Enabling Human-Robot Collaboration through Representation Learning

by

Mohammad Samin Yasar

A dissertation defense document submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
in
Department of Electrical and Computer Engineering
University of Virginia
September, 2024

Doctoral Committee:

Tariq Iqbal (Advisor), Assistant Professor, SIE/CS
Qing Chang (Chair), Professor, MAE/SIE
Yangfeng Ji, Assistant Professor, CS
Daniel Quinn, Associate Professor, MAE/ECE
Yixin Sun, Assistant Professor, CS/ECE

ABSTRACT

Robots are transitioning from working in isolated chambers to close-proximity collaboration with humans as part of human-robot teams. This transition is pivotal for the development of human-robot teams, where robots are not just tools but active collaborators that must seamlessly integrate into human workflows. In such collaborative settings, the success of these teams hinges on the robots' ability to effectively model and understand both human-human and human-robot team dynamics. These capabilities are crucial for anticipating human intent, making informed and timely decisions, and taking appropriate actions within a constantly changing environment.

The core challenge addressed in this research is the representation learning gap that currently limits robots' ability to fully anticipate human intentions and respond with closed-loop actions. This gap also impedes their ability to adapt to non-stationary conditions—a critical requirement for real-world applications where environments and tasks can change unpredictably. To overcome these challenges, robots must not only possess advanced perception capabilities but also the ability to retain and build upon past experiences without suffering from catastrophic forgetting.

This research is structured around three main pillars: modeling humans, modeling robot's decision-making, and joint human-robot interaction. Each pillar addresses a fundamental aspect of the robot's role in a collaborative setting.

Modeling Humans: One of the primary challenges in human-robot interaction is the accurate modeling and prediction of human motion and intent. Human behavior is inherently complex, characterized by variability, adaptability, and context-dependent actions that are difficult to capture with traditional models. Moreover, existing perception models, which are often trained solely on datasets featuring human-only interactions, struggle to generalize when applied to mixed-team environments where robots and humans interact together. To overcome these limitations, our research focused on developing advanced architectural frameworks that enhance the robot's ability to perceive and predict human actions.

We have developed several architectures which improved interpretability of motion prediction architectures [1] and incorporated multimodal and interactional context when predicting human motion [2]. Furthermore, we have also addressed some of the optimization challenges when training such generative models for motion [3]. Finally, we have introduced the PoseTron framework, a novel approach leveraging transformer-based architecture, which incorporates sequence learning and generative modeling techniques to predict human motion. PoseTron uses specialized attention mechanisms that efficiently weigh motion information from all agents in a scene, integrating this data to create a robust representation of team dynamics. This framework significantly improves the prediction of human motion in both single-agent and multi-agent settings, allowing robots to better understand and anticipate the actions of human partners in collaborative environments.

Modeling Robot's Decision Making: For robot decision-making and control, the challenge lies in enabling robots to operate effectively in dynamic and uncertain environments. Traditional approaches to robot control often involve a clear separation between planning and execution: robots generate a plan based on their current knowledge of the environment and then execute it. However, in real-world human-robot interaction scenarios, this separation can be problematic. Environmental conditions and human behaviors can change rapidly, rendering pre-established plans obsolete before they can be executed.

To address this, we developed control mechanisms that integrate planning and execution, allowing robots to continuously update their plans in real-time as they gather new information from their surroundings. This approach ensures that robots remain flexible and adaptive, capable of handling the unpredictability of human behavior. A key contribution in this area is the development of the LASSO algorithm, which decouples representation learning from policy learning. This modular architecture allows robots to learn robust representations of environmental dynamics that can be used to inform their actions, enabling them to navigate and operate effectively across a wide range of tasks and situations. LASSO, by focusing on the representation needed to forecast future states, allows for the flexible and efficient handling of both known and novel environments, ensuring robots can maintain high performance even in unfamiliar settings.

Joint Human-Robot Interaction: The ultimate goal of human-robot interaction is to achieve seamless and effective collaboration between humans and robots, where both parties can work together fluidly, in-

fluencing and responding to each other's actions in real-time. Traditional models have often assumed a unidirectional flow of information, where robots passively respond to human actions. However, real-world scenarios demand a more dynamic, bidirectional interaction model, where robots not only react to but also anticipate and influence human behavior.

To address the identified gaps, this dissertation makes several novel contributions aimed at enabling close-proximity collaboration between humans and robots.

INTERACT Dataset: One of the primary contributions is the introduction of the INTERACT dataset. Unlike traditional datasets that focus solely on human interactions, INTERACT captures the dynamic interplay between humans and robots, providing a richer, more relevant source of data for training perception models. This inclusion is critical for developing robust models that can generalize to real-world collaborative environments. The dataset encompasses a variety of scenarios, from 3 humans collaborating to 3 humans and a robot collaborating in long-horizon navigation plus manipulation tasks, ensuring that the models trained on it can adapt to different collaborative scenarios.

PoseTron Framework: The second contribution is the development of PoseTron, a novel framework for human motion prediction. PoseTron addresses the challenge of human motion prediction in collaborative environments by leveraging advanced sequence learning and generative modeling techniques. Traditional models often fail to capture the variability in human behavior, offering limited predictive accuracy. With PoseTron, we introduce a novel transformer-based architecture to address the gap in learning algorithms. PoseTron introduces a conditional attention mechanism in the encoder enabling efficient weighing of motion information from all agents to incorporate team dynamics. The decoder features a novel multimodal attention mechanism, which weights representations from different modalities and the encoder outputs to predict future motion.

CollabPolicy Benchmark: The final contribution is the introduction of CollabPolicy, a multi-agent policy learning benchmark designed to facilitate the development of collaborative strategies among agents. This benchmark provides a testing ground for evaluating langauge-conditioned imitation learning alogirthms as well as foundation models on critical challenges such as spatial reasoning, object localization, physical coordination, and collaborative decision-making and policy learning. Through this framework, we aim to advance the capabilities of policy learning models in tackling complex, real-world multi-agent scenarios.

The contributions and key findings of this work has the potential to significantly advance the field of Human-Robot Interaction (HRI) due to its focus on addressing the critical challenges of modeling human motion, enhancing robotic control, and facilitating joint human-robot collaboration. Through the development of frameworks such as the PoseTron and IMPRINT architectures, this research has improved the accuracy and robustness of human motion prediction, enabling robots to better understand and anticipate human actions in both single-agent and multi-agent settings. The introduction of the LASSO algorithm represents allows for the seamless integration of planning and execution in dynamic environments, thus enhancing the robot's adaptability and autonomy. Furthermore, the creation of the INTERACT dataset and the CollabPolicy Benchmark has the potential to provide valuable resources for the research community, enabling the development and validation of models that capture the complexities of real-world human-robot interactions. Collectively, these contributions not only address existing gaps in HRI but also pave the way for more effective, efficient, and intuitive collaboration between humans and robots in diverse and unpredictable environments.

TABLE OF CONTENTS

Acknowledgments . . . 8
List of Figures . . . 9
List of Tables . . . 10

1 Introduction **12**
1.1 Problem Space: Modeling Humans and Robots in Interaction . . . 12
1.2 Challenges . . . 12
1.2.1 Modeling Humans . . . 13
1.2.2 Modeling Robots . . . 14
1.2.3 Joint Modeling of Human-Robot Interaction . . . 14
1.3 Thesis Statement . . . 14
1.4 Completed Work . . . 15
1.5 Contributions . . . 17
1.6 Publications . . . 17

2 A Scalable Approach to Predict Multi-Agent Motion for Human-Robot Collaboration **19**
2.1 Introduction . . . 19
2.2 Related Work . . . 21
2.3 Problem Formulation . . . 22
2.4 Human Motion Prediction . . . 22
2.4.1 Single-agent Motion Prediction . . . 22
2.4.2 Multi-agent Motion Prediction . . . 25
2.5 Experimental Setup . . . 26
2.5.1 Datasets . . . 26
2.5.2 State-of-the-art methods and baselines . . . 26
2.5.3 Evaluation Metric . . . 27
2.5.4 Implementation Details . . . 27
2.6 Results and Discussion . . . 28
2.6.1 Single agent Motion Prediction . . . 28
2.6.2 Multi-agent Motion Prediction . . . 28
2.6.3 Human-Robot Collaboration Experiments . . . 29
2.6.4 Latent Space Interpretation . . . 30
2.6.5 Ablation Study of Learning Modules . . . 30
2.7 Limitations . . . 31

3 Having dynamic and learnable priors for human motion (VADER) **32**
3.1 Introduction . . . 32
3.2 Related Works . . . 34
3.3 Problem Formulation . . . 35
3.4 VADER: Vector-Quantized Generative Adversarial Networks for Motion Prediction . . . 36
3.5 Experimental Setup . . . 38
3.5.1 Datasets . . . 38
3.5.2 State-of-the-art methods and baselines . . . 38
3.5.3 Evaluation Metric . . . 39

3.6 Results and Discussion . . . 39
3.6.1 Single-agent scenario: . . . 39
3.6.2 Multi-agent scenario: . . . 40
3.6.3 Human-robot collaboration scenario: . . . 40
3.7 Ablation Study . . . 41
3.8 Limitations . . . 42

4 IMPRINT: Interactional Dynamics-aware Motion Prediction in Teams using Multimodal Context 44
4.1 Introduction . . . 44
4.2 Related Work . . . 46
4.3 Problem Formulation . . . 50
4.4 Proposed Approach . . . 51
4.4.1 Motion Encoder . . . 52
4.4.2 Interaction Module . . . 53
4.4.3 Multimodal Context Module . . . 54
4.4.4 Motion Decoder . . . 55
4.5 Experimental Setup . . . 55
4.5.1 Datasets . . . 55
4.5.2 State-of-the-art methods . . . 56
4.5.3 Evaluation Metric . . . 57
4.5.4 Implementation Details . . . 57
4.6 Results . . . 59
4.6.1 Multi-human motion Prediction . . . 59
4.6.2 Human-Robot Collaboration Experiments . . . 60
4.6.3 Ablation Experiments . . . 61
4.6.4 Significance Analysis . . . 62
4.6.5 Attention Weights Interpretation . . . 63
4.7 Limitations . . . 64

5 PoseTron: Enabling Close-Proximity Human-Robot Collaboration Through Multi-human Motion Prediction 65
5.1 Introduction . . . 65
5.2 Related Work . . . 67
5.2.1 Multimodal Datasets in HRI . . . 67
5.2.2 Human Motion Prediction . . . 68
5.3 INTERACT: HHC and HRC Dataset . . . 68
5.3.1 Study Apparatus and Implementation . . . 68
5.3.2 Human Ethics . . . 69
5.3.3 Participants . . . 69
5.3.4 Data Collection Procedure . . . 70
5.4 Multi-agent Motion Prediction . . . 71
5.5 PoseTron . . . 71
5.5.1 Multimodal Pose Encoder . . . 71
5.5.2 Multimodal Pose Decoder . . . 73
5.6 Experiments . . . 74
5.6.1 Experimental Setup and Evaluation Metric . . . 74
5.6.2 Implementation Details . . . 75
5.6.3 Results and Discussion . . . 75
5.6.4 Overall Discussion . . . 77
5.7 Interact Dataset Details . . . 78
5.7.1 Participant Roles . . . 78
5.7.2 Dataset Details . . . 79
5.7.3 Dataset Collection and Synchronization . . . 79

5.8 Surveys . . . 80
5.9 Learning Architecture Details . . . 82

6 Improving Human Motion Prediction Through Continual Learning 84
6.1 Introduction . . . 84
6.2 Problem Formulation . . . 85
6.3 Continual Learning for Human Motion Prediction . . . 86
6.3.1 Overall model for motion prediction . . . 86
6.3.2 Curriculum learning for the decoder: . . . 88
6.4 Experimental Setup . . . 89
6.4.1 Dataset . . . 89
6.4.2 Generalized Representation Learning . . . 89
6.4.3 Curriculum Learning for a specific subject . . . 89
6.4.4 State-of-the-art method and baseline . . . 89
6.4.5 Evaluation Metric . . . 89
6.5 Results and Discussion . . . 89

7 CoRaL: Continual Representation Learning for Overcoming Catastrophic Forgetting 91
7.1 Introduction . . . 91
7.2 Related Work . . . 92
7.3 Problem Formulation . . . 93
7.4 CoRaL: Continual Representation Learning . . . 94
7.4.1 Representation Learning . . . 95
7.4.2 Knowledge Distillation . . . 96
7.4.3 Overall Objective for End-to-End Learning . . . 97
7.5 Experimental Setup . . . 97
7.5.1 Datasets . . . 97
7.5.2 Continual Learning Scenarios . . . 98
7.5.3 Architectures . . . 98
7.6 Training Details . . . 98
7.6.1 Augmentation . . . 98
7.6.2 Input to the encoder . . . 99
7.6.3 Learning Architecture Details . . . 99
7.7 Sampling strategy for rehearsal . . . 100
7.8 Efficacy of the modified cosine similarity . . . 100
7.9 Details of the Continual Learning Scenarios . . . 101
7.10 Hyper-parameters in CoRaL . . . 101
7.11 Training Environment . . . 102
7.11.1 Evaluation Protocol . . . 102
7.12 Results and Discussion . . . 102
7.12.1 Incremental Task . . . 102
7.12.2 Incremental Class . . . 103
7.12.3 Incremental Domain . . . 103
7.12.4 Backward Transfer . . . 103
7.13 Ablation Study . . . 104
7.13.1 Analyzing Different Representation Learning Approaches . . . 104
7.13.2 Impact of CoRaL's Learning Modules . . . 104
7.14 Analysis of the Stability-Plasticity . . . 105
7.14.1 Effect of Varying the Plasticity on the Accuracy . . . 105
7.14.2 Effect of Varying the Stability on the Accuracy . . . 106
7.14.3 Discussion on the Stability-Plasticity Trade-off . . . 106
7.15 Conclusion . . . 107

8 LASSO: Learning Policies via State Space Modeling 108

8.1 Introduction . . . 108
8.2 Related Work . . . 110
8.3 Problem Formulation . . . 110
8.4 LASSO: Learning Latent Policies via State Space Modeling . . . 111
8.4.1 Representation Learning Module . . . 111
8.4.2 Policy Learning . . . 113
8.5 Experimental Setup . . . 114
8.5.1 Evaluation . . . 114
8.5.2 Environments . . . 114
8.5.3 State-of-the-Art Methods . . . 115
8.5.4 Learning Architecture Details . . . 115
8.6 Results and Discussion . . . 115
8.6.1 Static Environments . . . 115
8.6.2 Dynamic Environments . . . 116
8.6.3 Ablation study . . . 117
8.7 Conclusion . . . 118

9 CollabPolicies: Policy Learning for Collaborative Systems 119
9.1 Introduction . . . 119
9.2 Related Work . . . 120
9.3 Problem Formulation . . . 121
9.3.1 Single-Agent RL . . . 121
9.3.2 Multi-Agent RL . . . 122
9.4 CollabPolicy . . . 122
9.5 Experimental Setup . . . 124
9.5.1 Evaluated Algorithms . . . 124
9.6 Results and Discussion . . . 124
9.6.1 Translation Loss . . . 124
9.6.2 Gripper Loss . . . 124
9.6.3 Collision Loss . . . 125
9.6.4 Discussion . . . 125
9.7 Future Directions . . . 125
9.8 Conclusion . . . 126

10 Conclusion 127
10.1 Summary of Contributions . . . 127
10.1.1 Modeling Human Motion . . . 127
10.1.2 Robot Control . . . 127
10.1.3 Joint Human-Robot Interaction . . . 127
10.2 Lessons Learned . . . 128
10.2.1 Importance of Multimodal Data . . . 128
10.2.2 Scalability and Generalization . . . 128
10.2.3 Balancing Real-Time Performance and Model Complexity . . . 128
10.2.4 Addressing Long-Horizon Predictions . . . 128
10.3 Future Work . . . 128
10.3.1 Expanding Multimodal Fusion Techniques . . . 128
10.3.2 Enhancing Scalability for Large Teams . . . 128
10.3.3 Real-World Application and Generalization . . . 129
10.3.4 Improving Long-Horizon Predictions . . . 129
10.3.5 Developing Explainable AI in HRI . . . 130
10.4 Conclusion . . . 130

Acknowledgments:

First and foremost, I would like to express my deepest gratitude to my advisor, Professor Tariq Iqbal, for his unwavering support, guidance, and mentorship throughout the course of my research. His insights and encouragement have been invaluable, and this work would not have been possible without his expertise and dedication. From welcoming me into his lab in 2020 to supporting me through both the personal and professional challenges of my Ph.D., he has been a pillar of strength at crucial times.

I extend my heartfelt thanks to the members of my dissertation committee—Professor Qing Chang, Professor Yangfeng Ji, Professor Daniel Quinn, and Professor Yixin Sun—for their valuable feedback and suggestions, which have significantly enhanced the quality of this work. Their time and effort in reviewing my research and providing constructive criticism have been instrumental in shaping this thesis.

I am deeply grateful to my colleagues and lab mates at the Collaborative Robotics Lab, including Md. Mofijul Islam, Shaid Hasan, Sujan Sarker, and Haley Green, for creating an inspiring and collaborative environment. I also want to thank Sheikh Ziauddin Ahmed for guiding me towards the University of Virginia for my Ph.D., and Abu Sayeed Mondol, Zakaria Mehrab, and Sudipta Saha for their unwavering support as friends and mentors throughout this journey. My sincere appreciation also goes to my undergraduate collaborators—Hashneet Bhatia, T.J. Vitchutripop, Brandon Yang, Wesley Lewis, and Michael Fatemi—who have inspired me with their youthful energy and wisdom beyond their years.

This Ph.D. journey has spanned seven years, and it would not have been possible without the aspirations and unconditional trust of my parents. Their belief in my abilities nudged me to pursue higher studies in the United States, and their positivity and perseverance have been my guiding star during moments of doubt.

I would also like to acknowledge the financial support provided by the Commonwealth Center for Advanced Manufacturing (CCAM), Advanced Robotics for Manufacturing (ARM), and University of Virginia's Endowed Fellowship. This support was crucial in providing the resources and opportunities necessary to conduct my research.

Lastly, I must express my deepest gratitude to my wife, Fairuz Nawar, who is the most integral part of my life and my greatest champion. Her presence and companionship have been the driving force that pushed me to the finish line, and her unwavering faith in me continues to inspire me as I move towards the next chapter of my life.

Finally, I want to thank everyone who contributed to this thesis, directly or indirectly. Your support and encouragement have meant the world to me, and I am truly grateful for all that you have done to help me reach this milestone.

