eq. (3.4), the second term of the teaching volume will be 0. Thus, to better approximate ν^* , in practice, we plug in $\nu^{t-1} - \eta_t \frac{\partial l(\langle \tilde{x}, \nu^{t-1} \rangle, y)}{\partial \nu^{t-1}}$. That is, the learner first updates ν^{t-1} just like a naive learner. Then he calculates the gradient of $\log q_\nu$ w.r.t. the new ν and does an additional gradient descent corresponding to the last term in Eq. (3.6). Also, in supervised learning settings, the teacher needs to provide labels of the whole dataset for the learner to calculate the expectation. This is a mild requirement easy to be satisfied in practice. In the iterative process, D^t is a mini-batch sampled from a large dataset with a small batch size, say 20 examples. Thus, $\mathbb{E}_{x \sim q_\nu}[g_x(\nu)]$ can be calculated exactly, and, compared with the standard mini-batch gradient descent, the only additional information needed from the teacher is the index of (x_t, y_t) . In fact, we can further relax this condition by letting the learner estimate $\mathbb{E}_{x \sim q_\nu}[g_x(\nu)]$ with only a subset $\hat{D}^t \subseteq D^t$. In our

experiments, we show that with only one random unchosen example provided, i.e. $|\widehat{D}^t| = 1$, the teacher-aware learner outperforms the naive learner. See Algorithm 13 for details.

Algorithm 13: Iterative Teacher-Aware Learning

Input: Data distribution $\mathcal{D}$, teacher parameter ω^* , learning rate η_t , teacher estimation scale β_t

Result: $\nu^{(T)}$

- 1 Randomly initialize student model $\nu^{(0)} \sim \text{Uniform}(N)$; Set $t = 1$ and T as the maximum iteration number;
 - 2 **while** $t < T$ **do**
 - 3 Teacher gets data batch $D^t \sim \mathcal{D}$
 - 4 Learner reports $\alpha_x = \langle \nu^{(t-1)}, \tilde{x} \rangle$ for all $x \in D^t$ to the teacher
 - 5 Teacher selects data for time t :

$$(x^t, y^t) = \arg \max_{(x,y) \in D^t} \left(-\eta_t^2 \left\| \frac{\partial l(\alpha_x, y)}{\partial \alpha_x} x \right\|^2 + 2\eta_t \left(l(\alpha_x, y) - l(\langle \omega^*, x \rangle, y) \right) \right)$$
 - 6 Learner uses the selected data $(\tilde{x}^t, y^t)$ and D^t to calculate
 - 7 $\hat{\nu}^{(t)} = \nu^{(t-1)} - \eta_t \frac{\partial l(\langle \tilde{x}^t, \nu^{(t-1)} \rangle, y^t)}{\partial \nu^{(t-1)}}$
 - 8 $\nu^{(t)} = \hat{\nu}^{(t)} - 2\beta_t \eta_t^2 \left(\frac{\partial l(\langle \tilde{x}^t, \hat{\nu}^{(t)} \rangle, y^t)}{\partial \hat{\nu}^{(t)}} - \mathbb{E}_{(\tilde{x}, y) \sim q_{\hat{\nu}^{(t)}}(\tilde{x}, y | \nu^{(t-1)}, D^t)} \left[\frac{\partial l(\langle \tilde{x}, \hat{\nu}^{(t)} \rangle, y)}{\partial \hat{\nu}^{(t)}} \right] \right)$
 - 9 where $q_{\hat{\nu}^{(t)}}(\tilde{x}, y | \nu^{(t-1)}, D^t) = \frac{\exp \left(\beta_t \widehat{TV}_{\hat{\nu}^{(t)}}(\tilde{x}, y | \nu^{(t-1)}, D^t) \right)}{\sum_{(x', y') \in D^t} \exp \left(\beta_t \widehat{TV}_{\hat{\nu}^{(t)}}(\tilde{x}', y' | \nu^{(t-1)}, D^t) \right)}$
 - 10 with $\widehat{TV}_{\hat{\nu}^{(t)}}(\tilde{x}, y | \nu^{(t-1)}, D^t)$ defined in Eq. (3.5).
 - 11 $t = t + 1$
 - 12 **end**
-

We now prove the teacher-aware learner can always perform better than a naive learner given proper conditions.

Theorem 1 (Local Improvement). *Denote $\tilde{\nu}^t = \nu^{t-1} - \eta_t g_{x^t}(\nu^{t-1})$. For a specific loss function l , given the same learning status ν^{t-1} and a teacher following Eq. (3.3), suppose x^t satisfies that x^t itself maximizes $\widehat{TV}_{\tilde{\nu}^t}(x, y | \nu^{t-1})$. Denote $\hat{x}^t$ as the $x \in D^t$ which achieves the second largest $\widehat{TV}_{\tilde{\nu}^t}(x, y | \nu^{t-1})$. Suppose that $\|g_{x^t}(\tilde{\nu}^t)\|_2 \leq G$ for any $x \in D^t$. If $\langle \tilde{\nu}^t - \nu^*, g_{x^t}(\tilde{\nu}^t) - g_{\hat{x}^t}(\tilde{\nu}^t) \rangle > 0$, then with large enough β_t , the teacher-aware learner using Eq. (3.6) is guaranteed to make no smaller progress than a naive learner using Eq. (3.1).*

One intuition for the assumption is that the best example selected by the teacher does

bring more benefits to the learner than the other examples do. Suppose we have $\hat{\nu}^t = \tilde{\nu}^t - \eta_t g_{\hat{x}^t}(\tilde{\nu}^t)$, then moving from $\hat{\nu}^t$ to ν^t follows $\eta_t(g_{\hat{x}^t}(\tilde{\nu}^t) - g_{x^t}(\tilde{\nu}^t))$. The assumption $\langle \tilde{\nu}^t - \nu^*, g_{x^t}(\tilde{\nu}^t) - g_{\hat{x}^t}(\tilde{\nu}^t) \rangle = \langle \nu^* - \tilde{\nu}^t, g_{\hat{x}^t}(\tilde{\nu}^t) - g_{x^t}(\tilde{\nu}^t) \rangle > 0$ simply suggests that updating with x^t gives the learner an advantage over updating with $\hat{x}^t$. The advantage points to ν^* (the two vectors $\nu^* - \tilde{\nu}^t$ and $g_{\hat{x}^t}(\tilde{\nu}^t) - g_{x^t}(\tilde{\nu}^t)$ form an acute angle).

For simplicity, let g_x denote $g_x(\tilde{\nu}^t)$ in this proof. First, we provide an intuition for the assumption. Suppose we have $\hat{\nu}^t = \tilde{\nu}^t - \eta_t g_{\hat{x}^t}$, then moving from $\hat{\nu}^t$ to ν^t follows $\eta_t(g_{\hat{x}^t} - g_{x^t})$. The assumption $\langle \tilde{\nu}^t - \nu^*, g_{x^t} - g_{\hat{x}^t} \rangle = \langle \nu^* - \tilde{\nu}^t, g_{\hat{x}^t} - g_{x^t} \rangle > 0$ simply suggests that updating with x^t gives the learner an advantage over updating with $\hat{x}^t$. The advantage points to ν^* (the two vectors $\nu^* - \tilde{\nu}^t$ and $g_{\hat{x}^t} - g_{x^t}$ form an acute angle).

Next, we start the proof. We need the following lemma:

Lemma 2. Denote $\hat{x}^t$ as the x which achieves the second largest $\widehat{TV}_{\nu^*}(\tilde{x}, y | \nu^{t-1})$. Suppose that $\langle \tilde{\nu}^t - \nu^*, g_{x^t} - g_{\hat{x}^t} \rangle > 0$, then there exists $\alpha > 0$ such that with large enough β_t , we have

$$\|\beta_t \eta_t^2(g_{x^t} - \mathbb{E}_{x \sim q_{\tilde{\nu}^t}}[g_x])\|_2 \leq \alpha \|\tilde{\nu}^t - \nu^*\|_2 \quad (3.7)$$

and

$$\langle \tilde{\nu}^t - \nu^*, g_{x^t} - \mathbb{E}_{x \sim q_{\tilde{\nu}^t}}[g_x] \rangle \geq \alpha \|\tilde{\nu}^t - \nu^*\|_2 \|g_{x^t} - \mathbb{E}_{x \sim q_{\tilde{\nu}^t}}[g_x]\|_2. \quad (3.8)$$

Proof of Lemma 2. We set $\alpha = \langle \tilde{\nu}^t - \nu^*, g_{x^t} - g_{\hat{x}^t} \rangle / (2\|\tilde{\nu}^t - \nu^*\|_2 \|g_{x^t} - g_{\hat{x}^t}\|_2) > 0$.

First we show that $\|\beta_t \eta_t^2(g_{x^t} - \mathbb{E}_{x \sim q_{\tilde{\nu}^t}}[g_x])\|_2 \leq \alpha \|\tilde{\nu}^t - \nu^*\|_2$. For simplicity we denote $s(x) = \widehat{TV}_{\tilde{\nu}^t}(\tilde{x}, y | \nu^{t-1})$. Then by assumption on the selection of x^t we have $x^t = \arg \max_{x \in D^t} s(x)$ and

$$\begin{aligned} & (g_{x^t} - \mathbb{E}_{x \sim q_{\tilde{\nu}^t}}[g_x]) \int_{x' \in D} \exp(\beta_t s(x')) \\ &= \left(g_{x^t} - \frac{\int_{x' \in D} \exp(\beta_t s(x')) g_{x'} }{\int_{x' \in D} \exp(\beta_t s(x'))} \right) \int_{x' \in D} \exp(\beta_t s(x')) \end{aligned}$$

$$\begin{aligned}
&= \exp(\beta_t s(x^t)) [g_{x^t} - g_{x^t}] + \exp(\beta_t s(\hat{x}^t)) [g_{x^t} - g_{\hat{x}^t}] + \sum_{x \neq x^t, \hat{x}^t} \exp(\beta_t s(x)) [g_{x^t} - g_x] \\
&= \exp(\beta_t s(\hat{x}^t)) [g_{x^t} - g_{\hat{x}^t}] + \sum_{x \neq x^t, \hat{x}^t} \exp(\beta_t s(x)) [g_{x^t} - g_x].
\end{aligned} \tag{3.9}$$

Therefore, denote $\xi_{t-1} = s(\hat{x}^t) - s(x^t) < 0$, we have that when $\beta_t \rightarrow \infty$,

$$\begin{aligned}
&\beta_t \|g_{x^t} - \mathbb{E}_{x \sim q_{\tilde{\nu}^t}}[g_x]\|_2 \\
&\leq \beta_t \|g_{x^t} - \mathbb{E}_{x \sim q_{\tilde{\nu}^t}}[g_x]\|_2 \int_{x' \in D} \exp(\beta_t [s(x') - s(x^t)]) \\
&= \beta_t \exp(-\beta_t s(x^t)) \left\| (g_{x^t} - \mathbb{E}_{x \sim q_{\tilde{\nu}^t}}[g_x]) \int_{x' \in D} \exp(\beta_t s(x')) \right\|_2 \\
&= \beta_t \left\| \exp(\beta_t [s(\hat{x}^t) - s(x^t)]) [g_{x^t} - g_{\hat{x}^t}] + \sum_{x \neq x^t, \hat{x}^t} \exp(\beta_t [s(x) - s(x^t)]) [g_{x^t} - g_x] \right\|_2 \\
&\leq \beta_t \sum_{x \neq x^t} \left\| \exp(\beta_t [s(x) - s(x^t)]) [g_{x^t} - g_x] \right\|_2 \\
&\leq \beta_t |D| \exp(\beta_t \xi_{t-1}) \max_{x' \in D} \|g_{x^t} - g_{x'}\|_2 \\
&\leq \beta_t |D| \exp(\beta_t \xi_{t-1}) \max_{x' \in D} (\|g_{x^t}\|_2 + \|g_{x'}\|_2) \\
&= 2\beta_t |D| \exp(\beta_t \xi_{t-1}) G \\
&\rightarrow 0,
\end{aligned}$$

where the first inequality holds due to the fact $s(x') < s(x^t)$ for any $x' \in D$, the second inequality holds due to triangle inequality, the third inequality holds due to the facts $(s(x) - s(x^t)) < \xi_{t-1}$ for any $x \neq x^t$, the fourth inequality holds due to the assumption that $\|g_x\|_2 \leq G$, the last line holds due to the fact that $x \exp(ax) \rightarrow 0$ for $a < 0$ and $x \rightarrow \infty$. Therefore, taking large enough β_t , we have

$$\|\beta_t \eta_t^2 (g_{x^t} - \mathbb{E}_{x \sim q_{\tilde{\nu}^t}}[g_x])\|_2 \leq \alpha \|\tilde{\nu}^t - \nu^*\|_2.$$

Next we show that $\langle \tilde{\nu}^t - \nu^*, g_{x^t} - \mathbb{E}_{x \sim q_{\tilde{\nu}^t}}[g_x] \rangle \geq \alpha \|\tilde{\nu}^t - \nu^*\|_2 \|g_{x^t} - \mathbb{E}_{x \sim q_{\tilde{\nu}^t}}[g_x]\|_2$. From (3.9)

we have

$$\begin{aligned}
& (g_{x^t} - \mathbb{E}_{x \sim q_{\hat{\nu}^t}}[g_x]) \exp(-\beta_t s(\hat{x}^t)) \int_{x' \in D} \exp(\beta_t s(x')) \\
&= g_{x^t} - g_{\hat{x}^t} + \underbrace{\sum_{x \neq x^t, \hat{x}^t} \exp(\beta_t [s(x) - s(\hat{x}^t)]) [g_{x^t} - g_x]}_{g(\beta_t)},
\end{aligned}$$

For $g(\beta_t)$, denote $\hat{\xi}_{t-1} = \min_{x \neq x^t, \hat{x}^t} [s(x) - s(\hat{x}^t)] < 0$. Then when $\beta_t \rightarrow \infty$, we have

$$\begin{aligned}
\|g(\beta_t)\|_2 &\leq \sum_{x \neq x^t, \hat{x}^t} \left\| \exp(\beta_t [s(x) - s(\hat{x}^t)]) [g_{x^t} - g_x] \right\|_2 \\
&\leq |D| \exp(\beta_t \hat{\xi}_{t-1}) \max_{x \in D} \|g_{x^t} - g_x\|_2 \\
&\leq 2G|D| \exp(\beta_t \hat{\xi}_{t-1}) \\
&\rightarrow 0,
\end{aligned} \tag{3.10}$$

where the first inequality holds due to triangle inequality, the second inequality holds due to the fact $s(x) - s(\hat{x}^t) < \hat{\xi}_{t-1}$, the third inequality holds due to the assumption that $\|g_x\|_2 \leq G$, the last line holds because $\exp(-x) \rightarrow 0$ when $x \rightarrow \infty$. Thus when $\beta_t \rightarrow \infty$, we have

$$\begin{aligned}
\frac{g_{x^t} - \mathbb{E}_{x \sim q_{\hat{\nu}^t}}[g_x]}{\|g_{x^t} - \mathbb{E}_{x \sim q_{\hat{\nu}^t}}[g_x]\|_2} &= \frac{(g_{x^t} - \mathbb{E}_{x \sim q_{\hat{\nu}^t}}[g_x]) \exp(-\beta_t s(\hat{x}^t)) \int_{x' \in D} \exp(\beta_t s(x'))}{\|(g_{x^t} - \mathbb{E}_{x \sim q_{\hat{\nu}^t}}[g_x]) \exp(-\beta_t s(\hat{x}^t)) \int_{x' \in D} \exp(\beta_t s(x'))\|_2} \\
&= \frac{g_{x^t} - g_{\hat{x}^t} + g(\beta_t)}{\|g_{x^t} - g_{\hat{x}^t} + g(\beta_t)\|_2} \\
&\rightarrow \frac{g_{x^t} - g_{\hat{x}^t}}{\|g_{x^t} - g_{\hat{x}^t}\|_2},
\end{aligned}$$

where the last line holds due to $g(\beta_t) \rightarrow 0$ from (3.10). Therefore, we know that for large enough β_t , we have

$$\left| \left\langle \frac{\tilde{\nu}^t - \nu^*}{\|\tilde{\nu}^t - \nu^*\|_2}, \frac{g_{x^t} - \mathbb{E}_{x \sim q_{\hat{\nu}^t}}[g_x]}{\|g_{x^t} - \mathbb{E}_{x \sim q_{\hat{\nu}^t}}[g_x]\|_2} \right\rangle - \left\langle \frac{\tilde{\nu}^t - \nu^*}{\|\tilde{\nu}^t - \nu^*\|_2}, \frac{g_{x^t} - g_{\hat{x}^t}}{\|g_{x^t} - g_{\hat{x}^t}\|_2} \right\rangle \right| \leq \alpha,$$

Finally, due to the fact $\langle \tilde{\nu}^t - \nu^*, g_{x^t} - g_{\hat{x}^t} \rangle = 2\alpha \|\tilde{\nu}^t - \nu^*\|_2 \|g_{x^t} - g_{\hat{x}^t}\|_2$ from (3.8), we have

$$\left\langle \frac{\tilde{\nu}^t - \nu^*}{\|\tilde{\nu}^t - \nu^*\|_2}, \frac{g_{x^t} - \mathbb{E}_{x \sim q_{\tilde{\nu}^t}}[g_x]}{\|g_{x^t} - \mathbb{E}_{x \sim q_{\tilde{\nu}^t}}[g_x]\|_2} \right\rangle \geq \left\langle \frac{\tilde{\nu}^t - \nu^*}{\|\tilde{\nu}^t - \nu^*\|_2}, \frac{g_{x^t} - g_{\hat{x}^t}}{\|g_{x^t} - g_{\hat{x}^t}\|_2} \right\rangle - \alpha = \alpha.$$

□

Now we prove the main theorem.

Proof of Theorem 1. The naive learner and the teacher-aware learner, after receiving (x^t, y^t) , will update their model to $\tilde{\nu}^t = (\nu^{t-1} - \eta_t g_{x^t})$ and $\nu^t = (\nu^{t-1} - \eta_t g_{x^t} - 2\beta_t \eta_t^2 (g_{x^t} - \mathbb{E}_{x \sim q_{\tilde{\nu}^t}}[g_x]))$ respectively. Then with large enough β_t , we have

$$\begin{aligned} & \|\nu^t - \nu^*\|_2^2 \\ &= \|\tilde{\nu}^t - \nu^* - 2\beta_t \eta_t^2 (g_{x^t} - \mathbb{E}_{x \sim q_{\tilde{\nu}^t}}[g_x])\|_2^2 \\ &= \|\tilde{\nu}^t - \nu^*\|_2^2 - 4\langle \tilde{\nu}^t - \nu^*, \beta_t \eta_t^2 (g_{x^t} - \mathbb{E}_{x \sim q_{\tilde{\nu}^t}}[g_x]) \rangle + 4\|\beta_t \eta_t^2 (g_{x^t} - \mathbb{E}_{x \sim q_{\tilde{\nu}^t}}[g_x])\|_2^2 \\ &\leq \|\tilde{\nu}^t - \nu^*\|_2^2 - 4\alpha \|\tilde{\nu}^t - \nu^*\|_2 \|\beta_t \eta_t^2 (g_{x^t} - \mathbb{E}_{x \sim q_{\tilde{\nu}^t}}[g_x])\|_2 + 4\|\beta_t \eta_t^2 (g_{x^t} - \mathbb{E}_{x \sim q_{\tilde{\nu}^t}}[g_x])\|_2^2 \\ &\leq \|\tilde{\nu}^t - \nu^*\|_2^2 - 4\alpha \|\tilde{\nu}^t - \nu^*\|_2 \|\beta_t \eta_t^2 (g_{x^t} - \mathbb{E}_{x \sim q_{\tilde{\nu}^t}}[g_x])\|_2 \\ &\quad + 4\alpha \|\tilde{\nu}^t - \nu^*\|_2 \|\beta_t \eta_t^2 (g_{x^t} - \mathbb{E}_{x \sim q_{\tilde{\nu}^t}}[g_x])\|_2 \\ &= \|\tilde{\nu}^t - \nu^*\|_2^2, \end{aligned}$$

where the first inequality holds due to (3.8) in Lemma 2, the second inequality holds due to (3.7) in Lemma 2. □

Corollary 3 (Global Improvement). *Under the same condition of Theorem 1, suppose that $\|\partial l(\langle \tilde{x}, \nu \rangle, y) / \partial \nu\|_2^2$ and $l(\langle \tilde{x}, \nu \rangle, y)$ are L -Lipschitz for x with any ν . Suppose the sample set D^t satisfies that for any $x \in D^t$, there exists $x' \in D^t$ such that $\|x' - x\|_2 \leq \epsilon / (TL(\eta_t^2 + 4\eta_t))$ for any t , where T is the total number of iterations. Then if the inequality*

$$\|\nu_1 - \nu^*\|_2^2 - \max_{(x,y) \in D^t} \widehat{TV}_{\nu^*}(x, y | \nu_1) \leq \|\nu_2 - \nu^*\|_2^2 - \max_{(x,y) \in D^t} \widehat{TV}_{\nu^*}(x, y | \nu_2) \quad (3.11)$$

holds for any ν_1, ν_2 that satisfy $\|\nu_1 - \nu^*\|_2^2 \leq \|\nu_2 - \nu^*\|_2^2$, then with the same parameter initialization, learning rate and a teacher following Eq. (3.3), a teacher-aware learner can always converge not slower than a naive learner up to ϵ error.

Proof. Let ν_a and ν_b be the model parameter of the naive learner and the teacher-aware learner. Denote $\widehat{TV}_{\nu^*}^E(\nu) = \max_{x \in E} \widehat{TV}_{\nu^*}(x, y|\nu)$, E is some dataset. Let x^* denote the argmax of $\max_{x \in D} \widehat{TV}_{\nu^*}(x, y|\nu)$, then by the assumption on D^t , we know that there exists x' such that $\|x' - x^*\|_2 \leq \epsilon/(TL(\eta_t^2 + 4\eta_t))$. Then we have

$$\begin{aligned}
& \widehat{TV}_{\nu^*}^D(\nu) - \widehat{TV}_{\nu^*}^{D^t}(\nu) \\
&= \max_{x \in D} (-\eta_t^2 g_x(\nu) + 2\eta_t(l(\nu, x) - l(\nu^*, x))) - \max_{x \in D^t} (-\eta_t^2 g_x(\nu) + 2\eta_t(l(\nu, x) - l(\nu^*, x))) \\
&\leq (-\eta_t^2 g_{x^*}(\nu) + 2\eta_t(l(\nu, x^*) - l(\nu^*, x^*))) - (-\eta_t^2 g_{x'}(\nu) + 2\eta_t(l(\nu, x') - l(\nu^*, x'))) \\
&\leq L(\eta_t^2 \|x^* - x'\|_2 + 4\eta_t \|x^* - x'\|_2) \quad (L\text{-Lipschitz}) \\
&\leq \epsilon/T. \quad (3.12)
\end{aligned}$$

Now we prove that $\|\nu_b^t - \nu^*\|_2^2 \leq \|\nu_a^t - \nu^*\|_2^2 + t/T\epsilon$ for all $1 \leq t \leq T$. As ν^0 is the same for both learners, knowing theorem 1, we have $\|\nu_b^1 - \nu^*\|_2^2 \leq \|\nu_a^1 - \nu^*\|_2^2 + 1/T\epsilon$. Suppose $\|\nu_b^t - \nu^*\|_2^2 \leq \|\nu_a^t - \nu^*\|_2^2 + t/T\epsilon$, then we have

$$\begin{aligned}
& \|\nu_b^{t+1} - \nu^*\| \\
&\leq \|\nu_b^t - \nu^*\|_2^2 - TV_{\nu^*}^{D^t}(\nu_b^t) \quad (\text{Theorem 1}) \\
&\leq \|\nu_b^t - \nu^*\|_2^2 - \widehat{TV}_{\nu^*}^{D^t}(\nu_b^t) \quad (\text{convexity of } l) \\
&\leq \|\nu_b^t - \nu^*\|_2^2 - \widehat{TV}_{\nu^*}^D(\nu_b^t) + \epsilon/T \quad \widehat{TV}_{\nu^*}^D(\nu) - \widehat{TV}_{\nu^*}^{D^t}(\nu) \leq \epsilon \\
&\leq \|\nu_a^t - \nu^*\|_2^2 - \widehat{TV}_{\nu^*}^D(\nu_a^t) + (t+1)/T\epsilon \quad (\text{condition}) \\
&\leq \|\nu_a^t - \nu^*\|_2^2 - \widehat{TV}_{\nu^*}^{D^t}(\nu_a^t) + (t+1)/T\epsilon \quad \widehat{TV}_{\nu^*}^D(\nu) - \widehat{TV}_{\nu^*}^{D^t}(\nu) \geq 0 \\
&= \|\nu_a^{t+1} - \nu^*\|_2^2 + (t+1)/T\epsilon
\end{aligned}$$

Therefore, we have $\|\nu_b^t - \nu^*\|_2^2 \leq \|\nu_a^t - \nu^*\|_2^2 + t/T\epsilon$, which suggests that the teacher-aware learner can always converge no slower than the naive learner up to an ϵ factor. $\square$

To guarantee that $\|x' - x\|_2 \leq \epsilon/(TL(\eta_t^2 + 4\eta_t))$ for any $x \in D$, we need the subset $D^t \subseteq D$ to be ‘uniform distributed’ on D . To achieve this goal, we can uniformly sample point $x \in D$ and let D^t to be the set of these points. It is easy to verify that the ‘uniform distributed’ property holds with high probability when $|D^t|$ is large enough. Meanwhile, Eq. (3.11) in Corollary 3 is defined as teaching monotonicity in [LDH17], and they proved that the squared loss satisfies teaching monotonicity given a dataset $D = \{x \in \mathbb{R}^d, \|x\| \leq R\}$ [LDH17, see, proposition 3]. The main difference between Eq. (3.11) and that in [LDH17] is that Eq. (3.11) works for the non-omniscient teacher setting, while [LDH17] focuses on the omniscient teacher setting.