List of Figures

1.1 Examples of multi-agent human-robot and human-human interactions. 14

2.1 Qualitative performance of motion prediction methods for *walking* on UTD-MHAD. 20
2.2 Proposed framework for single-agent setting. 23
2.3 Adversarial training over the latent space. 25
2.4 Proposed framework for multi-agent setting. 26
2.5 Continuous latent space visualization using t-SNE plots on UTD-MHAD (Left) and NTU RGB+D 60 (Right) datasets. 29
2.6 Action primitives for *wave* on UTD-MHAD. 30

3.1 Qualitative evaluation of the predicted motion of VADER and the next best performing model. 33
3.2 VADER:Vector-Quantized Generative Adversarial Network for Motion Prediction. 35

4.1 Qualitative Performance of IMPRINT compared to the Ground Truth. 47
4.2 IMPRINT: Interactional Dynamics-aware Motion Prediction in Teams using Multimodal Context. 51
4.3 Change in the Multimodal Context Weights over time for each agent for the task of *handshake* 63

5.1 Samples of Close-proximity Human-Human and Human-Robot Collaboration from the INTERACT Dataset. 66
5.2 Human-Robot Collaboration samples from the INTERACT dataset. 69
5.3 Overall Architecture of PoseTron. 72
5.4 Setup for two scenarios: HHC and HRC, and thir corresponding variations. 79

6.1 Qualitative performance of different motion prediction methods for walking on UTD-MHAD. 85
6.2 Motion prediction architecture 86

7.1 CoRaL: Continual Representation Learning for Overcoming Catastrophic Forgetting. 94
7.2 The three types of Continual Learning scenarios that were considered in our experiments. . . 102
7.3 Plasticity Analysis on the S-CIFAR-10 dataset. 106
7.4 Stability Analysis on the S-CIFAR-10 dataset. 107

8.1 A Dynamic-goal Environment. 109
8.2 LASSO: Learning Policies via State Space Modeling. 111
8.3 Qualitative comparison between LASSO (left) and SAC (right) on Static Slide. 116
8.4 Performance comparison of all the evaluated benchmarks 116
8.5 Impact of forecasted states on the overall performance 117

9.1 Collaborative Policy Learning Environments 123

List of Tables

2.1 MSE (in cm^2) comparison of different single-agent methods on UTD-MHAD and KTH-HRC datasets (Lower is better). . . . 27
2.2 MSE (in cm^2) comparison of different multi-agent methods on NTU RGB+D 60 and CMU Panoptic datasets (Lower is better). . . . 28
2.3 Ablation Study of our method on UTD-MHAD. Here, SPL: Using Structured Prediction Layer, TF: Teacher Forcing. . . . 31

3.1 MSE (in cm^2) comparison of different single-agent methods on the UTD-MHAD (Lower is better). . . . 39
3.2 MSE (in cm^2) comparison of different multi-agent methods on the NTU-RGBD 60 Dataset (Lower is better). . . . 40
3.3 MSE (in cm^2) comparison of different multi-agent methods on the CMU Panoptic Dataset (Lower is better). . . . 41
3.4 MSE (in cm^2) comparison of different multi-agent motion prediction methods on the KTH-HRC Dataset (Lower is better). . . . 42
3.5 Ablation results on the UTD-MHAD Dataset (Lower is better). . . . 42

4.1 MSE (in cm^2) comparison of different multi-agent methods on the NTU RGB+D dataset (Lower is better). . . . 58
4.2 MSE (in cm^2) comparison of different multi-agent methods on the CMU Panoptic Dataset (Lower is better). . . . 59
4.3 MSE (in cm^2) comparison of different multi-agent motion prediction methods on the KTH-HRC Dataset (Lower is better). . . . 60
4.4 The results of the ablation experiments on the NTU RGB+D Dataset (Lower is better). . . . 61
4.5 Significance analysis of different motion prediction models on the NTU RGB+D Dataset (Lower is better). J+S+C: Joint Learning + Social + Context, S+I: Scalable + Interpretable, IMPRINT. § We conducted significance analysis at level $\alpha = 0.05$ (Following the procedure proposed by Dror, Shlomov, and Reichart [4]). . . . 63

5.1 Summary of Publicly Available Multimodal Human-Robot Interaction Datasets. . . . 67
5.2 Summary Statistics of the INTERACT Dataset. . . . 70
5.3 PA-MPJPE of different methods for the evaluation setup: HHC Train, HHC Test. . . . 75
5.4 PA-MPJPE of different multi-agent motion prediction methods for the evaluation setup: HRC Train, HRC Test. . . . 76
5.5 PPA-MPJPE of different multi-agent motion prediction methods for the evaluation setup: HHC Train, HRC Test. . . . 77
5.6 Ablation Study: HRC Train, HRC Test. . . . 78
5.7 Summary Statistics of the INTERACT Dataset. . . . 79
5.8 Pre-task survey questionnaire 80
5.9 Post-task survey questionnaire for Participant 1 in HHC 80
5.10 Post-task survey questionnaire for Participant 2 in HHC 80
5.11 Post-task survey questionnaire for Participant 3 in HHC 81

5.12 Post-task survey questionnaire for Participant 2 in HRC 81
5.13 Post-task survey questionnaire for Participant 3 in HRC 81
5.14 List of common notations used 82

6.1 MSE (in cm^2) comparison of fune-tuning vs no fine-tuning on UTD-MHAD for different test subjects) (Lower is better) 88

7.1 Performance comparison (averaged across 10 runs) of various CL methods on different scenarios (Accuracy in %) 97
7.2 Backward Transfer (BWT) comparison (averaged across 10 runs) on **Incremental Domain** (in %). 98
7.3 List of common notations used 100
7.4 Impact of different Cosine Similarity loss on S-CIFAR-10, averaged over 10 runs (in %). . . . 100
7.5 Hyper-parameters for CoRaL 101
7.6 Impact of Representation Learning techniques. 104
7.7 Ablation results (top-1) over different learning modules. 105
7.8 Effect of varying the plasticity parameter (α) on the average accuracy (after 5 independent runs) for S-CIFAR-10. 105
7.9 Effect of varying the stability parameter (β) on the average accuracy (after 5 independent runs) for S-CIFAR-10. 107

8.1 Success Rate on static environments (Higher is better). 114
8.2 Success Rate on dynamic environments (Higher is better). 115
8.3 Ablation of LASSO's learning modules 118

9.1 Performance of state-of-the-art approaches on multi-agent benchmark 126

Chapter 1

Introduction

Human-robot interaction (HRI) represents one of the most compelling and challenging frontiers in artificial intelligence and robotics today. As robotic systems become more integrated into everyday human environments—from manufacturing floors to healthcare facilities and domestic spaces—the dynamics of interaction between humans and robots have taken on increasing importance [5, 6, 7, 8]. Unlike interactions between only humans [9] or between isolated robotic systems [10, 11], HRI is characterized by a unique interdependence: every movement and decision made by a person influences the robot's actions, and conversely, the robot's behavior affects the human's responses [12, 13, 14, 15, 16, 17, 18, 19, 20]. This complex interplay brings forward a need to explicitly model human behavior, and robot behavior separately (see Figure 1.1), and then investigate an effective mechanism for modeling the interplay between humans and robots that transcends time, space and how these spatiotemporal factors influence the interaction dynamics [21, 2, 22]. These challenges are multi-faceted, encompassing issues of perception, prediction, decision-making, and real-time adaptation.

With the rise of foundation models trained on vast amounts of internet-scale data [23, 24, 25], we are seeing remarkable progress in core areas such as computer vision and natural language processing. However, these breakthroughs have yet to fully translate to human-robot interaction (HRI), where achieving seamless and effective collaboration remains an unresolved challenge. Central to this challenge is the need to model the dynamic and reciprocal nature of human-robot interactions. Unlike traditional human-computer interaction, where the computer typically acts as a passive receiver of commands, HRI involves continuous feedback loops where the actions of both human and robot are interdependent. This complexity is especially evident in collaborative tasks, where humans and robots must work together to achieve a common goal.

1.1 Problem Space: Modeling Humans and Robots in Interaction

In this thesis, we address the complex problem space of human-robot interaction (HRI) by methodically breaking it down into three interrelated components: modeling humans, modeling robots, and ultimately closing the loop by modeling human-robot interactions jointly. This structured approach allows us to tackle the multifaceted nature of HRI by examining each element in detail, thereby contributing to a more comprehensive understanding and on each of the facets.

1.2 Challenges

The first component of our research focuses on modeling human behavior, which is a fundamental requirement for effective HRI. Human behavior is inherently stochastic and multimodal and is a function of multiple latent variables spanning a combination of sensory inputs, cognitive processes, and social interactions. In the context of HRI, modeling human behavior involves predicting how a human will move and act in response to both the environment and the presence of a robot and/or other humans.

1.2.1 Modeling Humans

Human motion prediction is widely considered as an essential component of robotic intelligence that can enhance robot perception and enable them to react rapidly and accurately to complex changes in the environment [26, 27, 1, 28, 29]. The significance and challenges of this problem have led to extensive study in computer vision and machine intelligence. Human behavior is highly variable and context-dependent. Individuals may react differently to the same stimulus based on their personal experiences, emotions, and situational factors. This variability makes it difficult to develop models that can accurately predict human actions across different scenarios. For instance, predictive models often struggle to generalize across diverse populations or situations [30]. Moreover, humans are capable of rapid, sometimes unpredictable, changes in behavior, which can complicate the task of real-time prediction in dynamic environments [31].

Architectural Challenges: Some of the earliest methods for modeling motion prediction involved using Hidden Markov Models and Gaussian processes. Lehrmann et al. [32] proposed latent-variable models that follow state-space equations modeled by Hidden Markov Models. Taylor et al. [33] introduced the use of conditional restricted Boltzmann machines (RBM) for motion prediction. Wang et al. [34] used Gaussian-Processes to perform non-linear motion prediction.

These methods have given way to more data-driven and learning-based architectures [35, 29, 36, 1, 3]. When training these networks, the core assumption is to either learn a distribution that can fit a static prior [1, 37, 38, 39], or to learn a point estimate over the past observed data which will be used to predict future human motion [29, 36, 28]. Learning a static prior introduces an auxiliary objective that acts as a regularizer which requires careful tuning, whereas learning a point estimate leads to less robust representations. Furthermore, prior works have relied on the reconstruction error as the sole objective for training these networks. Such an objective function may cause the predictions to regress to the mean and, as such, may not be able to capture the spatial and temporal correlations in human motion.

Predicting motion in group settings: Previous works on human-robot collaboration have explored the concepts of joint action by modeling human activities and using that knowledge as an input to the robot's anticipatory action planning mechanism [40, 41, 42, 43, 44]. However, the majority of these methods have been modeled from the perspective of dyadic interaction, comprising one human and one robot [45, 38]. Transitioning from dyadic interactions to interactions involving multiple humans, as seen in (Figure 1.1b) and potentially multiple robots constitutes a fundamental change in complexity that is difficult to solve using existing dyadic algorithms [46, 47]. Going beyond dyadic interactions would require robots to understand the inter-agent and intra-agent dynamics within the team while also being cognizant of any external stimuli that may affect the behavior of the team as a whole. Prior approaches for modeling multi-agent interactional dynamics have approached the problem from a social navigation perspective and primarily focused on predicting the trajectories (2-D global positions) of multiple humans in a scene [30, 48, 49, 50]. However, when humans and robots collaborate in close-proximity, robots would require accurate forecasting of human motion (3-D skeletal joint positions) instead of just the global 2-D trajectories.

Multimodal Data: Another challenge is effectively integrating multi-modal data to build comprehensive models of human behavior. Humans rely on various sensory modalities such as visual, auditory, proprioceptive to interact with their environment. However, processing and combining these diverse data streams into a coherent model that accurately reflects human behavior remains a significant challenge [51]. Furthermore, the data from different modalities may be noisy, asynchronous, or incomplete, complicating the task of reliable behavior prediction [35].

Interaction Dynamics: Modeling human behavior becomes even more complex in multi-agent settings where social dynamics play a crucial role. Interactions between multiple humans involve subtle cues, such as body language and gaze, which are challenging for robots to perceive and interpret accurately [52]. Additionally, the presence of a robot can alter human behavior, introducing further unpredictability that models must account for [53, 54].

(a) Dyadic (Hugging [55]) vs Multi-agent (Haggling [56]) human-human interaction

(b) Dyadic (Hand-shaking [38]) vs Multi-agent (Dancing) human-robot interaction [57]

Figure 1.1: Examples of multi-agent human-human and human-robot interactions. In each scenario, there is a need to model the interactional dynamics and the multimodal context in order to understand and predict the behavior of all the humans accurately.

1.2.2 Modeling Robots

The second component of our research focuses on modeling robotic behavior, particularly in the context of HRI. Unlike humans, robots do not inherently possess the ability to understand or predict human behavior; they must be programmed or trained to do so. This necessitates the development of sophisticated algorithms that enable robots to plan and execute actions in a way that anticipates human movements and adapts to changing circumstances.

Dynamics-Aware Decision-Making: One of the key challenges is enabling robots to make decisions that are safe, effective, and contextually appropriate. In HRI, robots must continuously update their understanding of the environment and the human's state, requiring real-time processing and adaptation [58]. However, the computational complexity of real-time decision-making, especially in uncertain and dynamic environments, poses significant challenges [59].

Learning from Interaction: For robots to collaborate with humans, they must also be capable of learning from their interactions with humans, and/or other robots. However, learning in real-time from noisy, ambiguous, or sparse feedback is challenging [60]. Reinforcement learning approaches, for example, can require large amounts of data and may struggle with the complexities of human-robot interaction, such as interpreting non-verbal cues or understanding implicit goals.

1.2.3 Joint Modeling of Human-Robot Interaction

The final component of our research focuses on the joint modeling of human-robot interaction. In true HRI, the actions of the human and the robot are tightly coupled, with each influencing the other in a continuous feedback loop. Effective joint modeling requires not only an understanding of human behavior and robotic capabilities but also an ability to predict and manage the interactions between the two.

One of the key challenges in joint modeling is the need to account for the uncertainty and variability in human behavior. Humans are not always predictable, and their actions can be influenced by a wide range of factors, including emotions, social norms, and environmental conditions [61]. To address this challenge, we have explored the use of probabilistic models and deep learning techniques to capture the uncertainty and variability in human behavior, and to integrate this information into the robot's decision-making process.

1.3 Thesis Statement

As robots transition from isolated operation to becoming integral members of human-robot teams, there is a pressing need for novel frameworks that can model the complex, dynamic interactions between humans and robots, integrate multi-modal data for enhanced situational awareness, and generalize across diverse scenarios. This thesis addresses the need for novel frameworks that model complex human-robot interactions, integrate multi-modal data for enhanced situational awareness, and generalize across diverse scenarios, while introducing two comprehensive datasets and benchmarks to simulate real-world collaboration.

1.4 Completed Work

Modeling Humans: For the first thrust, we aimed to address some of the open challenges of Robot Perception: anticipating human motion and intent, and learning representations under non-stationarity. Human motion prediction is widely considered one of the essential parts of robotic intelligence that would enhance robot perception. Towards this end, we have made several architectural contributions.

- Firstly, we proposed a scalable and interpretable encoder-decoder approach for predicting human motion in both single and multi-agent settings [1]. Our encoder explicitly incorporates velocity and acceleration features alongside skeletal positions, improving representation by leveraging an attention mechanism that adaptively weighs these features for more robust information capture, as compared to traditional pooling mechanisms. The resulting latent representation, composed of continuous and categorical variables, is used by the decoder to forecast future trajectories in an auto-regressive manner, conditioned on a subset of past sequences rather than just the last predicted frame. For multi-agent settings, we employ separate encoders and decoders for each agent, with a novel attention-based mechanism that disentangles and weighs relevant features to produce a shared latent representation, modeling both categorical and continuous aspects of inter-agent dynamics, which each agent-specific decoder uses to refine its predictions.

- Next, we introduced VADER, [3], a novel approach that aims to close two critical gaps in motion prediction: 1) learning a robust representation of the past motion and 2) improving temporal and spatial correlation in the prediction. VADER is built on top of the encoder-decoder framework but augments it with codebook learning and distribution matching. Our approach uses the expressive powers of codebooks to learn discrete representations over the observed motion data. Furthermore, as motion prediction introduces high data dependencies, it makes any regular MSE-based objectives ill-posed. As such, we propose a novel discriminator-based loss to increase the temporal and spatial coherency by penalizing predictions that deviates from the ground-truth distribution.

- Next, we explicitly consider interaction dynamics, and multimodal representation by proposing IMPRINT: Interactional Dynamics-aware Multi-agent Motion Prediction in Teams using Multimodal Context, a model that explicitly captures team interaction dynamics and fuses multimodal context from non-skeletal modalities to predict human motion in team settings. Drawing inspiration from joint action [9] research and integrating it with current motion prediction methods [1, 3], IMPRINT uses an encoder-decoder architecture augmented with specialized modules. These include: i) an encoder to learn rich representations of past observations, ii) an inter-agent attentional mechanism that models interactional dynamics and fuses relevant information adaptively, iii) a multimodal attentional mechanism to integrate complementary information from various modalities, and iv) a decoder that uses these dynamics and context to predict future motion of all team members auto-regressively.

- Following the development of IMPRINT, we addressed the need for improved multi-agent motion prediction by introducing Posetron, a novel and efficient transformer-based architecture. Posetron uses an encoder-decoder framework to extract spatio-temporal representations of human motion, fuse diverse modalities, and capture interaction dynamics among agents. The encoder employs self-attention to identify agent-specific motion patterns and conditional attention to incorporate team dynamics by querying other agents' representations.

  The encoder's output, which includes both skeletal and non-skeletal representations, is combined with the last observed motion and passed to the decoder. The decoder uses an auto-regressive approach to predict future motion, further refining its predictions through conditional attention that integrates key features from the encoder and its own generated outputs. Posetron effectively enhances multi-agent motion prediction, making it a valuable addition to the field of Human-Robot Interaction.

Modeling Robots: We have also explored the interleaving of planning and execution in robot control. Traditional approaches to robot control often involve a clear separation between planning and execution: the robot first generates a plan based on its current knowledge of the environment, and then executes that plan [59]. However, in dynamic and uncertain environments, this separation can be problematic, as the plan may become outdated or irrelevant by the time it is executed.

To address this issue, we have investigated approaches that integrate planning and execution, allowing the robot to continually update its plan as it gathers new information from the environment. This approach is particularly important in HRI, where the robot must be able to adapt to the unpredictable and changing behavior of the human partner [62]. Our proposed algorithm LASSO aims to learn a robust representation of the underlying dynamics of the environment, which can then be reliably used for policy learning. LASSO comprises a Representation Learning module and a Policy Learning module. We decouple Policy Learning from Representation Learning, thus allowing for a modular architecture, which is trained with specific objective functions. This modular architecture provides the flexibility of learning a representation that is geared toward obtaining a model of the environment dynamics. The learned representation can then be used by the Policy Learning module as a reliable approximation of the state. LASSO can be viewed as lightweight extension to model-free approaches by learning a robust representation from the experience buffer, similar to prior representation learning approaches such as CURL [63]. Unlike model-based approaches [64, 65] which perform planning on the world model and as such may not require a policy network, LASSO does not propagate stochastic gradients of returns to its learned representation, instead only focusing on the representation to forecast future states.

Datasets for Human-Robot Collaboration: One of the primary challenges in advancing Human-Robot Interaction (HRI) is the scarcity of comprehensive real-world datasets that capture scenarios involving robots collaborating with multiple humans. This lack of data represents a significant bottleneck, as the development and validation of algorithms for accurately predicting human motion and intention rely heavily on the availability of such datasets [66, 38, 67].

To address these challenges, we developed the INTERACT dataset, a comprehensive and large-scale collection of multimodal data designed to advance the field of human-human and human-robot collaboration. INTERACT is unique in its focus on both Human-Human Collaboration (HHC) and Human-Robot Collaboration (HRC) within team-based tasks, marking a significant shift from traditional dyadic datasets to more complex, multi-participant interactions. The dataset features assembly tasks involving three participants in HHC scenarios and four participants in HRC scenarios, with one of the participants being a robot.

INTERACT includes a rich array of synchronized multimodal data, such as 3-D human skeletal joint positions, RGB and depth data of the workspace from two viewpoints, ego-view data from human participants, eye-tracking and gaze data, and robot joint data. This comprehensive dataset provides an exhaustive view of the collaboration tasks, enabling the development of more sophisticated models for predicting human motion and understanding interaction dynamics. We collected data from 63 participants, organized into 21 groups for both HHC and HRC scenarios, resulting in approximately 1 million samples of synchronized multimodal data. The availability of this dataset will support the development of algorithms that better reflect the complexities of real-world HRI, moving beyond dyadic interactions to encompass the collaborative dynamics of teams.

Joint modeling of human-robot interaction: Traditional approaches to HRI often assumed a unidirectional flow of information, where the robot responds to the human's actions without influencing them in return [68]. However, in many real-world scenarios, the interaction is bidirectional, with the robot's actions influencing the human's behavior and vice versa [69, 20].

To address this challenge, we have explored the use of multi-agent scenarios where two robots must collaborate on shared objectives. These agents interact with each other in a shared environment, and their behavior is influenced by both their own goals and the actions of the other agent.

Our preliminary work in this area has focused on scenarios where multiple robots collaborate in a shared task environment. In these scenarios, the robots must not only coordinate their actions with each other but also with the human participants. This adds a layer of complexity to the interaction, as the robots must be able to understand the intentions and actions of both their human and robotic counterparts.

1.5 Contributions

This thesis makes important contributions to the field of Human-Robot Interaction by addressing critical challenges in modeling human behavior, enhancing robotic control, and developing frameworks for joint human-robot interactions. In the area of human motion modeling, we developed several novel architectural frameworks that enhance the prediction of human motion in both single-agent and multi-agent settings. Our contributions include a scalable encoder-decoder approach that incorporates velocity and acceleration features alongside skeletal positions, improving the robustness and accuracy of motion predictions through adaptive attention mechanisms. Additionally, we introduced VADER, a framework that addresses the challenges of learning robust motion representations and maintaining temporal and spatial coherence in predictions. We further expanded on this by developing IMPRINT and Posetron, architectures designed to model complex interaction dynamics and multimodal contexts in team-based settings, thereby advancing the state of the art in multi-agent motion prediction.

In addition to human modeling, this thesis contributes to the modeling of robotic behavior, particularly in dynamic and uncertain environments. We proposed a novel approach that integrates planning and execution, allowing robots to continuously update their plans based on real-time information, thus enhancing their adaptability and autonomy in Human-Robot Interaction (HRI) settings. Our work culminated in the development of the LASSO algorithm, which decouples representation learning from policy learning, enabling robots to learn a robust representation of environmental dynamics that can be reliably used for policy learning. This modular architecture provides flexibility and improves the robot's ability to handle a wide range of tasks in unpredictable environments.

Finally, this thesis also contributes to the understanding and modeling of joint human-robot interactions, particularly in scenarios where multiple agents must collaborate to achieve shared objectives. We recognized the limitations of traditional unidirectional HRI models and explored bidirectional interaction dynamics, where the actions of both humans and robots influence each other. Our work focused on multi-agent scenarios, including the development of the INTERACT which captures complex human-human and human-robot collaboration tasks, and CollabPolicy datasets which involves close-proximity manipulation among two robots. By addressing the challenges of modeling these interactions, our research paves the way for more effective and seamless collaboration between humans and robots in real-world environments.

1.6 Publications

1. S. Hasan, **M. S. Yasar**, and T. Iqbal, "M2RL: A Multimodal Multi-Interface Dataset for Robot Learning from Human Demonstrations," in *Proceedings of the International Conference on Multimodal Interaction (ICMI)*, 2024.
2. **M. S. Yasar**, M. M. Islam, and T. Iqbal, "PoseTron: Enabling Close-Proximity Human-Robot Collaboration Through Multi-human Motion Prediction," in *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2024.
3. **M. S. Yasar**, M. M. Islam, and T. Iqbal, "IMPRINT: Interactional Dynamics-aware Motion Prediction using Multimodal Context," *Transactions on Human-Robot Interaction (THRI)*, 2023.
4. **M. S. Yasar** and T. Iqbal, "VADER: Vector-Quantized Generative Adversarial Network for Motion Prediction," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.
5. **M. S. Yasar**, "Learning Transferable Representations for Non-stationary Environments," in *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems Doctoral Consortium (DC-AAMAS)*, 2023.
6. **M. S. Yasar** and T. Iqbal, "CoRaL: Continual Representation Learning for Overcoming Catastrophic Forgetting," in *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2023.

7. M. M. Islam, **M. S. Yasar**, and T. Iqbal, "MAVEN: A Memory Augmented Recurrent Approach for Multimodal Fusion," *IEEE Transactions on Multimedia*, 2022.
8. **M. S. Yasar** and T. Iqbal, "Robots That Can Anticipate and Learn in Human-Robot Teams," in *ACM/IEEE International Conference on Human-Robot Interaction (HRI) Pioneers*, 2022.
9. **M. S. Yasar** and T. Iqbal, "Improving Human Motion Prediction Through Continual Learning," in *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, LEAP-HRI Workshop, 2021.
10. **M. S. Yasar** and T. Iqbal, "A Scalable Approach to Predict Multi-Agent Motion for Human-Robot Collaboration," *IEEE Robotics and Automation Letters*, 2021. Presented at the *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

Chapter 2

A Scalable Approach to Predict Multi-Agent Motion for Human-Robot Collaboration

Human motion prediction is considered a key component for enabling fluent human-robot collaboration. The ability to anticipate the motion and subsequent intent of the partner(s) remains a challenging task due to the complex and interpersonal nature of human behavior. In this work, we propose a novel sequence learning approach that learns a robust representation over the observed human motion and can condition future predictions over a subset of past sequences. Our approach works for both single and multi-agent settings and relies on an interpretable latent space that has the implicit benefit of improving human motion understanding. We evaluated the proposed approach by comparing its performance against state-of-the-art motion prediction methods on single, multi-agent, and human-robot collaboration datasets. The results suggest that our approach outperforms other methods over all the evaluated temporal horizons, for single-agent and multi-agent motion prediction. The improved performance of our approach for both single and multi-agent settings, coupled with an interpretable latent space, can enable close-proximity human-robot collaboration.

2.1 Introduction

Understanding human motion is a crucial skill for robots to coexist and collaborate with humans [15, 20, 70]. Humans develop the ability to engage in joint action during infancy and early childhood, through a combination of observation, active participation and explicit teaching [38, 9]. As such, humans are innately adept at anticipating the motion and intent of other persons over varying horizons [71, 72]. This is best observed in team activities, where two or more individuals can understand and predict each other's motion [73, 9]. Along these lines, for robots to fluently collaborate with humans, they need to combine aspects of perception, representation, and motion analysis, to accurately anticipate the motion of surrounding individuals [74, 75, 76, 77, 78]. In addition, the robot's perception and decision-making processes need to be explainable to human collaborators for enabling close human-robot collaboration.

Human motion is often modeled by tracking the movement of the skeletal joints over time [79, 35, 29, 36, 37]. Several approaches have modeled the problem of predicting human motion as that of forecasting future trajectories, conditioned on past observed trajectories, in a sequence-to-sequence manner [35, 29, 36]. Prior work can be broadly categorized into deterministic approaches: learning point estimates over future trajectory [35, 29, 36], and probabilistic approaches: learning a distribution over future trajectories using latent variables [37, 39, 38]. Although the aforementioned works have shown promising results, predicting human motion *quantitatively*, in terms of some evaluation metric such as Mean Squared Error and *qualitatively*

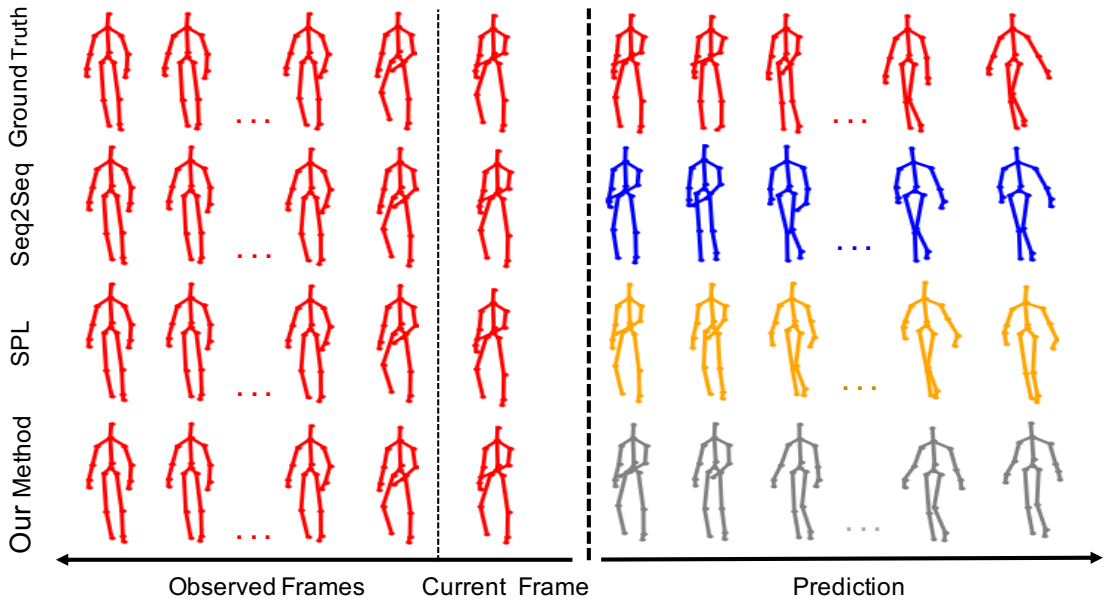


Figure 2.1: Qualitative Performance of different motion prediction methods for *walking* on UTD-MHAD. Our method produces more feasible joint poses by maintaining relative orientation of each joints, while achieving the best quantitative performance.

(see Fig. 2.1), in terms of generating feasible and realistic motion remains a challenging task [29, 36]. This highlights the need to learn a more robust representation of observed trajectories.

Predicting the motion of just one person is not enough for a robot to be successful in a team. It is expected to work with multiple people and needs to capture the inter-agent dynamics to accurately predict the motion of all individuals. Prior work on multi-agent forecasting has primarily used the global motion (2D positions) for modeling the interaction among all the agents [30, 48, 80, 50]. These approaches do not consider the local pose or skeletal joints of the humans, thus only modeling coarse information about human trajectory. To incorporate skeletal pose, recent work introduced a joint-learning framework that models both skeleton positions and global 2D positions, for multi-agent settings [28]. However, the approach relies on pooling mechanisms (e.g., [30, 48]) to model the interaction among multiple agents, which are prone to losing valuable information while being invariant to small changes in input [81, 82].

To address the above challenges, we propose an encoder-decoder approach that is *scalable*: predicting human motion for single and multiple agents, and *interpretable*: disentangling relevant aspects of human motion. The encoder architecture of our approach differs from prior works [35, 29, 48, 83] by explicitly considering velocity, and acceleration features in addition to skeletal positions, to obtain a more salient representation over past motion. These features are fed to an attention mechanism [84], which learns to adaptively weigh the different motion features and is more robust at capturing relevant information, compared to pooling mechanisms. The output from the attention mechanism is used to obtain the latent representation, which comprises of continuous and categorical random variables. The decoder then uses these latent variables to forecast future trajectories in an auto-regressive manner. We differ from previous work by learning to condition the decoder output on a subset of the past sequences, instead of just the last predicted frame.

In settings of more than one agent, we use separate encoders and decoders to model the motion for each agent. To model inter-agent dynamics our approach relies on a novel attention-based mechanism that learns to weigh relevant features from each agent to produce a disentangled multi-agent representation. This is used to compute the shared latent representation for all agents, which models the categorical and continuous aspects of multi-agent interaction. The output of the latent space is then used by each agent-specific decoder to condition its prediction between the immediate agent-specific prediction and the latent variables that

represent multi-agent interaction.

We evaluated the performance of our approach on single-agent settings on the UTD-MHAD [85], multi-agent settings on the NTU RGB+D 60 [55] and CMU Panoptic [56] datasets, and human-robot collaboration scenarios on the KTH Human-Robot Collaboration (KTH-HRC) dataset [38]. The results suggest that our approach outperformed state-of-the-art human motion prediction methods over all the evaluated horizons for single-agent and multi-agent settings. Finally, we provide an interpretation of the underlying generative process of human motion by exploring the latent space. Our findings suggest that the categorical latent variables learn to segment an action into separate action primitives while the continuous latent variables learn to cluster activities with similar spatial semantics.

2.2 Related Work

Human motion prediction: Recent work on human motion prediction has predominantly posed the problem as that of sequence learning, modeled using Recurrent Neural Nets in an encoder-decoder framework [35, 79, 29, 37, 36]. Martinez et al. [29] showed that weight sharing between the encoder and decoder results in quicker convergence. Furthermore, they model velocity representation at the decoder by introducing a residual connection. To explicitly encode the skeletal hierarchy, prior work has modeled the kinematics chain at the encoder by dividing the skeleton into 5 major clusters [79] or following the kinematic chain starting from the end-effectors [86]. Aksan et al. [36] proposed structured prediction at the decoder, by introducing a Structured Prediction Layer which decomposes the model prediction into individual skeletal joints, each predicted in a hierarchical sequence. While most works on motion prediction adopt a deterministic approach, recent work has approached the problem as that of learning a probability density function of future human poses conditioned on previous poses [39, 83, 37, 38].

Butepage et al. [37] and Toyer et al. [39] adopted the Variational Autoencoder (VAE) framework for motion prediction, which rely on learning a functional mapping from the data space to the latent space at the encoder, with the decoder sampling from this latent space to generate future human motion. Barsoum et al. [83] proposed a modified version of Wasserstein GAN (WGAN-GP) with the model input being a sequence of past human poses plus a random vector z.

Multi-agent motion prediction: Multi-agent forecasting is widely considered a challenging problem as the agents' policies are not directly accessible. Several data-driven approaches have been applied to forecast complex interactions in social navigation [30, 87, 48], autonomous vehicles [49, 80, 50] and HRI settings [88, 89]. Alahi et al. [30] introduced social-LSTM, which uses agent-specific LSTMs to summarize past observations of each agent. The hidden states of the neighboring LSTMs are connected through a social pooling strategy and used as the input to the LSTM cell at the next timestep. Gupta et al. [48] proposed Social GAN, which introduced a computationally efficient pooling mechanism comprising of a Multi-Layer Perceptron followed by max pooling. While the aforementioned works only consider the global motion of the agents, in particular 2D locations, Adeli et al. [28] jointly modeled global and local movement by incorporating skeleton positions. Despite the promising performances of these methods, the pooling mechanism commonly used in these approaches runs the risk of losing valuable information while being invariant to small changes in input [81], thereby learning a sub-optimal representation.

Human motion interpretation: Interpreting the learned representation of deep learning frameworks is crucial to their acceptability for any application. The problem of latent space learning and interpretation for images has been extensively studied and introduced several seminal approaches [90, 91, 92, 93]. In comparison, work on understanding the underlying generative process of human motion is less explored. For human-robot collaboration, robot perception needs to be explainable. Prior work has modeled various aspects of human-robot collaboration from human motion [37] to robot motion [94] and emotion [95] using continuous latent variables, while providing an intuitive explanation of the learned latent representation. However, these approaches learn the latent representations over simple motion (*reaching* or *pouring*) and cannot not capture the high level dynamics of human motion.

Although the aforementioned works show promising results, learning effective representations that summarize the observed trajectory at the encoder remains an open problem. In addition, the decoder network in

most approaches condition only on the past generated frame. This results in performance degradation over long-term horizons and is not suited for multi-agent settings where there is a need to consider cross-agent interaction. To this end, prior approaches rely on pooling over encoder representations of multiple agents, which can lead to losing relevant information. Finally, prior works on human motion interpretation focus on learning representations over simple actions and fail to capture the high level dynamics of human motion. To address these challenges, we propose an encoder-decoder approach for human motion prediction, which we describe in section 2.4.

2.3 Problem Formulation

Our goal is to accurately predict the motion of all agents in a given workspace. We assume that the number of agents, m is known. In all our formulations, we use superscript to represent agents and subscript to represent time.

For simplicity, let us first assume that there is one agent a and we have access to the agent's trajectory, spanning time $t = 1$ to τ, with observed trajectory frames: $\mathbf{X}^a = \{x_1^a, \ldots, x_\tau^a\}$. We pose the motion prediction problem as predicting future trajectory frames over a horizon H: $\mathbf{Y}^a = \{y_{\tau+1}^a, \ldots, y_{\tau+H}^a\}$, conditioned on the observed frames $\mathbf{X}^a$. Each frame $x_t^a \in \mathbb{R}^N$ denotes the N-dimensional body pose. N depends on the number of joints in the skeleton, J and the dimension of the joints D, where $N = J \times D$.

We assume that future human pose is conditioned on the past observed or generated poses, and predict each frame in an auto-regressive manner as formulated below:

$$p_\theta(\mathbf{Y}^a) = \prod_{\delta=\tau+1}^{\tau+H} p_\theta(y_\delta^a | y_{\tau:\delta-1}^a, x_{1:\tau}^a) \tag{2.1}$$

where the joint distribution is parameterized by θ.

In the case of multiple agents, we assume that the future pose for each agent is conditioned on the observed poses of all the agents and generated pose of the specific agent. As such, we can extend Eq. 2.1 for each agent a as follows:

$$p_\theta(\mathbf{Y}^a) = \prod_{\delta=\tau+1}^{\tau+H} p_\theta(y_\delta^a | y_{\tau:\delta-1}^a, x_{1:\tau}^{1:m}); \forall a = 1, \ldots, m \tag{2.2}$$

2.4 Human Motion Prediction

Our approach has the overarching goal of accurately predicting human motion while being scalable and interpretable. It comprises of an encoder-decoder, trained end-to-end, with adversarial regularization on the latent variables. To address the challenges of learning a robust representation, the encoder explicitly models position, velocity, and acceleration information. The decoder conditions its output on both the latent representation and the immediate past frame, thus attaining performance gain over long horizons. For multi-agent settings, our approach uses an attention mechanism to model the inter-agent dynamics, thus learning a more robust representation. We will first describe our framework for single-agent motion prediction and then discuss its scalability for predicting the motion of multiple agents.

2.4.1 Single-agent Motion Prediction

Our framework for a single-agent setting comprises of one encoder-decoder, along with adversarial training (see Fig. 2.2).

Multi-stream Encoder: The encoder aims to learn a spatio-temporal representation over the past observation for a given agent. To obtain a rich and more robust representation over the past trajectories, we extract the past velocity and acceleration features along with the provided positional values, thus forming

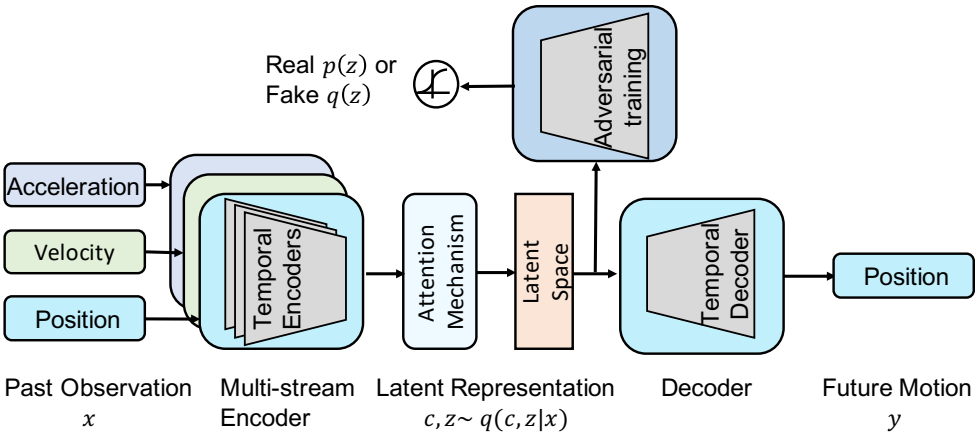


Figure 2.2: Proposed framework for single-agent setting.

a multi-stream input for the encoder. The velocity and acceleration features are first and second-order derivative of the position values for each skeleton joint.