3.3 Results

3.3.1 Machine Teacher

To justify the effectiveness of iterative teacher-aware learning (ITAL), we compared it with IMT with a naive learner on regression, classification, and IRL tasks. The coverage of the squared loss, cross-entropy loss, and negative log-likelihood proves the robustness of our algorithm on various selections of l . For regression tasks, we measured the performance using the difference between $\|\omega^t - \omega^*\|_2$ and the mean squared loss of the test set. For the classification task, we measured the difference, the cross-entropy loss, and the classification accuracy of the test set. For online IRL problems, we measured the parameter difference, the total variance between the teacher’s and the learner’s policies, and the average rewards achieved by the learner. The feature dimension of the teachers can be different from that of the learners in some experiments. In Fig. 3.1, we show the results of the teacher having a smaller feature dimension than the learner does. The rest can be found in [YZS21, see

sec. B]. **Batch** means the learner uses all the data in the mini-batch to calculate the mean gradient. **SGD** means the learner randomly selects an example in the mini-batch to calculate the gradient. **ITAL- M s** represent our algorithm with M indicates $|\widehat{D}^t|$. The mini-batch D^t is randomly sampled at every step with batch size 20. The learning rate is **1e-3** for all the experiments. β_t is in the scale of **1e4**, varying for different settings. We grid search β_t starting from **1e4** and use the largest one inducing Eq. (3.4) that is no longer a delta function. We used two types of loss functions in all the experiments. For regression tasks, our loss function is

$$\min_{\omega \in \mathbb{R}^d, b \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^n \frac{1}{2} (\omega^T x_i + b - y_i)^2 + \frac{\lambda}{2} \|\omega\|_2^2$$

For classification tasks, our loss function is

$$\min_{\omega \in \mathbb{R}^d, b \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^n \sum_{k=1}^K -\mathbf{1}(y_i = k) \log p_{ik} + \frac{\lambda}{2} \|\omega\|_2^2$$

$$p_{ik} = \frac{\exp(\omega_k^T x_i + b_k)}{\sum_{k'=1}^K \exp(\omega_{k'}^T x_i + b_{k'})}$$

where $\omega \in \mathbb{R}^{K \times d}$ and ω_k is the k -th row of ω , $b \in \mathbb{R}^K$ and b_k is the k -th element of b . The norm is the Frobenius norm. In both regression and classification tasks, we refer to $[\omega, b]$ as ω^* . In all the following experiments, we used a constant learning rate 10^{-3} for all the algorithms. The size of the mini-batch was set to 20. As the gradient scale is different in different experiments, we used different β s. We chose the hyperparameter β so that at the beginning of the learning, the data in the mini-batch with the smallest teaching volume has above 80% probability of being selected. In the supplementary material, we show additional results of our experiments. All plots are consistent with the results in the main text. All of our experiments were run on machines with 16 I9-9900K cores and 64GiB RAM. The longest setting is the Tiny ImageNet classification, which takes about 12 hours to finish (2000 iterations for 8 methods and 20 random seeds). We ran each experiment with 20

different random seeds to calculate the mean and its standard error, shown in Fig. 3.1.

It can be seen that using the full mini-batch gives almost identical learning performance as using only one random sample from it. IMT has noticeable but limited improvements compared with Batch and SGD, suggesting not necessarily substantial advantage brought by the helpful teacher. ITAL, on the other hand, significantly outperforms all other baselines, even with only 2 data points as the approximation of the full mini-batch. The learner modeled by these baselines only learns from the examples, but when the examples are no longer acquired randomly but from an intentional teacher, the example selection of the teacher also conveys a large amount of information. In particular, the teacher-aware learner can absorb information from not only the selected examples but also the unselected ones. As the learner has access to more unselected examples, he has a better approximation of the teaching process and learns more efficiently.

3.3.1.1 Supervised Learning

Linear Models of Synthetic Data: In these experiments, we explored the convergence of our method in linear regression and multinomial logistic regression. For linear regression, we randomly generated a M -dimensional vector and a bias term as the ω^* , and $X \in \mathbb{R}^{N \times (d+1)}$ as the training set, with the last column being all 1s. The labels are $Y = X\omega^*$. For the classification task, we randomly generated K points in the d -dimensional space, each of which is used as the mean of a normal distribution. Then we sampled N/K points from each Gaussian distribution together as the training data. The labels are the indices of these distributions. With these data, we trained a logistic regression model using Scikit-learn [PVG11], and used the coefficients as the teacher’s ω^* . We used a random orthogonal projection matrix to generate the teacher’s feature space from the student’s. At every step, a subset of the training data is randomly selected as the mini-batch. The data points in that mini-batch along with their labels and the index of the data selected by the teacher are sent to the student.

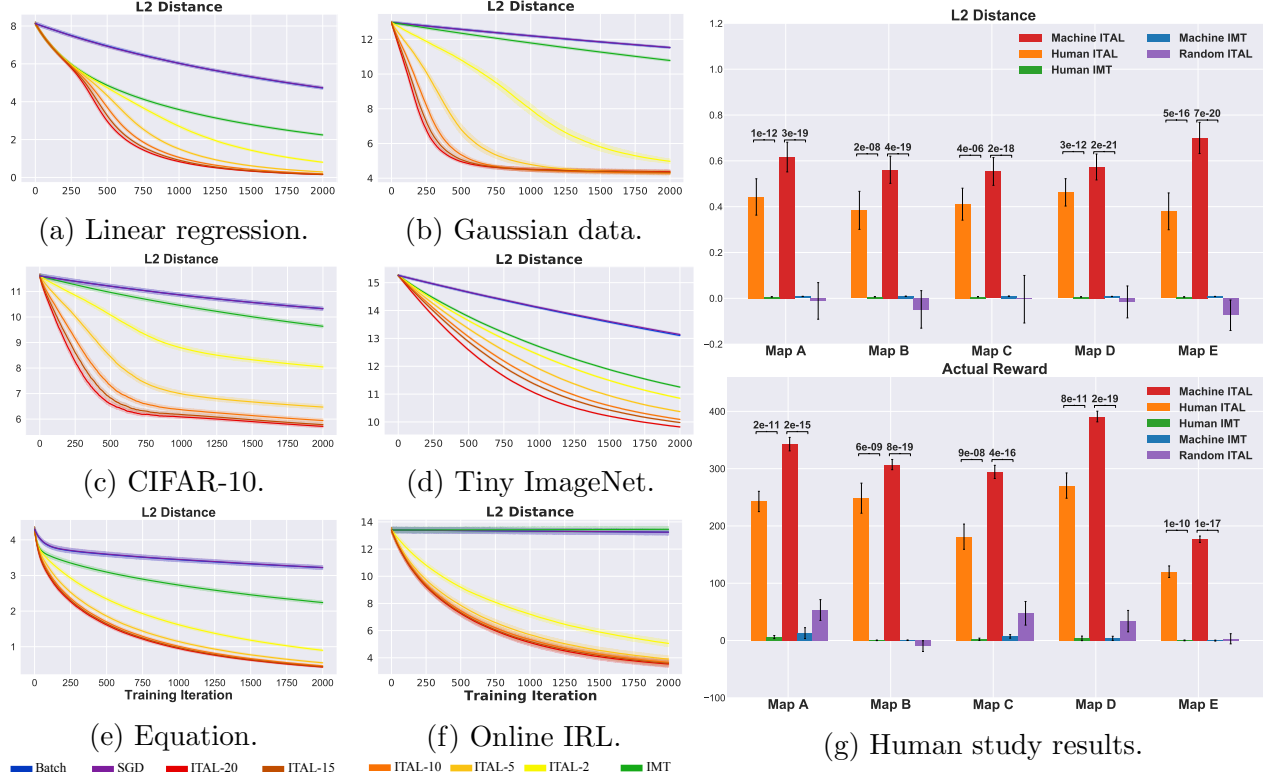


Figure 3.1: **Fig. 3.1a-Fig. 3.1f**: Cooperative teacher results. Our method always gives a substantial improvement over IMT, showing the effect of teacher-awareness. Within 2000 steps, ITAL already show convergence, while a naive learner only learns to a limited extent in most tasks. **Fig. 3.1g**: In the top plot, the height of each bar represents the **decrease** of the L2-distance between the learner’s reward parameter and the ground-truth parameter. In the bottom, the height represents the accumulated reward. Paired t-tests were conducted between Human (Machine) ITAL & IMT respectively.

Linear Classifiers on Natural Image Datasets: We further evaluated our teacher-aware learner on image datasets, CIFAR-10 [KH09] and Tiny ImageNet [tin17] (an adaptation of ImageNet [DDS09b] used in Stanford 231n with 200 classes and 500 images in each class). In these experiments, the teacher tried to teach the parameters of the last fully connected (FC) layer in a convolutional neural network (CNN) trained on the dataset. We trained 3 baseline CNNs independently to do CIFAR-10 10-class and ImageNet 200-class classification. All of them achieved reasonable accuracy ($\geq 82\%$ for CIFAR-10 and $\geq 58\%$ top-1, $\geq 85\%$ top-5 for ImageNet). For CIFAR-10, we trained three different types

of CNNs, CNN-6/9/12. For ImageNet we used VGG-13/16/19 [SZ15]. The features fed into the last FC layers are extracted to be the teaching dataset. The learner’s feature is from CNN-9/VGG-16 and we set the teacher as either CNN-6/12 or VGG-13/19.

Linear Regression for Equation Simplification: In this experiment, we learned a linear value function that can be used to guide action selections. Given polynomial equations with fraction coefficients and unmerged terms, we want to simplify them into cleaner forms with all the terms merged correctly, all the coefficients rescaled to integers without common factors larger than 1, and all the terms sorted by the descending power. For example, equation $-\frac{1}{2}x^2y + \frac{1}{3}xy = -\frac{1}{5}y^3 + \frac{1}{3}x^2y$ will be simplified to $-25x^2y + 10xy + 6y^3 = 0$. We defined a set of equation editing actions and a set of simplification rules. For a given equation, we applied the rules, recorded every editing action, and collected a simplifying trajectory. With all the trajectories of the training equations, we trained a value function by assuming that the value monotonically increases in each trajectory. Then the teacher tried to teach the student this value function. We used three different feature dimensions: 40D, 45D, and 50D. The learner always used 45D, and the teacher used 40D or 50D.

3.3.1.2 Online Inverse Reinforcement Learning

In this experiment, we changed from labeled data in standard supervised learning to demonstrations in IRL. The learner wanted to learn the parameter for a linear reward function $r(s, \omega^*)$ so that the likelihood of the demonstrations is maximized [BMS11, Vro14, ML15]. One challenge is that the max function in Bellman equations [SB18] is non-differentiable. Thus, we approximated max with soft-max, namely: $\max(a_0, \dots, a_n) \approx \frac{\log(\sum_{i=0}^n \exp ka_i)}{n}$, with k controlling the level of approximation and leveraged the online Bellman gradient iteration [LB17].

Specifically, we want to learn a reward function $r(s, \omega^*)$. We can define a Markov Decision Process $\langle S, A, r, P, \gamma \rangle$, where S is the state space, A is the action space, $r : S \rightarrow R$ is a reward function mapping from state to a real number as the reward. $P_{ss'}^a$ is the transition model

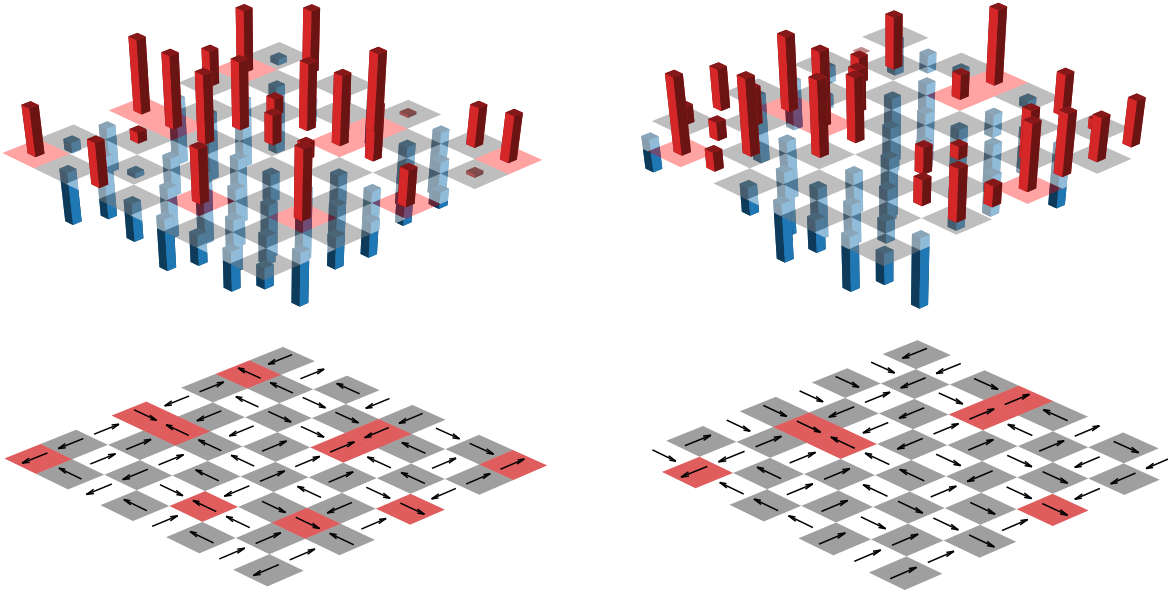


Figure 3.2: IRL map examples. Each map has 8×8 grids. Every grid contains a reward. Maps in the first row plots the **ground truth** rewards in each grid. Red bars represent positive rewards and blue bars represent negative rewards. The learner tries to learn a policy to walk in the map and collect the most accumulative rewards. The arrows below indicate the most probable action taken by the learner after he learned the reward function. The red grids are targets of all their neighbors.

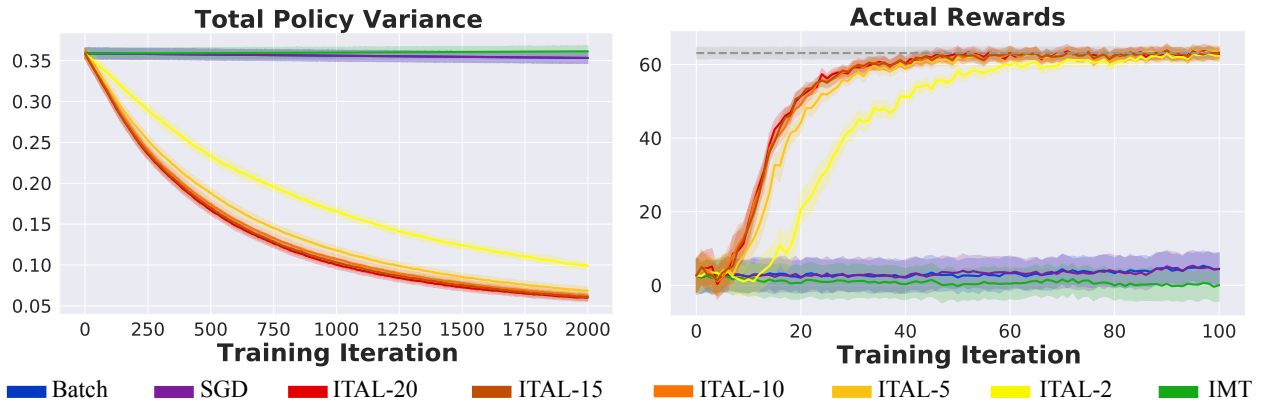


Figure 3.3: Total variance between the learner's policy and the teacher's policy and the actual gain of the learner during the learning process. The gray horizontal dash line represents teacher's expected accumulative reward.

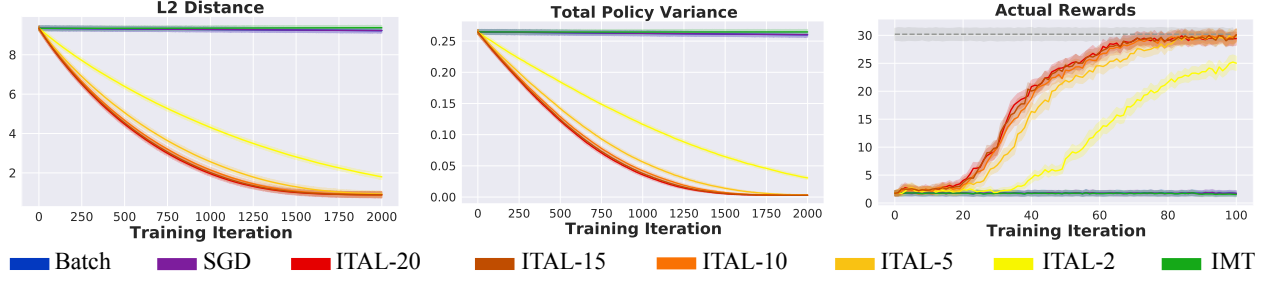


Figure 3.4: Learning results of sparse reward maps. All grids in the 8×8 map have 0 reward but 3 grids with reward 1. In every experiment, we randomly selected the 3 grids. Curves drawn with results using 20 different random seeds. The gray horizontal dash line represents teacher's expected accumulative reward.

that state s becomes s' after the agent conducts action a . γ is a discount factor that ensures the convergence of the MDP over an infinite horizon. Given a reward function, using the Bellman equation we have

$$V^*(s) = \max_{a \in A} \sum_{s' | s, a} P_{ss'}^a [r(s') + \gamma V^*(s')]$$

$$Q^*(s, a) = \max_{a' \in A} \sum_{s' | s, a} P_{ss'}^a [r(s') + \gamma \max_{a' \in A} Q^*(s', a')]$$

Suppose an agent behaves by following Boltzmann rationality:

$$\pi(a^t | s^t; \omega) = \frac{\exp(\alpha Q^*(s^t, a^t; \omega))}{\sum_{a' \in A} \exp(\alpha Q^*(s^t, a'; \omega))}$$

Take log-likelihood of this function we can have an objective function that the learner can optimize to learn ω^* .

$$l(s^t, a^t; \omega^{t-1}) = \alpha Q^*(s^t, a^t; \omega^{t-1}) - \log \sum_{a' \in A} \exp(\alpha Q^*(s^t, a'; \omega^{t-1}))$$

$$\omega^t = \omega^{t-1} + \eta_t \frac{\partial l(s^t, a^t; \omega^{t-1})}{\partial \omega^{t-1}} \quad (3.13)$$

Then, the online IRL process can be accommodated by our learning framework. We ap-

proximated non-differentiable max with soft-max, namely: $\max(a_0, \dots, a_n) \approx \frac{\log(\sum_{i=0}^n \exp ka_i)}{n}$, with k controlling the level of approximation and leveraged the online Bellman gradient iteration [LB17] to calculate the gradient for each step.

$$\begin{aligned}\frac{\partial V_{g,k}(s; \omega^t)}{\partial \omega^t} &= \sum_{a \in A} \frac{\exp(kQ_{g,k}(s, a; \omega^t))}{\sum_{a' \in A} \exp(kQ_{g,k}(s, a'; \omega^t))} \frac{\partial Q_{g,k}(s, a; \omega^t)}{\partial \omega^t} \\ \frac{\partial Q_{g,k}(s, a; \omega^t)}{\partial \omega^t} &= \sum_{s'|s,a} P_{ss'}^a \left(\frac{\partial r(s'; \omega^t)}{\partial \omega^t} + \gamma \frac{\partial V_{g,k}(s'; \omega^t)}{\partial \omega^t} \right)\end{aligned}$$

In every round, we randomly sample 20 (s, a) pairs from $|S| \times |A|$ state-action pairs as our minibatch. Then the teacher will conduct Bellman gradient iteration. The learner will return his reward estimation for each grid to the teacher.

We used an 8×8 grid map as the environment, and the action space A includes four actions up, down, left, right. See figure 3.2 for map examples. 80% of the time, the agent goes to its target, 18% of the time ends up in another random neighbor grid and 2% of the time dies abruptly (game ends). We set $\gamma = 0.5$. The reward in each grid is randomly sampled from a uniform distribution, $U[-2, 2]$. If we encode each grid with a one-hot vector, then the reward parameter is a 64D vector with the i -th entry corresponding to the reward of the i -th grid. The teacher uses a shuffled map encoding as the student's. For instance, if the first grid is $[1, 0, \dots, 0]$ to the learner, then it becomes $[0, \dots, 0, 1, 0, \dots]$ to the teacher. See figure 3.3 for the actual accumulative reward acquired by the agent during learning.

In addition to the environment with random dense rewards, we tested the teacher-aware learner in a sparse reward environment. Each time, we only pick 3 grids randomly to assign a non-zero reward. Our algorithm still shows robust performance. Results in figure 3.4. The setup of this IRL can be further scaled-up, as we'll see in Chapter 5.

3.3.1.3 Adversarial Teacher

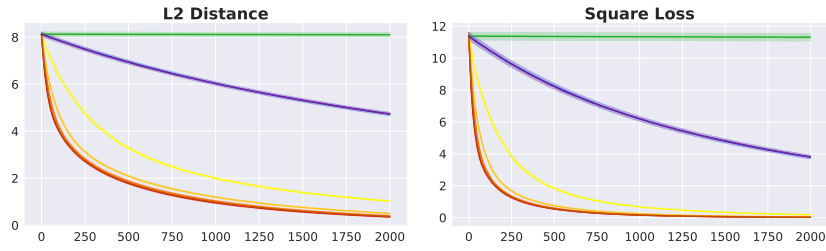
In addition to the cooperative setting that we assumed throughout the discussion above, we also explored if the learner can still learn given an adversarial teacher. An adversarial teacher doesn't mean that she gives fake data to the student, but she uses $\arg \min$ in Eq. (3.4) instead of using $\arg \max$. That is, she always chooses the least helpful data for the learner. Hence, a learner, being aware of this unhelpful pedagogy, will adjust Eq. (3.3) accordingly by using $\beta_t \leq 0$. See table 3.1 for the β used in these experiments. We redid all previous experiments with an adversarial teacher and showed that our learner can still learn effectively given an adversarial teacher, while a naive learner barely improves. See Fig. 3.5 for results. These experiments justify the universal utility of modeling the teacher's intention, regardless of the informativeness of the teaching examples.

Table 3.1: Selection of β s in the adversarial teacher experiments. For cooperative teachers, the absolute values of the β s are the same, only the signs are flipped.

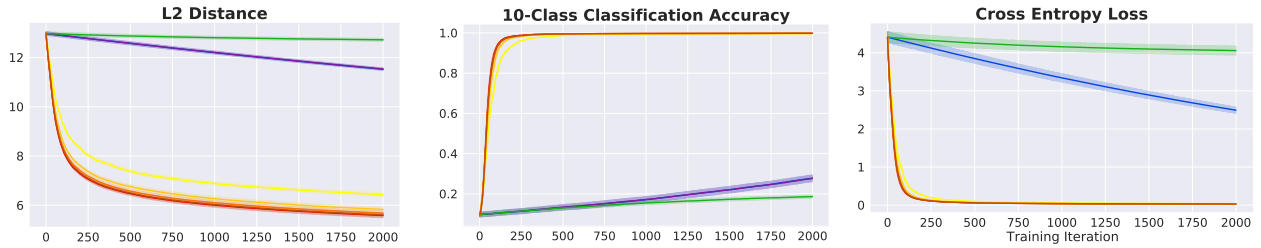
Experiment	Value of β
Linear Classifiers on Synthesized Data	-60000
Linear Regression on Synthesized Data	-5000
Linear Classifiers on MNIST Dataset	-30000
Linear Classifiers on CIFAR Dataset	$-50000(1 - 5e^{-6})^t$
Linear Classifiers on Tiny ImageNet Dataset	-1000
Linear Regression for Equation Simplification	-5000
Online Inverse Reinforcement Learning (Random Rewards)	-25000
Online Inverse Reinforcement Learning (Sparse Rewards)	-30000

3.3.2 Human Teacher

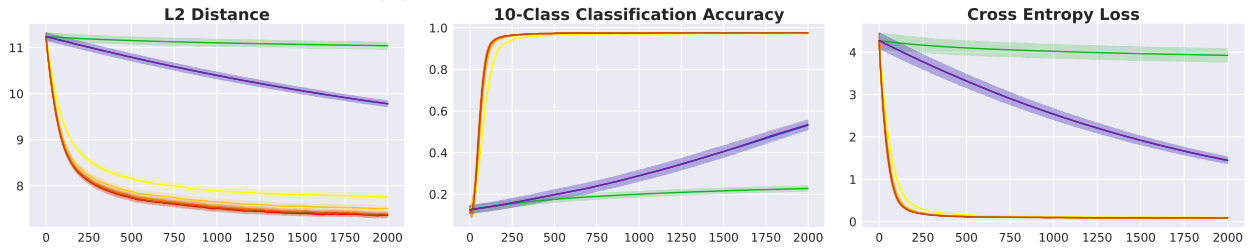
In the previous section, we showed that teacher-awareness substantially accelerates learning, given a machine teacher. In this section, we further investigate if our teacher-aware learner can show an advantage in scenarios where humans play the role of the cooperative teacher. We hypothesize that despite the discrepancy between the pedagogical pattern of human and machine teachers, our learner can still benefit from his teacher-awareness modeled with



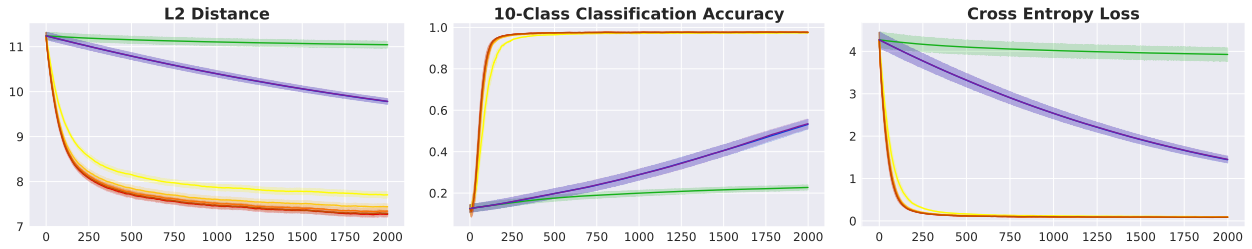
(a) Linear Regression



(b) Gaussian data 10-class classification.