As we pose this as a sequence learning problem, we employ Recurrent Neural Networks, in particular unidirectional Gated Recurrent Units (GRU), to extract temporal feature representations for each stream. Our choice of unidirectional GRUs over a bi-directional architecture is motivated by our need to predict human motion in real-time. We choose GRUs due to their comparative performance to LSTMs while having computational advantages. For each stream, the stream-specific GRU aims to encode the spatio-temporal information over the input sequence, which is formulated as:

$$h_{s,t} = GRU(h_{s,t-1}, x_{s,t}, \phi_s) \tag{2.3}$$

where s represents position, velocity, or acceleration. Here, $x_{s,t}$ represents the input to the GRU at time t and will take the value of $x_{pos,t}, x_{vel,t}, x_{acc,t}$ for position, velocity and acceleration, respectively. $h_{s,t-1}$ represents the past hidden output and ϕ_s represents the stream-specific encoder weights for the GRU. The output from each GRU is passed to a multi-head self-attention module [84]. The attention module is tasked to sparsely and adaptively extract the salient features from the three streams.

$$h_t = \text{Concat}(h_{\text{pos},t}, h_{\text{vel},t}, h_{\text{acc},t});\ h_{\text{att},t} = \text{Att}(h_t, \phi_{\text{att}}) \tag{2.4}$$

In the self-attention module the concatenated output, h_t is at first linearly projected to query (Q), key (K), and value (V) embedding for each head. The embeddings are used to compute attention weights using the scaled-dot product softmax (sf) approach. The overall functions for each head in the multi-head self-attention module are formulated below:

$$\begin{aligned} Q &= h_t W^Q; \\ K &= h_t W^K; \\ V &= h_t W^V \\ \text{Att}(Q, K, V) &= \text{sf}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \end{aligned} \tag{2.5}$$

where, W^Q, W^K, W^V represent the linear projection weights and $\frac{1}{\sqrt{d_k}}$ is the scaling factor for calculating the attention weights.

Latent Variables: Our proposed approach aims to learn a distribution over past observations similar to previous work [39, 86, 38], but differs in terms of the latent space representation and regularization. The core

assumption underlying such approaches is that the past observations and future trajectories are generated by some random process involving unobserved latent variables. Unlike prior approaches, our framework models both continuous Z and categorical random variables C as part of the latent space.

In line with prior work on representation learning for images [92, 91], our framework augments the continuous latent distribution with a relaxed discrete distribution, but for human motion modeling. The motivation here is to disentangle and model continuous aspects of human motion such as the style of the agent, as well as discrete information such as class activity or action primitive.

To obtain the continuous latent variable z_t, the output from the self-attention module is passed through a linear layer (Lin), whereas in the case of the categorical latent variable c_t, the output from the self-attention module is passed through a linear layer followed by a softmax (sf) layer.

$$z_t = Lin(h_{att,t});\ h_{c,t} = Lin(h_{att,t});\ c_t = sf(h_{c,t}) \tag{2.6}$$

Discriminators: In line with previous frameworks on latent space learning and regularization, such as Variational Autoencoders (VAEs) [90], Joint-VAE [92] and Adversarial Autoencoders (AAE) [91], we enforce a prior on the latent variables. We differ from prior work on motion generation that use KL-divergence for enforcing a prior [86, 39], instead using adversarial training, thus adopting the AAE framework for motion generation [91]. Our choice of using adversarial training is to avoid tuning the KL-divergence loss that is often small compared to the reconstruction loss and requires a scaling factor β as well as an annealing schedule.

In our framework, the encoder aims to confuse the discriminators by trying to ensure that its output is similar to the aggregated prior. The discriminators are trained to distinguish the true samples generated using a given prior, from the latent space output of the encoder, thus establishing a min-max adversarial game between the networks [93, 91].

We use two discriminators, one for the continuous latent variable and the other for the categorical latent variable, as shown in Fig. 2.3. The discriminators compute the probability that a point z_t or c_t is a sample from the prior distribution that we are trying to model (positive samples), or from the latent space (negative sample). The discriminator loss, which is high if the generated sample from the encoder is coming from a different distribution compared to the prior, is used to update the parameters of the encoder, thus enforcing it to produce samples similar to the prior. We use a Gaussian prior for continuous latent variables and a uniform distribution prior for categorical latent variables.

Decoder: The decoder is auto-regressive, i.e., it uses the output of previous timesteps to predict the current pose, and has only one stream: position. The input to the decoder is the latent representation, summarizing the past observations as well as the immediate hidden representation of the last predicted frame. This is passed to a multi-head self-attention module, similar to one at the encoder, which learns the attention weights between the previous output and the latent variables that summarize past frames.

The first part of the decoder is a GRU cell, that takes as input the output of the multi-head self-attention module as well as the output of the last timestep. This is followed by either a fully connected layer or a Structured Prediction Layer (SPL) [36], which aim to explicitly model the spatial structure of the joints by hierarchically predicting each joint, instead of treating each joint individually. The operations at the decoder are formulated as follows:

$$\begin{aligned} p_t &= Concat(z_t, c_t, h_{dec,t-1});\ p_{att,t} = Att(p_t, \phi_{att}) \\ h_{dec,t} &= GRU(S_{t-1}, p_{att,t}, \phi_{pos});\ S_t = \gamma(h_{dec,t}) \end{aligned} \tag{2.7}$$

where z_t and c_t are the latent variables, $h_{dec,t-1}$ is the previous hidden output of the GRU. $p_{att,t}$ is the output of the attention mechanism in the decoder, which is passed to the GRU along with the previous GRU output S_{t-1}. ϕ_{att} and ϕ_{pos} represents the weights of the attention module and GRU cell respectively. γ represents the output layer of the decoder with $\mathbf{S_t}$ being the *predicted motion at time t*. We add a residual connection between decoder output at the last and current timestep, which improves short-term prediction and result in smoother output sequence [29].

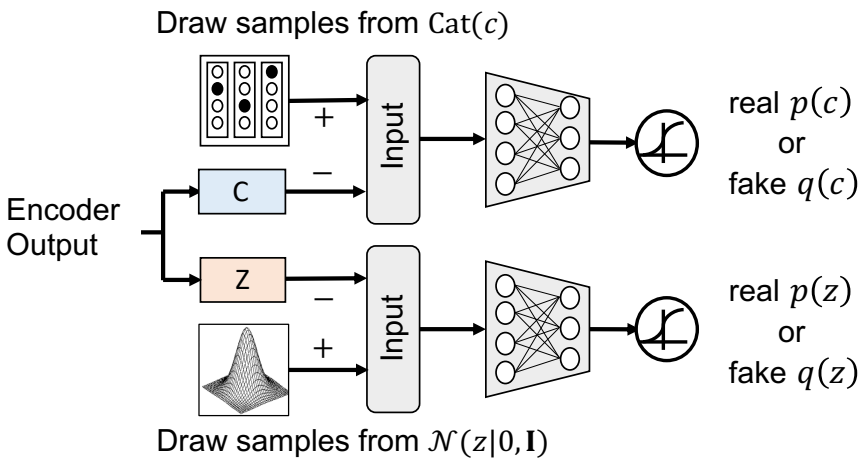


Figure 2.3: Adversarial training over the latent space.

2.4.2 Multi-agent Motion Prediction

In addition to addressing the challenges of learning a robust representation and improving the interpretability of single-agent motion prediction, our approach can be scaled to predict motion for multiple agents.

Multi-stream Encoder: Each agent's motion is modeled by an agent-specific multi-stream encoder that learns a spatio-temporal representation over the past trajectories. The operations per-agent are similar to the ones in Eq. 2.3. For m agents, there will be m number of encoders and decoders, matching the number of agents (see Fig. 2.4).

To obtain a robust cross-agent interaction, the output of all the encoders is passed to a multi-head self-attention module. The operations can be summarized as:

$$\begin{aligned} h_t^a &= Concat(h_{pos,t}^a, h_{vel,t}^a, h_{acc,t}^a) \\ h_t &= Concat(h_t^a, \ldots, h_t^m); h_{att,t} = Att(h_t, \phi_{att}) \end{aligned} \tag{2.8}$$

where h_t^a represents the agent-specific multi-stream output from each encoder. h_t is the concatenated representation for all agents and $h_{att,t}$ is the output of the attention module, representing the cross-agent interaction. We use the attention mechanism to disentangle and extract relevant multi-agent features from agent-specific representations while addressing the limitations of (max, average) pooling, which tend to summarize and thereby lose valuable information.

Latent variables and Discriminators: For multi-agent settings, we use the formulations of Eq. 2.6 to obtain the latent variables. Here, the latent variables represent the joint motion segment and the spatial semantics of all the agents. As the underlying functions for the discriminators and latent space remain unchanged, our approach is robust to the number of agents and can model interactions among all the agents.

Decoder: Each agent will have a specific decoder that auto-regressively predicts the motion for that agent only. The inputs to the decoder are the latent variables as well as the hidden representation of the last frame. This is passed to a self-attention module that learns the attention weights between the immediate agent-specific past output and the latent variables that represent multi-agent interaction. This allows the decoder to better capture inter-agent dynamics as well as condition its output on a subset of past frames.

The representation obtained from the self-attention module is fed to the GRU cell along with output of the last timestep. The output from the GRU is passed to a linear layer, with S_t^a being the predicted motion of agent a at time t. The operations at each decoder are formulated as follows:

$$\begin{aligned} p_t^a &= Concat(z_t, c_t, h_{dec,t-1}^a);\ p_{att,t}^a = Att(p_t^a, \phi_{att}^a) \\ h_{dec,t}^a &= GRU(S_{t-1}^a, p_{att,t}^a, \phi_{pos}^a);\ S_t^a = \gamma(h_{dec,t}^a) \end{aligned} \tag{2.9}$$

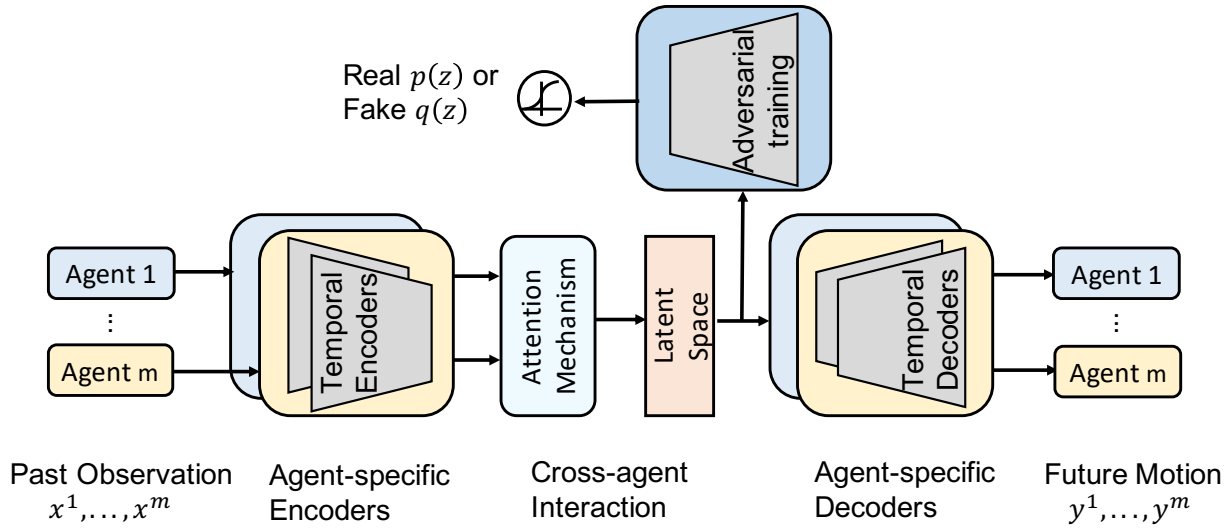


Figure 2.4: Proposed framework for multi-agent setting.

2.5 Experimental Setup

2.5.1 Datasets

We evaluated the performance of our approach by applying it on three widely used human-activity and social interaction datasets: UTD-MHAD [85], NTU RGB+D 60 [55] and CMU Panoptic [56]. Furthermore, we evaluated our approach on the KTH Human-Robot Collaboration (KTH-HRC) dataset [38]. For single-agent motion prediction, we conducted experiments on the UTD-MHAD. The dataset contains 27 action classes covering activities from hand gestures to training exercises: providing a range of relevant and diverse activities for human-robot collaboration. We used skeleton data for predicting human motion, following previous work in this domain [79, 35, 29, 37, 36], and considered each of the 20 provided joints. We used the cross-subject evaluation scheme, training and validating on odd-numbered subjects while testing on even-numbered subjects.

For multi-agent motion prediction, we conducted experiments on the NTU-RGB+D 60 [55] and CMU Panoptic [56] datasets. For NTU-RGB+D 60 dataset, we focused on the action classes involving more than one agent, resulting in 11 joint actions in total, ranging from punching to hugging, similar to previous work [28]. We used the cross-subject evaluation scheme [55], with 20 subjects for training and validation and a separate 20 for testing. For the CMU Panoptic dataset, we focused on the Haggling action, which consisted of more than two agents and had a defined training and testing protocol. Similar to the single-agent setup, we used the skeleton modality and all provided joints of each agent for motion prediction across all methods. While having access to a different modality, such as RGB data, can potentially improve model performance, prior work has shown that the improvement is only marginal due to the constrained environmental setup in which the data were collected [28].

Finally, the KTH-HRC dataset [38] comprised 4 human-robot collaboration actions ranging from handshaking to hand wave. We used two experimental setups. First, we train our model on Human-Robot Collaboration (HRC) data and set aside the last 20% of all the trials for testing, in keeping with [38]. Second, we train the model on Human-Human Collaboration (HHC) data and test on HRC data. We used the same four joint positions as the original paper [38].

2.5.2 State-of-the-art methods and baselines

For evaluating our model on single-agent settings, we compared against two state-of-the-art approaches: Seq2Seq-sampling [29], Seq2Seq-sampling-SPL [36], and the zero-velocity baseline [29]. The Seq2Seq-sampling

Table 2.1: MSE (in cm^2) comparison of different single-agent methods on UTD-MHAD and KTH-HRC datasets (Lower is better).

Approaches/Frames	UTD-MHAD						KTH-HRC (Trained and tested on HRC data)						KTH-HRC (Trained on HHC, tested on HRC data)					
	2	4	8	10	13	15	5	10	20	30	35	40	5	10	20	30	35	40
Zero-Velocity [29]	11.31	27.91	68.79	89.09	116.95	133.05	0.11	0.34	1.18	2.38	3.07	3.81	0.09	0.32	1.14	2.33	3.02	3.76
Seq2Seq [29]	8.90	19.09	39.03	47.45	57.84	63.30	0.18	0.55	1.67	3.11	3.91	4.74	0.14	0.36	1.09	2.17	2.81	3.49
Seq2Seq-SPL [36]	8.17	17.63	36.86	45.02	55.20	60.72	0.17	0.42	1.20	2.33	2.98	3.66	0.09	0.25	0.86	1.97	2.73	3.62
Our method	**6.39**	**14.33**	**31.63**	**39.12**	**48.57**	**53.74**	**0.06**	**0.20**	**0.72**	**1.61**	**2.21**	**2.91**	**0.07**	**0.24**	**0.78**	**1.52**	**1.95**	**2.28**

approach is based on the sequence-to-sequence learning framework but introduces a skip connection between the final model prediction and the past predicted frame. In the Seq2Seq-sampling-SPL approach [36], the authors introduce a Structural Prediction Layer at the decoder that results in a hierarchical prediction of joints, based on the structural prior of human joints. In addition, we compared against the zero-velocity baseline used in many other work for comparison and demonstrated to be a high-performance baseline that is hard to outperform [29, 35, 39]. The baseline assumes that all the future predictions are identical to the last observed pose and is difficult to outperform for short-term prediction.

Similar to single agent, we compared our multi-agent approach against two state-of-the-art methods, Joint Learning and Joint Learning + Social [28]. In case of Joint Learning + Social, a permutation invariant pooling mechanism is applied to pool social features across all agents with max-pooling providing the best results [28]. To ensure a fair comparison, we fine-tuned hyper-parameters for all the approaches.

2.5.3 Evaluation Metric

We evaluated the performance of all models using the Mean Squared Error (MSE), which is the l_2 distance between the ground-truth and predicted poses at each timestep, averaged over the number of joints and sequence length, similar to prior work [86, 37, 36, 28]. The MSE is calculated as:

$$\mathcal{L}(\mathcal{X}, \hat{\mathcal{X}}) = \frac{1}{T \times K} \sum_{t=1}^{T} \sum_{i=1}^{K} (x_t^i - \hat{x}_t^i)^2 \tag{2.10}$$

where, T and K are the total number of frame and joints respectively. The MSE jointly encodes global body motion and skeletal movements [28], making it an ideal metric.

2.5.4 Implementation Details

Our approach is divided into four modules: the encoder, latent variables, discriminators and decoder. The training has two phases: reconstruction and regularization, in line with the AAE framework [91]. In the reconstruction phase, the encoder-decoder is trained end-to-end, using reconstruction loss. In the regularization phase, the discriminators are trained using the cross-entropy loss. The discriminator loss is used to update the weights of the encoder. We provide details on the training of all experiments in the supplementary video.

Encoder: For single-agent experiments on UTD-MHAD and HRC data, we use one multi-stream encoder to encode past observations. The encoder comprises of three GRUs for position, velocity, and acceleration. The hidden state dimension is 200 for velocity and acceleration. For position, the hidden state dimension is the same as the input dimension.

For multi-agent experiments on the NTU RGB+D 60 and CMU Panoptic datasets, we varied the number of encoders depending on the number of agents. We use dropout regularization for all GRUs with a dropout probability of 0.1.

Latent variables: We empirically evaluated the ideal combination for the continuous and categorical latent variables, while ensuring that they are smaller than the intrinsic dimension of the data. The dimensions for continuous and categorical latent variables vary depending on the datasets and are provided in the supplementary video.

Decoder: For single-agent experiments, we use one decoder and implement weight sharing between the position-specific encoder and decoder GRU. The output of the GRU is followed by a Structured Prediction Layer (SPL) [36].

Table 2.2: MSE (in cm^2) comparison of different multi-agent methods on NTU RGB+D 60 and CMU Panoptic datasets (Lower is better).

Approaches/Frames	NTU RGB+D 60						CMU Panoptic					
	2	4	8	10	13	15	2	4	8	10	13	15
Joint Learning [28]	9.68	15.84	29.88	37.52	49.55	57.93	1.334	2.29	4.15	5.09	6.55	7.56
Joint Learning + Social [28]	9.71	15.97	30.36	38.25	50.70	59.38	1.396	2.39	4.35	5.35	6.87	7.90
Our method	**9.66**	**15.66**	**29.05**	**36.16**	**47.20**	**54.84**	**1.327**	**2.22**	**3.94**	**4.79**	**6.07**	**6.94**

For multi-agent experiments, we varied the decoders depending on the number of agents. We simplify the decoder operations by using a linear layer as the final output for each decoder. In both experiments, we use Teacher Forcing [96] to aid the learning *only during training*, whereby we feed the actual output at the last timestep to prevent prediction errors from severely propagating into the future.

Discriminator: We use feedforward neural networks with 2 linear layers each, followed by sigmoid activation for both discriminators. The hidden size of both layers is 200 and 100 for single and multi-agent experiments respectively.

Training environment: We used Pytorch v1.5.1 running on Nvidia Titan v100 and Cuda 10.1 for all our experiments. The encoder-decoder architecture is trained end-to-end using the Adam optimizer [97]. We used an initial learning rate of 1e-3 for experiments on UTD-MHAD, KTH-HRC & CMU Panoptic datasets and 5e-4 for experiments on the NTU RGB+D 60 dataset. For all experiments, we used weight decay on plateau with a decay factor of 0.1 and early stopping on the validation set. For the discriminators, we used Adam optimizer with learning rates of 2e-6 onUTD-MHAD, KTH-HRC and CMU-Panoptic and 2e-7 on NTU RGB+D 60.

2.6 Results and Discussion

2.6.1 Single agent Motion Prediction

Results: We present the results of all models on single-agent motion prediction on the UTD-MHAD in Table 2.1. We report the performance of all approaches at distinct frame intervals to circumvent the problem of frame drops during data collection and subsequent evaluation. Our frame intervals aim to evaluate all models on short (2 & 4), mid (8 & 10), and long-term motion prediction (13 & 15). The results in Table 2.1 suggest that our approach outperforms all other methods and the zero-velocity baseline for short, mid, and long-term prediction. Our proposed model performs particularly well for long-term prediction with the performance of all models deteriorating as the prediction horizon increases.

Discussion: Our proposed approach outperformed state-of-the-art models on all evaluated benchmarks, suggesting improved representation learning and sequence modeling. The results from Table 2.1 suggest that all models outperform the zero-velocity baseline [29]. For long-term motion prediction (13 & 15 frames), our method outperforms other approaches, firstly demonstrating the robust learning capability of the multistream encoder. Furthermore, the latent variables learn a distribution over the observed trajectory, which is used to predict future frames. As such, they learn long-term representation over a horizon. As the decoder conditions its output on the last predicted frame *and* the latent variables, it achieves performance gains over the long-term. Fig. 2.1 underscores the fact that our approach generates more feasible motion compared to other methods by accurately modeling joint position and orientation.

2.6.2 Multi-agent Motion Prediction

Results: We present the results of all models on multi-agent motion prediction on the NTU RGB+D 60 and CMU Panoptic datasets in Table 2.2. Similar to the single-agent setup, we measured all models' quantitative performance at the same distinct frame intervals (2, 4, 8, 10, 13 & 15). The results in Table 2.2 suggest that our approach outperforms all models over all evaluated horizons, with particularly improved performance over longer horizons.

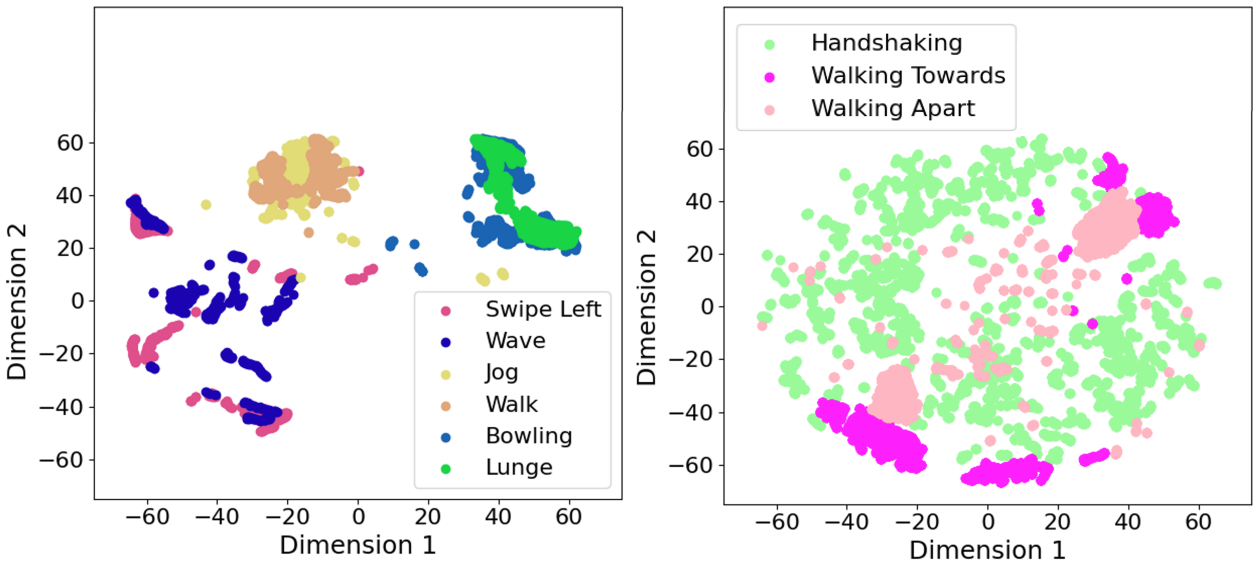


Figure 2.5: Continuous latent space visualization using t-SNE plots on UTD-MHAD (Left) and NTU RGB+D 60 (Right) datasets.

Discussion: Our proposed approach outperformed other methods over all the evaluated horizons. This suggests that our approach learns a more robust representation for each agent, while also capturing relevant inter-agent dynamics among all the agents. The multi-stream encoder provides a salient representation for each agent, which is then used by the self-attention mechanism to adaptively weigh relevant agent-specific features for modeling the interaction dynamics among all the agents. In addition, the decoder module learns the attention weights between the immediate agent-specific past output and the latent variables representing the observed multi-agent interaction. This further contributes to the performance gain, especially over longer horizons, as the decoder conditions over a subset of past frames and interaction among all the agents.

2.6.3 Human-Robot Collaboration Experiments

Results: We present the results of all models on the human-robot collaboration experiments in Table 2.1. We first trained and evaluated all models on HRC data. Next, we trained all models on HHC data and evaluated them on HRC data. Here, we measured the MSE over larger frame intervals due to the tasks' duration being longer (approx. 11 seconds). We evaluated all models on short (5 & 10), mid (20 & 30) and long-term horizons (35 & 40).

Discussion: The results in Table 2.1 (KTH-HRC (Trained and tested on HRC data)) suggest that our proposed method outperformed all other approaches over all the horizons. Similar to the single and multi-agent conditions, our approach's performance gains increase over longer horizons.

When training on HHC data and testing on HRC data, the results in Table 2.1 suggest a similar pattern, with our proposed approach outperforming other methods. We also observed that the models generalize better when training on HHC data and testing on HRC data. We attribute this to there being greater and more diverse training samples, which allowed the models to learn a more robust representation.

The above results demonstrate our model's ability to best predict human motion, even in the presence of a collaborative robot. Having superior short-term performance would allow the robot to prevent collisions and be more responsive, thus enhancing collaboration safety. On the other hand, having superior long-term performance would allow the robot to plan its actions more efficiently.

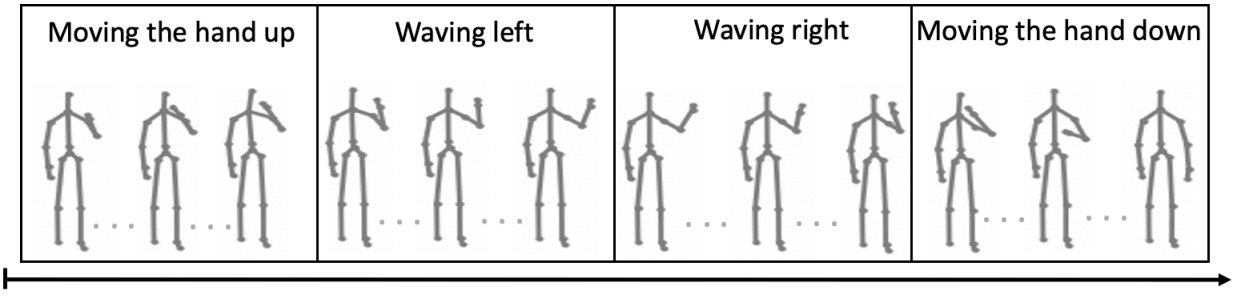


Figure 2.6: Action primitives for *wave* on UTD-MHAD.

2.6.4 Latent Space Interpretation

Results: We visualized the learned latent space of our framework to improve our understanding of the generative process. For this purpose, we analyzed the learned latent manifold for single and multi-agent settings on the UTD-MHAD and NTU RGB+D 60 respectively. For each temporal window of observations, our proposed framework maps the high-dimensional data into a low-dimensional manifold, represented by the continuous and categorical latent variables.

To visualize the continuous latent variables, we project them to a 2-D plane using t-SNE [98] as shown in Fig. 2.5. We then segment the 2-D plane by action class labels, which were not provided during training. For analyzing the categorical latent variables, we look at their distribution over a trajectory for each action. Fig. 2.6 presents the predicted frames for the action *wave* on UTD-MHAD, along with the distribution of the categorical latent variable over the duration of the action.

Discussion: Our results suggest that the continuous latent variables learn spatial embedding for each temporal window. Fig. 2.5 shows that activities that share similar spatial semantics, such as *walking and jogging* on UTD-MHAD (Fig. 2.5-Left) and *walking towards and walking away* on NTU RGB+D 60 (Fig. 2.5-Right) have overlapping clusters. Similarly, other sets of activities such as *bowling and lunging*, and *wave and swipe left* on UTD-MHAD also have separate overlapping clusters. Additionally, as seen in Fig. 2.5-Left for UTD-MHAD, our framework learns to separate activities that have different spatial semantics: the clusters for *bowling and lunging*, *walking and jogging*, and *wave and swipe left*, do not overlap. Similar segmentation is observed for multi-agent activities such as *handshaking* and *walking towards/apart* on NTU RGB+D 60 (Fig. 2.5-Right).

In case of categorical latent variable, our results on UTD-MHAD indicate that it takes on different values over time, which coincides with different action primitives. As can be seen in Fig. 2.6, the action class *wave* is segmented into four action primitives: *moving the hand up*, *waving left*, *waving right* and finally *moving the hand down*.

The above results demonstrate how our framework interprets each temporal window, modeling the continuous and categorical aspects of human motion. The learned representation of our framework can be used for various facets of robot perception, from activity segmentation and recognition to learning from demonstration. Crucially, it can be viewed as a step towards closer human-robot collaboration, by providing an explainable robot perception.

2.6.5 Ablation Study of Learning Modules

Results: We conducted an ablation study on the UTD-MHAD to evaluate the importance of various learning modules in our approach. Table 2.3 shows the impact of specific learning practices, given the same backbone framework of the multi-stream encoder and decoder.

Discussion: For a baseline, we have no SPL at the decoder, replacing it with a linear layer, while also not using Teacher Forcing (TF) during training. This architecture provided the worst performance in terms of MSE loss. Adding TF with a probability of 0.5 resulted in a large improvement in the short-term prediction, with marginal gains over the long-term. This highlights the importance of TF especially for short-term

Table 2.3: Ablation Study of our method on UTD-MHAD. Here, SPL: Using Structured Prediction Layer, TF: Teacher Forcing.

Approaches/Frames	2	4	8	10	13	15
No-SPL + No-TF	8.31	17.32	35.29	43.09	52.81	57.90
No-SPL + TF	6.52	14.43	32.79	41.02	51.89	57.84
SPL + No-TF	7.61	16.14	34.29	42.27	51.97	57.01
SPL + TF	**6.39**	**14.33**	**31.63**	**39.12**	**48.57**	**53.74**

prediction while also suggesting that the benefit decreases with an increase in time. We next assess the impact of having the SPL, firstly with no TF. Consistent with previous results, the short-term performance of the model is worse when compared with No-SPL + TF; however it is better across all evaluated horizon when compared against No-SPL + No-TF. This suggests the benefit of hierarchically predicting each joint when using SPL as the final layer instead of using a linear layer that assumes all joints are independent. Our best performing model is the SPL + TF, which combines the benefit of using structured prediction as well as having the short-term improvement of TF.

2.7 Limitations

In this work, we introduced a novel sequence-learning approach for human motion prediction that outperformed state-of-the-art methods on single and multi-agent settings. Our framework for multi-agent motion prediction introduces an attention-based mechanism that can better represent the inter-agent dynamics of human motion. Despite the many improvments in modeling human motion, this work has some limitations.

One key limitation lies in mechanism used to regularize the latent space, which relies on adversarial regularization. This introduces a min-max optimization problem that can be difficult to balance, potentially leading to unstable training dynamics or suboptimal representations that may not generalize well across different tasks or environments. In addition, this approach of regualarizing the latent space with a static prior may hinder the actual objective function of leaning to predict human motion. Finally, this work uses a simple mean-squared error objective, which does not account for the spatial and temporal patterns of human motion.

Chapter 3

Having dynamic and learnable priors for human motion (VADER)

While our previous work introduced a promising sequence-learning approach for human motion prediction, it also revealed some limitations, particularly in the use of static priors and adversarial regularization in the latent space. These approaches, while innovative, introduced challenges such as balancing the min-max optimization problem, which could lead to unstable training and suboptimal representations. Additionally, the reliance on a simple mean-squared error objective did not fully capture the intricate spatial and temporal patterns inherent in human motion.

In this chapter, we address these limitations by proposing a novel approach that incorporates dynamic and learnable priors for human motion prediction, as well as augmenting the mean-squared error objective. This new methodology aims to enhance the robustness and generalization of the model, ensuring more reliable predictions in diverse and complex environments. By moving beyond static priors and refining the objective functions, we aim to overcome the challenges identified in our previous work, paving the way for more effective and adaptive human-robot collaboration.

3.1 Introduction

Robots that can collaborate with humans mark a shift from traditional industrial robots, which operated in isolation from their human co-workers [99]. This collaboration seeks to combine the flexibility and decision-making capabilities of humans with the strength and endurance of robots [100]. Human beings rely heavily on predicting the behavior of others, as observed in activities of daily living, from walking in crowded areas to shaking hands to handing over or receiving objects [5, 6]. Along this line, for robots to operate safely and reliably in the presence of humans, they need to perceive, anticipate and adapt to the changes in their environment, particularly the motion and intent of humans in the vicinity [101, 102, 103, 15, 104]. Despite significant advances in robot perception for detecting changes and adapting to new environments [70, 78, 105, 76], the ability to predict changes in environment dynamics in a reliable manner remains an open challenge.

The concept of anticipation has been extensively studied in the field of robotics, particularly in the context of motion planning, where the main objective is to navigate safely around humans, thus avoiding any interference [44, 106]. Other works have investigated anticipatory planning of robot action based on inferred goals [107, 108, 109]. However, as robots are expected to collaborate with humans over prolonged periods, there is a need to anticipate human motion at a higher spatial and temporal granularity [20, 75, 74]. This involves predicting future human poses conditioned on past motion (Fig. 3.1), enabling the robot to plan around the human without disrupting their natural flow. However, accurate prediction of human motion remains challenging due to the complex and interpersonal nature of human behavior [1, 45].

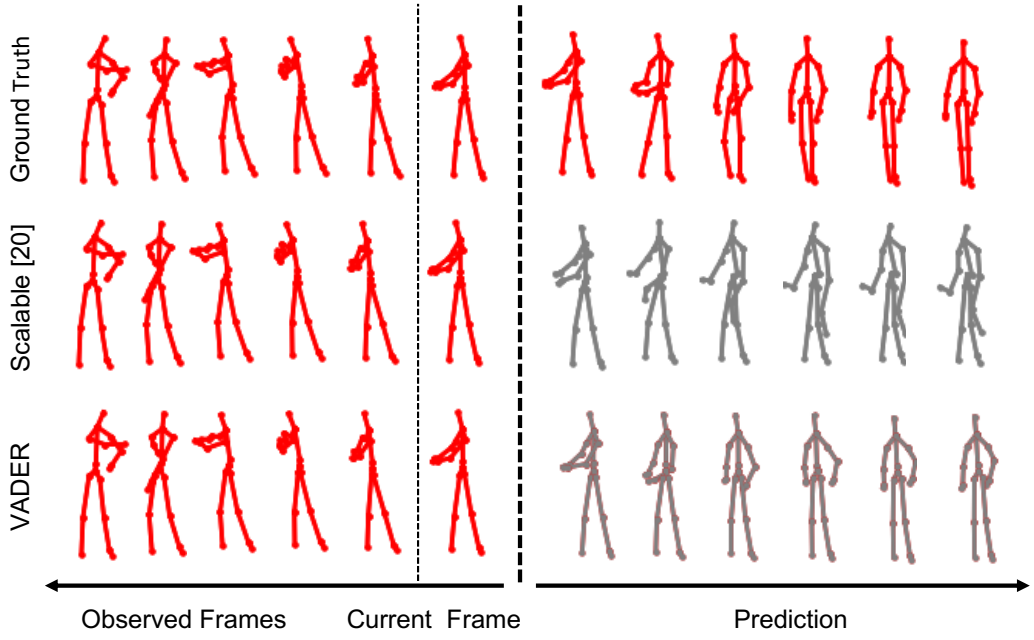


Figure 3.1: Qualitative evaluation of the predicted motion of VADER and the next best performing model, Scalable + Interpretable [1] on the Baseball Swing task on UTD-MHAD. VADER's use of a flexible discrete latent space and the introduction of the discriminator loss results in predictions which are closer to the ground-truth.

Prior work has framed the task of predicting future poses as a sequence learning problem to address the aperiodic and stochastic nature of human motion [35, 29, 36, 1]. These approaches aim to learn a unified representation from training samples that are expected to generalize for test data. When training these networks, the core assumption is to either learn a distribution that can fit a static prior [1, 37, 38, 39], or to learn a point estimate over the past observed data which will be used to predict future human motion [29, 36, 28]. Learning a static prior introduces an auxiliary objective that acts as a regularizer which requires careful tuning, whereas learning a point estimate leads to less robust representations. Furthermore, prior works have relied on the reconstruction error as the sole objective for training these networks. Such an objective function may cause the predictions to regress to the mean and, as such, may not be able to capture the spatial and temporal correlations in human motion.

To address the above challenges, we propose VADER, a novel approach that aims to close two critical gaps in motion prediction: 1) learning a robust representation of the past motion and 2) improving temporal and spatial correlation in the prediction. VADER is built on top of the encoder-decoder framework but augments it with codebook learning and distribution matching. Our approach uses the expressive powers of codebooks to learn discrete representations over the observed motion data. Furthermore, as motion prediction introduces high data dependencies, it makes any regular MSE-based objectives ill-posed. As such, we propose a novel discriminator-based loss to increase the temporal and spatial coherency by penalizing predictions that deviates from the ground-truth distribution.