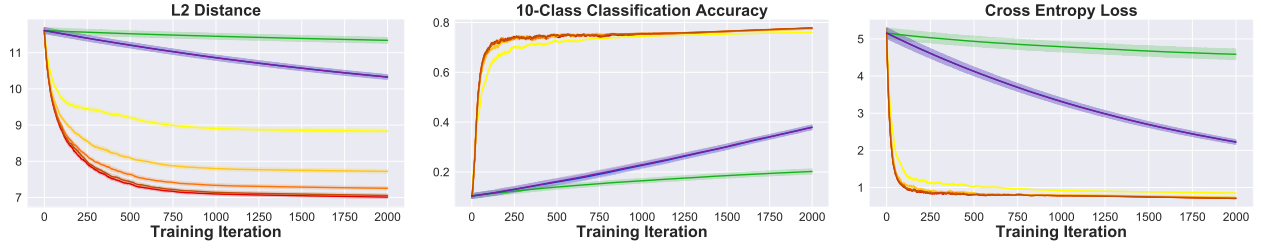
(c) MNIST 10-class, 20D teacher features



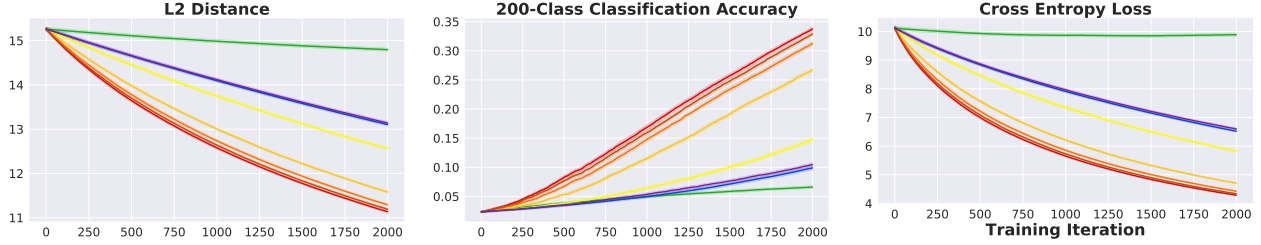
(d) MNIST 10-class, 30D teacher features



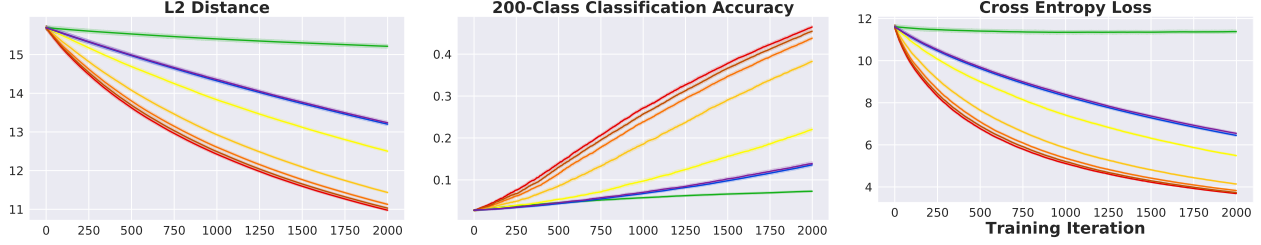
(e) CIFAR-10, teacher feature from CNN-9.



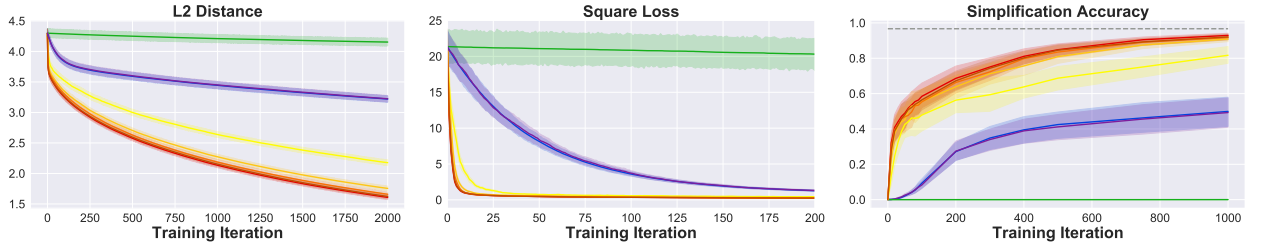
(f) CIFAR-10, teacher feature from CNN-12.



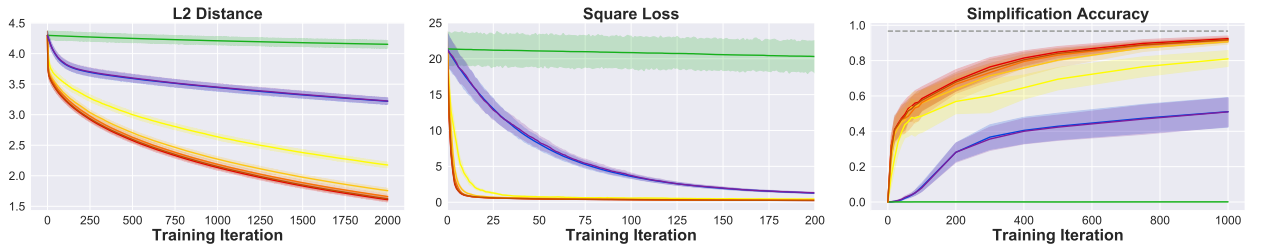
(g) Tiny ImageNet, teacher feature from VGG-13, showing top-5 accuracy.



(h) Tiny ImageNet, teacher feature from VGG-19, showing top-5 accuracy.



(i) Equation simplification, 40D teacher features



(j) Equation simplification, 50D teacher features

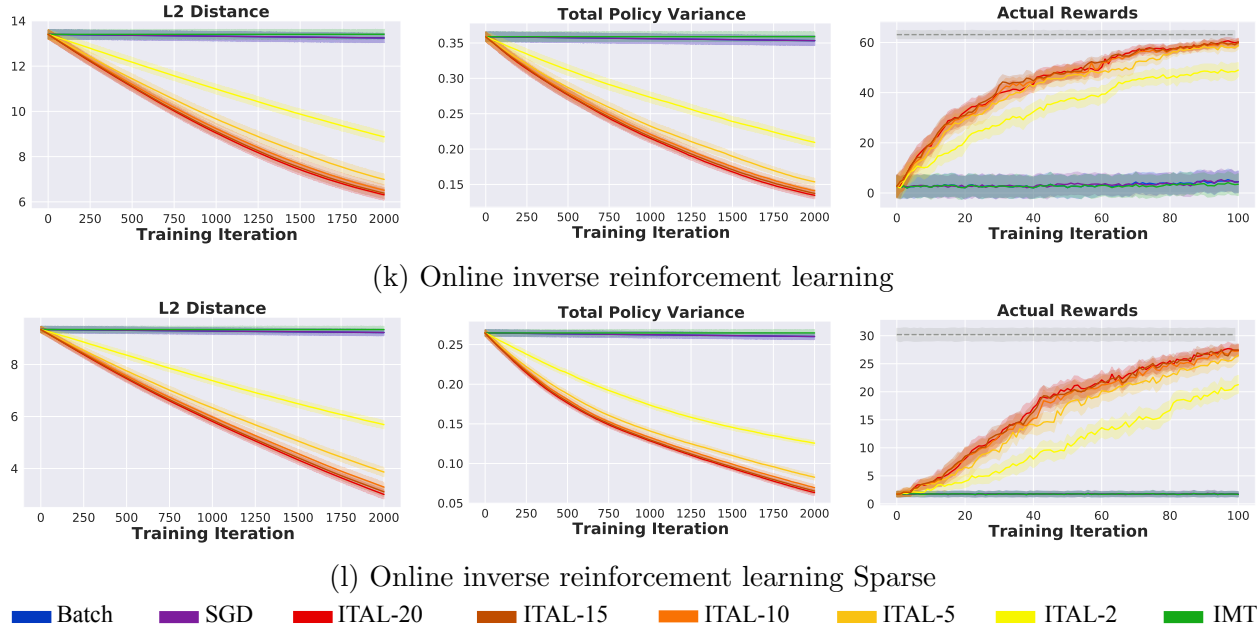


Figure 3.5: Adversarial teacher results. With an adversarial teacher, a naive learner can no longer learn effectively. ITAL still learns efficiently. SGD and batch learning are included for comparison.

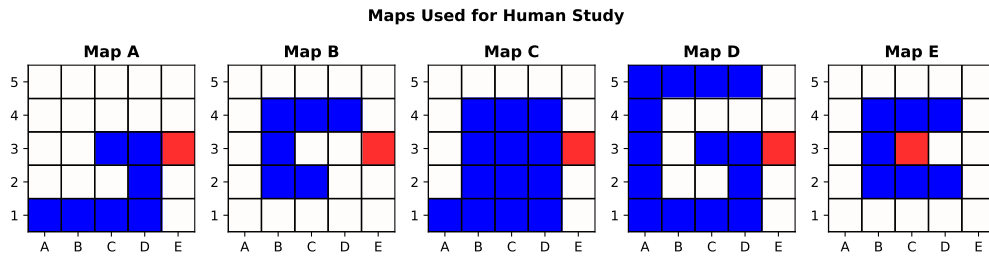


Figure 3.6: Map configurations

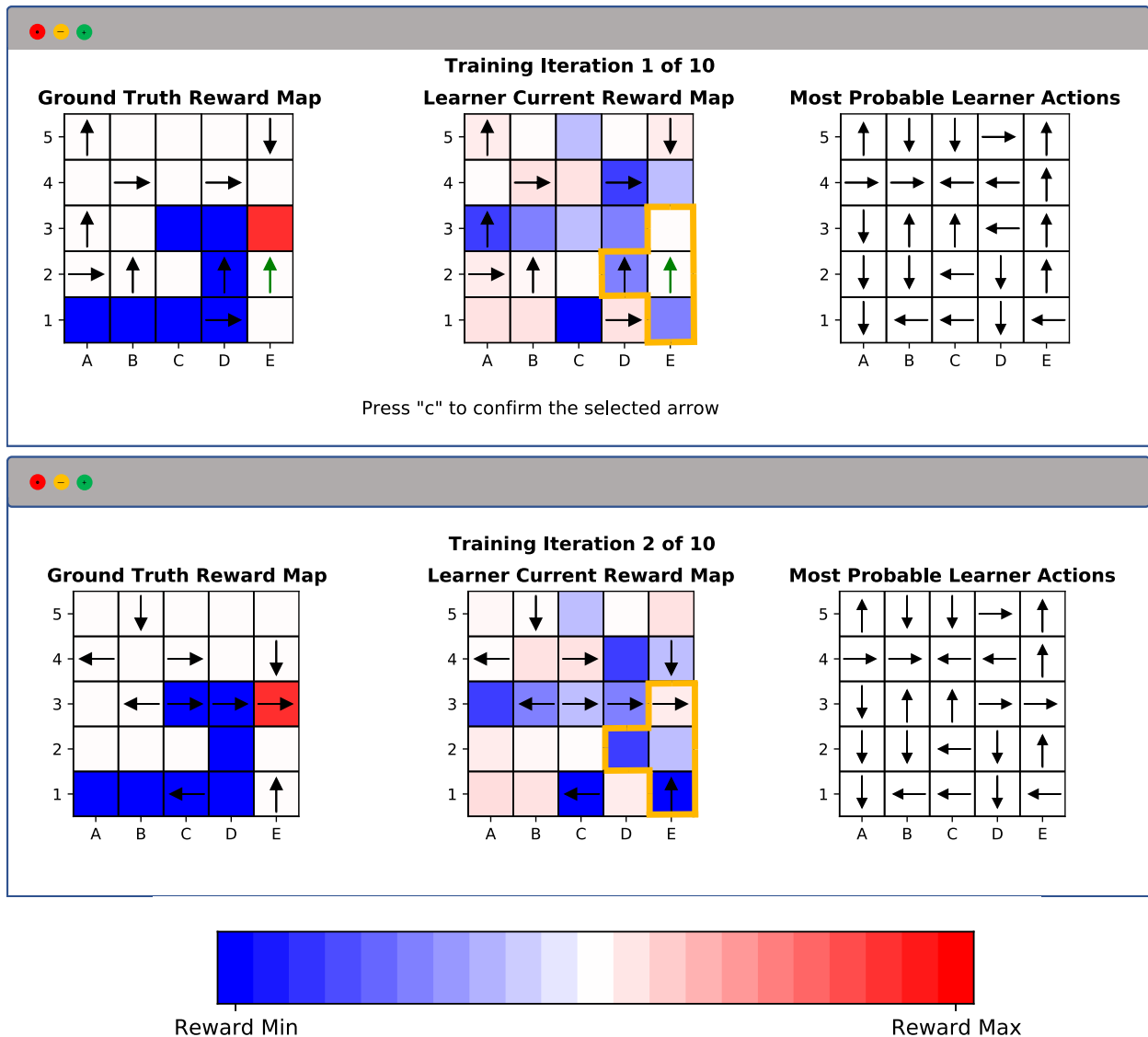


Figure 3.7: Actual interface used in the human study. The reward spectrum will be shown to the subjects at the experiment introduction. In this example, the subject chose the green arrow as the example at the 1-st iteration. Then, as we annotated with the orange T-shaped boxes, the estimated reward of the target tile of the green arrow increased, while rewards of the arrow source and the surrounding neighbors decreased. The orange boxes were not included in the human study.

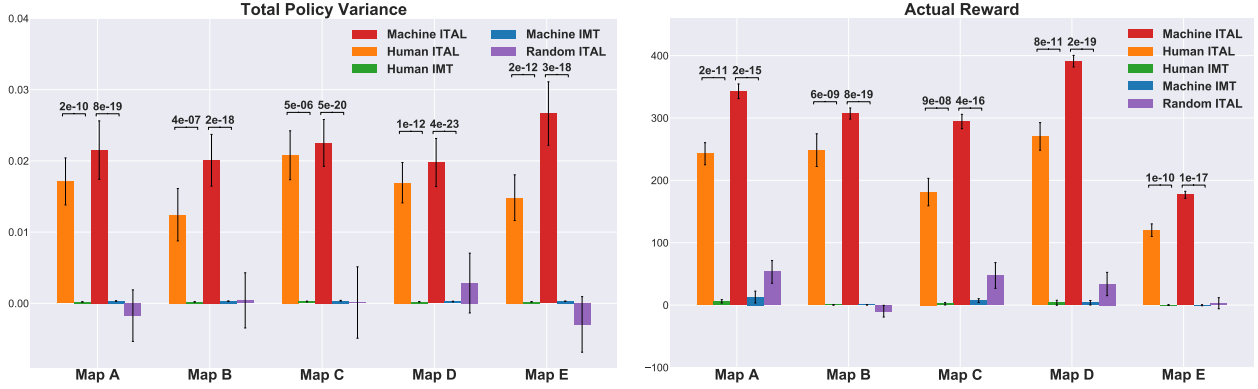


Figure 3.8: Human study results. All the p -values are calculated with paired t-test.

Eq. (3.4).

We conducted a proof-of-concept human study with a similar but simplified version of the IRL experiments in Section 3.3.1.2. To better suit human participants, we first change the maps from 8×8 to 5×5 . Second, instead of assigning random continuous rewards to the grids, we color them with white, blue, and red, representing neutral, bad, and good tiles. In each teaching session, a participant is given one of the five reward maps as the ground truth and a randomly initialized learner to be taught. Then, the participant will be asked to teach the learner about the ground truth reward in each grid by providing (s, a) examples as in Section 3.3.1.2. In each time step, we construct the examples by randomly sampling a set of 10 grids and drawing an arrow on each sampled grid indicating which direction the learner should go to if in that grid. The human teacher is asked to choose the most helpful arrow given the learner’s current reward map. The map configurations are in Fig. 3.6. A similar map setup for reward teaching was used by [HLM16]. Every participant will teach both the naive and the teacher-aware learner about the same map. We then run a paired sample t-test to compare the learning effect of the two types of learners. We show the improvement of the L2-distance between the learner’s reward parameters and the ground-truth reward parameters and the accumulated reward in Fig. 3.1g. Comparison of policy total variance and learning curves are in Fig. 3.9.

For all maps, the ITAL method has a significant (p -value ≈ 0) advantage over its IMT

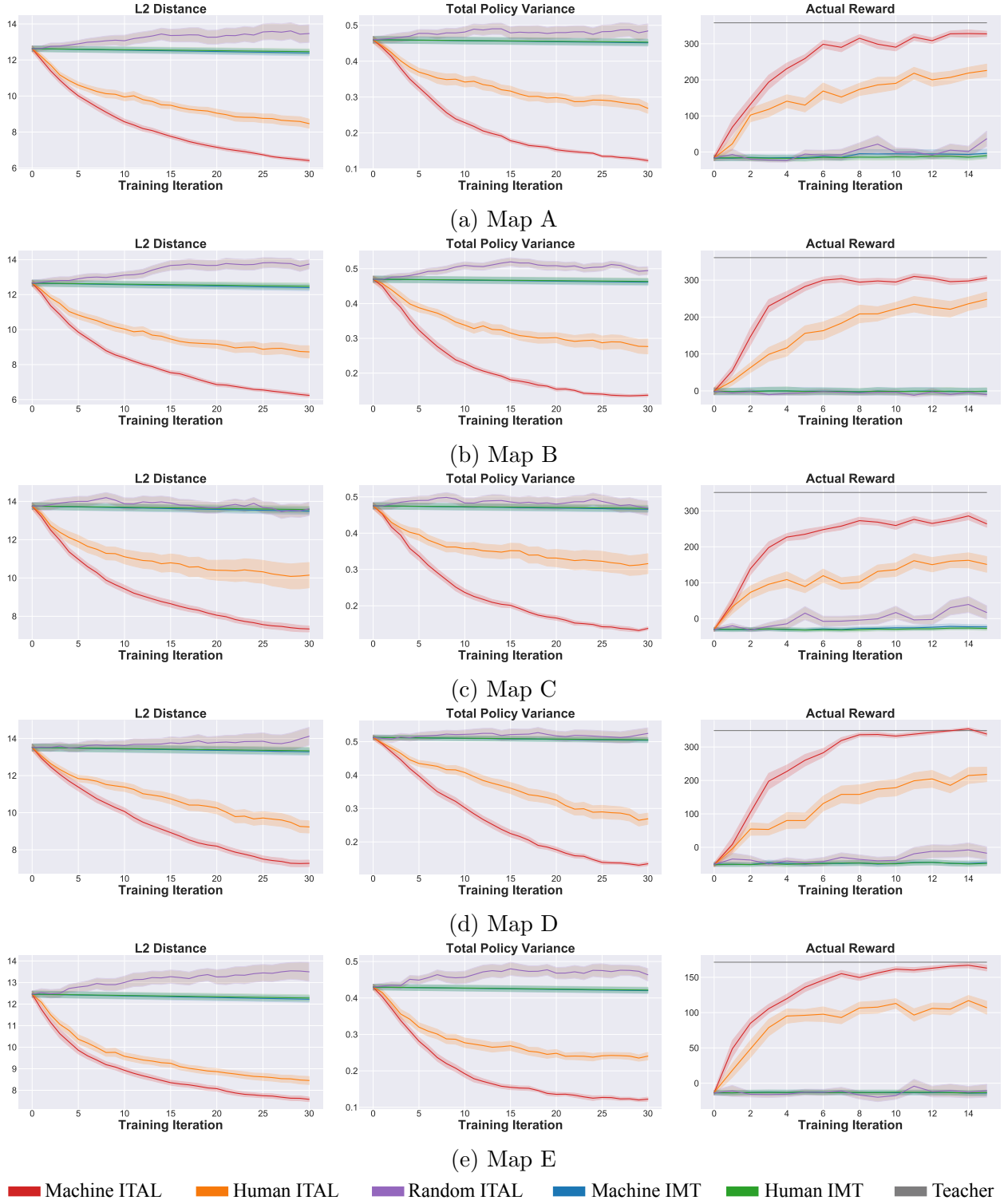


Figure 3.9: Learning curves for each map.

counterparts. The human ITALs all perform worse than machine ITALs. This is as expected as we directly reuse the machine teacher model to simulate humans. There is no guarantee that all the participants follow the same teaching pattern as the machine teacher, or even have a consistent teaching pattern at all. Yet, we still manage to grasp human cooperation to some extent. To illustrate the influence of the teacher model, we also teach the ITAL learner with a random teacher, who samples the example uniformly every time and is not cooperative at all. As shown in Fig. 3.1g, this combination doesn't benefit the learner, because the mismatch between the imagined cooperative teacher and the actual random teacher will very likely introduce over-interpretation of the examples. To summary, these results justify that human teachers do have cooperative (contrary to uniform) pedagogy patterns and the current teacher-aware model can take advantage of them. Finding a comprehensive and accurate human-robot communication model will be an open question for future works.

3.4 Discussion and Conclusion

Pedagogy has a profound cognitive science background, but it hasn't received much attention in machine learning works until recently. The learning algorithm in this chapter adopts the CL formalism to integrate pedagogy with parameter learning. Our algorithm changes the model update step for the gradient learner to accommodate the intention of the teacher. We provide theoretical and empirical evidence to justify the advantage of the teacher-aware learner over the naive learner.

To be aware of the teacher, the learner needs an accurate estimation of the teaching model. In many cases, such a model is not directly accessible, *e.g.* when there is a human-in-the-loop. In this chapter, we model the teacher in a heuristic manner. Our human study proved the generality of this model, especially when the learner only assumes a suboptimal teacher with Boltzmann rationality. In realistic settings, a more advanced teacher model should be investigated, acquired through task-specific data and/or interactions between the

agents. In the next chapter, we'll explore the emergence of learning protocols between agents interacting with the CL formalism.

CHAPTER 4

Emergence of Learning Protocols

In this chapter, we use a proof-of-concept referential game as an example to demonstrate the usage of the CL formalism. Despite its simplicity, the referential game encompasses the necessary components of standard communication between collaborative agents. It has been shown that effective communication protocols could emerge between agents playing various forms of referential games [LPB17, LHT18], but, in this example, we illustrate how the CL formalism can facilitate the emergence of a more efficient protocol, which we termed as the pragmatic protocol. Also, we show how the belief-update and value functions defined in Section 2.3 can be learned via cooperative interactions supervised with the outcomes of referential games.

4.1 Background: Referential Game

There is a teacher and a student in a referential game. The teacher has a target in mind and aims to send a message to the student so that the student can identify the target out of a set of distractors after receiving this message. The motivation of our algorithm is the rational speech act (RSA) model between ToM agents (also termed as bilateral optimality [Blu00]): in order to establish proper communication, the speaker has to take into account the perspective of the listener, while the listener has to take into account the perspective of the teacher [GF16]. Fig. 4.1a shows an example. There are three objects, a blue sphere, a red sphere, and a blue cone. Suppose the target is the blue sphere. If the only allowed messages are colors and shapes, then, for a literal student, there is no unique identifier for the

blue sphere, because both “blue” and “sphere” have more than one consistent candidates. Nonetheless, a pragmatic student, after hearing “blue” from the teacher, should be able to do counterfactual reasoning and identify the blue sphere instead of the blue cone, because he knows the teacher is helpful and would have used cone to refer to the blue cone unambiguously. During pragmatic communication, a message conveys more information than the message itself. The usage of that message can usually suggest the intention of the teacher.

The referential game can be formally defined by a tuple $\langle A, B, \Omega, \mathcal{M}, \mathcal{A} \rangle$, where A and B stand for a teacher and a student. Ω is the instance space, where the distractors and targets are sampled from. $\mathcal{M}$ is the message space and $\mathcal{A}$ is the student’s action space. In a specific game, a set of instances $O \subseteq \Omega$ is sampled from Ω as candidates, and one of the candidates $o^* \in O$ is designated as the target, while the rest, $O/\{o^*\}$, are distractors. The candidates O are available to both of the agents, while only the teacher knows the target, o^* . Agents take turns in this game. In every round, the teacher first sends a message $m_t \in \mathcal{M}$ to the student, followed by an action $a_t \in \mathcal{A} = \{1, 2, \dots, |O|, \Xi\}$ taken by the student, where number 1 to $|O|$ represent “identify a certain instance as the target”, Ξ means “wait for next message” and t stamps the t -th round. Every message comes with a message cost, c_m , and the total gain for both the teacher and the student, given a_T the first non Ξ action, is $R = \sum_{t=1}^T -c_{m_t} + \mathbf{1}(o^* = O[a_T])$. Notice that the game ends when the student performs a non Ξ action. We define a protocol between A and B as a set of policies

$$\Pi = \langle \pi_A : \mathbb{P}(\Omega) \times O \times \mathcal{M}^* \times \mathcal{M} \rightarrow [0, 1], \pi_B : \mathbb{P}(\Omega) \times \mathcal{M}^* \times \mathcal{M} \times \mathcal{A} \rightarrow [0, 1] \rangle$$

$\mathbb{P}(\Omega)$ is the power set of Ω , where O sampled from, and $*$ is the kleene star, standing for the history of message. Intuitively, the teacher selects a message based on the distractors, the target and communication history. The student chooses an action according to candidates, history and the latest message. The goal for both of the agents is to maximize the expected

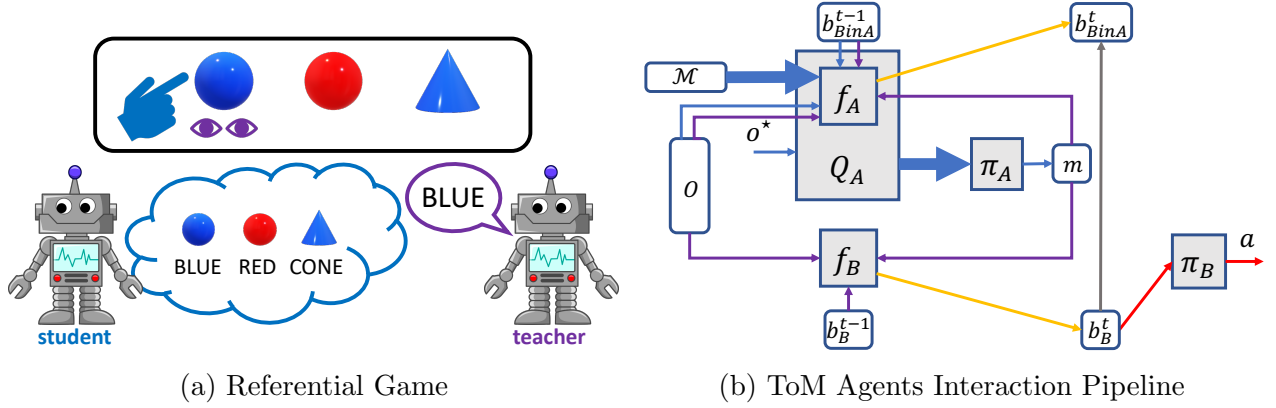


Figure 4.1: (a) An example referential game. There are three objects, a blue sphere, a red sphere, and a blue cone. If a student hears “blue” from the teacher, he should be able to identify the blue sphere instead of the blue cone. (b) ToM Agents Interaction Pipeline. First, the teacher chooses a message according to the context and her prediction of the student’s reaction (blue arrows). After a message is sent, the student updates his belief and the teacher updates her estimation of the student’s belief (purple and orange arrows). Then, the student either waits or selects a candidate (red arrows). Only in the training phase, the actual student belief will be returned to the teacher (gray arrow). Bold arrows stand for the whole message space being passed. Notice ψ_A is part of the Q_A . O and b_{BinA}^{t-1} are passed to ψ_A twice, for message selection and teacher’s new belief estimation. Empty boxes are game and time variants; shadowed boxes are agents’ mental structures. Notations are introduced in Section 4.2 with θ omitted from subscripts.

gain:

$$\mathbf{E}_{O \sim \mathbb{P}(\Omega), o^* \sim O, m_{1:T} \sim \pi_A, a_{1:T} \sim \pi_B} \left[- \sum_{t=1}^T c_{m_t} + \mathbf{1}(o^* = O[a_T]) \right] \quad (4.1)$$

4.2 Adaptive Emergence of Pragmatic Learning Protocol

Emergence of Pragmatic Protocol: Our goal is to learn a protocol for agent A and B so that they can communicate with the contextual information being considered. To avoid tracking the message history, which scales exponentially with time, we use beliefs as sufficient statistics for the past. Hence, ToM can be embodied as estimating partner’s current and future belief, then choose the most ideal action to manipulate them as needed. In the referential game, since the teacher knows the target, only the student holds a belief,

b_B , about the target. Utilizing the obverter technique [BGB18, CLF18], we let the teacher holds a belief b_{BinA} as her estimation of the student’s belief. As b_{BinA} is an estimation of the student’s belief, it should be a belief over belief, i.e. a distribution over a distribution over the candidates. Since a distribution is a continuous random variable, distribution over a continuous variable can be represented as a set of particles. However, to avoid the complexity, we only use one particle to approximate. That is, b_{BinA} is still a distribution over the candidates. This is reasonable because the belief update process is deterministic for rational agents following the Bayesian rule [VBP13, FGH17]. Given b_B^0 a uniform distribution over candidates, $P(b_B^t)$ is a single-mode distribution and can be approximated with a particle.

Before speaking, the teacher traverses all messages and predicts the student’s new belief after receiving each message. She then sends the message leading to the most optimal student’s new belief. Hearing the message, the student updates his belief and takes action. This process is visualized in Fig. 4.1b and formalized in Algorithm 14 Line 7 to Line 17. The recursive mutual modeling in ToM is integrated within the belief update process. $\psi_{\theta_i}, i \in \{A, B\}$ are belief update functions parameterized by θ_i , taking in candidates, current belief, message and returning a new belief. The beliefs in our model are semantically meaningful hidden variables in the teacher’s Q-function and student’s policy network, as the student directly samples an action according to his belief. The evolution of the belief update function reflects the protocol dynamics between the agents. Within f , we code in the Bayesian rule with the likelihood function varying across different training phases. In each phase, we first train the teacher for a fixed student, then adapt the student to the teacher.

Difference from multiagent Q-learning: Our algorithm considers both the physical state and agent’s mental state in the value function, and has a dynamic belief update function. Moreover, since agents are never trained simultaneously, our algorithm doesn’t suffer from non-stationarity [FAF16, LWT17].

Teacher: The teacher selects messages according to her Q-values and belief update function. We use $\psi_{\theta_A}(O, b, m)$ to denote the teacher’s belief update function, which takes in

the candidates set, current belief estimation and a message. The return value of this function is a new belief estimation $b' \in \Delta O$. ΔO represents all probabilistic distributions over the candidates. This function can be parameterized as a neural network with weighted candidates encoding and messages as inputs and softmax as the output layer. The return value of the belief update function is directly fed into the Q-function. In practice, we implement it as a submodule of the Q-net. That is, the output of the belief update function is used in A 's Q-function and to predict the student's belief in the next step during testing. The teacher chooses messages according to her Q-value following Eq. (4.2).

$$\pi_{\theta_A}(m|O, o^*, b) = \frac{\exp(\beta Q_{\theta_A}(O, o^*, b, m))}{\sum_{m' \in \mathcal{M}} \exp(\beta Q_{\theta_A}(O, o^*, b, m'))} \quad (4.2)$$

$$Q_{\theta_A}(O, o^*, b, m) = \mathbf{E}_{a \sim \pi_{\theta_B}(\psi_{\theta_B}(O, b, m))} \left[\mathbf{1}(O[a] = o^*) + \right. \\ \left. \mathbf{1}(a = \Xi) \max_{m'} Q_{\theta_A}(O, o^*, \psi_{\theta_B}(O, b, m), m') \right] - c_m \quad (4.3)$$

Eq. (4.3) defines the teacher's Q-function. $\mathbf{1}(O[a] = o^*)$ indicates whether the student makes a correct prediction. $\mathbf{1}(a = \Xi)$ indicates if the game is still ongoing. Student's belief b is the state of teacher's MDP. Since student's actions determine the game states, the expectation is over the student's policy. By definition, the teacher's Q-function relies on student's policy and belief update function. She has no access to these functions of the student, but since we never train the teacher and student simultaneously, the expectation can be approximated through Monte-Carlo (MC) sampling. To form a protocol, the agent A needs to learn two functions, her belief update function ψ_{θ_A} and Q_{θ_A} . In the training phase, every time the student receives a message, he returns his new belief b_B^t to the teacher. During testing, she needs to use the output of ψ_{θ_A} to approximate the student's new belief. We train ψ_{θ_A} by minimizing the cross-entropy, H , between b_B^t and teacher's prediction, denoted as L^{Obv} , the obverter loss. Teacher's Q-function is learned with Q-learning [WD92]. The λ in Line 30

controls the scale of the two losses.

Student: We directly learn the belief update function and policy of the student through the REINFORCE algorithm [Wil92]. In the referential game, student’s policy is quite simple. If his belief is certain enough, he will choose the target based on his belief; otherwise, wait for further messages. The output of the policy network is a distribution with $|O| + 1$ dimensions. The last dimension is a function of the entropy of the original belief. If the belief is uncertain, this value will be dominant after normalization. ψ_{θ_B} has the same structure as ψ_{θ_A} . ψ_{θ_B} and π_{θ_B} can be parameterized as an end-to-end trainable neural network, with the candidates encoding, original belief and received a message as the input and returning an action distribution.

Adaptive Training: The whole training process can then be summarized as Algorithm 14. Both the teacher and student are adaptively trained to maximize their expected gain defined in Eq. (4.1). The training details for the teacher and the student are illustrated in Line 25-Line 31 and Line 32-Line 35 of Algorithm 14 respectively.

4.3 Results

We evaluated our algorithm with two datasets, number set and 3D objects, and played referential games with four or seven candidates. The number set is a symbolic dataset, with an instance as a set of categorical numbers. For example, $[(1, 2, 3, 9), (1, 2, 4), (2, 3), (3, 4, 5)]$ consists a referential game with four candidates. Notice that the numbers are merely symbols without numerical order. If there are four candidates, we randomly choose numbers from 0 to 9, with maximum four numbers in a set; if seven candidates, we choose from 0 to 11, with maximum five numbers in a set. Each set is encoded by multi-hot encoding. There are 385 and 1585 different possible number sets, consisting up to 9.0×10^9 and 4.9×10^{18} different games with four and seven candidates. Number sets make a generic referential game prototype, where each instance can be disentangled into independent attributes perfectly.

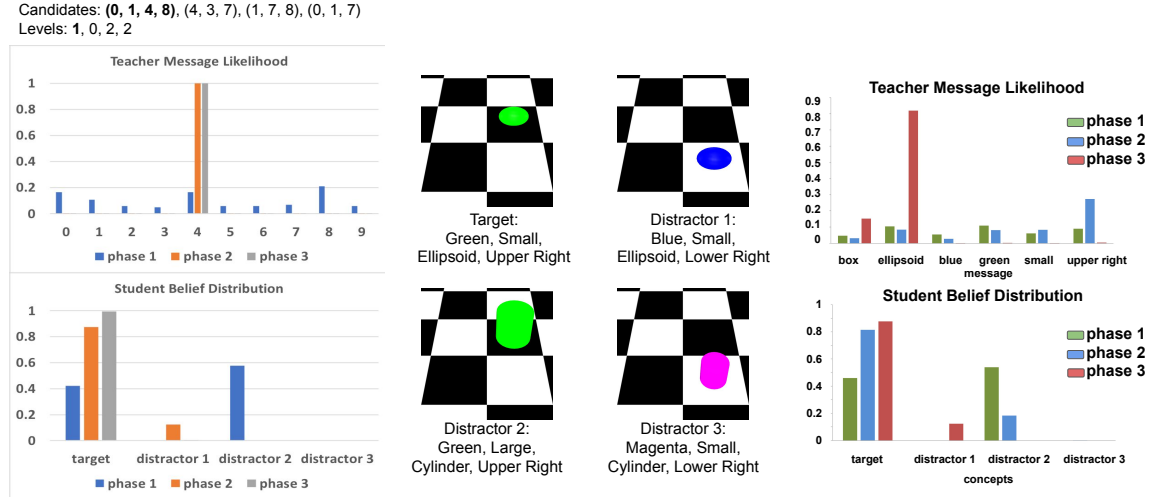


Figure 4.2: 4 distractors referential game example. Number set on the left (candidates listed in the title with the target in bold fonts) and 3D objects on the right. Due to the space limit, we only show the message distribution for the target and student’s new belief after receiving the most probable message. As for the teacher’s message distribution for distractors, all probability weights concentrate on the unique identifiers after the first phase of training. Student’s belief illustrates that teacher’s most probable message, though consistent with multiple candidates, can successfully indicate the target with more confidence as training goes. In general, both agents’ behavior becomes more certain, and the certainty coordinates.

To verify the generality of our algorithm on more complicated candidates, we used MoJoCo physical engine to synthesize RGB images of resolution 128×128 depicting single 3D object scenes. For each object, we pick one of six colors (blue, red, yellow, green, cyan, magenta), six shapes (box, sphere, cylinder, pyramid, cone, ellipsoid), two sizes and four locations, resulting in 288 combinations. In every game, candidates are uniformly sampled from the instances space. We use a message space with the same size as the number of attributes appeared in the dataset, i.e., 10 or 12 for number set, and 18 for 3D objects. In every game, we only allow one round of communication with one message. To prevent collusion using trivial position indicators, candidates are presented to the agents in different orders.

4.3.1 Referential Game with Symbolic and Pixel Input

For number set, we encode candidates with multi-hot vectors and messages with one-hot vectors. For 3D objects, we used a CNN to extract features of the candidates. For all

datasets, we manually generated 600k and 100k mutually exclusive games for training and testing. Namely, the same instances can appear in both datasets, but not any identical candidates combinations. To win a game, the same instance needed to be handled differently given different contexts, so, as long as the games are exclusive between training and testing sets, sharing instances won't cause over-fitting. To test the robustness, we also report results using exclusive testing instances in Table 4.1. We compared the pragmatic protocol developed using our algorithm against previous works on referential game [LPB17, LHT18]. Both utilized RL to train a protocol, but neither modeled recursive mind reasoning between agents. In [LPB17], only the teacher considers the context, while no context is included in [LHT18]. We trained our model for 3 phases, with 20k iterations for each phase and switch the training agent in the middle of every phase. Both benchmarks were trained for 100k iterations. Since there is no official code released by the authors, we implemented their model by ourselves and did thorough hyper-parameter grid search. Results shown in Table 4.1. Our experimental results in all settings are significantly better than both of the benchmarks. Using the paired T-test, the one-tail p -value is smaller than 0.001 for all settings in Table 4.1. We found that even with simpler representation, number set games are more difficult than 3D objects, because we don't have any limitations generating the instances in number sets. 3D objects, on the other hand, form special cases of number sets, as some attributes can never coexist. *E.g.* a shape cannot be a sphere and a cone simultaneously.

4.3.2 Connection with RTD

Definition 1. Define the teaching hierarchy for $\mathcal{C}$ as the sequence $((\mathcal{C}_1, d_1), \dots, (\mathcal{C}_h, d_h))$ such that for all $j \in \{1, \dots, h\}$, $\mathcal{C}_j = \{c \in \bar{\mathcal{C}}_j | d_j = TD(c, \bar{\mathcal{C}}_j) \leq TD(c', \bar{\mathcal{C}}_j), \forall c' \in \bar{\mathcal{C}}_j\}$ where $\bar{\mathcal{C}}_1 = \mathcal{C}$ and for all $i \in \{1, \dots, h-1\}$, $\bar{\mathcal{C}}_{i+1} = \mathcal{C} \setminus (\mathcal{C}_1 \cup \dots \cup \mathcal{C}_i)$. RTD of c in $\mathcal{C}$ is defined as $RTD(c, \mathcal{C}) = d_j$. RTD of $\mathcal{C}$ is $RTD(\mathcal{C}) = \max\{d_j | 1 \leq j \leq h\}$.

The iterative adaptive idea of our algorithm is similar to the definition of RTD, which

measures the number of examples needed for concept learning between a pair of cooperative and rational agents [CCT16]. We included the formal definition of RTD in Definition 1. Intuitively, in a concept class, there is a subset of concepts which are the simplest to learn i.e. has the minimum sized teaching set among all concepts. One can first learn those concepts and remove them from the concept class. Now, for the remaining set of concepts, one can recursively learn the simplest concepts and so on. The teaching complexity of this learning schema lower bounds classic teaching dimension [DFS14]. In every phase of our iterative training, the agent learns to identify the optimal teaching set for the “simplest” remaining candidates. In our experiments, candidates identifiable with a unique message are the simplest. If a candidate becomes the simplest after k times of removal, then we call it a level k candidate. To better illustrate the connection, we show two example referential games in Fig. 4.2 and the accuracy improvement after each phase of training in Fig. 4.3. We can see from Fig. 4.3 that after one phase of training all level 0 targets can be perfectly identified. Thus, the student, if shown the four 3D objects in Fig. 4.2, will know that the teacher will send “Blue” for distractor 1, “Large” for distractor 2 and “magenta” for distractor 3. Hence, “Upper Right” and “Ellipsoid”, though consistent with multiple objects, must indicate the target. The accuracy for higher-level targets in Fig. 4.3 keeps increasing as they become uniquely identifiable after lower-level targets are pruned out. We can observe the emergence of pragmatics from these results. From the student’s perspective, the messages from the teacher are no longer merely comprehended by their literal meanings, and from the teacher’s perspective, she selects the most helpful message to teach. In the 4-candidate scenario, most questions with level 0 and 1 are correctly answered, a similar capability shown in human one-shot referential game study [BGL12].

Notice that RTD is derived under the assumption that both agents only decipher the candidates as sets of discrete attributes. To eliminate the usage of other hidden patterns in the candidates, we need to pretrain the agents to ground messages to instance’s attributes. This can be easily achieved by initializing agents’ belief update function as Bayesian belief

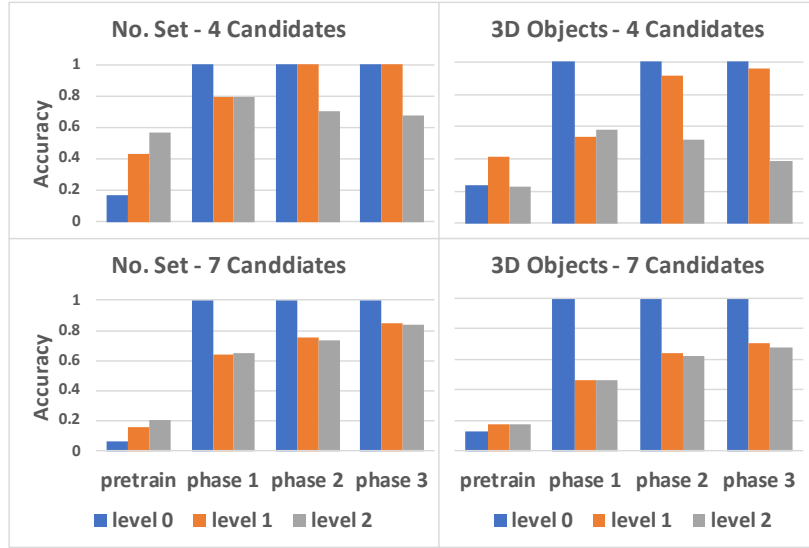


Figure 4.3: Finishing one phase of adaptation, agents can correctly identify level 0 targets. Namely, the teacher always sends a unique identifier as long as there is one. After phase 2, the student knows how the teacher will teach level 0 targets. Thus, he can prune out all level 0 candidates if he doesn't receive their unique identifiers, leaving level 1 candidates the "simplest". The decrease of level 2 accuracy among 4 candidates might be caused by the lack of level 3 targets. Only 3% of the number set and 0.02% 3D Objects games have level 2 targets, so agents have no motivation to explore for a higher level of reasoning.

update. That is, before running Algorithm 14, we train the belief neural network f with cross-entropy loss between the generated new belief and ground truth Bayesian belief. Afterward, every message grounds to an attribute.

Bayesian pretraining provides human decipherable examples for failure cases. Most of our model’s mistakes are on targets with $RTD > 1$ or $RTD = 1$ but requiring high (≥ 2) level unique identifiers (hard games). Since we only allow one message, targets with $RTD > 1$ are theoretically impossible to be certainly identified, even between cooperative agents with ToM. As for the hard games, their relatively low frequency in the training set may impinge the acquiring of high-level best response. Failure to handle a certain type of scenarios is a common empirical defect of the current RL algorithms, in our case, the hard games. Another benefit of Bayesian pretrain is that the initial message grounding is decipherable to human. In the next section, we show this interpretability can be preserved by our algorithm.

4.3.3 Stability of the Protocol

We also explored how much the communication changes after the mutual adaptation process. Suppose the agents are initialized with human-understandable message groundings, ideally, we want the emerged protocol preserves its human interpretability. To test this property, we give the teacher and student different but equivalent candidates. Namely, we randomly generate a one-to-one mapping from attributes to attributes, such as replacing all 1 with 7 in number sets or replacing all red with blue in 3D objects. After the teacher sends a message to the student, both the message and candidates are converted with the same mapping before presented to the student. The converted candidates form an equivalent game for the original one. For example, if the teacher sees $[(0, 1, 5, 8), (0, 4, 6), (4, 8, 9), (0, 1, 4, 5)]$ and sends 4. Converted by a mapping which adds 1 to all attributes, the student gets $[(1, 2, 6, 9), (1, 5, 7), (5, 9, 0), (1, 2, 5, 6)]$ and 5.

We pretrain agents with Bayesian belief update separately, then train them together without equivalent games but test them with equivalent games. See Table 4.2 for the results. We

can see that our algorithm preserves human interpretability the most. The iterative adaptation contributes to stability because, in every phase, one agent is fixed while optimizing the other. Hence, the evolving of the protocol is not arbitrary and will maintain the effective part of the existing protocol while improving the rest. This property can facilitate human-robot communication, as we only need to provide natural language grounding to robots, and they can self-evolve to take the best advantage of this grounding without developing human undecipherable protocols.

4.3.4 Global Mapping and Local Selection

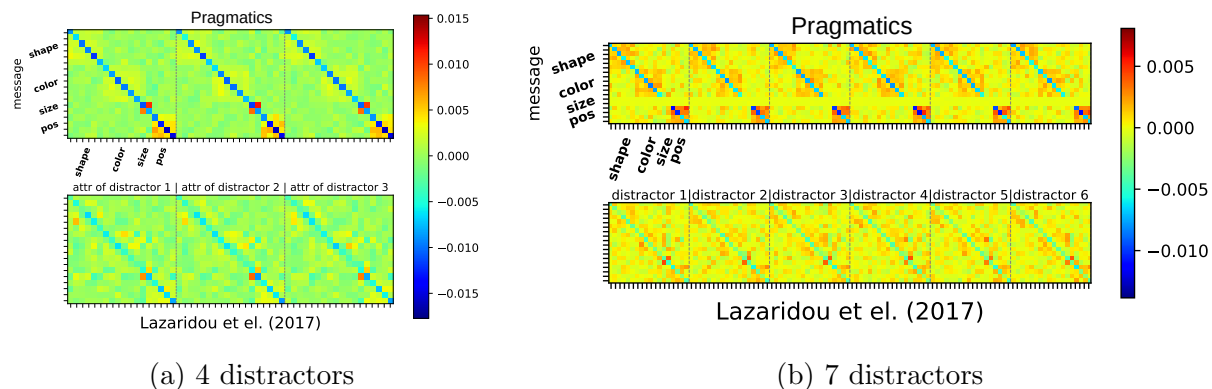


Figure 4.4: Covariance between messages and distractor attributes given the same target but different distractors. In both experiments, our algorithm has a more clear pattern. The blue diagonals show that if an attribute appears among distractors the teacher tends to avoid the corresponding message. The brighter sub-squares illustrate that messages are significantly influenced by distractors’ attributes in the same category. Namely, color messages have stronger covariance with color attributes than with shapes, sizes or positions. Notice that size messages are seldom used in 7-distractor game, thus their covariance with distractors is close to 0. Better be viewed in high resolution, *e.g.* using Adobe Acrobat Reader.

The core of pragmatics is the consideration of the context while comprehending the language. In this experiment, we want to show that the teacher using a pragmatic protocol can learn global mappings from instances to messages and select messages dynamically according to the context. First, we test global mapping by checking if messages used by the teacher for a target are consistent with this target’s attributes. Then we evaluate message selection

through seeing whether the same target yields different messages given different distractors. To make the message grounding easier to understand, we still pretrain agents with Bayesian belief update. We took 80% of the instances to generate training data. In the testing phase, the rest 20% of images are used as targets, with 3/6 distractors randomly selected from all images. Results in Table 4.3 justify that the pragmatic protocols achieve the best balance between message validity and referential accuracy. Then we calculated covariance between messages and distractors attributes, given the same target but changing distractors. For a target, the covariance is calculated using 100 games. In Fig. 4.4, we visualized mean covariance for 58 targets. Since the teacher doesn’t have access to distractors in [LHT18], we only compared our model and [LPB17].

4.4 Discussion and Conclusion

The emerged protocol demonstrates pragmatic reasoning, which brings significantly better referring performance than other emerged protocols. In Fig. 4.2, we show some example referential games together with the teacher’s and the student’s beliefs during training. It is clear that the student can differentiate targets from distractors, even when messages from the teacher have consistent literal meanings with all of them. That is, both the literal and the pragmatic meanings of messages are correctly grasped. Mathematically, the emerged pragmatic protocol approximating the RTD of the candidate concept space, which measures the number of examples needed for concept learning between a pair of cooperative and rational agents [CCT16, YFS19].

In a nutshell, the referential game example illustrates how the CL formalism can be used to develop new learning protocols from scratch with only signals about communication outcomes. Next, we give an exemplar usage of CL in a more realistic and more complicated setting than the referential games to mark the generality and scalability of the formalism.

Algorithm 14: Iterative Adaption Protocol Emergence

Input: Number of candidates K , Learning rate η , Batch size N

```
1 Randomly initialize  $\theta_A, \theta'_A, \theta_B$ 
2 for each phase do
3   for  $i \in \{A, B\}$  do
4     Initialize replay buffer  $\mathcal{D} \leftarrow \emptyset$ 
5     while train agent  $i$  do
6        $t = 1$ , Initialize  $\mathcal{E} \leftarrow \emptyset$ 
7       repeat
8         if  $t = 1$  then
9           Sample  $O = \{\omega_1, \dots, \omega_K\}$ 
10          Random select  $o^\star = \omega_j$ 
11          Initialize  $b_B^0, b_{BinA}^0$  as uniform distribution
12        end
13         $m_t \sim \pi_{\theta_A}(\cdot | O, o^\star, b_{BinA}^{t-1})$ 
14         $b_B^t = \psi_{\theta_B}(O, b_B^{t-1}, m_t)$ 
15         $a_t \sim \pi_{\theta_B}(b_B^t)$ 
16         $r_t = -c_{m_t} + \mathbf{1}(O[a_t] = o^\star)$ 
17         $b_{BinA}^t = b_B^t$ 
18        if  $i = A$  then
19           $\mathcal{D} \leftarrow \mathcal{D} \cup \{(O, o^\star, b_{BinA}^{t-1}, m_t, b_B^t, r)\}$ 
20        else
21           $\mathcal{E} \leftarrow \mathcal{E} \cup \{(O, b_B^{t-1}, m_t, a_t, r)\}$ 
22        end
23         $t \leftarrow t + 1$ 
24      until  $a_t \neq \Xi$ ;
25      if  $i = A$  then
26        Sample  $\{(O, o^\star, b_{BinA}^{t-1}, m_t, b_{BinA}^t, r)\}_N \sim \mathcal{D}$ 
27         $\xi = r + \gamma \arg \max_m Q_{\theta'_A}(O, o^\star, b_{BinA}^t, m)$ 
28         $L^Q = \frac{1}{N} \sum_N \|\xi - Q_{\theta_A}(O, o^\star, b_{BinA}^{t-1}, m_t)\|^2$ 
29         $L^{Obv} = \frac{1}{N} \sum_N H(b_{BinA}^t, \psi_{\theta_A}(O, b_{BinA}^{t-1}, m_t))$ 
30         $\theta_A \leftarrow \theta_A - \eta \nabla_{\theta_A}(L^Q + \lambda L^{Obv})$ 
31        Update  $\theta'_A \leftarrow \theta_A$  periodically
32      else
33        Compute  $R_t = \sum_{k=t}^{|\mathcal{E}|} \gamma^{k-t} r_k$  for  $t$  in  $1, \dots, |\mathcal{E}|$ 
34         $J = \frac{1}{|\mathcal{E}|} \sum_{t=1}^{|\mathcal{E}|} \log \pi_{\theta_B}(a_t | \psi_{\theta_B}(O, b_B^{t-1}, m_t)) R_t$ 
35         $\theta_B \leftarrow \theta_B + \eta \nabla_{\theta_B} J$ 
36      end
37    end
38  end
39 end
```

Table 4.1: The **first three rows** are the total accuracy, while the **second three rows** measure the accuracy for difficult games (numbers in percentage). We define game difficulty with average cosine similarity between the target and the distractors. The larger the cosine similarity, the harder the game is. We report the accuracy for top 10% hard games. L.A. stands for Lazaridou et al. To verify the generalization of our algorithm, we form training sets with 70% of all instances and test using the rest 30% unseen instances. The performance doesn’t exhibit noticeable degradation: $99.1 \pm 0.1\%$ and $93.0 \pm 1.7\%$ for 4 and 7 candidates number set respectively (comparable with the No. Set results in the third line). Mean and std calculated using 3 different random splits, 2 experiments per split.

No. Set	3D objects	$\leftarrow 4 \mid 7 \rightarrow$	No. Set	3D Objects
79.1 ± 3.3	86.9 ± 4.1	[LHT18]	64.2 ± 6.0	77.3 ± 2.8
96.8 ± 0.2	97.0 ± 0.5	[LPB17]	80.8 ± 3.1	88.2 ± 1.7
98.9 ± 0.1	99.6 ± 0.2	Pragmatics	93.2 ± 1.0	97.4 ± 1.2
79.3 ± 3.1	86.9 ± 4.5	[LHT18]	67.2 ± 5.8	77.2 ± 2.6
91.5 ± 0.4	88.0 ± 1.9	[LPB17]	66.2 ± 3.1	68.2 ± 2.7
98.1 ± 0.3	98.8 ± 0.3	Pragmatics	88.3 ± 0.6	94.1 ± 2.3

Table 4.2: All models are pretrained with Bayesian belief update. The three rows above are tested with agents seeing different but equivalent candidates, while the three rows below are tested with agents sees identical candidates. The larger the performance difference is, the more the protocol diverges from the initial grounding.