Similar to past approaches in motion prediction, the input to VADER is the past human pose data. As pose data only captures positional information, we follow recent approaches [1] to augment the pose data with motion features by extracting velocity and acceleration information. These represent different motion data streams that are separately fed to the encoder network. The output from the encoder is used to learn a discrete latent space using vector quantization (VQ) [110], which allows for a flexible prior instead of a static one. This allows the framework to remain flexible in learning representations, with no restriction of a static prior, which may over-regularize the learning.

The quantized latent representations are fed to an autoregressive decoder. We choose an auto-regressive decoder to encourage conditional dependency between the predicted frames. However, if the input prediction

is erroneous, this may also end up propagating errors over time. To encourage spatial and temporal consistency among the predicted frames, we use a discriminator, which is tasked to penalize predictions that are further from the ground-truth. This creates a min-max game between the encoder-decoder network and the discriminator network. The discriminator penalizes outputs that are not from the ground-truth distribution, and the encoder-decoder generates samples that are progressively closer to the ground-truth distribution.

We performed extensive experiments to evaluate the efficacy of VADER, across three motion prediction scenarios: single-agent, multi-agent and human-robot collaboration, on four popular datasets: UTD-MHAD [85], NTU RGB+D 60 [55], and CMU Panoptic [56] and the KTH Human-Robot Collaboration (KTH-HRC) dataset [38]. The results underscore VADER's effectiveness in addressing the open challenges in motion prediction, as it consistently outperformed all evaluated algorithms across all benchmarks attaining the lowest prediction error across all temporal horizons quantitatively and qualitatively. Furthermore, through extensive experiments, we demonstrated that the novel discriminator objective could be combined with the reconstruction loss in a complementary manner for motion prediction. Finally, we conducted extensive ablation analyses to assess the efficacy of each of our modules across different scenarios and datasets. The ablation studies underline the importance of VADER's learning modules and justify VADER's novel objective function.

3.2 Related Works

Human motion prediction: Human motion prediction is widely considered as an essential component of robotic intelligence that can enhance robot perception and enable them to react rapidly and accurately to complex changes in the environment [26, 27, 1, 28, 29]. The significance and challenges of this problem have led to extensive study in computer vision and machine intelligence. Some of the earliest methods for motion prediction involved using Hidden Markov Models and Gaussian processes. Lehrmann et al. [32] proposed latent-variable models that follow state-space equations modeled by Hidden Markov Models. Taylor et al. [33] introduced the use of conditional restricted Boltzmann machines (RBM) for motion prediction. Wang et al. [34] used Gaussian-Processes to perform non-linear motion prediction.

In recent years, there has been a trend towards using data-driven approaches such as Recurrent Neural Networks (RNNs) or other types of neural networks for motion prediction [28, 36, 35, 1, 29, 111, 112]. Jain et al. [79] proposed the Structural RNN, which uses graphs to capture the spatio-temporal relationship of skeletal joints. The authors divided the skeletal hierarchy into five clusters, each representing a specific portion of the body joints. Fragkiadaki et al. [35] introduced the Encoder-Recurrent-Decoder (ERD), which uses multi-layer feed-forward networks to extract pose features before passing them to the LSTM cell to encode history information through its recurrent architecture. Martinez et al. [29] extended this scheme by modeling the velocity component of motion prediction. The authors introduced residual connections in the decoder part of the architecture, which improves motion prediction in terms of smoothness and accuracy. Furthermore, the authors also demonstrated how previous works failed to outperform a simple zero-velocity baseline, which constantly predicts the last observed pose. Aksan et al. [36] further improved the structural prediction of the models and enabled the generation of more feasible human joints by introducing a Structural Layer at the decoder (SPL). The SPL layer predicts each joint hierarchically, imposing conditional dependence between joints.

Due to the sequential nature of motion prediction, there has been a recent trend of applying the attention mechanism [84] to either the encoder or decoder or both [1, 113, 114, 112]. Mao et al. [112] introduced an attention-based feed-forward network to capture the similarity between the current motion context and the historical sub-sequences, where each sub-sequence is represented using the Discrete Cosine Transform (DCT). Yasar et al. [1] used the attention mechanism at the encoder to learn a more salient representation from the different encoder streams and at the decoder to allow the auto-regressive decoder to weigh between the latent space and its past output. Liu et al. [114] used spatial and temporal attention to model relationships between the skeletal joints and the temporal frames, respectively.

Latent representation for motion prediction: Although the majority of research on motion prediction has employed a deterministic approach, a growing number of studies have begun to view the problem as

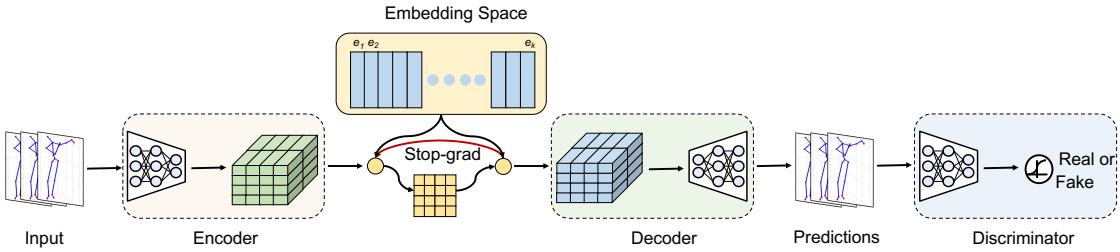


Figure 3.2: VADER:Vector-Quantized Generative Adversarial Network for Motion Prediction. The encoder-decoder framework is tasked to generate future human motion, using a discrete latent codebook. The latent codebook allows for a flexible prior which is learned. The discriminator aims to distinguish between the ground-truth distribution and decoder's prediction, penalizing deviation from the ground-truth.

learning a probability density function for future human poses, based on the past poses. [37, 38, 49, 83, 39, 115, 116]. Butepage et al. [37] and Toyer et al. [39] adopted the Variational Autoencoder framework for predicting human motion, which relies on learning a functional mapping from the input data space to the latent space at the encoder, and from the latent space to the reconstructed data space at the decoder. Barsoum et al. [83] proposed HP-GAN, which is trained with a modified version of the improved Wasserstein generative adversarial networks (WGAN-GP). Kundu et al. [116] proposed the BiHMP-GAN where a bi-directional GAN was designed to avoid the problem of mode-collapse. Yasar et al. [1, 117] proposed the use of discriminators on the latent space to enforce a prior instead of using KL-divergence or other distribution matching losses such as JS-divergence [37, 39].

While prior works in motion prediction have significantly advanced the state-of-the-art, generating feasible and temporally coherent human motion remains an open research problem [29, 36, 112, 1]. Additionally, unlike other areas of machine intelligence, such as computer vision or machine translation, there is no consensus on the optimal framework for capturing the spatial and temporal dynamics of human motion. Although recent approaches have adopted an encoder-decoder framework, learning a robust representation at the encoder that best captures the past human pose is still an ongoing research effort. Moreover, while learning a distribution over past observed motion has benefits, identifying a prior that can accurately model this distribution remains a challenge, resulting in difficulties with an optimization that require both reconstruction and distribution matching.

3.3 Problem Formulation

Our objective is to improve the robot's perception by providing it with the capability to forecast the motion of all human collaborators in the team. Human motion prediction is formally described as the task of estimating the future human pose for a certain period, given their past pose. We will begin by presenting the problem for single-agent motion prediction and later extend the formulation to multiple humans. Our notation employs superscripts to denote agents and subscripts to denote time in all formulations.

Single Agent: To begin with, we consider the case of a single agent, denoted as a. The task is to predict the future pose trajectory of the agent, given their observed pose trajectory from time $t = 1$ to τ: $\mathbf{X}^a = x_1^a, \ldots, x_\tau^a$. Each pose frame $x_t^a \in \mathbb{R}^N$ represents the N-dimensional body pose, where N is determined by the number of joints J in the skeleton and the dimension of each joint D, with $N = J \times D$.

The expected output of the model is the future trajectory frames over horizon H : $\mathbf{Y}^a = \{y_{\tau+1}^a, \ldots, y_{\tau+H}^a\}$. Our first objective is to learn the underlying representation which would allow the model to predict accurate and feasible future human poses $\hat{\mathbf{Y}}^a = \{\hat{y}_{\tau+1}^a, \ldots, \hat{y}_{\tau+H}^a\}$. We assume that future human pose is conditioned on the past observed and generated poses and predict each frame in an auto-regressive manner as formulated below:

$$p_\theta(\hat{\mathbf{Y}}^a) = \prod_{\delta=\tau+1}^{\tau+H} p_\theta(\hat{y}_\delta^a | \hat{y}_{\tau:\delta-1}^a, x_{1:\tau}^a) \tag{3.1}$$

Multiple Agent: In multi-agent scenarios, we assume that the number of agents in the scene, denoted by K, is known beforehand. The input to the model consists of the observed pose information of all agents in the scene from time $t = 1$ to τ: $\mathbf{X} = \{X^1, \ldots, X^K\} = \{x_1^{1:K}, x_2^{1:K}, \ldots, x_\tau^{1:K}\}$. The model aims to predict the future trajectory frames over horizon H: $\mathbf{Y} = \{Y^1, \ldots, Y^K\} = \{y_{\tau+1}^{1:K}, y_{\tau+2}^{1:K}, \ldots, y_{\tau+H}^{1:K}\}$.

We assume that the future human pose of each agent is conditioned on the observed poses of all agents, and predict each frame in an auto-regressive manner. Thus, the multi-agent motion prediction problem is formulated as follows:

$$p_\theta(\hat{\mathbf{Y}}^a) = \prod_{\delta=\tau+1}^{\tau+H} p_\theta(\hat{y}_\delta^a | \hat{y}_{\tau:\delta-1}^a, x_{1:\tau}^{1:K}); \quad \forall a = 1, \ldots, K \tag{3.2}$$

3.4 VADER: Vector-Quantized Generative Adversarial Networks for Motion Prediction

We now introduce our proposed framework, VADER: Vector-Quantized Generative Adversarial Network for Motion Prediction. VADER comprises two primary components: the encoder-decoder architecture with discrete latent representation and the discriminator network (see Fig. 3.2). The objective of the encoder-decoder architecture is to predict future human pose. The discriminator is tasked to distinguish frames that are coming from the ground-truth distribution, from frames that are not, penalizing the latter. This creates a min-max game between the two networks, which combine to provide more accurate motion prediction. VADER's represents a unified framework for single, multi-agent, and human-robot collaboration scenarios, with the main difference between the number of encoders and decoders, which scale with the number of agents that require modeling.

Encoder: The encoder seeks to learn a salient representation over the raw input space. The input to the encoder is the multi-stream raw data of the past observed motion. The streams comprise the skeletal joint position, velocity, and acceleration of the observed motion.

We use separate encoders to process the position, velocity, and acceleration streams. The goal of each encoder is to learn a spatio-temporal representation of the observed motion data. To address this sequence learning problem, we employ unidirectional Gated Recurrent Units (GRU) in the encoders to extract temporal feature representations for each stream. We chose to use unidirectional GRUs rather than a bi-directional architecture to reduce the computational load on the robot while predicting human motion. Our decision to use GRUs [118] instead of other RNNs, such as LSTMs [119], was based on their comparable performance to LSTMs, and relative computational efficiency. For each stream, the stream-specific GRU aims to encode the spatio-temporal information over the input sequence, which is formulated as:

$$h_{s,t} = GRU(h_{s,t-1}, x_{s,t}, \phi_s) \tag{3.3}$$

where s represents position, velocity, or acceleration. Here, $x_{s,t}$ represents the input to the GRU at time t and will take the value of $x_{pos,t}, x_{vel,t}, x_{acc,t}$ for position, velocity, and acceleration at time t, respectively. $h_{s,t-1}$ represents the past hidden output at time $t-1$ and ϕ_s represents the stream-specific encoder weights for the GRU. The output from each encoder is passed to a self-attention module [84]. The attention module is tasked to sparsely and adaptively extract the salient features from the three streams.

$$h_t = \text{Concat}(h_{\text{pos},t}, h_{\text{vel},t}, h_{\text{acc},t}); \; z_t = \text{Att}(h_t, \phi_{\text{att}}) \tag{3.4}$$

where ϕ_{att} represents weights of the attention module. In the self-attention module, we first linearly project the concatenated output h_t into a separate query (Q), key (K), and value (V) embeddings for each head. These embeddings are then used to calculate attention weights using the scaled-dot product softmax (sf) approach. The functions for each head in the multi-head self-attention module are defined as follows:

$$\begin{gathered} Q = h_t W^Q; \; K = h_t W^K; V = h_t W^V \\ Att(Q, K, V) = sf\left(\frac{QK^T}{\sqrt{d_k}}\right) V \end{gathered} \tag{3.5}$$

where, W^Q, W^K, W^V represent the linear projection weights and $\frac{1}{\sqrt{d_k}}$ is the scaling factor for calculating the attention weights. The output of the attention module is passed to the discrete codebook to obtain the latent space.

Latent codebook: In VADER, we propose the use of a codebook for calculating the latent space, similar to the Vector Quantization approach in VQ-VAE [110], which has been successfully applied for image synthesis. Unlike prior approaches in motion prediction [1, 86, 37, 39], which uses variation bottleneck or discriminators to enforce a prior, we propose the use of a flexible prior, which is learned dynamically as the training progresses. This has the benefit of not restricting the learning procedure by forcing the latent space to conform to a static distribution while also reducing the likelihood of mode collapse.

In VADER, we define the latent embedding as a codebook $e \in \mathbb{R}^{LXM}$, where L is the size of the categorical latent space, and M is the dimension of each categorical embedding vector e_i. The output of the encoder is used to calculate the discrete latent space using the nearest neighbor lookup from the shared embedding space e. Thus, the latent space z can be posed as a posterior categorical distribution $q(z|x)$, where the probabilities of the categorical vector are one-hot and defined as follows:

$$q(z = k|x) = \begin{cases} 1 & \text{for k} = \text{argmin}_j ||z - e_j||_2, \\ 0 & \text{otherwise,} \end{cases} \tag{3.6}$$

where, $z(x)$ is the output of the encoder network, e_j represents a vector from the codebook e.

Decoder: To predict the current pose, we use an auto-regressive decoder that depends on the output of previous time steps. The decoder has only one output stream: the 3-D skeletal joint position. It takes as input a discrete embedding vector e_k that summarizes the past motion observations. The decoder also has access to the last predicted frame, which is passed to a Keyless attention module [120]. The attention module calculates the weights between the immediate past output and the latent representation of the observed motion.

The first part of the decoder is a GRU cell that takes as input the output of the Keyless Attention module [120], as well as the output of the last timestep. This is followed by a fully connected layer. The operations at the decoder are formulated as follows:

$$\begin{aligned} p_t = Concat(z_t, h_{dec,t-1});\ p_{att,t} = Att(p_t, \phi_{att}) \\ h_{dec,t} = GRU(S_{t-1}, p_{att,t}, \phi_{pos});\ S_t = \gamma(h_{dec,t}) \end{aligned} \tag{3.7}$$

where, the latent representation is denoted as z_t, while $h_{dec,t-1}$ represents the previous hidden output of the GRU. The output of the attention mechanism in the decoder is denoted by $p_{att,t}$, which is passed along with the previous GRU output S_{t-1} to the GRU cell. ϕ_{att} and ϕ_{pos} denote the weights of the attention module and GRU cell, respectively. The decoder's output layer is represented by γ, and $\mathbf{S_t}$ represents the *predicted pose at time t*.

Multi-agent implementation: In the presence of multiple agents, the input to the decoder remains the same. However, as multiple agents need to be modeled, the decoder now conditions its output on the latent space and the *previous hidden state of all the agents*. As such, Eq 3.7 is now modified to include the previous hidden states of all the agents $h^a_{dec,t-1}, \ldots, h^n_{dec,t-1}$. This allows the decoder to model the interaction between the agents explicitly.

Discriminator: The discriminator consists of a separate encoder and aims to distinguish samples from the ground-truth distribution from samples generated from the decoder. As such, the discriminator takes as input the ground-truth data, $T_{real} = Y$, and the predicted motion $T_{fake} = \hat{Y}$ and classifies them as real and fake. The inputs are passed through an encoder, similar to Eq .3.3. The encoder's output is passed to the linear layer to obtain the classification results.

Overall objective function for VADER: In VADER, there are two distinct modules that are trained in opposition to each other, following the min-max setup in GANs [121]: the encoder-decoder architecture

and the discriminator. As such, the overall training procedure can be summarized by the following objective function:

$$\min_G \max_D V(G,D) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p(z)}[\log(1 - D(G(z)))] \tag{3.8}$$

where G represents the encoder-decoder architecture and D denotes the discriminator network. Furthermore, as we use a discrete codebook for representing the latent space, there are additional terms in the objective for the encoder-decoder architecture, which reflects the vector quantization algorithm that is used to learn the discrete representation, along with additional commitment loss to ensure that the encoder commits to an embedding, instead of arbitrarily growing in embedding space. The objective function can be defined as follows:

$$\mathcal{L} = \log p(y|z(x)) + ||z(x) - e||_2^2 + \beta||z_e(x) - sg[e]||_2^2 \tag{3.9}$$

where, the first term is the reconstruction loss. sg represents a stop-gradient operator that prevents the flow of gradient through the codebook. The second term is the nearest-neighbor embedding loss for selecting the embedding vector from the codebook. The third term is the commitment cost, ensuring the encoder commits to a specific embedding.

3.5 Experimental Setup

3.5.1 Datasets

We evaluated the performance of our approach by applying it to three different scenarios for motion prediction: 1) single-agent motion, where we evaluated on the popular human activity dataset UTD-MHAD [85], 2) multi-agent motion, where we evaluated on two multi-agent datasets NTU RGB+D 60 [55], and CMU Panoptic [56], and 3) human-robot collaboration dataset, where we evaluated on the KTH Human-Robot Collaboration (KTH-HRC) dataset [38].

For single-agent motion prediction, we conducted experiments on the UTD-MHAD. The dataset contains 27 action classes covering activities from hand gestures to training exercises: providing a range of relevant and diverse activities for human-robot collaboration. We used skeleton data for predicting human motion, following previous work in this domain [79, 35, 29, 37, 36], and considered each of the 20 provided joints. We used the cross-subject evaluation scheme, training and validating on odd-numbered subjects while testing on even-numbered subjects.

Our experiments in multi-agent motion prediction were conducted on two datasets: NTU-RGB+D 60 [55], and CMU Panoptic [56]. For the NTU-RGB+D 60 dataset, we focused on 11 joint actions that involve more than one agent, similar to previous work. We followed the cross-subject evaluation scheme and used 20 subjects for training and validation and another 20 for testing. As for the CMU Panoptic dataset, we focused on the Haggling action that involved more than two agents and had a defined training and testing protocol. We used the skeleton modality and all the provided joints of each agent for motion prediction, as prior work has shown that using RGB data only provides marginal improvements due to the constrained environmental setup in which the data were collected [28].

The third scenario involves human-robot teams, and the evaluation was performed on the KTH-HRC dataset [38] that contains tasks such as hand-shaking and hand-waving between a human and a robot. The prediction task involves only the human team member's motion, given the past motion of both the robot and the human, since the robot's policy is fully observable. The human is represented by four joints, "RightShoulder," "RightArm," "RightForeArm," and "RightHand" while the robot is represented by seven joint angles. We performed training and validation on 80% of each trial, while the remaining 20% was used for testing. The human joints were modeled in 3D Cartesian space, resulting in a 12-dimensional vector. The implementations of all the experiements are provided in the supplementary materials.