No. Set	3D Objects	$\leftarrow 4 \mid 7 \rightarrow$	No. Set	3D objects
59.5 ± 0.4	53.3 ± 1.8	[LHT18]	35.0 ± 0.2	37.7 ± 1.8
52.3 ± 2.9	60.8 ± 2.9	[LPB17]	33.3 ± 2.0	46.0 ± 1.4
97.5 ± 0.4	98.7 ± 0.5	Pragmatics	73.9 ± 0.3	84.2 ± 0.1
81.5 ± 2.6	83.7 ± 3.4	[LHT18]	74.9 ± 0.5	75.3 ± 2.9
98.6 ± 1.0	98.1 ± 0.2	[LPB17]	93.0 ± 0.5	91.2 ± 0.5
98.1 ± 0.3	99.7 ± 0.3	Pragmatics	88.2 ± 0.8	91.2 ± 0.1

Table 4.3: A message is valid if it corresponds to one of the attributes of a novel target. Std calculated with 3 different random splits.

	Valid %		Accuracy	
	4	7	4	7
Pragmatics	98.8 ± 0.29	98.1 ± 0.36	98.6 ± 0.08	87.9 ± 0.61
L.A. [2017]	74.5 ± 5.69	79.1 ± 4.73	97.0 ± 0.02	88.8 ± 0.25

CHAPTER 5

Application of Communicative Learning in Human-machine Interaction

The overarching purpose of CL is to fulfill human-like learning ability and, thus, achieve generic human-machine teaming. The machine would adopt the human user’s input and change its behavior in real-time so that the system and the human user would cooperatively accomplish a common task. To do so, the machine needs to actively infer the human user’s belief, desire, and goal [GGZ20, YLF20]. This inference process can be naturally modeled as a learning problem and fit into the CL formalism. In this section, we introduce a human-machine collaborative game, in which a team of machine scouts needs to constantly align their value to the human commander’s value to finish certain tasks. We then demonstrate how the CL formalism can be applied to this scenario. As one will see, the success of the game completely relies on effective communications between the commander and the scouts, enabled by representations in Section 2.1 and modeling in Section 2.3.

5.1 Communication Paradigm for Generic Human-robot Teaming

To achieve generic human-robot teaming, robots must be able to adopt their users’ values and change their behaviors in *real-time* so that human-machine teams can cooperatively achieve a set of common goals. To grasp the user’s messages instantaneously, conventional data-driven machine learning approaches need to be replaced by CL within a cooperative team. Such a cooperation-oriented human-machine teaming would require the machine to

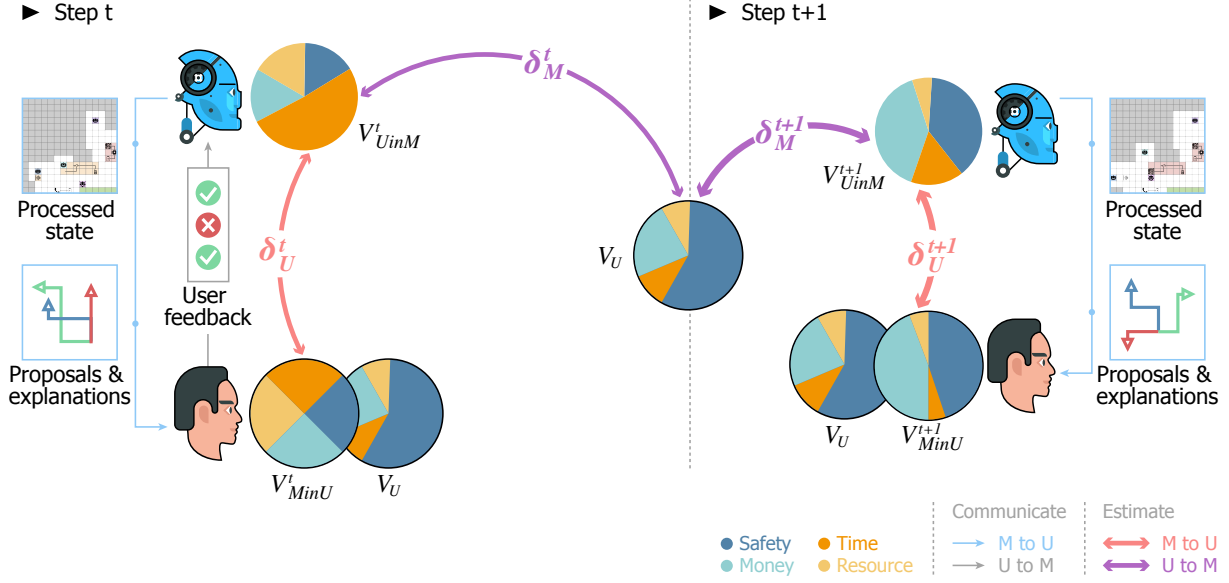


Figure 5.1: **Overview of bidirectional human-robot value alignment.** Pie charts represent the values, *i.e.*, the importance of different goals in a collaboration task, such as simultaneously considering safety, gaining money, saving time, and reserving resources. t in the superscript represents the time step. A and B in the subscript represent “user” and “machine”, respectively. U_A is the user’s true value, U_{AinB} is the robot’s estimation of the user’s value, and U_{BinA} is the user’s estimation of the robot’s current value. δ denotes the distance between values in the task value space. In every round of interaction, the machine first receives signals from the physical environment and processes its observations to form an abstract state of the environment. Next, the machine presents the processed map together with movement proposals and explanations to human users, who will provide feedback to the system accepting/rejecting the proposals according to human values and current map state. Given the user’s feedback, the machine then updates its estimation of human values and takes actions w.r.t. the new values. Cooperative human-robot communication with appropriate explanation aligns the team values in two directions by diminishing the distance between U_{AinB} and U_A , as well as U_{BinA} and U_{AinB} , resulting in final convergence to the true value U_A .

possess a certain level of ToM: A machine would *actively* infer the user’s beliefs, desires, goals [GGZ20, YLF20], and human’s need for cooperation, therefore forming a *human-centric* and *human compatible* process [Rus19]. As the example in Section 1.2.2 suggests, the essence of establishing such cooperation lies in *shared agency* [TSZ20, SZZ20] or *common mind* [Tom10, BCC18, FQZ21].

5.1.0.1 Prototype setting for generic human-robot teaming

We devised a human-machine teaming system presented as a collaborative game, in which the human user needs to work together with a group of robot scouts to accomplish some tasks and optimize the group gain. In this game, the human user and robot scouts communicate on a constrained channel: Only the robot team directly interacts with the physical world; the human user does not directly access the physical world or direct control over robot scouts' behavior. Meanwhile, only the human user has access to the ground-truth value function of the task (*e.g.*, to minimize overall time); the robot team has to infer this value function through human-machine interaction. Such a setting realistically mimics real-world human-machine teaming tasks, as many systems perform autonomously in dangerous settings under human users' supervision.

To complete a game successfully, robots are expected to accomplish bidirectional alignment by both “listening” and “speaking” wisely. First, robots need to extract useful information from human feedback to infer the user's values and adjust their policies accordingly. Second, robots are required to effectively explain what they have done and plan to do based on their current value inference, so that the user knows whether the team shares the human values. In the meantime, the human user is tasked to command robot scouts to reach the destination while maximizing the team's score. Hence, the human user's evaluation of the robots is also a *bidirectional* process: The human user has to infer the goal of robot scouts, check if it aligns with the given value function of the task and choose proper instructions to adjust their goals if not aligned. Ultimately, if the system works well, the robot scouts' value function should align well with the ground-truth value function given only to the human user, and the human user should gain high trust from the system. Fig. 5.1 illustrates the bidirectional value alignment process in the game. There are three values in the interactive process:

- U_A : the user's true value.

- U_{AinB} : the robot’s estimation of the user’s value. In this game, the scouts don’t have their own value, so they will act according to U_{AinB} .
- U_{BinA} : the user’s estimation of the robot’s value. This is the ToM structure held by the user, essential for feedback and trust formation.

Among these three values, two alignments take place:

- $U_{BinA} \rightarrow U_A$: the robots learn the user’s value from feedback.
- $U_{AinB} \rightarrow U_{BinA}$: the user learns the robots’ value from explanation.

Eventually, three values will converge to U_A , at which moment will the human-machine team forms mutual trust and effective collaboration.

Our design encourages natural human-machine teaming and bidirectional reasoning, because both parties have crucial but private information at the beginning of the game. The robot scouts possess information about the map but lack access to the human user’s value function, which determines mission goals, hindering the robot scouts’ ability to make proper decisions that reflect the human user’s intent. Meanwhile, the human user, who knows the task’s value function that governs the decision-making process, lacks direct access to the environment. By allowing constrained communication to fulfill human-machine collaboration, the robot scouts can make sporadic action proposals to the human user, and the human user provides a binary accept/reject feedback, which the robot scouts will then utilize to infer the human user’s value function and adjust robot scouts’ behaviors accordingly.

Viewing this task from the perspective of CL framework, we have the human user as the teacher A and the robot scouts team as the student B . The goal is for the robots’ utility function to align with the user’s and the user trusts in the robots, *i.e.* $u_A = u_B$ and $u_{BinA} = u_A$. As the robot team shares the environment information with the user in real-time, we assume their state beliefs are identical (so do their belief over state beliefs). This game doesn’t entail model learning within the human-robot team, but the same algorithm

for value alignment can be easily applied to model alignment with minor adjustments. m^t are robots’ proposals and accompanied explanations, while d^t are user’s feedback (acceptance or rejection) to the robot team. For a fast alignment, the students need to know when and how to make proposals, such that feedback from the teacher is most informative to estimate the value function correctly. The feedback directly changes the belief of the robots. To obtain instructive feedback from the human teacher, the robots must establish a shared agency or common mind—what the human user knows and believes, what the human user intends to do, and what are aligned and misaligned. Only based on this shared agency could the robot scouts provide explanations that properly justify previous actions and current proposals, and influence the BoB of the human user properly.

5.2 Collaborative Game Setup

Our collaborative game has a minimal design and involves one human commander and three robot scouts. The game’s objective is to find a safe path on an unknown map from the base (located in the bottom right corner of the map) to the destination (located in the upper left corner of the map). The map is represented as a partially observed 20×20 tile board, with each tile potentially holding one of the various devices and remaining unobserved until a robot scout moves closer to it.

We define a set of goals for the robot scouts to pursue as they find the path to reach the destination, including (i) saving time used to reach the destination, (ii) investigating suspicious devices on the map, and (iii) exploring tiles, and (iv) collecting resources. The game’s performance is measured by the accomplishment of these goals by the robot scouts and their relative importance (weights), defined as the human user’s value function. Again, this value function is only revealed to the human user, not the robot scouts. See Fig. 5.2 for a snapshot of the game.

The game’s performance is measured by these factors, *i.e.* the accomplishment of these

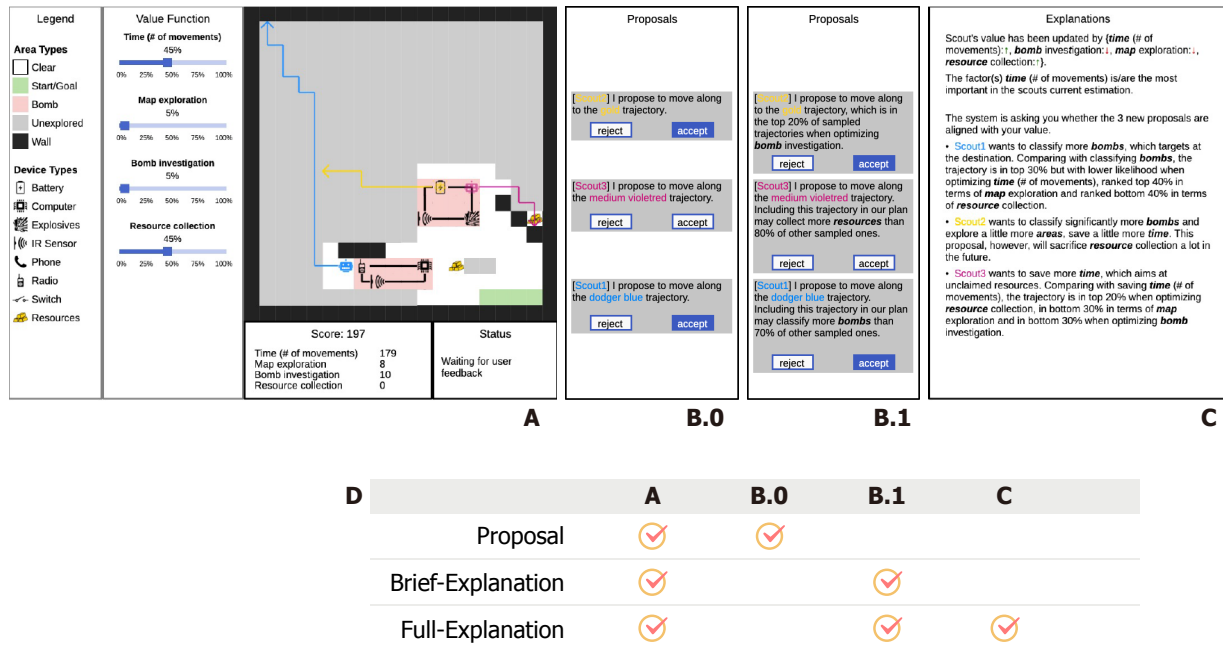


Figure 5.2: **User interface for the scout exploration game.** (A) From left to right: Legend panel in the first column explains the meaning of various icons used in the game. The value function panel in the second column shows the true values indicating the relative importance of various goals; the values are unknown to the robot scouts and cannot be modified by the user. The four right panels change dynamically over the course of the game. The central panel in the third column shows the current status of the map in the game. The score panel at the bottom shows the current scores for achieving individual goals. The overall score is the sum of the scores for individual goals, weighted by the values known to human users in the value function panel. The Status panel provides a text summary of the current status of the robot system. (B) The Proposal panel shows the robot scouts' current proposals; human users can accept or reject proposals of individual scouts. In the proposal-only group, participants only see a descriptive sentence for each proposal (B.0), whereas, in the brief-explanation and full-explanation groups, participants are presented with a brief explanation about the proposal's purpose (B.1). (C) The Explanation panel shows detailed explanations provided by the scouts, only displayed to the full-explanation group. (D) The bottom table summarizes key components of the game display included in each group.

goals by the robot scouts, and their relative importance (weights), defined as the human user's value function. Although all goals have intrinsic benefits, a trade-off among the goals has to be made according to the value function. For instance, if time is valued more in the value function than resources, the scouts should ignore some resources along the way to the

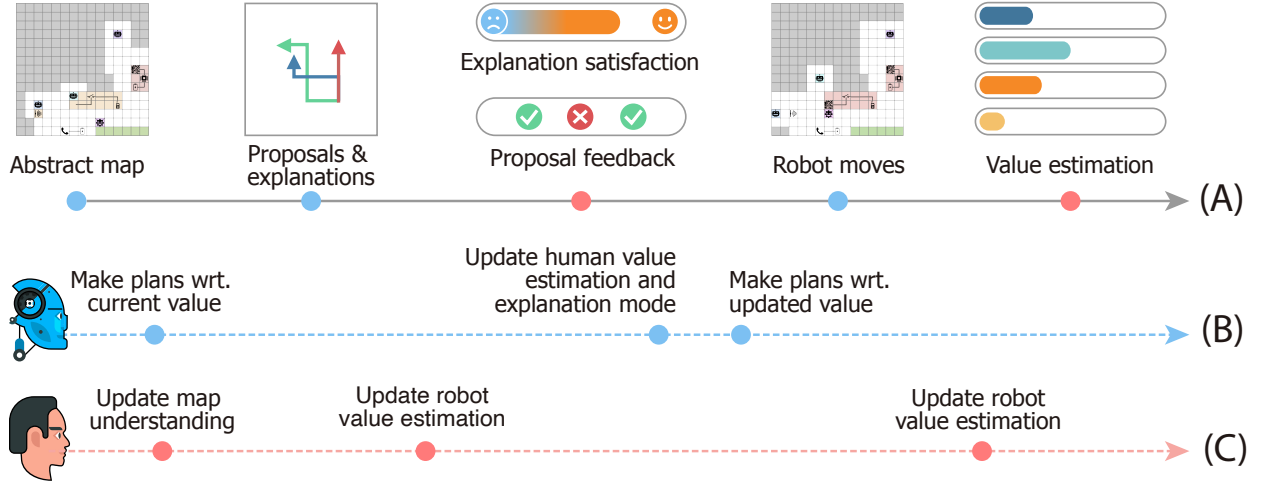


Figure 5.3: **Study design of the Scout Exploration Game.** Timeline (A) denotes events happening in a single round of the game, starting from scouts receiving environment signals and ending with their next move. Proposals and explanations are presented differently to users depending on their experimental group. The value estimation asks users to infer scouts’ value at current time. Answers to these questions will not be used by the scouts during the game, but only for inspecting users’ mental model after the game completes. Timelines (B) and (C) depict mental dynamics of the robots and the user, respectively.

destination for the sake of time. To emphasize the trade-off essence of value functions, we represent the importance of each factor with a percentage and the four percentages sum to 1. Before the interaction begins, a value function is assigned **only** to the human user as the mission for the game. Just like in the real world, various tasks can be specified with distinctive functions defined on a unified set of features [BDM17]. Without loss of generality, we coined seven value functions in this study: four of them have one dominant goal, two of them have two equally important goals, and the rest values all but resources.

We mimic a realistic scenario, where human needs can be too diverse to code in the robot beforehand and value functions can be difficult to transfer between human and machine due to different mental representations. Without knowing the value function, to complete a task, the robot scouts (as a team) must quickly infer the commander’s value. In each step, we let the robot team make three movement proposals, one for each scout, to the user, and the user can either accept or reject a proposal. To help the commander make decisions, the robot team

also explains the reason for every proposal. With the user’s feedback, conditioned on the interaction history as well as the current map status, the robot team adjusts its estimation of the human value and takes actions accordingly. Specifically, if a plan is accepted, the proposer will follow that plan as much as possible (a plan may be interrupted by unexpected blocks in the partially observable map); otherwise, the robot will execute a new plan with the updated value estimation. This concludes one round of interaction, and this process will be repeated. Of note, to avoid inefficient communication, the robot only makes proposals when necessary and acts according to the basis of the latest human value estimation. Fig. 5.3 summarizes the human-machine interaction flow.

5.3 Value Alignment with CL

To estimate the human user’s value during the communication process, we integrate two levels of ToM into our computation model. The level-1 ToM encodes the cooperative assumption. Namely, given a cooperative human user, the accepted proposals are more likely to align with the correct value function than the rejected ones. The level-2 ToM further accommodates users’ pedagogy into the model. That is, the feedback that drives robots’ value closer to the true value is preferred over other feedbacks. The pedagogical inclination requires an additional level of ToM because it demands recursive modeling of the user’s model of the robots. Combining both levels of ToM, we formalize human behavior with distributions parameterized by the value and develop a learning algorithm with a closed-form parameter update function.

5.3.1 Level-1 ToM

The robot scouts need to estimate the human user’s value from their interactions. In the collaborative game, the more a proposal facilitates goals with high values, the more it is likely to be accepted. Similar to Section 2.4, we refer to the ability to infer humans’ value

from their actions as level-1 ToM. Bearing level-1 ToM, the scouts can interpret the user's feedback and update the value estimation given the current map status. For example, if a trajectory towards a partially explored circuit is accepted, the scouts are likely to increase the value to bomb investigation and lower the other goals. We integrate level-1 ToM into our computation model and develop a learning algorithm with a closed-form parameter update function.

5.3.1.1 Belief Update with Level-1 ToM

Let $m^H(fb)$ denote the human user's feedback, which is a binary code with the i -th bit indicating the acceptance or rejection of the proposal from the i -th scout. Assuming the human user considers each proposal separately and follows a Bernoulli acceptance distribution [CNS18], the likelihood function of the human user's feedback is:

$$\begin{aligned} p(m^H(fb)|\tau; \bar{\theta}) &= \prod_{i=1}^K p(m^H(fb)_i|\tau_i; \bar{\theta}) \\ &= \prod_{i=1}^K \frac{\exp(\beta_1 \bar{\theta}^T \overline{f(\tau_i)})^{m^H(fb)_i} \exp(\beta_1 \bar{\theta}^T \overline{f(\neg\tau_i)})^{(1-m^H(fb)_i)}}{\exp(\beta_1 \bar{\theta}^T \overline{f(\tau_i)}) + \exp(\beta_1 \bar{\theta}^T \overline{f(\neg\tau_i)})}, \end{aligned} \quad (5.1)$$

where

$$\overline{f(\tau_i)} = \sum_{\tau \in \mathcal{T}: \tau_i \in \tau} \overline{f(\tau)}, \quad \text{and} \quad \overline{f(\neg\tau_i)} = \sum_{\tau \in \mathcal{T}: \tau_i \notin \tau} \overline{f(\tau)}. \quad (5.2)$$

That is, a proposal is more likely to be accepted if including it in the scouts' plan is more beneficial than excluding it when $\bar{\theta}$ is the value parameter. Given this likelihood function, we utilize MLE to learn $\bar{\theta}$ by maximizing $\log p(m^H(fb)|\tau; \bar{\theta})$ w.r.t. $\bar{\theta}$:

$$\bar{\theta} = \bar{\theta} + \eta \frac{\partial \log p(m^H(fb)|\tau; \bar{\theta})}{\partial \bar{\theta}}, \quad (5.3)$$

where η is the learning rate, and

$$\begin{aligned} \frac{\partial \log p(m^H(fb)|\tau; \bar{\theta})}{\partial \bar{\theta}} = & \beta_1 \sum_{i=1}^K \left[m^H(fb)_i \overline{f(\tau_i)} + (1 - m^H(fb)_i) \overline{f(\neg \tau_i)} \right. \\ & \left. - \mathbb{E}_{m \sim p(m^H(fb)_i|\tau_i; \bar{\theta})} [m \overline{f(\tau_i)} + (1 - m) \overline{f(\neg \tau_i)}] \right]. \end{aligned} \quad (5.4)$$

where acceptance/rejection selects the feature of including/excluding τ_i and the expectation is taken w.r.t. the feedback distribution given current $\bar{\theta}$. The expectation computes the average feature if the plan, τ , is randomly accepted/rejected according to current $\bar{\theta}$. The difference between the user’s designated feature and the expected feature forms the gradient. Since $\bar{\theta} > 0$ and $\|\bar{\theta}\|_1 = 1$, we perform MLE with the projected stochastic gradient ascent algorithm.

5.3.2 Level-2 ToM

Intuitive but limited, the comprehension of feedback endowed by level-1 ToM is constrained to its plain content, *i.e.*, the literal meaning of the feedback. In human communication, messages often convey both literal meanings and pragmatic meanings [SGF13]. In other words, one can acquire not only explicit information from what others said, but also implicit information from what others did not say. A typical concretization is the Gricean Maxims of quantity [Gri75] or the scalar implicature: When people say “I like drinking warm coffee,” though the lexical meaning of “warm” is semantically close to “hot”, they mean “not hot”; otherwise people would have said “hot” directly [Car98, VBP13]. Similarly, the human user’s selection of a certain combination of feedback but not other combinations can also help robot value alignment. To comprehend this process, it requires the robots to mentally simulate and plan based on human users’ pedagogical tendency and belief about the robots’ current plan. We refer to such a recursive inference ability as level-2 ToM.

5.3.2.1 Belief Update with Level-2 ToM

To enable level-2 ToM, robots need to conduct a recursive mental simulation in a counterfactual fashion and consider the advantage of the received feedback over others not being sent. Intuitively, suppose the user knows how the robots with level-1 ToM update the value given feedback; the more the feedback leads to changes towards the ground-truth value, the more it is likely to be selected. Computationally, the level-2 robots first simulate level-1 value update given all possible feedback. Next, the robots find a ground-truth value such that the update brought by the received feedback is better than the other alternative feedback. Mathematically, we formulate the human user providing feedback based on its anticipatory improvement following

$$q(m^H(fb)|\bar{\theta}, \tau; \theta^*) = \frac{\exp(-\beta_2 \|\bar{\theta} + \eta \frac{\partial \log p(m^H(fb)|\tau; \bar{\theta})}{\partial \theta} - \theta^*\|^2)}{\sum_{\widehat{m^H(fb)} \in FB} \exp(-\beta_2 \|\bar{\theta} + \eta \frac{\partial \log p(\widehat{m^H(fb)}|\tau; \bar{\theta})}{\partial \theta} - \theta^*\|^2)}, \quad (5.5)$$

where $\beta_2 \geq 0$ controls the extremeness of the Boltzmann rationality, η is the learning rate, and θ^* is the set of ground-truth parameters of the value function possessed by the human user. The intuition of this equation is: The feedback from the human user is sampled from a soft-min distribution of the distance between the updated parameters given the feedback and the ground-truth parameters. The smaller the distance is, the larger the improvement brought by that feedback, and the larger the improvement is, the more likely the feedback is provided. Further analysis of the above distance can be found in Liu *et al.* [LDL18]. Integrating this feedback function into our value learning algorithm, we can derive a new parameter update function:

$$\bar{\theta} = \bar{\theta} + \eta g(m^H(fb)) + 2\beta_2 \eta^2 (g(m^H(fb)) - \mathbb{E}_{m(fb) \sim q(m(fb)|\bar{\theta}, \tau; \theta^*)} [g(m(fb))]), \quad (5.6)$$

where

$$g(m(fb)) = \frac{\partial \log p(m(fb)|\tau; \bar{\theta})}{\partial \bar{\theta}}.$$

The derivation for Eq. (5.6) is as the following: Agents with level-2 ToM updates their belief by considering both the literal meaning and the pragmatic meaning of the human feedback. Hence, the agents try to maximize $p(m^H(fb)|\tau; \theta)$ and $q(m^H(fb)|\bar{\theta}, \tau; \theta)$ w.r.t. θ simultaneously. Of note, in both terms, θ denotes the user's true value function, not to be confused with the robots' value denoted by $\bar{\theta}$. $p(m^H(fb)|\tau; \theta)$ captures the literal meaning of the feedback, *i.e.*, the probability of a user with value function θ gives out $m^H(fb)$ based on how the proposal τ is compliant with θ . $q(m^H(fb)|\bar{\theta}, \tau; \theta)$ captures the pragmatic meaning of the feedback, namely the probability of a cooperative user with value θ gives out $m^H(fb)$ because it is more helpful to the scouts to align to the true value. Thus, agents with level-2 ToM maximize a new likelihood $p(m^H(fb)|\tau; \theta)q(m^H(fb)|\bar{\theta}, \tau; \theta)$ w.r.t. θ . Let $g(m) = \frac{\partial \log p(m|\tau; \theta)}{\partial \theta}$, we have

$$\begin{aligned} & \frac{\partial}{\partial \theta} \log p(m^H(fb)|\tau; \theta) q(m^H(fb)|\bar{\theta}, \tau; \theta) \\ &= g(m^H(fb)) + \frac{\partial \log q(m^H(fb)|\bar{\theta}, \tau; \theta)}{\partial \theta} \\ &= g(m^H(fb)) + 2\beta_2(\bar{\theta} + \eta g(m^H(fb)) - \theta^*) \\ & \quad - 2\beta_2 \sum_{m' \in FB} \frac{(\bar{\theta} + \eta g(m) - \theta^*) \exp(\beta_2 \|\bar{\theta} + \eta g(m) - \theta^*\|^2)}{\sum_{m' \in FB} \exp(\beta_2 \|\bar{\theta} + \eta g(m') - \theta^*\|^2)} \\ &= g(m^H(fb)) + 2\beta_2 \eta \left(g(m^H(fb)) - \mathbb{E}_{m \sim q(m|\bar{\theta}, \tau; \theta)} [g(m)] \right). \end{aligned} \tag{5.7}$$

Multiplying Eq. (5.7) with learning rate η , we have the update function in Eq. (5.6). Eq. (5.4) gives the closed-form of $g(m)$.

The first two terms in Eq. (5.6) are the same as the level-1 belief update, whereas the third term grasps the message's context by comparing the selected message against the also-

runs and leverages the advantage to further update the belief. Notice that θ^* is unknown to the agent, so q in the expectation does not have an exact solution. Thus, we use $\bar{\theta} + \eta g(m^H(fb))$ as an approximation of θ^* . That is, we calculate level-1 ToM update on the parameters of the value function and take an additional gradient ascent step for level-2 ToM update. In this work, we always initialize scouts’ value as uniform across all goals, *i.e.*, $\bar{\theta}^0 = [0.25, 0.25, 0.25, 0.25]$.

The difference between robots with level-1 ToM and level-2 ToM is the likelihood function they used to model the user. A level-1 robot assumes the user provides feedback only by thinking about how good the proposals are, whereas a level-2 robot is also aware of the pedagogical perspective of the human user in the collaborative game and accommodating the information of both the literal and the pragmatic meaning of user feedback.

Theoretically, the recursive reasoning between robots and the human user can continue infinitely with unlimited resources or up to a fixed-point of convergence [WWP20]. In this work, we only model the human user as knowing the value update mechanism of scouts with level-1 ToM, a manageable extent of reasoning for human cognitive capability [WVV17], which is also adopted by recent literature (*e.g.*, [PCD19]).

Of note, one comparable but different setting to our human-machine teaming framework is IRL [AD21]. Nevertheless, IRL aims to recover an underlying reward function given pre-recorded expert demonstrations in an offline passive learning setting. As we summarize in Table 2.1 and specify in Section 2.4.1.3, the standard IRL is a level-0 paradigm. In contrast, the robot scouts in our setting are designed to learn *interactively* from *scarce* supervisions given by the human user. Crucially, our design requires the robots to actively infer the human user’s value in **real-time** and **in-situ** as the task proceeds, a unique property of *human-centric* learning schemes falling into the category of level-2 paradigms. Furthermore, to consummate a collaboration, not only must the robot scouts quickly comprehend the human user’s intent, but also elucidate themselves to ensure smooth communication with the human user throughout the entire game. In brief, the robots are tasked to perform

value alignment by inferring the human user’s mental model, actively making proposals, and evaluating the human user’s feedback, which requires complex and recursive mind modeling of the human user.

5.4 Human Experiment Design

The human study examines whether our explainable artificial intelligence (XAI) system achieves real-time bidirectional value alignment between the human and the machine. In particular, we evaluate the efficacy of different forms of explanation of the robots’ plans to human users. We conducted a psychological study with 135 participants. Participants were randomly assigned to one of three groups: (i) a proposal-only group (ii) a brief-explanation group, and (iii) a full-explanation group, each with 45 participants. In the proposal-only group, the scouts only make proposals and give no explanations to the human. In the brief-explanation group, every proposal consists of a one-sentence brief explanation, explicating its positive outcomes. In the full-explanation group, a more detailed explanation accompanies every proposal, expounding the gains and costs of scouts’ tentative actions and the dynamics of their values for the importance of different goals. Across all three groups, the robot scouts follow the same action policy and decision process for belief updating. The three groups differ only in terms of the forms of explanations provided to the human participants. Fig. 5.2 compares the game interface that appeared in each group.

Our experimental setup consists of three phases: introduction, familiarization, and game playing. The first two phases prepare participants for the game. Game overview, rules, and user interface (UI) are explained in the introduction phase. In the familiarization phase, the system tests participants with a set of questions to validate their understanding of the game; only participants who correctly answer all the questions in the familiarization phase proceed to the game playing phase. During the game, participants are asked to accept or reject scouts’ proposals and assess satisfaction with the scouts’ communication after every

feedback. The feedback for proposals are given using buttons shown in Fig. 5.2B. The satisfaction are provided via Likert-scale questions. In addition, we also ask participants to estimate the machine’s internal states, such as the scouts’ current value function and their qualitative trust of the XAI system. The scouts’ value estimation questions are asked every 2 rounds of communication, and the trust questions are asked every 5 rounds. Note that human judgments about the value estimation and trust are not used to adjust scouts’ behavior; these additional measures are only used for evaluation purposes.

5.4.1 Experiment Details and Demographics

Human participants were recruited from University of California, San Diego (UCSD) undergraduate students taking psychology courses and the University of California, Los Angeles (UCLA) Department of Psychology subject pool. All subjects were compensated with course credit for their participation. A total of 167 students completed the introduction phase and passed the familiarization test (56, 53, 58 for the proposal-only , brief-explanation , full-explanation group, respectively). 19 subjects were removed from the analysis for failing the attention check during the game play, resulting in 148 subjects (49, 47, 52 for the proposal-only , brief-explanation , full-explanation group, respectively) considered in the final results. In Fig. 5.4, we report results after the removal of outliers that are 1.5 Interquartile Range Method (IQR) below the 25th percentile or above the 75th percentile, resulting in 135 valid subjects (45 subjects per group). Before the game starts, all subjects were assigned to one of the three explanation groups and given one of seven value functions randomly.

The experiment included three phases: introduction, familiarization, and game play. In the introduction phase, participants were presented with the context and rules of the Scout Exploration Game. Icons, scores and UI in the game were explained to the participants with both text descriptions and video demonstrations. Because subjects in different explanation groups will see different UI in the game, we guaranteed that the UI in the video demonstrations is consistent with the one in the actual game; video demonstrations for other

groups were not presented, which ensures the between-subject design. In the familiarization stage, participants were tested with multiple-choice questions about their understanding of the game flow, rules, and the UI. Participants who correctly answered all questions proceeded to game play. Participants having at least one wrong answer were asked to review the introduction and retake the familiarization test. Participants who could not pass the familiarization test twice or took more than 20 minutes before starting the game play were removed from the study. Credits were awarded to participants regardless of their familiarization test results. The computational model used for scouts' value alignment is the same across all groups to attribute the difference in the performance of bidirectional value alignment to the lack or distinction of explanations. Participants in the proposal-only , brief-explanation , and full-explanation group communicate with the scouts for 15.4, 16.4, 16.2 rounds on average, with the standard deviation of 3.2, 4.0, 3.7 rounds, respectively. The average time of game play is 20.8, 22.5, 32.3 minutes, with the standard deviation of 4.5, 6.3, 10.7 minutes for the proposal-only , brief-explanation , and full-explanation group, respectively.

5.5 Results

To measure the value alignment performance, we use the Kendall rank correlation coefficient, commonly referred to as the Kendall τ coefficient, to compare the goals' importance ranking in the target value with the ranking in the value estimation. The null hypothesis is that explanations yield the same value alignment across different groups, and therefore, no difference in the ranking statistics would be observed. The test is a two-tailed independent samples t test to compare performance from two groups of participants, because we used a between-subjects design in the study, with a commonly used significance level $\alpha = 0.05$, assuming t -distribution, and the rejection region is $P \leq 0.05$.

Fig. 5.4 illustrates the bidirectional human-robot value alignment results for all three

groups. We compute the Kendall’s τ coefficient to assess the value alignment between scouts and humans. To compare two sets of values (*e.g.*, values estimated by scouts versus true values known to human users), we first rank the task goals by their corresponding values and then calculate the Kendall’s τ coefficient between the two rankings. Perfectly agreed/disagreed rankings have $\tau = \pm 1$, and independent rankings expect $\tau \approx 0$. To demonstrate dynamic changes in bidirectional value alignment between scouts and humans, we record the scouts’ estimation of the user’s value and measure the user’s estimation of the scout’s value as the game proceeds. Since games have various lengths due to different explanation formats, group values, and individual differences in participants, we normalize game progress as a percentage calculated by dividing the number of current iterations by the total number of iterations. For all three groups, we remove subjects who fail attention checks and outliers whose alignment results are 1.5 IQR below the 25th percentile or above the 75th percentile at any game progresses.

Fig. 5.4A shows the alignment between robots’ estimated values and the true values known to human users. First, all groups show higher value alignment at the end of the game compared to the beginning of the game [paired t -test, $t_{Prop}(44) = 2.850, P_{Prop} = 0.007, t_{Brief}(44) = 10.148, P_{Brief} < 0.001, t_{Full}(44) = 11.452, P_{Full} < 0.001]$. Importantly, scouts that interacted with the brief-explanation and full-explanation groups show stronger value alignment, revealed by higher correlations between scouts’ estimated values and true values, than alignment in the proposal-only group ($\tau = 0.2, 0.4, 0.5$ for the proposal-only, brief-explanation and full-explanation groups respectively). The group differences emerge in early stages of the game (25% of the game progress), and are maintained to the end of the game, confirmed by analysis of variance (ANOVA) at a range of progress points [from game progress 25%, 50%, 75% and 100%, $F_{(2,132)} = 19.086, 14.202, 11.961, 11.622; P < 0.001, P < 0.001, P < 0.001, P < 0.001]$. Better value alignment in the two groups involving explanation than the baseline proposal-only group without explanation provides strong evidence that explanations about robot decision processes to human users enhance bidirectional communica-

tions between humans and machines. The enhanced communication, in turn, helps machines gain accurate estimates of human values, thereby fostering human-machine teaming. There is no significant difference between the brief-explanation group and full-explanation group, implying that the detail of the explanations may not critically influence humans' feedback in terms of accepting or rejecting robots' proposals, as long as these explanations provide sufficient contexts to justify the robots' intents.

Fig. 5.4B depicts how well the human users estimate the scouts' values over the progress of the game (*i.e.*, the accuracy with which humans assess the scouts' values). An ANOVA test revealed a significant main effect of groups in the later stage of the game playing at game progress 50%, 75% and 100%, [$F_{(2,132)} = 7.632, F_{(2,132)} = 8.339, F_{(2,128)} = 10.542, P = 0.001, P < 0.001, P < 0.001$ respectively]. The brief-explanation and full-explanation groups show significant enhancement of alignment between human estimates of scouts' values and scouts' values used in determining their decision and proposals at the end of the game compared to the beginning of the game [paired t -test, $t_{Brief}(44) = 3.272, P_{Brief} = 0.002, t_{Full}(39) = 2.810, P_{Full} = 0.007$], whereas the proposal-only group does not show any improvement [paired t -test, $t_{Prop}(38) = 0.286, P_{Prop} = 0.776$]. These results suggest that human users have difficulty understanding robots' intentions by only observing their situational behaviors, highlighting the central role of explanation in revealing the robots' intentions to its human user. Critically, humans show stronger alignment in estimating scout's values in the full-explanation group than in the other two groups in the second half of the game [independent sample t -test, from game progress 50%, 75% and 100%, $t(88) = 4.291, t(88) = 4.511, t(84) = 5.088, P < 0.001, P < 0.001, P < 0.001$ against proposal-only ; $t(88) = 2.387, t(88) = 2.219, t(86) = 2.196, P = 0.019, 0.030, 0.031$ against brief-explanation]. In comparison, the brief-explanation group only yields a more consistent human estimate of scouts' values than the proposal-only group at the end of the game [independent sample t -test, $t(86) = 2.274, P = 0.026$]. Taken together, these results indicate that both forms of explanation facilitate human estimates of robots' value estimates based on observation of

robots’ behavior and interactions with the robots. But the full explanation, which provides details about both the advantages and the disadvantages of a proposal, is more helpful to human judgments about the robots’ estimates than a brief explanation showing only the major benefit of a proposal.

Both results from robots’ estimate of human values in Fig. 5.4A and from human estimate of robots’ values Fig. 5.4B show that groups with brief and full explanations can maintain a stable trend of value alignments across the entire human-robot teaming process. However, the emergence of value alignment differs across different groups and varies for different alignment metrics over the course of the game. Robots’ value alignment metrics measured by the Kendall’s τ coefficient converges at 25% of the game. Alignment of human estimates and scouts’ values converges at 50% progress for the full-explanation group, and 75% game progress for the brief-explanation group. These results demonstrate our system’s capability to maintain the established team mental model during continuous human-robot teaming, with full explanations enabling faster convergence of users’ estimates of robots’ mental status (estimates of values). The convergence of both alignment metrics shows that (i) our value alignment algorithm enables the robot scouts to learn human values in an in-situ, real-time, and interactive manner, and (ii) explanations generated by the robots enable users to better perceive the machine’s values. These results demonstrate a bidirectional human-robot alignment. Moreover, our result pins down the contributions of explanation formats in different facets of human-robot communication. We found that brief and full explanations lead to similar effects in improving the way humans provide feedback to the machine via acceptance or rejection of robots’ proposals. However, the full-explanation group shows a significantly greater benefit for human accuracy in estimating robots’ values.

5.6 Discussion and Conclusion

Our proposed XAI system successfully demonstrates the feasibility of a bidirectional human-robot value alignment framework. From the listener’s perspective, robots in all three explanation groups can quickly align to the user’s value by correctly ranking at least 60% of goals’ importance as early as the 25% progress of the game. From the speaker’s perspective, by providing proper explanations, robots can reveal their intentions to the user and facilitate better human perception of the machine’s values, with convergence occurring at 50% (full-explanation) and 75% (brief-explanation) of the game. Together, both perspectives provide convincing evidence of a bidirectional process of value alignment. Specifically, (i) by receiving cooperative human feedback, robots gradually update their value function to align with the human values, and (ii) by continuously interacting with the robots, the human user gradually forms a coherent perception of the system’s capability and intentions. Although the system’s values have not converged in the first half of the game, the user’s perception of the robots’ estimate can still improve. Eventually, when the robots’ values become stable, the user’s estimation of the robots also becomes stable. The pairing of convergence from robots’ estimate of the user’s values to user’s true values, and from user’s estimate of the robots’ values to robots’ current values, forms a bidirectional value alignment anchored by the user’s true value.

Despite showing similar converging trends of value alignments, the three explanation groups differ in the precision of their alignments. In both directions of human-robot estimation (*i.e.*, scouts estimating human values and the human’s understanding of the scouts’ current value estimation), the Kendall’s τ coefficients of the proposal-only group are significantly lower than the coefficients of the other two groups. These gaps suggest that human-machine interactions alone are not sufficient to enhance the human perception of the machine, nor are they sufficient to evoke better human feedback/guidance to the robots. The extent of human-robot mutual understanding depends upon how an AI system explains

itself to the user. In the absence of informative explanations, certain misconceptions cannot be eliminated even with continuous interaction between humans and robots.

Compared with the brief-explanation group, the full-explanation group demonstrates significant enhancement in the human estimate of robots' values, but does not show a strong advantage in terms of scouts' value alignment. These results indicate that human users in the two explanation groups provide feedback to the scouts in similar ways, although the full-explanation group acquires a more accurate understanding of the system. One possible cause of this dissociation is that human users exhaust their cognitive resources when processing other complexities of the game, such as comprehending messages from the three scouts or analyzing information on the map. Thus, additional details in the full explanation cannot be accommodated to offer more rationale feedback. An alternative possible reason involves the design of the game, in particular, the granularity of the feedback. As there are only eight possible feedbacks (two for each scout's proposal) in every round, it is possible to identify the best response to the scouts with only a limited understanding of the system status and the proposals. The extra information provided in the full explanation, though beneficial to the user's perception of the machine, maybe useless for scouts' value learning.

To summarize, we use an XAI system to demonstrate a bidirectional human-robot value alignment framework. The proposed XAI system shows that, with the CL formalism as the backbone of the machine's learning module and appropriate explanations provided to the user, humans and robots are able to achieve alignment of mental models through an in-situ, real-time and interactive manner. After reviewing a wide range of applications of the CL formalism, in the next chapter, we'll discuss the implications of this formalism and its contribution to the foundation of learning theory.

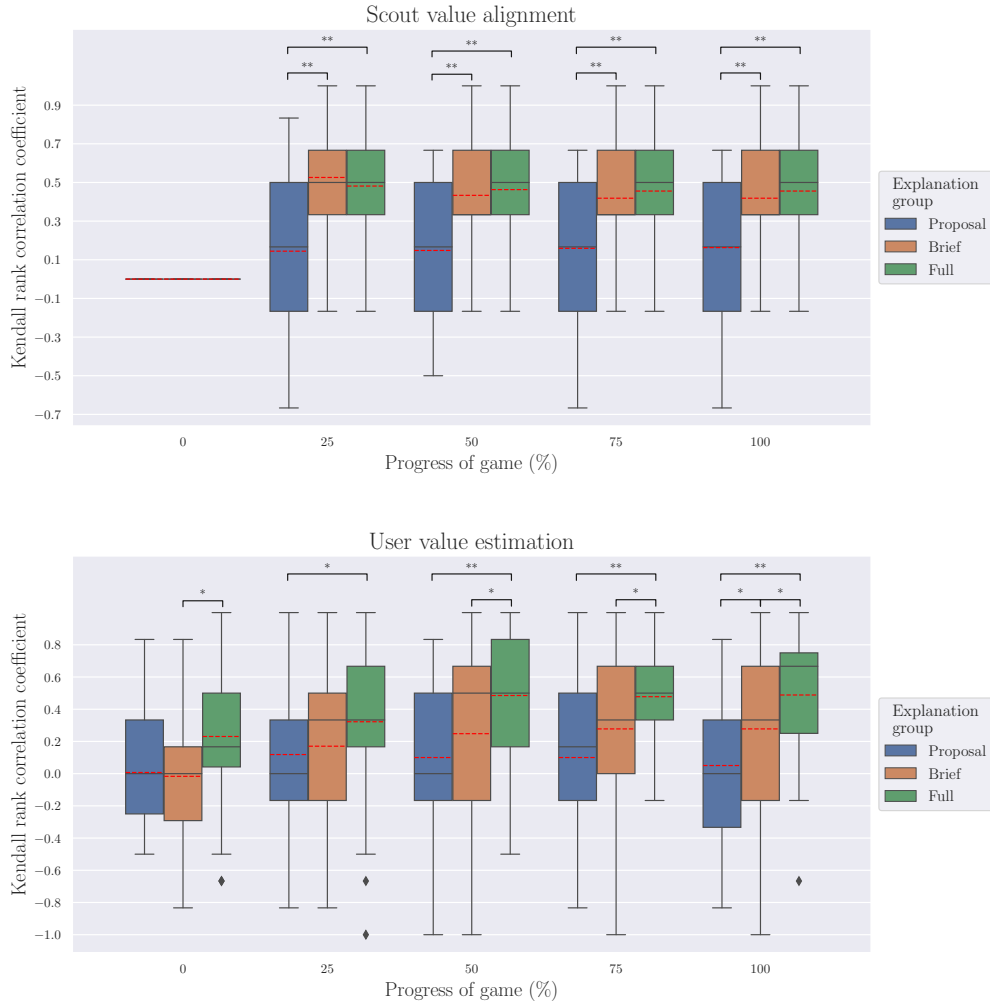


Figure 5.4: **Results of value estimation for scouts and humans in three groups.** The legends: proposal, brief, and full refer to the proposal-only group, the brief-explanation group, and the full-explanation group, respectively. Horizontal axis indicates the progress of the game for human participants; vertical axis indicates Kendall’s rank correlation coefficient between estimated values by scouts and humans; higher correlation indicates better value alignment. Top panel **A**: correlation between scouts’ value estimate and the true values that are known to human users as a function of game progress (*i.e.*, scout’s accuracy in estimating human values). Before the game starts, the scouts’ value estimate is initialized as uniform across all goals. Bottom panel **B**: correlation between the human estimate of the scouts’ values and scouts’ estimate of the true values as a function of game progress (*i.e.*, humans’ accuracy in estimating scouts’ values). Asterisks in the plot indicate significant group differences. The error bars indicate the observation minimum and maximum. The solid lines and red dashed lines in the bars respectively indicate the median and mean.

CHAPTER 6

Implications and Conclusion

As discussed in Section 1.2, current frameworks in communication, applied math and statistical machine learning are limited to special settings, and the derived performance bounds [GK95, BEH89, BBM02, CSZ06, Bal08, BPZ13] based on PAC-learning and VC dimensions are often overly pessimistic. Most concepts are “not learnable” in PAC-learning settings [Val84], while human intelligence can learn from small examples for daily tasks. The main reason is that the current methods do not account for many important aspects of human communication and end up with less effective learning protocols.

In this section, we will elaborate on a few topics to formulate and develop the theoretical foundation to support CL. We first introduce a new representation of learning, a starting point of learning protocols going beyond the Shannon communication limit [Sha48]. Then we discuss the hierarchies of learning and put forth the halting problem of learning, on top of these hierarchies.

6.1 When Aumann Meets Grice: from Distributed Knowledge to Common Knowledge

Realizing the insufficiency of Shannon’s communication model and Valiant’s PAC learning theory in modeling human learning, we seek another representation that is generic enough to accommodate both the scientist’s and the student’s way of learning. As discussed in Section 1.2.1, both types of learning can be interpreted as having information delivered

from one mind to another. Hence, a generic representation should model both the “mind of departure” and the “mind of destination” of information delivery. Particularly, we need a clear formulation for what is known and unknown in the learning process to model the transitions in the teacher’s and the student’s minds during learning. Fortunately, epistemic logic, *i.e.* the logic of knowledge [Hin62] introduces a rigorous definition and mathematical representation of knowledge and beliefs. In the 1970s, Aumann further expands the existing logical analysis of knowledge reasoning to multiple agents and applies the concept of common knowledge to economics and game theories [Aum76]. The notion of common knowledge, together with the later proposed concept of distributed knowledge [HM90], provides an ideal tool to represent agents’ mental status before and after learning, and thus model the whole learning process.

The framework for modeling knowledge is based on possible worlds [FMH03]. The intuitive idea behind the possible-worlds model is that besides the true state of affairs, there are a number of other possible states of affairs or “worlds”. Given its current information, an agent may not be able to tell which one of the possible worlds describes the actual state of affairs. An agent is then said to know a fact if it is true in all the possible worlds (given the agent’s current information). Concretely, we can imagine a robot with a blurry camera. Visual signals received by the blurry camera are not clear enough for the robot to differentiate every possible world, so the robot needs to maintain a set of worlds that are all possible given the views it has received. The facts known by the robot must be true in all the worlds it considers possible.

When there are two agents reasoning about the world, the concepts of common knowledge and distributed knowledge kick in.

- Common knowledge: facts that both agents know, and both agents know their partners know, and both agents know their partners know that they know, so on and so forth.
- Distributed knowledge: facts that both agents will know if they fully combine their

knowledge.

In other words, common knowledge is something known by both agents and neither of them can deny, whereas distributed knowledge may not be known by any of the agents individually before they exchange their knowledge. Hence, distributed knowledge is always at least as precise as and usually preciser than common knowledge. The goal of learning is for the facts in **the teacher and the student's distributed knowledge to be delivered to their common knowledge**. For instance, suppose there are two robots, whose cameras are blurred in different ways. Learning takes place when they communicate and share their knowledge to prune both of their sets of possible worlds. When their common knowledge becomes identical to their distributed knowledge, the learning terminates, as both robots have no private information the other robots don't know.

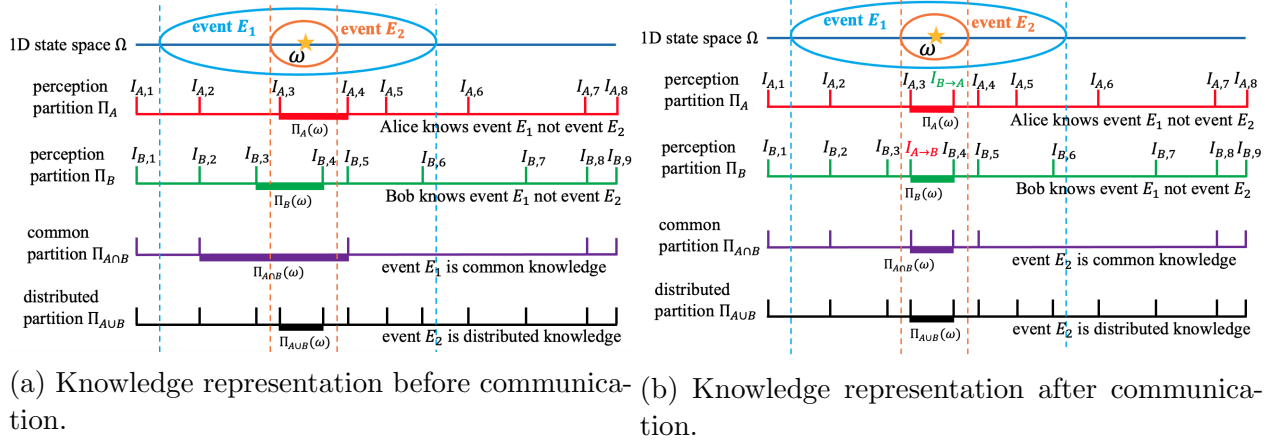


Figure 6.1: Common and distributed knowledge for inferring a state ω (star) in 1D space between Alice and Bob. Every segment represents a cell in the partition. States fell into the same segment cannot be differentiated. Alice's and Bob's perception partition become finer after communication because partners' messages enable further differentiation of worlds.