3.5.2 State-of-the-art methods and baselines

To evaluate our model in a single-agent scenario, we compared it with three state-of-the-art approaches: Seq2Seq-sampling [29], Seq2Seq-sampling-SPL [36], Scalable + Interpretable [1] and the zero-velocity baseline

Table 3.1: MSE (in cm^2) comparison of different single-agent methods on the UTD-MHAD (Lower is better).

Approaches	Frames					
	2	4	8	10	13	15
Zero-Velocity [29]	11.31	27.91	68.79	89.09	116.95	133.05
Seq2Seq [29]	8.90	19.09	39.03	47.45	57.84	63.30
Seq2Seq-SPL [36]	8.17	17.63	36.86	45.02	55.20	60.72
Scalable + Interpretable [1]	**6.39**	14.33	31.63	39.12	48.57	53.74
VADER	6.61	**14.22**	**29.82**	**36.23**	**43.83**	**47.81**

[29]. The Seq2Seq-sampling approach uses a sequence-to-sequence learning framework but includes a skip connection between the final model prediction and the past predicted frame. Additionally, we evaluated the Seq2Seq-sampling-SPL approach, which introduces a Structural Prediction Layer in the decoder to predict joints hierarchically based on the structural prior of human joints. The Scalable + Interpretable method comprised a multi-stream encoder, followed by a continuous and categorical latent space, which is then passed to a decoder. The zero-velocity baseline was included for comparison, and is a widely used and highly effective baseline that is challenging to outperform for short-term prediction, as reported by prior works [1, 29, 35, 39].

To assess the performance of VADER in multi-agent and human-robot collaboration scenarios, we compared it against various state-of-the-art models, including Joint Learning [28], Joint Learning + Social [28], Joint Learning + Social + Context [28], and Scalable + Interpretable [1]. All the Joint Learning models follow a sequence-to-sequence architecture and assume that agents do not interact, hence predicting each agent's motion independently. However, the Joint Learning + Social method applies a permutation-invariant pooling mechanism to aggregate social features across all agents, while the Joint Learning + Social + Context method employs an additional spatio-temporal context CNN module to extract RGB features from the scene. The Scalable + Interpretable method proposed an encoder-decoder approach with adversarial regularization on the latent space, incorporating an attention module to disentangle and extract multi-agent features. To ensure a fair comparison, we fine-tuned hyperparameters for all the models.

3.5.3 Evaluation Metric

We evaluated the performance of all models using the Mean Squared Error (MSE), which is the l_2 distance between the ground-truth and predicted poses at each timestep, averaged over the number of joints and sequence length, similar to prior work [86, 37, 36, 28]. The MSE is calculated as:

$$\mathcal{L}(\mathcal{X}, \hat{\mathcal{X}}) = \frac{1}{H \times K} \sum_{t=1}^{H} \sum_{i=1}^{D} (x_{t,i} - \hat{x}_{t,i})^2 \tag{3.10}$$

where, H and D are the total number of frame and joints respectively. The MSE jointly encodes global body motion and skeletal movements [28], making it an ideal metric.

3.6 Results and Discussion

3.6.1 Single-agent scenario:

Results: We first present the results of all evaluated models on the UTD-MHAD dataset in Tab. 3.1. We report the results in distinct frame intervals instead of seconds, similar to [1] to circumvent the problem of frame drops during data collection and subsequent evaluation. We use these frame intervals to evaluate the models' performance across short-term horizons (2 & 4 frames), mid-term horizons (8 & 10 frames), and long-term horizons (13 & 15 frames). The results shown in Tab. 3.1 indicate that VADER outperformed all other methods across the mid-term and long-term horizons. Additionally, the performance gains increase over time, with VADER outperforming the closest benchmarked approach (by up to 5.93 cm^2) for long-term horizons.

Discussion: The results in Tab. 3.1 suggest the efficacy of VADER's flexible latent space and the discriminator objective loss. Having the latent codebook with a flexible prior prevents any form of over-regularization

Table 3.2: MSE (in cm^2) comparison of different multi-agent methods on the NTU-RGBD 60 Dataset (Lower is better).

Approaches	Frames					
	2	4	8	10	13	15
Joint Learning [28]	9.68	15.84	29.88	37.52	49.55	57.93
Joint Learning + Social [28]	9.71	15.97	30.36	38.69	51.68	59.38
Joint Learning + Social + Context [28]	9.78	16.02	30.46	38.39	50.91	59.63
Scalable + Interpretable [1]	9.66	15.66	29.05	36.16	47.20	54.84
VADER	**9.65**	**15.48**	**28.57**	**35.64**	**46.71**	**54.39**

that is introduced by a static prior as used in previous approaches [1]. This is further complemented by the discriminator loss, which penalizes deviation from the ground truth, which along with the reconstruction loss, allows VADER to outperform all the evaluated methods. As can be observed in Tab. 3.1, VADER attains significant performance gain over the long-term compared to other approaches. One possible explanation for this is that the latent representation fed to the decoder effectively encapsulates the salient features from the past observed motion, thus reducing the error propagation of the auto-regressive decoder.

3.6.2 Multi-agent scenario:

Results: We next report the results of all evaluated models for multi-agent motion prediction scenarios on the NTU-RGB+D 60 [55] and the CMU Panoptic datasets [56]. The temporal intervals are the same as the single-agent scenarios. The results in Tab. 3.2 and Tab. 3.3 suggest that VADER outperformed all other methods across all the evaluated horizons for both the NTU RGB+D 60 and the CMU Panoptic dataset.

Discussion: VADER outperformed all evaluated models across all intervals on both datasets, indicating improved representation learning and sequence modeling for multi-agent scenarios. One possible reason for its superior performance is that VADER can model interaction dynamics in the decoder by computing attention weights over all agents' hidden states. Previous studies have attempted to model inter-agent dynamics using social pooling [28], or attention approaches [1], which have improved the state-of-the-art. However, these methods do not condition predictions on the past trajectories of all agents in the vicinity. In contrast, VADER explicitly conditions its predictions on the past hidden states and latent states of all agents. Additionally, the use of a discriminator loss encourages VADER to generate trajectories that are more accurate. VADER utilizes the Keyless Attention mechanism in the decoder, which enables effective modeling of interactions without losing relevant features while also being lightweight compared to the original self-attention mechanism.

3.6.3 Human-robot collaboration scenario:

Results: We next report the results of the Human-Robot Collaboration experiments on the KTH-HRC dataset in Tab, 3.4. Similar to the previous experiments, we report the performances of all models at different frame intervals. However, the frame rates in this dataset are different (40 vs. 30 fps for previous datasets. Thus, we report the results for 5 & 10 frames for short-term, 20 & 30 frames for mid-term, and 35 & 40 frames for long-term horizons. As the sampling frequency in this dataset is 40 frames-per-second, the predictions are made up to 1 second of human motion. It can be observed from Tab. 3.4 that VADER outperformed all other evaluated models for each horizon.

Discussion: The results in Tab. 3.4 underline VADER superiority over other evaluated approaches as it again outperformed all other evaluated approaches over all the horizons, achieving an improvement of up to 0.06 cm^2 over the long-term horizons. The results provide more evidence of VADER's superior representation learning and subsequent sequence modeling capabilities.

While prior approaches focus solely on modeling human skeleton data, VADER considers robot joint angle

Table 3.3: MSE (in cm^2) comparison of different multi-agent methods on the CMU Panoptic Dataset (Lower is better).

Approaches	Frames					
	2	4	8	10	13	15
Joint Learning [28]	1.33	2.29	4.15	5.09	6.55	7.56
Joint Learning + Social [28]	1.39	2.39	4.35	5.35	6.87	7.90
Joint Learning + Social + Context [28]	1.40	2.44	4.59	5.71	7.42	8.58
Scalable + Interpretable [1]	1.33	2.22	3.94	4.79	6.07	6.94
VADER	**1.32**	**2.19**	**3.84**	**4.66**	**5.89**	**6.75**

data when making predictions. The robot joint angle data is first processed by a motion encoder, similar to the one used for human motion data, to obtain a representation that is used to calculate the latent representation before being passed to the decoder. In situations where humans and robots collaborate in close proximity, human motion maybe influenced by the motion of robots. Many recent studies have shown that robots can affect human actions [74, 57]. Hence, VADER considers the past observed motion of the human and the past robot joint angles in its encoder to calculate the latent space, recognizing that the robot may significantly impact human motion in human-robot collaboration scenarios.

In contrast to VADER, the other evaluated methods did not consider the interaction between the human and robot. Of the evaluated models, Scalable + Interpretable [1] performed second best to VADER, potentially due to its multi-stream encoder, which incorporates velocity and acceleration along with skeletal joint position, and a self-attention mechanism that dynamically weighs these streams. Additionally, all evaluated models performed better than the zero-velocity baseline for long-term horizons (35 & 40 frames), while only VADER and Scalable + Interpretable outperformed the baseline for short-term horizons (5 & 10 frames), possibly due to their use of the multi-stream encoder.

3.7 Ablation Study

Results: We performed extensive ablation experiments on the UTD MHAD dataset to assess our design choices for VADER. For the ablation experiments, we performed two architectural ablations: first, at a modular level, where we removed the Discriminator module and subsequently the discriminator objective (VADER w/o GAN objective). Next, we explored some of the decisions at the architectural level, particularly the use of GRU cells over contemporary architectures for timeseries, such as Temporal Convolutional Networks (TCNs) [122] and the use of the attention mechanism. As such, we ablated the GRU cells from both the encoder and decoder (VADER with TCN encoder-decoder) and replaced them with TCNs. We also experimented with a hybrid architecture, where TCN was used for the encoder and GRU was used for the decoder (VADER with TCN encoder). We also ablated the attention mechanism from both the encoder and decoder. The results in Tab. 3.5 justify our design choices with VADER outperforming its ablated variants across all temporal horizons.

Discussion: The results in Table 3.5 justify the design choices of VADER. We first notice a drop in performance for all evaluated horizons when we ablate the attention mechanism and when we ablaate GRU cell and replace it with TCN for both the encoder and decoder. This implies the benefit of using GRU for motion prediction, as they are particularly effective for auto-regressive implementations. On the other hand, TCN struggles to effectively model the sequential and auto-regressive nature of motion prediction. We observe a performance improvement when we replace the decoder TCN with GRU, with the recurrent gating mechanism in the GRU cell proving effective for recalling past information that is salient for predicting the current frame.

We next ablate the discriminator module and the discriminator objective, thus making the overall objective similar to the original VQ-VAE implementation. As can be observed in Tab. 3.5, ablating the module

Table 3.4: MSE (in cm^2) comparison of different multi-agent motion prediction methods on the KTH-HRC Dataset (Lower is better).

Approaches	Frames					
	5	10	20	30	35	40
Zero-Velocity [29]	0.11	0.34	1.18	2.38	3.07	3.81
Seq2Seq [29]	0.14	0.36	1.09	2.17	2.81	3.49
Seq2Seq-SPL [36]	0.17	0.42	1.20	2.33	2.98	3.66
Scalable + Interpretable [1]	0.06	0.20	0.72	1.61	2.21	2.91
VADER	**0.06**	**0.20**	**0.69**	**1.55**	**2.15**	**2.88**

Table 3.5: Ablation results on the UTD-MHAD Dataset (Lower is better).

Approaches	Frames					
	2	4	8	10	13	15
VADER with TCN encoder-decoder	9.68	19.71	37.27	44.02	52.35	57.21
VADER with TCN encoder	7.85	16.29	33.49	40.68	49.47	54.27
VADER w/o GAN objective	8.08	16.76	33.57	40.39	48.54	52.87
VADER w/o attention mechanism	10.19	22.21	45.26	54.78	66.85	73.92
VADER	**6.61**	**14.22**	**29.82**	**36.23**	**43.83**	**47.81**

leads to a drop in overall performance across all the temporal horizons. This suggests that the standalone reconstruction loss, which in this case is the MSE, is not effective in capturing the spatial and temporal correlations of motion prediction. As such, having a discriminator-based loss that can complement the reconstruction loss can lead to significant performance gains.

3.8 Limitations

In this work, we introduced VADER, a novel sequence learning approach that seeks to overcome some of the long-standing challenges of motion prediction. To tackle the issue of learning a robust representation of past observed poses, we proposed the use of vector quantization to learn a discrete latent space, with no restrictions of a static prior. Additionally, we proposed using the discriminator loss to compliment the MSE objective to improve the accuracy of motion prediction.

The approach primarily focuses on single-modal input, specifically the positional data of human poses. This limitation restricts the model's ability to fully capture the rich and varied information available in real-world scenarios, where multiple modalities such as visual, auditory, and contextual cues play a critical role in understanding and predicting human behavior. In HRI, where robots must navigate and respond to complex, multi-sensory environments, relying solely on positional data can result in a narrow understanding of human intent and actions.

Another significant limitation is the model's lack of explicit mechanisms for handling interactions between multiple humans. The current approach is designed to predict the motion of individual humans without considering the interdependencies and interactions that are common in collaborative settings. In HRI scenarios, where robots frequently need to interact with or among groups of people, the absence of explicit modeling for multiple human agents can lead to inaccurate predictions and suboptimal performance in collaborative tasks.

These limitations highlight areas for future improvement, particularly in expanding the model to incorporate multi-modal data and developing explicit mechanisms for modeling multi-human interactions. Addressing these gaps is crucial for advancing the applicability of motion prediction models in complex, real-world HRI environments.

Chapter 4

IMPRINT: Interactional Dynamics-aware Motion Prediction in Teams using Multimodal Context

As robots increasingly transition from working in isolation to collaborating with humans in integrated teams, they face the challenge of understanding and predicting the actions of multiple human team members. Previous approaches, such as VADER, have made significant strides in motion prediction but were limited by their reliance on single-modality inputs and the lack of explicit modeling for interactions between multiple humans. These limitations are particularly critical in Human-Robot Interaction (HRI) scenarios, where robots must navigate complex, multi-agent environments and respond to diverse, multimodal sensory inputs.

To address these challenges, this chapter introduces IMPRINT, a multi-agent motion prediction framework specifically designed to overcome the limitations of previous methods. By incorporating an Interaction module that models both intra-agent and inter-agent dynamics and a Multimodal Context module that leverages data from various sensors, IMPRINT aims to provide a more accurate and comprehensive prediction of team dynamics. This chapter will explore how IMPRINT addresses the shortcomings of earlier approaches, offering a robust solution for improving motion prediction in real-world human-robot collaboration scenarios.

4.1 Introduction

With improvements in machine perception and intelligence, robots that are capable of holistically perceiving the environment and making autonomous decisions are becoming more prevalent in human-centric environments. Robot perception has made significant advances, allowing them to identify new or unobserved states [123, 124, 125, 104], detect the presence of humans in the workspace and predict future human trajectories [126, 127, 128, 48, 129]. Furthermore, recent works on planning and control have also enabled robots to act under partial observability over long-horizon tasks [130, 131, 132, 133, 134]. The improvements in robot capabilities have enabled robots to move out of the proverbial cage from controlled spaces in assembly lines to unstructured, open-ended environments where they are expected to interact with humans, often as part of human-robot teams [135, 136, 137]. This has encouraged the adoption of robots in highly stochastic collaborative environments such as assistive living, autonomous driving, manufacturing [138, 139, 140, 141].

Robots collaborating with humans can potentially improve task performance and efficiency while reducing the workload of humans. Prior works on robot perception, planning, and control have been developed under the assumption of the robot acting alone. Just as human behavior changes when working alone to when they are coordinating a joint action in a team [142], robot algorithms need to also evolve from their current state, where there is no assumption of collaborative agents in the environment, for them to be effective collaborators. Humans are inherently adept at *joint action*, defined as a form of social interaction where two

or more participants coordinate their actions in time and space while making changes to their environment [143, 9]. Humans acquire the ability to perform a successful joint action through a combination of observation, participation, and explicit teaching [38, 9]. Along these lines, for robots to work in human spaces, they need to capture how humans achieve a high level of mutual coordination and adaptation [144, 20] which in turn allows for a fluent meshing of actions [138, 145, 146, 147, 135].

Several works spanning multiple disciplines of psychology, neuroscience, and cognitive sciences have explored the underlying mechanisms of a joint action task [148, 149, 150, 151]. Sebanz et al. [148] in their seminal work described three essential components of a successful joint action task. The first component involves predicting the intent of the interactional partner. The second involves understanding when to perform the actions jointly, which is crucial for temporal coordination. The last part involves understanding where and how to perform the joint action. The authors described these characteristics as the "What, When, and Where" components of joint action. For robots to achieve human-level fluency when collaborating, their learning modules must capture these aforementioned components of collaboration [57, 135]. From a computational standpoint, this requires modeling the stochastic nature of individual human behaviors and team dynamics and predicting their future motion while ensuring temporal consistency, synchrony, and spatial feasibility.

Previous works on human-robot collaboration have explored the concepts of joint action by modeling human activities and using that knowledge as an input to the robot's anticipatory action planning mechanism [40, 41, 42, 43, 44]. However, the majority of these methods have been modeled from the perspective of dyadic interaction, comprising one human and one robot [45, 38]. Transitioning from dyadic interactions to interactions involving multiple humans, as seen in (Figure 1.1b) and potentially multiple robots constitutes a fundamental change in complexity that is difficult to solve using existing dyadic algorithms [46]. Going beyond dyadic interactions would require robots to understand the inter-agent and intra-agent dynamics within the team while also being cognizant of any external stimuli that may affect the behavior of the team as a whole. Prior approaches for modeling multi-agent interactional dynamics have approached the problem from a social navigation perspective and primarily focused on predicting the trajectories (2-D global positions) of multiple humans in a scene [30, 48, 49, 50]. However, when humans and robots collaborate in close-proximity, robots would require accurate forecasting of human motion (3-D skeletal joint positions), instead of just the global 2-D trajectories.

Recent works on human motion prediction have extended the problem of predicting human motion to multi-agent settings [21, 28]. Multi-agent motion prediction is especially challenging as it is necessary to model multiple human motions while also needing to disentangle any interactional dynamics between them, as observed in Figures 1.1 & 4.1. Recently, Yasar et al. [21] proposed an attention-based mechanism to model the inter-agent dynamics that learn to weigh the relevant features from each agent. Adeli et al. [28] proposed the use of permutation invariant operators such as average/max pooling to model the social interaction among the people while also incorporating scene context using the spatio-temporal representations from the video. The authors rely on social pooling mechanisms for modeling interaction that are prone to losing valuable information [82]. Although the aforementioned works have shown promising results, predicting multiple agents' motion in team settings remains an open problem, with challenges ranging from extracting robust representations of individual agents and modeling their dynamics to obtaining complementary information from multimodal input.

Mutlimodal Data: Another unresolved challenge of human motion prediction approaches is that existing models have predominantly used data from the unimodal sensor, such as a skeleton or visual sensor [35, 29, 36, 38, 21], which suffers from a single modality failure. For example, visual occlusion can degrade a learning model's performance solely depending on the data from a visual sensor. Similarly, skeletal data can sometimes be noisy or incomplete due to sensor errors or occlusions, which can adversely impact the performance of prediction models that rely solely on this type of data ([76, 141]). While skeletal data is a valuable resource for motion prediction, additional modalities can provide complementary information that enhances the robustness and accuracy of the predictions. For example, visual and depth data can provide additional information about the spatial relationships between different body parts and the surrounding environment, which can be useful for predicting complex motions involving interactions with objects or other people [152, 153, 101, 154, 155]. Multimodal learning models that combine data from multiple sensors have been shown to improve performance on a variety of tasks, including activity recognition [152, 156, 153,

101, 154, 155], affective states recognition [157], visual-language representations learning [158, 159, 160, 161], and visual question answering [162, 163]. This suggests that integrating multiple modalities can provide a more comprehensive and robust representation of human motion, leading to more accurate predictions.