We demonstrate an example learning process with logic knowledge representation in Fig. 6.1. Suppose Alice and Bob have imperfect perception of the world (like the robot with blurry camera). *I.e.*, they cannot observe ω , instead they observe some projections as input.

$$I_A = I_A(\omega) = (I_{A,1}, \dots, I_{A,8});$$

$$I_B = I_B(\omega) = (I_{B,1}, \dots, I_{B,9}).$$

Each input $I_{A,j}, I_{B,j} \in \{0, 1\}$ is binary: $= 1$ if ω is on its right side and $= 0$ if ω is on the left. Their perception partitions are Π_A and Π_B respectively.

$$\Pi_A(\omega) = \{\omega' : I_A(\omega') = I_A(\omega)\};$$

$$\Pi_B(\omega) = \{\omega' : I_B(\omega') = I_B(\omega)\}.$$

That is, when the world is $\omega \in \Omega$, Alice cannot differentiate worlds in $\Pi_A(\omega)$. *I.e.*, she knows the real world must be in $\Pi_A(\omega)$, but she doesn't know which element of $\Pi_A(\omega)$ it is. Similarly, Bob knows one world in $\Pi_B(\omega)$ has to be true but confuses about the exact element in $\Pi_B(\omega)$.

Then, with some preliminary definitions about partitions, we can define Alice and Bob's common and distributed knowledge. Given two partitions $\mathcal{P}$ and $\mathcal{P}'$ of a set S , then

- $\mathcal{P}$ is finer than $\mathcal{P}'$ if $\forall s \in S, \mathcal{P}(s) \subseteq \mathcal{P}'(s)$,
- $\mathcal{P}$ is coarser than $\mathcal{P}'$ if $\forall s \in S, \mathcal{P}'(s) \subseteq \mathcal{P}(s)$.

Intuitively, if partition $\mathcal{P}$ is finer than partition $\mathcal{P}'$, then the information sets given by $\mathcal{P}$ give at least as much information as the information sets given by $\mathcal{P}'$ (since considering fewer states possible corresponds to having more information). The *meet* of partitions $\mathcal{P}$ and $\mathcal{P}'$, denoted $\mathcal{P} \cap \mathcal{P}'$, is the finest partition that is coarser than $\mathcal{P}$ and $\mathcal{P}'$; the *join* of $\mathcal{P}$ and $\mathcal{P}'$, denoted $\mathcal{P} \cup \mathcal{P}'$, is the coarsest partition finer than $\mathcal{P}$ and $\mathcal{P}'$ [FMH03, Chapter 3]. Next, we can make use of the meet and the join to give nice characterizations of common knowledge and distributed knowledge.

As shown in Fig. 6.1a, before they communicate, the *meet* of the two partitions, $\Pi_{A \cap B}$, forms the common partition of Alice and Bob, and the *join* of the two partitions, $\Pi_{A \cup B}$, forms the distributed partition of them. That the event E_1 happens is the common knowledge,

because

$$\Pi_{A \cap B}(\omega) \subset E_1.$$

Namely, because the event E_1 happens in all worlds of $\Pi_{A \cap B}(\omega)$, both Alice and Bob know it happens, despite they don't know the exact world. Moreover, neither Alice nor Bob can deny they don't know E_1 , because $\Pi_{A \cap B}(\omega)$ contains all possible confused worlds for both Alice and Bob¹, making E_1 their common knowledge. On the contrary, neither Alice nor Bob knows that the event E_2 happens, because

$$\Pi_A(\omega) \not\subset E_2 \wedge \Pi_B(\omega) \not\subset E_2.$$

However, by combining their knowledge, E_2 is also commonly knowable, because

$$\Pi_{A \cup B}(\omega) \subset E_2.$$

In Fig. 6.1b, we show that by sharing one of their partition boundaries to their partners, Alice and Bob can transit the event E_2 from their distributed knowledge to their common knowledge. Namely,

$$m_{A \rightarrow B} = I_{A,3}(\omega), \text{ and } m_{B \rightarrow A} = I_{B,4}(\omega).$$

Then, by 1-round of messaging, E_2 becomes common knowledge. The perceived cell is compressed:

$$\Pi_A(\omega) = \{\omega' : I_A(\omega') = I_A(\omega) \wedge I_A(\omega') = m_{B \rightarrow A}\};$$

¹Intuitively, since Alice knows $\omega \in \Pi_A(\omega)$, she knows that Bob must know $\omega \in [I_{B,3}, I_{B,5}]$. Likewise, Bob knows Alice knows that $\omega \in [I_{A,2}, I_{A,4}]$. Together, the mutual knowledge forms $\Pi_{A \cap B}(\omega)$. Rigorously, the definition of common knowledge triggers infinite recursions. In the case of Fig. 6.1, the recursion converges after one round.

$$\Pi_B(\omega) = \{\omega' : I_B(\omega') = I_B(\omega) \wedge I_B(\omega') = m_{A \rightarrow B}\}.$$

Here we slightly abuse the notation and use the equal sign to indicate perception consistent with the received message. In a nutshell, using the framework of logical knowledge analysis, learning is modeled as agents communicating and sharing their individual knowledge so that their partition of the worlds is refined and information is delivered from distributed knowledge to common knowledge.

6.1.1 Beyond Shannon’s Limits: Better Learning Protocol Enabled by CL

The notion of common and distributed knowledge provides a formal representation of learning. Nevertheless, the knowledge representation alone still falls short of modeling the cooperation in human pedagogy. Specifically, information delivery from distributed knowledge to common knowledge answers the question of what is learning, but it doesn’t approach how the teacher and the student should send out and comprehend messages, *i.e.* what is an efficient learning protocol. To answer the question of how to learn efficiently, we must arm Aumann’s knowledge representation with pragmatics.

As we mentioned, pragmatics is the branch of linguistics studying how the context of language using contributes to the meanings [Gri75, SGF13, HRA16]. Recall the example illustrated in Fig. 1.4, Fig. 4.1a and Fig. 4.2. In the context of pedagogy, where the teacher and the student form a collaborative group, not only are the literal meanings of the teacher’s messages are considered, but her actions of choosing certain messages over others also facilitate the student’s learning. The student makes exquisitely sensitive inferences about what the utterance means given their knowledge of the situation, the context, and the teacher [GF16]. A famous concretization is the Gricean Maxims of quantity [Gri75] or the scalar implicature: When people say “I like drinking warm coffee,” though the lexical meaning of “warm” is semantically close to “hot,” they mean “not hot”; otherwise people would have said “hot” directly [Car98, VBP13]. This simple phenomenon entails two fundamental characteristics

of human communication, *i.e.* the collaborative common ground between the listener and speaker, and recursive ToM modeling. By embedding learning in a communication framework, CL can satisfy both of these conditions that unilateral machine learning paradigms are impossible to meet.

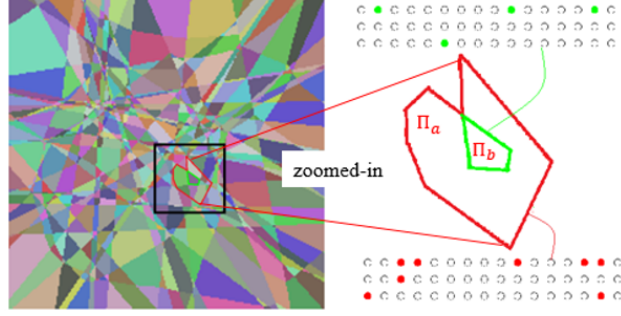


Figure 6.2: An example of partition in 2D space and inference of a state using the pragmatic protocol. **Left:** The partition of the 2D space. States within the same patch trigger identical neural responses. **Right:** Neurons fired for two cells π_a (red) and π_b (green).

In Fig. 6.2, we give an example demonstrating the advantage of integrating pragmatic reasoning into the learning protocol. Let state ω be an image, and Alice and Bob has N_A and N_B neurons as their observations:

$$I_A(\omega) = (h_A^1(\omega), \dots, h_A^{N_A}(\omega)),$$

$$I_B(\omega) = (h_B^1(\omega), \dots, h_B^{N_B}(\omega)).$$

h denotes neurons, which can be indicators or ReLU projections of the image. That is,

$$h(\omega) = \mathbf{1}(\langle \omega, \theta \rangle \geq 0) \text{ or } \max(0, \langle \omega, \theta \rangle),$$

where θ is the weight of the neuron. As shown in Fig. 6.2, π_a is surrounded by 8 red neurons while π_b is bounded by 4 green neurons. Suppose Alice knows an event $\omega \in \pi_a$ and tells Bob

by

$$m_{A \rightarrow B} = (h_{A,a1}, \dots, h_{A,a8}).$$

Then Bob will refine his perception from $\Pi_B(\omega)$ to π_a , achieving an information gain of

$$\delta_{Shannon} = \log \frac{|\Pi_B(\omega)|}{|\pi_a|},$$

as defined in Eq. (1.1). Namely, Bob's belief of the possible worlds narrows down from $\Pi_B(\omega)$ to π_a . Interestingly, if Bob has ToM and integrates pragmatics into the learning protocol, he will read-between-lines: Since Alice could but didn't send a shorter message using the 4 green neurons, she must imply that not π_b but π_a . Therefore, Bob can further refine his belief and achieve an information gain of

$$\delta_{CL} = \log \frac{|\Pi_B(\omega)|}{|\pi_a/\pi_b|},$$

where π_a/π_b means areas in π_a but not in π_b .

Proposition 4. *Pragmatic-protocol is more effective than Shannon's communication protocol as Bob gains more information than Shannon's information measurement, as we have*

$$\delta_{CL} > \delta_{Shannon}$$

The pragmatic-protocol goes beyond Shannon's information limit by integrating ToM. The extra information gain is brought by reflecting the minds of the other agents: Alice selects messages after deliberating what Bob knows, and Bob reasons why Alice sent this message instead of other plausible messages.

6.2 Halting Problem of Learning

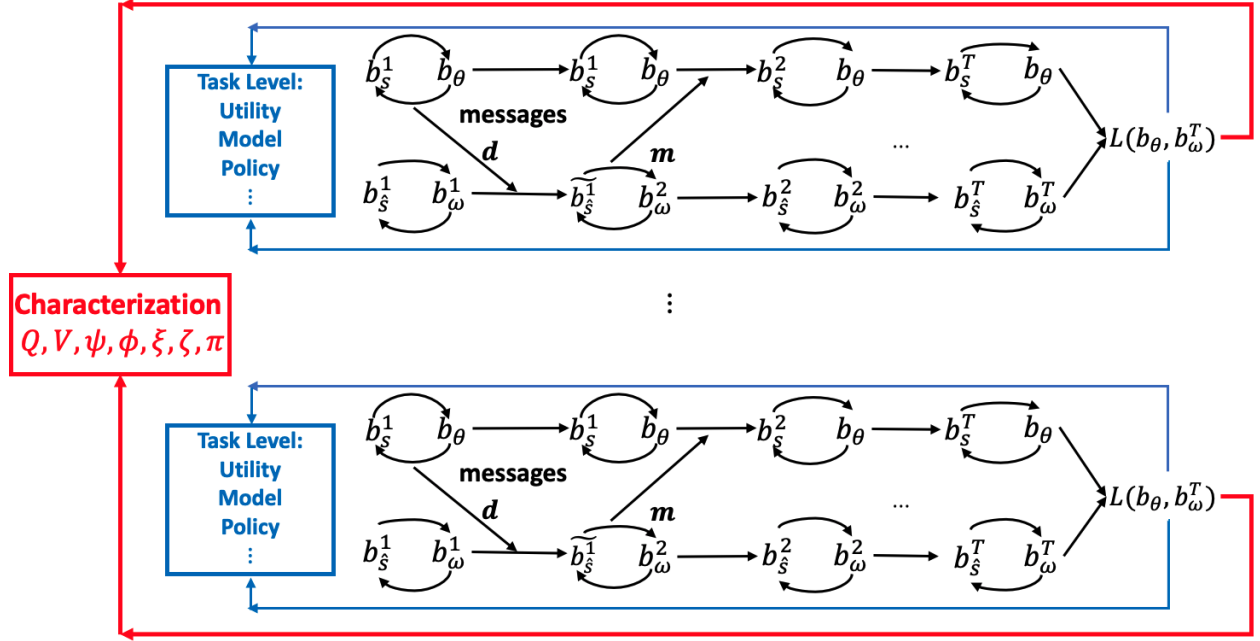


Figure 6.3: CL includes 3 nested loops: i) The **reflection loop** in black for deliberation of inferential messages to achieve common ground; ii) The **learning loop** in blue is to achieve a common model, utility, policy *etc.*; and iii) The **characterization loop** in red is to achieve better group understanding of each agent’s characterization, such as value and belief-update functions defined in Section 2.3. The norm of communication forms when group members’ estimations of their partners’ characterizations stabilize.

After studying the convergence from distributed knowledge to common knowledge, we raise the “halting problem of learning”, by analogy to the halting problem of computing [Tur37]. *I.e.* under what conditions, does the learning process terminate at various equilibria, which defines the fundamental limits of learning. Just like the pedagogy and learning in our everyday lives, CL proceeds iteratively. For iterative learning, the problem of halting is ubiquitous. So far in this article, we haven’t dived into this problem. In Algorithm 1, the algorithms terminate when a fixed number of steps is reached. However, deciding the proper rounds of interaction beforehand can be challenging, if not impractical. Thus, we need some criteria to monitor the learning and terminate the process when learning arrives at its limits.

To approach the halting problem of learning, we must know the underlying driving force of learning. Namely, what do the teacher and the student seek to agree on when they communicate with each other. Here, we recognize the three granularity of learning: *i)* Message Level, *ii)* Task Level and *iii)* Group Level. Every level has a distinctive goal and is controlled by a loop in CL, as depicted in Fig. 6.3. Within each loop, the teacher and the student aims to achieve an equilibrium. In the next section, we introduce each level in detail. As one purpose of CL is to facilitate the development of new learning paradigms, we also include opening questions along with the introduction to encourage future explorations in related topics.

6.2.1 Three Hierarchies in Learning

Message Level The message level indicates the interpretation of each message between the teacher and the student. As messages are the building blocks of a communication process, being able to fully comprehend the speaker is the prerequisite of an effective learning process. At this level, messages communicate about a state, a cell, or an event (set) in the state space to achieve a common ground and common belief. Although we only discuss a few types of messages such as labeled data (Section 2.4), projection on cells as partitions (Section 6.1), and linear neurons (Section 6.1.1), the message space of CL can be extended to nodes in a parse graph or logic predicates which correspond to compositions of the atomic cells or events. The **reflection loop** in CL is in charge of the equilibrium in the message level. The reflection processes involve loops because the agents bear ToM and conduct recursive mutual reasoning. In particular, the teacher considers what the student needs to know, and the student thinks about why the teacher sends one specific message instead of others, *etc.*

At this level, ideal agents should be capable of capturing both the literal and pragmatic meaning of the messages. So far, most works assume that the teacher and the student have a common ground about the literal meanings of all the messages. In other words, they speak the same language, or, at least, each has a dictionary for the language the other is speaking.

One relaxation of this assumption is to take away their dictionaries and see if they can still develop a valid codebook that they share as the common literal meanings of the messages. Namely, the teacher needs to learn ϕ, ψ and the student needs to learn ξ, ζ and π from scratch as a group or individually with the partner fixed. Another challenge at this level is the derivation of pragmatic meanings, which often requires counterfactual reasoning. That is, to understand the speaker, the listener needs to consider not only the explicit selected message but also the implicit unselected messages. When the message space is large, *e.g.* English, such counterfactual reasoning often becomes intractable and requires an efficient method to be fully interpreted.

Task Level The task level learning involves transmitting and understanding a sequence of messages to obtain the convergence of the teacher’s and the student’s mental components. The **learning loop** in CL aims to find the equilibrium in this level. Namely, for the student’s utility, model, and policy to be close enough to the teacher’s counterparts. Therefore, task level learning accommodates the planning for sequential pedagogy, in which the communication takes multiple steps. As the planning complexity is exponential w.r.t. the number of steps, and the message space is usually large or dynamic, most existing teachers use heuristics or short-horizon planning. Nevertheless, even for this inadequate planning, the student is treated as a naive learner who doesn’t model the teacher (level-1 paradigms). Developing a planning algorithm for the teachers in level-2 paradigms remains an active research question. It requires the teacher and the student to form a group level norm of communication, which falls into the scope of group level learning. Another potential improvement of task level learning is to generalize message level pragmatic reasoning to the task level. In other words, endow the student with more information from the unselected teaching sequences in addition to the selected one.

Group Level Finally, the group level entails the characterization of the teacher and the student. To conduct effective, CL, the value and belief-update functions of the teacher and

the student must work in accordance. In most prior learning paradigms, the characterizations of the agents are predefined by heuristics. In more generic and realistic settings, the teacher and the student needs to learn proper characterization to collaborate. The **characterization loop** in CL seeks to tackle this problem. Just like human teams demand sufficient collaboration experience to acquire tacit cooperation. It usually requires multiple learning tasks for the agents to learn appropriate characterization. As shown in Fig. 6.3, the outcomes of multiple learning tasks will shape the agents’ characterizations. The referential game in Chapter 4 follows this process. Afterward, norms of communication and a learning protocol will be established between the teacher and the student.

In all the works surveyed in this article, one teacher is exclusively designed for one or one type of student. A standard characterization loop also only sets up the learning protocol for one specific group. A more ambitious setting is to have a teacher that can adapt to different kinds of students, even those she never encountered during her training. Suppose we can characterize the student’s personality, such as IQ, memory, which corresponds to his mental dynamics, then a versatile teacher should be able to identify her students’ characterization and customize her pedagogy accordingly. To be more specific, the teacher (student) can parameterize her (his) dynamic functions, ψ, ϕ (ξ, ζ, π), and model a distribution of the parameters. Such setting is similar to the Ad-Hoc teamwork [SK10] and multitask/meta-RL in MDPs with different dynamics [ZSK20], but hasn’t been thoroughly studied in the context of pedagogical machine learning.

6.2.2 Halting Criteria

Knowing the three hierarchies of learning, we can discuss the proper halting criteria of learning. In every level, the CL loop terminates in different conditions. In the **message level**, a plausible halting criterion is fully understanding both the literal and pragmatic meaning of each message. The literal meaning is usually straightforward when agents use mutually recognizable message spaces. The pragmatic meaning depends on the context,

e.g. the situated state, the communication history, and so on. As pragmatic understanding requires ToM, the reflection loop usually stops when the recursive reasoning converges, or the cognitive burden of going to the next recursion layer exceeds the cost of sending an explicit query message to clarify.

In the **task level**, the ideal halting occurs when the six minds converge to one, and validated by oracle (God's mind). Recall the high-level mind notation we used in Fig. 1.1. That is,

$$P_t = Q_t = \widehat{P}_t = \widehat{Q}_t = C_t = G.$$

This is the strictest halting condition. In many cases, some alternative halting conditions can be defined. For example, let $\mathcal{D}(X, Y)$ denote the distance between two minds X and Y , such as Kullback–Leibler divergence, total variance distance, earth mover's distance, *etc.* Then, we can name a few halting conditions:

- $\mathcal{D}(P_t, \widehat{Q}_t) \leq \epsilon$, *i.e.* the teacher thinks the student has already known what she knows and stops teaching.
- $\mathcal{D}(Q_{t-1}, Q_t) \leq \epsilon$, *i.e.* the student becomes complacent and thinks he cannot acquire new knowledge. Thus, stops learning.
- $\mathcal{D}(\widehat{Q}_{t-1}, \widehat{Q}_t) \leq \epsilon$, *i.e.* the teacher thinks the student cannot make further promising progress and stops teaching.
- $\mathcal{D}(C_t, \emptyset) \leq \epsilon$, *i.e.* the teacher and the student find it hard to reach common ground and terminate communication.
- $\mathcal{D}(Q_t, G) \leq \epsilon$, *i.e.* the student achieves satisfactory performance in the real world, and stops learning. Notice that in some cases, G may not be directly accessible, so certain surrogate functions may be needed. *E.g.*, let the student finish some unseen tasks as tests.

The above conditions are by no means comprehensive, more can be proposed to meet distinctive needs and circumstances. In addition, conditions more complicated than a single distance function can also be defined, *e.g.*, comparing the gain of minimizing the divergence between the student and the teacher against the cost of transmitting teaching messages.

In the **group level**, the teacher and the student can stop interacting with each other after they know their partners’ characteristics, *i.e.* dynamic functions, state, model, and value spaces, *etc.* This usually requires collaboration over several pedagogy tasks to accomplish. The assessment of the convergence needs the teacher and the student to communicate given unseen learning objectives. Only when they can cooperate effectively in diverse tasks, can they terminate the characterization loop and halt group level learning. Suppose we also want a flexible teacher who can adapt to various students quickly. Then, the halting condition will be for the teacher to successfully conduct CL with multiple students, each covered multiple learning objectives, resembling the evaluation used in ad-hoc teaming [BRK17].

6.3 Conclusion

In this dissertation, we study a CL formalism, which inspects learning from a communication perspective. We review existing learning paradigms such as passive learning, active learning and algorithmic teaching, and examine their limitations. The new formalism has the potential to overcome these limitations and integrate prior machine learning algorithms into a unified framework. With concrete usage examples, we demonstrate how efficient learning protocols can emerge from CL and verify the formalism’s suitability for complicated HRI tasks and generic human-machine collaboration. Moreover, the CL formalism makes two contributions to the foundation of learning. First, it introduces new representations of learning and puts forth protocols beyond Shannon’s communication limits. Second, it sheds light on the universal halting problem of learning by teasing out three hierarchies in learning and defining possible halting criteria. Overall, we see CL as providing conceptual clarity for

existing and future methods of machine learning from the perspective of mutual reasoning between the teacher and the student, and a fruitful base for future work on more advanced learning paradigms.

REFERENCES

- [ACF02] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. “Finite-time analysis of the multiarmed bandit problem.” *Machine learning*, **47**(2):235–256, 2002.
- [ACV09] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. “A survey of robot learning from demonstration.” *Robotics and autonomous systems*, **57**(5):469–483, 2009.
- [AD21] Saurabh Arora and Prashant Doshi. “A survey of inverse reinforcement learning: Challenges, methods and progress.” *Artificial Intelligence*, p. 103500, 2021.
- [Ang88] Dana Angluin. “Queries and concept learning.” *Machine learning*, **2**(4):319–342, 1988.
- [AS18] Stefano V Albrecht and Peter Stone. “Autonomous agents modelling other agents: A comprehensive survey and open problems.” *Artificial Intelligence*, **258**:66–95, 2018.
- [Aum76] R. Aumann. “Agreeing to disagree.” *The annals of statistics*, pp. 1236–1239, 1976.
- [Bal08] Frank J Balbach. “Measuring teachability using variants of the teaching dimension.” *Theoretical Computer Science*, **397**(1-3):94–113, 2008.
- [BBL09] Maria-Florina Balcan, Alina Beygelzimer, and John Langford. “Agnostic active learning.” *Journal of Computer and System Sciences*, **75**(1):78–89, 2009.
- [BBM02] Peter L Bartlett, Olivier Bousquet, and Shahar Mendelson. “Localized rademacher complexities.” In *International Conference on Computational Learning Theory*, pp. 44–58. Springer, 2002.
- [BCC18] Cristian-Paul Bara, Sky CH-Wang, and Joyce Chai. “MindCraft: Theory of Mind Modeling for Situated Dialogue in Collaborative Tasks.” In *Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.
- [BDL09] Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. “Importance weighted active learning.” In *Proceedings of the 26th annual international conference on machine learning*, pp. 49–56, 2009.
- [BDM17] André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado van Hasselt, and David Silver. “Successor features for transfer in reinforcement learning.” In *Advances in Neural Information Processing Systems*, 2017.

- [BEH89] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. “Learnability and the Vapnik-Chervonenkis dimension.” *Journal of the ACM (JACM)*, **36**(4):929–965, 1989.
- [BGB18] Ben Bogin, Mor Geva, and Jonathan Berant. “Emergence of communication in an interactive world with consistent speakers.” *arXiv preprint arXiv:1809.00549*, 2018.
- [BGL12] Leon Bergen, Noah Goodman, and Roger Levy. “That’s what she (could have) said: How alternative utterances affect language use.” In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 34, 2012.
- [Bis06] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.
- [Blu00] Reinhard Blutner. “Some aspects of optimality in natural language interpretation.” *Journal of semantics*, **17**(3):189–216, 2000.
- [BMS11] Monica Babes, Vukosi Marivate, Kaushik Subramanian, and Michael L Littman. “Apprenticeship learning about multiple intentions.” In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 897–904, 2011.
- [BN19] Daniel S Brown and Scott Niekum. “Machine teaching for inverse reinforcement learning: Algorithms and applications.” In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 7749–7758, 2019.
- [BPZ13] Adrian Barbu, Maria Pavlovskaja, and Song Chun Zhu. “Rates for inductive learning of compositional models.” In *Workshops at the Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.
- [BRK17] Samuel Barrett, Avi Rosenfeld, Sarit Kraus, and Peter Stone. “Making friends on the fly: Cooperating with new teammates.” *Artificial Intelligence*, **242**:132–171, 2017.
- [BST09] Chris L Baker, Rebecca Saxe, and Joshua B Tenenbaum. “Action understanding as inverse planning.” *Cognition*, **113**(3):329–349, 2009.
- [BT52] Ralph Allan Bradley and Milton E Terry. “Rank analysis of incomplete block designs: I. The method of paired comparisons.” *Biometrika*, **39**(3/4):324–345, 1952.
- [BT14] Chris L Baker and Joshua B Tenenbaum. “Modeling human plan recognition using Bayesian theory of mind.” *Plan, activity, and intent recognition: Theory and practice*, pp. 177–204, 2014.
- [CAL94] David Cohn, Les Atlas, and Richard Ladner. “Improving generalization with active learning.” *Machine learning*, **15**(2):201–221, 1994.

- [Car98] Robyn Carston. “Informativeness, relevance and scalar implicature.” *Pragmatics And Beyond New Series*, pp. 179–238, 1998.
- [CCT16] Xi Chen, Yu Cheng, and Bo Tang. “On the recursive teaching dimension of vc classes.” In *Advances in Neural Information Processing Systems*, pp. 2164–2171, 2016.
- [CG06] Gergely Csibra and György Gergely. “Social learning and social cognition: The case for pedagogy.” *Processes of change in brain and cognitive development. Attention and performance XXI*, **21**:249–274, 2006.
- [CG09] Gergely Csibra and György Gergely. “Natural pedagogy.” *Trends in cognitive sciences*, **13**(4):148–153, 2009.
- [CL12] Maya Cakmak and Manuel Lopes. “Algorithmic and human teaching of sequential decision tasks.” In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.
- [CLF18] Edward Choi, Angeliki Lazaridou, and Nando de Freitas. “Compositional obverter communication learning from raw visual input.” *arXiv preprint arXiv:1804.02341*, 2018.
- [CM05] Aron Culotta and Andrew McCallum. “Reducing labeling effort for structured prediction tasks.” In *AAAI*, volume 5, pp. 746–751, 2005.
- [CMS18] Yuxin Chen, Oisín Mac Aodha, Shihan Su, Pietro Perona, and Yisong Yue. “Near-optimal machine teaching via explanatory teaching sets.” In *International Conference on Artificial Intelligence and Statistics*, pp. 1970–1978. PMLR, 2018.
- [CNS18] Min Chen, Stefanos Nikolaidis, Harold Soh, David Hsu, and Siddhartha Srinivasa. “Planning with trust for human-robot collaboration.” In *Proceedings of ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 307–315, 2018.
- [CSM18] Yuxin Chen, Adish Singla, Oisín Mac Aodha, Pietro Perona, and Yisong Yue. “Understanding the role of adaptivity in machine teaching: The case of version space learners.” *arXiv preprint arXiv:1802.05190*, 2018.
- [CSZ06] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. *An Augmented PAC Model for Semi-Supervised Learning*, chapter 22, pp. 397–419. MIT Press, 2006.
- [Das09] Sanjoy Dasgupta. “The Two Faces of Active Learning.” In *Discovery science*, p. 35, 2009.
- [DCL18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “Bert: Pre-training of deep bidirectional transformers for language understanding.” *arXiv preprint arXiv:1810.04805*, 2018.

- [DDS09a] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. “Imagenet: A large-scale hierarchical image database.” In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.
- [DDS09b] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. “ImageNet: A large-scale hierarchical image database.” In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.
- [DFS14] Thorsten Doliwa, Gaojian Fan, Hans Ulrich Simon, and Sandra Zilles. “Recursive teaching dimension, VC-dimension and sample compression.” *The Journal of Machine Learning Research*, **15**(1):3107–3131, 2014.
- [DG09] Prashant Doshi and Piotr J Gmytrasiewicz. “Monte Carlo sampling methods for approximating interactive POMDPs.” *Journal of Artificial Intelligence Research*, **34**:297–337, 2009.
- [DHM07] Sanjoy Dasgupta, Daniel J Hsu, and Claire Monteleoni. *A general agnostic active learning algorithm*. Citeseer, 2007.
- [DVV14] Harmen De Weerd, Rineke Verbrugge, and Bart Verheij. “Theory of Mind in the Mod Game: An Agent-Based Model of Strategic Reasoning.” In *ECSI*, pp. 128–136, 2014.
- [DVV15] Harmen De Weerd, Rineke Verbrugge, and Bart Verheij. “Higher-order theory of mind in the tacit communication game.” *Biologically Inspired Cognitive Architectures*, **11**:10–21, 2015.
- [EFG16] Baxter S Eaves Jr, Naomi H Feldman, Thomas L Griffiths, and Patrick Shafto. “Infant-directed speech is consistent with teaching.” *Psychological review*, **123**(6):758, 2016.
- [EFT13] H-D Ebbinghaus, Jörg Flum, and Wolfgang Thomas. *Mathematical logic*. Springer Science & Business Media, 2013.
- [EGX17] Mark Edmonds, Feng Gao, Xu Xie, Hangxin Liu, Siyuan Qi, Yixin Zhu, Brandon Rothrock, and Song-Chun Zhu. “Feeling the force: Integrating force and pose for fluent discovery through imitation learning to open medicine bottles.” In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3530–3537. IEEE, 2017.
- [ESS15] Baxter Eaves, April Schweinhart, and Patrick Shafto. “Tractable bayesian teaching.” *Big Data in Cognitive Science*, **10**, 2015.
- [FAF16] Jakob Foerster, Ioannis Alexandros Assael, Nando de Freitas, and Shimon Whiteson. “Learning to communicate with deep multi-agent reinforcement learning.” In *Advances in Neural Information Processing Systems*, 2016.

- [FCA18] Jakob Foerster, Richard Y Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. “Learning with opponent-learning awareness.” In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 122–130. International Foundation for Autonomous Agents and Multiagent Systems, 2018.
- [FG12] Michael C Frank and Noah D Goodman. “Predicting pragmatic reasoning in language games.” *Science*, **336**(6084):998–998, 2012.
- [FGH17] Jaime F Fisac, Monica A Gates, Jessica B Hamrick, Chang Liu, Dylan Hadfield-Menell, Malayandi Palaniappan, Dhruv Malik, S Shankar Sastry, Thomas L Griffiths, and Anca D Dragan. “Pragmatic-pedagogic value alignment.” *arXiv preprint arXiv:1707.06354*, 2017.
- [FMH03] Ronald Fagin, Yoram Moses, Joseph Y Halpern, and Moshe Y Vardi. *Reasoning about knowledge*. MIT press, 2003.
- [FQZ21] Lifeng Fan, Shuwen Qiu, Zilong Zheng, Tao Gao, Song-Chun Zhu, and Yixin Zhu. “Learning Triadic Belief Dynamics in Nonverbal Communication from Videos.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [Fri17] Jerome H Friedman. *The elements of statistical learning: Data mining, inference, and prediction*. springer open, 2017.
- [FTQ18] Yang Fan, Fei Tian, Tao Qin, Xiang-Yang Li, and Tie-Yan Liu. “Learning to Teach.” In *International Conference on Learning Representations*, 2018.
- [FZ15] Amy Fire and Song-Chun Zhu. “Learning perceptual causality from video.” *ACM Transactions on Intelligent Systems and Technology (TIST)*, **7**(2):1–22, 2015.
- [GCS13] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. CRC press, 2013.
- [GDD14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. “Rich feature hierarchies for accurate object detection and semantic segmentation.” In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.
- [GF16] Noah D Goodman and Michael C Frank. “Pragmatic language interpretation as probabilistic inference.” *Trends in cognitive sciences*, **20**(11):818–829, 2016.
- [GGZ20] Xiaofeng Gao, Ran Gong, Yizhou Zhao, Shu Wang, Tianmin Shu, and Song-Chun Zhu. “Joint Mind Modeling for Explanation Generation in Complex Human-Robot Collaborative Tasks.” In *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 1119–1126. IEEE, 2020.

- [Gir15] Ross Girshick. “Fast r-cnn.” In *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.
- [GK95] Sally A Goldman and Michael J Kearns. “On the complexity of teaching.” *Journal of Computer and System Sciences*, **50**(1):20–31, 1995.
- [Gri75] Herbert P Grice. “Logic and conversation.” In *Speech acts*, pp. 41–58. Brill, 1975.
- [GS13] Noah D Goodman and Andreas Stuhlmüller. “Knowledge and implicature: Modeling language understanding as social cognition.” *Topics in cognitive science*, **5**(1):173–184, 2013.
- [GSS18] Hyowon Gweon, Patrick Shafto, and Laura Schulz. “Development of children’s sensitivity to overinformativeness in learning and teaching.” *Developmental psychology*, **54**(11):2113, 2018.
- [GTS10] Hyowon Gweon, Joshua B Tenenbaum, and Laura E Schulz. “Infants consider both the sample and the sampling process in inductive generalization.” *Proceedings of the National Academy of Sciences*, **107**(20):9066–9071, 2010.
- [Han07] Steve Hanneke. “A bound on the label complexity of agnostic active learning.” In *Proceedings of the 24th international conference on Machine learning*, pp. 353–360, 2007.
- [HGD17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. “Mask r-cnn.” In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.
- [HHP20] Sandy H. Huang, Isabella Huang, Ravi Pandya, and Anca D. Dragan. “Nonverbal Robot Feedback for Human Teachers.” In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura, editors, *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1038–1051, 2020.
- [HHT95] Keith J Holyoak, Keith James Holyoak, and Paul Thagard. *Mental leaps: Analogy in creative thought*. MIT press, 1995.
- [Hin62] Jaakko Hintikka. “Knowledge and Belief: An Introduction to the Logic of the Two Notions.” *Studia Logica*, **16**:119–122, 1962.
- [HLC18] Mark K Ho, Michael L Littman, Fiery Cushman, and Joseph L Austerweil. “Effectively learning from pedagogical demonstrations.” In *Proceedings of the annual conference of the cognitive science society*, 2018.
- [HLM16] Mark K Ho, Michael Littman, James MacGlashan, Fiery Cushman, and Joseph L Austerweil. “Showing versus doing: Teaching by demonstration.” In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

- [HM90] Joseph Y Halpern and Yoram Moses. “Knowledge and common knowledge in a distributed environment.” *Journal of the ACM (JACM)*, **37**(3):549–587, 1990.
- [HRA16] Dylan Hadfield-Menell, Stuart J Russell, Pieter Abbeel, and Anca Dragan. “Co-operative Inverse Reinforcement Learning.” In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.
- [HZR16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition.” In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- [Jag12] Gerhard Jäger. “Game theory in semantics and pragmatics.” *Semantics: An international handbook of natural language meaning*, **3**:2487–2516, 2012.
- [Jay57] Edwin T Jaynes. “Information theory and statistical mechanics. II.” *Physical review*, **108**(2):171, 1957.
- [KH09] Alex Krizhevsky, Geoffrey Hinton, et al. *Learning multiple layers of features from tiny images*. Citeseer, 2009.
- [Kiw01] Krzysztof C Kiwiel. “Convergence and efficiency of subgradient methods for quasiconvex minimization.” *Mathematical programming*, **90**(1):1–25, 2001.
- [KMT93] Sanjeev R Kulkarni, Sanjoy K Mitter, and John N Tsitsiklis. “Active learning using arbitrary binary valued queries.” *Machine Learning*, **11**(1):23–35, 1993.
- [KPK10] M Pawan Kumar, Benjamin Packer, and Daphne Koller. “Self-Paced Learning for Latent Variable Models.” In *NIPS*, volume 1, p. 2, 2010.
- [KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks.” *Advances in neural information processing systems*, **25**:1097–1105, 2012.
- [LB17] Kun Li and Joel W Burdick. “Online inverse reinforcement learning via Bellman Gradient Iteration.” *arXiv preprint arXiv:1707.09393*, 2017.
- [LBB98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. “Gradient-based learning applied to document recognition.” *Proceedings of the IEEE*, **86**(11):2278–2324, 1998.
- [LBD89] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. “Backpropagation applied to handwritten zip code recognition.” *Neural computation*, **1**(4):541–551, 1989.

- [LCS16] Changsong Liu, Joyce Y Chai, Nishant Shukla, and Song-Chun Zhu. “Task learning through visual demonstration and situated dialogue.” In *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [LDH17] Weiyang Liu, Bo Dai, Ahmad Humayun, Charlene Tay, Chen Yu, Linda B Smith, James M Rehg, and Le Song. “Iterative machine teaching.” In *International Conference on Machine Learning*, pp. 2149–2158. PMLR, 2017.
- [LDL18] Weiyang Liu, Bo Dai, Xingguo Li, Zhen Liu, James Rehg, and Le Song. “Towards black-box iterative machine teaching.” In *International Conference on Machine Learning*, pp. 3141–3149. PMLR, 2018.
- [LG94] David D Lewis and William A Gale. “A sequential algorithm for training text classifiers.” In *SIGIR’94*, pp. 3–12. Springer, 1994.
- [LHP16] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. “Continuous control with deep reinforcement learning.” In *International Conference on Learning Representations*, 2016.
- [LHT18] Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. “Emergence of Linguistic Communication from Referential Games with Symbolic and Pixel Input.” In *International Conference on Learning Representations*, 2018.
- [LPB17] Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. “Multi-agent cooperation and the emergence of (natural) language.” In *International Conference on Learning Representations*, 2017.
- [LSL00] Stephen C Levinson, C Stephen, and Stephen C Levinson. *Presumptive meanings: The theory of generalized conversational implicature*. MIT press, 2000.
- [LST15] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. “Human-level concept learning through probabilistic program induction.” *Science*, **350**(6266):1332–1338, 2015.
- [LWT17] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. “Multi-agent actor-critic for mixed cooperative-competitive environments.” In *Advances in Neural Information Processing Systems*, pp. 6379–6390, 2017.
- [LYS16] Changsong Liu, Shaohua Yang, Sari Saba-Sadiya, Nishant Shukla, Yunzhong He, Song-Chun Zhu, and Joyce Chai. “Jointly learning grounded task structures from language instruction and visual demonstration.” In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1482–1492, 2016.

- [LZO16] Ji Liu, Xiaojin Zhu, and Hrag Ohannessian. “The teaching dimension of linear learners.” In *International Conference on Machine Learning*, pp. 117–126. PMLR, 2016.
- [LZZ19] Laurent Lessard, Xuezhou Zhang, and Xiaojin Zhu. “An optimal control approach to sequential machine teaching.” In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2495–2503. PMLR, 2019.
- [MAM17] Smitha Milli, Pieter Abbeel, and Igor Mordatch. “Interpretable and pedagogical examples.” *arXiv preprint arXiv:1711.00694*, 2017.
- [MDK11] Tomas Mikolov, Anoop Deoras, Stefan Kombrink, Lukás Burget, and Jan Cernocký. “Empirical Evaluation and Combination of Advanced Language Modeling Techniques.” In *INTERSPEECH*, pp. 605–608. ISCA, 2011.
- [Mey01] Jacob L Mey. *Pragmatics: an introduction*. Blackwell Publishing Company, 2001.
- [Mit97] Tom M Mitchell et al. *Machine learning*. McGraw-hill New York, 1997.
- [MKS13] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. “Playing atari with deep reinforcement learning.” *arXiv preprint arXiv:1312.5602*, 2013.
- [ML15] James MacGlashan and Michael L Littman. “Between imitation and intention learning.” In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.
- [MN98] Andrew Kachites McCallumzy and Kamal Nigamy. “Employing EM and pool-based active learning for text classification.” In *Proc. International Conference on Machine Learning (ICML)*, pp. 359–367. Citeseer, 1998.
- [Mur12] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [NR00] Andrew Y Ng, Stuart J Russell, et al. “Algorithms for inverse reinforcement learning.” In *Icml*, volume 1, p. 2, 2000.
- [NY78] Arkadi S Nemirovski and David Berkovich Yudin. “Cesari convergence of the gradient method of approximating saddle points of convex-concave functions.” In *Doklady Akademii Nauk*, volume 239, pp. 1056–1059. Russian Academy of Sciences, 1978.
- [OKL19] Shayegan Omidshafiei, Dong-Ki Kim, Miao Liu, Gerald Tesauro, Matthew Riemer, Christopher Amato, Murray Campbell, and Jonathan P How. “Learning to teach in cooperative multiagent reinforcement learning.” In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 6128–6136, 2019.

- [PCD19] Tomi Peltola, Mustafa Mert Çelikok, Pedram Daei, and Samuel Kaski. “Machine Teaching of Active Sequential Learners.” In *Advances in Neural Information Processing Systems*, pp. 11202–11213, 2019.
- [Pea09] Judea Pearl. *Causality*. Cambridge university press, 2009.
- [PVG11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. “Scikit-learn: Machine Learning in Python.” *Journal of Machine Learning Research*, **12**:2825–2830, 2011.
- [PW78] David Premack and Guy Woodruff. “Does the chimpanzee have a theory of mind?” *Behavioral and brain sciences*, **1**(4):515–526, 1978.
- [QZH18] Siyuan Qi, Yixin Zhu, Siyuan Huang, Chenfanfu Jiang, and Song-Chun Zhu. “Human-centric Indoor Scene Synthesis Using Stochastic Grammar.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [RA07] Deepak Ramachandran and Eyal Amir. “Bayesian Inverse Reinforcement Learning.” In *IJCAI*, volume 7, pp. 2586–2591, 2007.
- [RDS15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. “ImageNet Large Scale Visual Recognition Challenge.” *International Journal of Computer Vision (IJCV)*, **115**(3):211–252, 2015.
- [RHW85] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. “Learning internal representations by error propagation.” Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
- [Riv91] R. Rivest. “Cryptography and machine learning.” In *International Conference on the Theory and Application of Cryptology*, pp. 427–439. Springer, 1991.
- [Rus19] Stuart Russell. *Human compatible: Artificial intelligence and the problem of control*. Penguin, 2019.
- [Saa09] David Saad. *On-line learning in neural networks*. Cambridge University Press, 2009.
- [Sam59] Arthur L Samuel. “Some studies in machine learning using the game of checkers.” *IBM Journal of research and development*, **3**(3):210–229, 1959.
- [SB98] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

- [SB14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.
- [SB18] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- [SC08] Burr Settles and Mark Craven. “An analysis of active learning strategies for sequence labeling tasks.” In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 1070–1079, 2008.
- [SCF08] Burr Settles, Mark Craven, and Lewis Friedland. “Active learning with real annotation costs.” In *Proceedings of the NIPS workshop on cost-sensitive learning*, volume 1. Vancouver, CA:, 2008.
- [Sch89] Michael H Schneider. “Matrix scaling, entropy minimization, and conjugate duality. I. Existence conditions.” *Linear Algebra and its Applications*, **114**:785–813, 1989.
- [SCR07] Burr Settles, Mark Craven, and Soumya Ray. “Multiple-instance active learning.” *Advances in neural information processing systems*, **20**:1289–1296, 2007.
- [Set09] Burr Settles. “Active learning literature survey.” In *Computer Sciences Technical Report 1648*. University of Wisconsin-Madison Department of Computer Sciences, 2009.
- [SG08] Patrick Shafto and Noah Goodman. “Teaching games: Statistical sampling assumptions for learning in pedagogical situations.” In *Proceedings of the 30th annual conference of the Cognitive Science Society*, pp. 1632–1637. Cognitive Science Society Austin, TX, 2008.
- [SGF13] Nathaniel J Smith, Noah Goodman, and Michael Frank. “Learning and using language via recursive pragmatic reasoning about other agents.” In *Advances in Neural Information Processing Systems*, 2013.
- [SGG14] Patrick Shafto, Noah D Goodman, and Thomas L Griffiths. “A rational account of pedagogical reasoning: Teaching by, and learning from, examples.” *Cognitive psychology*, **71**:55–89, 2014.
- [Sha48] Claude E Shannon. “A mathematical theory of communication.” *The Bell system technical journal*, **27**(3):379–423, 1948.
- [SHC17] Nishant Shukla, Yunzhong He, Frank Chen, and Song-Chun Zhu. “Learning human utility from video demonstrations for deductive planning in robotics.” In *Conference on Robot Learning*, pp. 448–457. PMLR, 2017.

- [SHM16] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. “Mastering the game of Go with deep neural networks and tree search.” *nature*, **529**(7587):484–489, 2016.
- [Shu87] Lee Shulman. “Knowledge and teaching: Foundations of the new reform.” *Harvard educational review*, **57**(1):1–23, 1987.
- [SK67] Richard Sinkhorn and Paul Knopp. “Concerning nonnegative matrices and doubly stochastic matrices.” *Pacific Journal of Mathematics*, **21**(2):343–348, 1967.
- [SK10] Peter Stone and Sarit Kraus. “To teach or not to teach?: decision making under uncertainty in ad hoc teams.” In *AAMAS*, pp. 117–124, 2010.
- [SM91] Ayumi Shinohara and Satoru Miyano. “Teachability in computational learning.” *New Generation Computing*, **8**(4):337–347, 1991.
- [SOS92] H Sebastian Seung, Manfred Oppel, and Haim Sompolinsky. “Query by committee.” In *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 287–294, 1992.
- [SW86] Dan Sperber and Deirdre Wilson. *Relevance: Communication and cognition*, volume 142. Harvard University Press Cambridge, MA, 1986.
- [SZ15] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition.” In *International Conference on Learning Representations*, 2015.
- [SZZ20] Stephanie Stacy, Qingyi Zhao, Minglu Zhao, Max Kleiman-Weiner, and Tao Gao. “Intuitive signaling through an “Imagined We”.” In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci)*, 2020.
- [TFL16] Yulia Tsvetkov, Manaal Faruqi, Wang Ling, Brian MacWhinney, and Chris Dyer. “Learning the Curriculum with Bayesian Optimization for Task-Specific Word Representation Learning.” In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 130–139, Berlin, Germany, August 2016. Association for Computational Linguistics.
- [Tho33] William R Thompson. “On the likelihood that one unknown probability exceeds another in view of the evidence of two samples.” *Biometrika*, **25**(3/4):285–294, 1933.
- [tin17] <https://www.kaggle.com/c/tiny-imagenet>, 2017.
- [Tom10] Michael Tomasello. *Origins of human communication*. MIT press, 2010.

- [TSZ20] Ning Tang, Stephanie Stacy, Minglu Zhao, Gabriel Marquez, and Tao Gao. “Bootstrapping an imagined We for cooperation.” In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci)*, 2020.
- [Tur37] Alan Mathison Turing. “On computable numbers, with an application to the Entscheidungsproblem.” *Proceedings of the London mathematical society*, **2**(1):230–265, 1937.
- [Val84] Leslie G Valiant. “A theory of the learnable.” *Communications of the ACM*, **27**(11):1134–1142, 1984.
- [VBP13] Adam Vogel, Max Bodoia, Christopher Potts, and Dan Jurafsky. “Emergence of Gricean maxims from multi-agent decision theory.” In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1072–1081, 2013.
- [Vro14] Monica C Vroman. *Maximum likelihood inverse reinforcement learning*. PhD thesis, Rutgers University-Graduate School-New Brunswick, 2014.
- [WD92] Christopher JCH Watkins and Peter Dayan. “Q-learning.” *Machine learning*, **8**(3-4):279–292, 1992.
- [WDV18] Harmen de Weerd, Denny Diepgrond, and Rineke Verbrugge. “Estimating the use of higher-order theory of mind using computational agents.” *The BE Journal of Theoretical Economics*, **18**(2), 2018.
- [Wea49] W. Weaver. “Recent contributions to the mathematical theory of communication.” *ETC: A Review of General Semantics*, pp. 261–281, 1949.
- [Wil92] Ronald J Williams. “Simple statistical gradient-following algorithms for connectionist reinforcement learning.” *Machine learning*, **8**(3-4):229–256, 1992.
- [WM17] Jacob Whitehill and Javier Movellan. “Approximately optimal teaching of approximately optimal learners.” *IEEE Transactions on Learning Technologies*, **11**(2):152–164, 2017.
- [WPS19] Pei Wang, Pushpi Paranamana, and Patrick Shafto. “Generalizing the theory of cooperative inference.” In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1841–1850. PMLR, 2019.
- [WTX18] Lijun Wu, Fei Tian, Yingce Xia, Yang Fan, Tao Qin, Lai Jian-Huang, and Tie-Yan Liu. “Learning to teach with dynamic loss functions.” In *Advances in Neural Information Processing Systems*, pp. 6466–6477, 2018.
- [WVV17] Harmen de Weerd, Rineke Verbrugge, and Bart Verheij. “Negotiating with other minds: the role of recursive theory of mind in negotiation with incomplete information.” *Autonomous Agents and Multi-Agent Systems*, **31**(2):250–287, 2017.

- [WWP20] Pei Wang, Junqi Wang, Pushpi Paranamana, and Patrick Shafto. “A mathematical theory of cooperative communication.” In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pp. 17582–17593. Curran Associates, Inc., 2020.
- [WWS20] Junqi Wang, Pei Wang, and Patrick Shafto. “Sequential Cooperative Bayesian Inference.” In *International Conference on Machine Learning*, pp. 10039–10049. PMLR, 2020.
- [XD09] Fei Xu and Stephanie Denison. “Statistical inference and sensitivity to sampling in 11-month-old infants.” *Cognition*, **112**(1):97–104, 2009.
- [XT07] Fei Xu and Joshua B Tenenbaum. “Sensitivity to sampling in Bayesian word learning.” *Developmental science*, **10**(3):288–297, 2007.
- [YDF08] Wako Yoshida, Ray J Dolan, and Karl J Friston. “Game theory of mind.” *PLoS Comput Biol*, **4**(12):e1000254, 2008.
- [YFS19] Luyao Yuan, Zipeng Fu, Jingyue Shen, Lu Xu, Junhong Shen, and Song-Chun Zhu. “Emergence of Pragmatics from Referential Game between Theory of Mind Agents.” *Emergent Communication Workshop, 33rd Conference on Neural Information Processing Systems*, 2019.
- [YLF20] Tao Yuan, Hangxin Liu, Lifeng Fan, Zilong Zheng, Tao Gao, Yixin Zhu, and Song-Chun Zhu. “Joint inference of states, robot knowledge, and human (false-) beliefs.” In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5972–5978. IEEE, 2020.
- [YYW18] Scott Cheng-Hsin Yang, Yue Yu, Pei Wang, Wai Keen Vong, Patrick Shafto, et al. “Optimal cooperative inference.” In *International Conference on Artificial Intelligence and Statistics*, pp. 376–385. PMLR, 2018.
- [YZS21] Luyao Yuan, Dongruo Zhou, Junhong Shen, Jingdong Gao, Jeffrey Chen, Quanquan Gu, Ying Nian Wu, and Song-Chun Zhu. “Iterative Teacher-Aware Learning.” *Advances in Neural Information Processing Systems*, **34**, 2021.
- [ZGF20] Yixin Zhu, Tao Gao, Lifeng Fan, Siyuan Huang, Mark Edmonds, Hangxin Liu, Feng Gao, Chi Zhang, Siyuan Qi, Ying Nian Wu, et al. “Dark, beyond deep: A paradigm shift to cognitive ai with humanlike common sense.” *Engineering*, **6**(3):310–345, 2020.
- [ZHG15] Yibiao Zhao, Steven Holtzen, Tao Gao, and Song-Chun Zhu. “Represent and Infer Human Theory of Mind for Human-Robot Interaction.” In *AAAI Fall Symposia*, p. 158, 2015.

- [Zhu13] Xiaojin Zhu. “Machine teaching for bayesian learners in the exponential family.” *Advances in Neural Information Processing Systems*, p. 1905–1913, 2013.
- [Zhu15] Xiaojin Zhu. “Machine teaching: An inverse problem to machine learning and an approach toward optimal education.” In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.
- [ZJ12] Shao Zhifei and Er Meng Joo. “A survey of inverse reinforcement learning techniques.” *International Journal of Intelligent Computing and Cybernetics*, 2012.
- [ZLG03] Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani. “Combining active learning and semi-supervised learning using gaussian fields and harmonic functions.” In *ICML 2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining*, volume 3, 2003.
- [ZLH11] Sandra Zilles, Steffen Lange, Robert Holte, Martin Zinkevich, and Nicolò Cesa-Bianchi. “Models of cooperative teaching and learning.” *Journal of Machine Learning Research*, **12**(2), 2011.
- [ZM07] Song-Chun Zhu and David Mumford. *A stochastic grammar of images*. Now Publishers Inc, 2007.
- [ZMB08] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. “Maximum entropy inverse reinforcement learning.” In *Aaai*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.
- [ZSK20] Amy Zhang, Shagun Sodhani, Khimya Khetarpal, and Joelle Pineau. “Learning Robust State Abstractions for Hidden-Parameter Block MDPs.” In *International Conference on Learning Representations*, 2020.
- [ZZC15] Yixin Zhu, Yibiao Zhao, and Song Chun Zhu. “Understanding tools: Task-oriented object modeling, learning and recognition.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2855–2864, 2015.