Recently, a few works have utilized multimodal sensor data to improve human motion prediction [28, 50]. However, there remain several challenges to extracting robust multimodal representations for motion prediction. Although these models can improve motion prediction involving dyadic interactions, these models can not effectively utilize the multimodal representations to improve motion predictions in multi-agent interactions. The primary lacking of these models is that existing learning models fuse data from multimodal sensors without considering inter-agent and intra-agent dynamics. Multimodal representations can effectively help to extract these dynamics, which can improve motion prediction in multi-agent settings. This is in line with human sensing, which utilizes multiple modalities to perceive the environments to effectively collaborate in teams ([164, 165, 166, 167, 168]).

To address the above challenges, we propose IMPRINT: Interactional Dynamics-aware Multi-agent Motion Prediction in Teams using Multimodal Context [1], which can explicitly model the interactional dynamics of all the team members and the multimodal context from non-skeletal modalities and fuse them adaptively to predict human motion in team settings. With IMPRINT, we take inspiration from prior works on joint action [9, 148, 150, 169] and connect these findings with the current convention of motion prediction [117, 21, 29, 112]. In line with current approaches to sequence learning and motion prediction, we propose the encoder-decoder architecture for IMPRINT and augment it with findings from joint action by introducing specialized modules to capture the *interactional dynamics* and the *multimodal context*. Along this line, we have designed IMPRINT's architecture to contain four distinct modules: i) an encoder that is tasked to learn a rich representation over the past observation of each agent, ii) an inter-agent attentional mechanism that uses conditional attention to model the interactional dynamics by extracting relevant information regarding individual agents and the team dynamics and fusing them adaptively, iii) a multimodal attentional mechanism that extracts complementary information from other modalities to generate a robust multimodal context and finally, iv) a decoder that is tasked to use the interactional dynamics and multimodal context to auto-regressively predict the future human motion of all the agents in the team.

We evaluated the performance of IMPRINT, first in scenarios involving only human-human interaction scenarios, on popular multi-agent datasets: NTU RGB+D 60 ([55]) and CMU Panoptic ([56]), comprising two or more people involved in several joint activities. Next, we evaluated our approach to human-robot collaboration scenarios on the KTH-HRC dataset [38], which involves collaborative activities between a human and a robot. Our experiments evaluated the generalizability of IMPRINT and other methods across all these different scenarios while assessing the performance of these sequence learning methods over different temporal horizons, ranging from short to long-term. The results suggest that IMPRINT outperformed state-of-the-art human motion prediction methods over every evaluation scenario and tested time horizons, achieving significant improvements on the NTU RGB+D 60 dataset (up to 1.20 cm^2), the CMU Panoptic dataset (up to 0.35 cm^2) and the KTH-HRC dataset (up to 0.49 cm^2) for long-term horizons. In addition, we perform extensive ablation studies on two multi-agent datasets. The ablation studies justify our framework's architectural and optimization design choices. Additionally, our significance analysis further instantiates that our proposed multimodal representation learning model can significantly outperform all the evaluated human motion prediction models over multiple iterations of the experimental runs. Finally, to complement our extensive quantitative analyses, we provide an interpretation of how IMPRINT robustly models the influence of the interactional dynamics and the multimodal contextual information when predicting human motion in multi-agent settings.

4.2 Related Work

Human motion prediction: Human motion prediction is widely considered as one of the essential parts of robotic intelligence that would enhance robot perception and allow for rapid and high fidelity reactions towards complex environment changes [26, 27, 21, 28, 29]. The problem has been extensively studied due to

[1] Code availability: https://github.com/MohammadYasar/IMPRINT

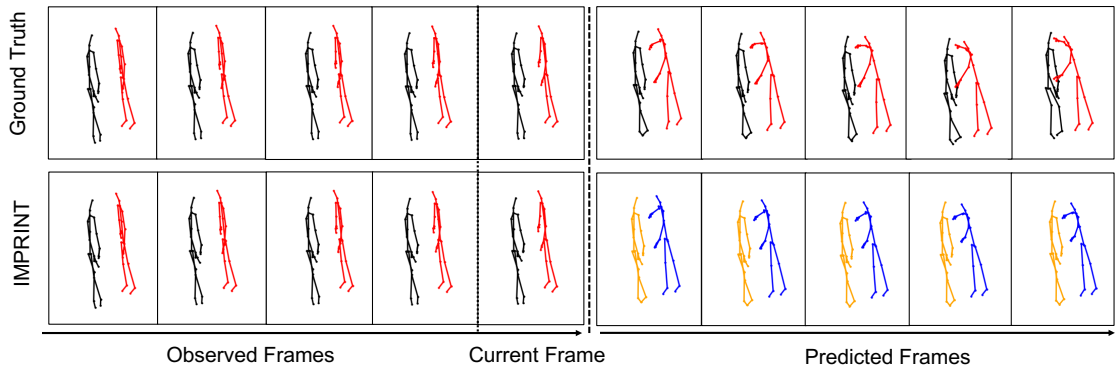


Figure 4.1: Qualitative Performance of IMPRINT compared to the Ground Truth. IMPRINT's modeling of Interactional Dynamics and Mulitmodal Context allows it to generate feasible joint poses which is similar to the ground-truth poses for a *hugging* activity.

its significance and challenges in computer vision and machine intelligence. Some of the earliest approaches to forecasting motions are designed using Hidden Markov Models and Gaussian processes. Lehrmann et al. [32] proposed the use of latent-variable models that follow state-space equations modeled by Hidden Markov Models. Taylor et al. [33] introduced the use of conditional restricted Boltzmann machines (RBM) for the task of motion prediction, which assumed a binary latent space and performed prediction by decoding the RBMs. Wang et al. [34] used Gaussian-Processes to perform non-linear motion prediction.

More recently, several works in motion prediction have adopted a data-driven approach using Recurrent, or other forms of Neural Networks [28, 36, 35, 21, 29, 111, 112]. Jain et al. [79] proposed structural RNN that tries to capture the spatio-temporal relationship of skeletal joints using graphs. The authors model the skeletal hierarchy by dividing it into five major skeletal clusters, each representing a specific portion of the body joints. Moreover, Fragkiadaki et al. [35] introduced an Encoder-Recurrent-Decoder (ERD) that includes non-linear transformations using multi-layer feed-forward networks to extract the pose features before passing them to the LSTM cell to encode the history information through its recurrent architecture. Additionally, Martinez et al. [29] further extended this scheme by modeling the velocity component of motion prediction. The authors introduced residual connections in the decoder part of the architecture, which significantly improves the motion prediction in terms of smoothness and accuracy. Furthermore, the authors demonstrated how previous works fail to outperform a simple zero-velocity baseline, i.e., constantly predicting the last observed pose. To improve the structural prediction of the models and allow the generation of more feasible human joints, Aksan et al. [36] introduced a structural layer at the decoder (SPL). The SPL layer imposes a structural prior on the output by predicting each joint hierarchically, thus imposing conditional dependence between joints. Moving away from skeleton input, Zhang et al. [170] proposed predicting future 3D mesh from past video input. The authors proposed using autoregressive models that learn a latent representation of 3D human dynamics using a causal temporal encoder. The latent representation is then fed to an autorgressive model to predict 3D human motion.

While most works on motion prediction have adopted a deterministic approach, recent works have also approached the problem as learning a probability density function of future human poses, conditioned on past pose [37, 38, 49, 83, 39, 115, 116]. Butepage et al. [37] and Toyer et al. [39] adopted the Variational Autoencoder framework for predicting human motion, which relies on learning a functional mapping from the input data space to the latent space at the encoder, and from the latent space to the reconstructed data space at the decoder. Barsoum et al. [83] proposed HP-GAN, which is trained with a modified version of the improved Wasserstein generative adversarial networks (WGAN-GP). Kundu et al. [116] proposed the BiHMP-GAN where a bi-directional GAN was designed to avoid the problem of mode-collapse. Yasar et al. [21, 117] proposed the use of discriminators on the latent space to enforce a prior instead of using KL-divergence or other distribution matching losses such as JS-divergence [37, 39].

Due to the sequential nature of motion prediction, there has been a recent trend of using the attention

mechanism in motion prediction [21, 113, 114, 112, 3], which has been popularized by recent progress in Natural Language Processing [84, 171, 172]. Mao et al. [112] introduced an attention-based feed-forward network that captures *motion attention* to capture the similarity between the current motion context and the historical sub-sequences, where each sub-sequence is represented using the Discrete Cosine Transform (DCT). Yasar et al. [21] used the attention mechanism at the encoder to learn a more salient representation from the different encoder streams and at the decoder to allow the auto-regressive decoder to weigh between the latent space and its past output. Liu et al. [114] used spatial and temporal attention to model relationships between the skeletal joints and the temporal frames, respectively.

While prior works in motion prediction have significantly advanced the state-of-the-art, generating feasible and temporally coherent human poses remains an open research problem [29, 36, 112, 21]. Furthermore, unlike other explored areas of machine intelligence, such as computer vision or machine translation, there is still a lack of consensus on the best framework to capture the spatial and temporal dynamics of human motion. Although recent approaches have adopted an encoder-decoder framework, learning a robust representation at the encoder that best captures the past poses is still an ongoing research effort.

Multi-agent motion prediction: People seldom work in solitude, and their behavior is often conditioned on other objects or humans in the team [103, 21, 9]. Thus, for any intelligent agent to interact with humans closely, there needs to be a provision for modeling the group dynamics [103]. Multi-agent motion prediction poses the same challenges as single-agent motion prediction while also introducing newer requirements due to the need to model human dynamics in team settings. Although multi-agent motion prediction is a recently explored problem with only a few notable works [28, 21], there is a parallel line of research in social navigation that focuses on predicting the trajectory of multiple agents in a given scene [30, 48, 128, 50]. Unlike multi-agent motion prediction, in social navigation, the problem breaks down to predicting the trajectory of multiple agents, where the trajectory is represented by the global locations of all the agents, particularly their 2D locations.

Alahi et al. [30] introduced social-LSTM, which uses agent-specific LSTM to summarize past observations of each agent. The hidden states of the neighboring LSTMs are connected through a social pooling strategy and used as the input to the LSTM cell at the next timestep. Instead of using agent-specific LSTMs and pooling for every timestep, Gupta et al. [48] proposed Social GAN, which reused the LSTM across all agents and introduced a computationally efficient pooling mechanism comprising a Multi-Layer Perceptron followed by max pooling. To explicitly model the cross-agent and scene interaction, Park et al. [50] proposed the use of the attention mechanism to model the agent-to-agent and agent-to-scene interactions for predicting diverse and admissible trajectories of multiple agents. To enforce dynamic constraints and incorporate environment information, Salzmann et al. [128] proposed Trajectron++, which uses a modular, graph-structured recurrent model to predict the trajectory of multiple agents while also incorporating agent dynamics and other modalities of data.

Towards addressing the challenges of multi-agent motion predictions, recent works have proposed extending the single-agent motion prediction architectures by incorporating social, and scene context using pooling mechanisms [28] or attention mechanism [21]. Adeli et al. [28] extended the seq2seq architecture [29] to multi-agent scenes, where each agent's past motion was first encoded using GRU cells before being passed to a permutation-invariant pooling mechanism while pools the aggregated features from all the agents. The pooled features were passed along with the agent-specific representation to the decoder. Yasar et al. [21] also relied on GRU cells to obtain the representations of past motion for all the agents. The representations were passed to an attention module which was tasked to disentangle and extract relevant multi-agent features from agent-specific representation.

Although prior works on multi-agent motion prediction have provided initial directions for modeling the motion of multiple humans in a shared workspace, they lack the provision to explicitly model the interactional dynamics between the agents. Thus, there is no concrete approach to understanding the influence of one agent on another. Furthermore, multi-agent motion prediction inherits the same sets of challenges as single-agent motion prediction, such as learning a robust representation at the encoder. These problems, however, get amplified in multi-agent scenarios due to the presence of multiple agents.

Multimodal representation learning: Humans naturally use multisensory systems (visual, auditory,

haptic, and verbal) to perceive the action and motion of others to collaborate in teams effectively [154, 101]. For robots to be effective team members in collaborative settings, robots need to perceive the action and motion of human team members using multimodal context [141, 76]. Several multimodal representation learning models have been proposed in the literature for various tasks, such as human activity recognition [152, 76, 155, 101, 154], affective states recognition [157], visual question answering [163, 162], and video recognition [173, 174, 175, 176]. These approaches fuse multimodal information in three ways: early, late, and intermediate fusion [76, 177, 168, 178]. Early fusion approaches fuse raw sensor data before extracting feature representations [141, 177]. Although early fusion is a straightforward approach to implement, this fusion approach is not suitable for fusing data from multiple modalities with heterogeneous feature distributions, such as fusing data from video, skeletal and physical sensor modalities [76, 163]. Unlike the early fusion approaches, late fusion approaches extract unimodal feature representation to produce the task outputs and fuse these outputs to produce combined task outputs [179, 173, 76, 177]. Although late fusion approaches allow for combining task decisions from multiple heterogeneous modalities, these approaches can not extract complementary multimodal representations [154]. Several intermediate fusion approaches have been proposed to address the deficiency of early and late fusion approaches where the feature representations are fused at the intermediate layers [76, 180, 178, 168, 179, 173, 174, 181, 182].

As intermediate fusion approaches show improved performance over the early and late fusion approaches, state-of-the-art multimodal representations learning model predominately uses intermediate fusion approaches [154, 177, 179, 183, 166, 152, 184, 120, 185, 186, 187]. For example, Lee et al. [188] developed a multimodal learning approach to combine visual and haptic feedback to extract complementary multimodal representations for task policy learning. Moreover, several works have been proposed to combine visual and haptic data modalities to extract multimodal representation for various tasks, such as manipulation, material recognition, and object categorization [189, 190, 191, 192]. Additionally. several datasets [153, 193, 141, 152, 156] and multimodal representation learning approaches have been proposed in the literature for activity and affective states recognition [157, 120, 152, 101]. Islam and Iqbal [76] proposed a multimodal attention-based fusion approach to combine unimodal representations from multiple modalities to recognize activities. Although this multimodal attention approach can improve the activity recognition accuracy, this approach did not consider the inter-modality interaction, which can help extract robust representation. To address this issue, Islam and Iqbal [155] developed a multimodal graphical attention approach that calculates inter-modality interaction to extract robust multimodal representation. Their experimental analysis showed that inter-modality interaction outperformed self-attention-based multimodal fusion approaches.

Additionally, intermediate fusing approaches either used feedforward [120, 152, 182, 173, 162, 163, 174] or recurrent model architecture [154]. For example, Islam et al. [154] proposed a recurrent information processing-based multimodal fusion approach, and their experimental results suggest that recurrent multimodal fusion outperformed the feedforward-based multimodal fusion approaches. However, the authors also noted that recurrent information processing increases the complexity of the model compared to the feedforward-based fusion approaches. This model complexity issue can be resolved by fusing multimodal representations using auxiliary information, such as activity group label [101]. However, state-of-the-art datasets may not contain this auxiliary information for all tasks. In those situations, feedforward multimodal fusion approaches are suitable to fuse heterogeneous modalities to extract robust representations.

Although feedforward model architecture reduces model complexity compared to the recurrent or multitask models, the existing model of human motion prediction naively fused (concatenation or sum) multimodal representations to predict human motion. Thus, these models can not effectively utilize the multimodal information for robust human motion prediction. Additionally, we can use the multimodal information to extract inter-agent and intra-agent dynamics which can help to predict human motion in multi-agent settings.

Motion prediction for Human-Robot Interaction (HRI): The fields of Human Motion Prediction and Human-Robot Interaction have independently garnered significant attention and advances in recent years. The synergy between the two fields is natural: by leveraging motion prediction, we can equip robots with the ability to anticipate and interpret human actions and intentions [21, 103]. Predicting human motion patterns empowers robots to anticipate future movements, enabling them to proactively adjust their behavior and respond appropriately in dynamic and collaborative environments [15]. This capability significantly enhances the perception and understanding of human behavior, thereby fostering more natural and intuitive

interactions.

The concept of prediction has been explored across various applications of HRI, from shared autonomy [194, 195] to social navigation [126, 127] and autonomous vehicles [49, 128, 48]. Aronson et al. highlighted the use of multimodal information for teleoperation, where the authors combine user joystick input and user gaze information to more accurately predict user intent [194]. Schmerling et al. [49] demonstrate the benefit of human intent prediction in settings where multimodality, i.e., the possibility of multiple highly distinct futures, plays a critical role in decision-making. The authors learn a multimodal probability distribution over future human action to perform a real-time robot policy construction for the domain of traffic weaving, where cars need to swap lanes in a short distance. Tang et al. [26] demonstrate the applicability of motion prediction for planning via computing a conditional probability density over the trajectories of other agents given a hypothetical rollout of the 'self' agent. The authors introduced an attention-based state encoder that learns latent variables to jointly model the multi-step future motions of agents in a scene, and demonstrate the efficacy of their technique for simulated and real-world vehicle trajectory datasets.

As human motion prediction plays a significant role in safe human-robot interaction, there exists an intrinsic connection between HRI and Human Motion Prediction. However, the bulk of prior works at the intersection of the two domains has leaned towards the autonomous vehicle domains, where prior work models future rollouts of human drivers, or shared-autonomy, where prior work models discrete human goals, often towards a target object. Leveraging human motion prediction at a granular skeletal level would present a starting point for conditional robot policy generation and is crucial to unlocking the potential of close-proximity human-robot collaboration.

4.3 Problem Formulation

Our goal is to enable fluent collaboration in human-robot teams by equipping the robot with the ability to anticipate the motion of all the human collaborators in the team. We assume access to skeleton data, which can be accessed directly from off-the-shelf sensors such as Kinect, or extracted from any RGBD sensors. Formally defined, human motion prediction is the problem of predicting the future human pose over a horizon, given their past pose and any additional contextual information. For simplicity, let us first introduce the problem for single-agent motion prediction and then extend the formulation to multiple humans. In all our formulations, we use superscripts to represent agents and subscripts to represent time.

Let us first assume there to be one agent, a. The model has access to the past or *observed* pose trajectory of the agent, spanning time $t = 1$ to τ: $\mathbf{X}^a = \{x_1^a, \ldots, x_\tau^a\}$ and any additional contextual information: $\mathbf{C} = \{c_1, \ldots, c_\tau\}$. Each pose frame $x_t^a \in \mathbb{R}^N$ denotes the N-dimensional body pose. N depends on the number of joints in the skeleton, J and the dimension of the joints D, where $N = J \times D$. The context frame c_t provides additional contextual information and comprises raw data from complementary modalities, such as RGB, Depth etc.

The expected output of the model is the future trajectory frames over horizon H, i.e., the ground truth pose over the horizon $t = \tau + 1$ to $\tau + H$: $\mathbf{Y}^a = \{y_{\tau+1}^a, \ldots, y_{\tau+H}^a\}$. Our first objective is to learn the underlying representation which would allow the model to predict accurate and feasible future human poses $\hat{\mathbf{Y}}^a = \{\hat{y}_{\tau+1}^a, \ldots, \hat{y}_{\tau+H}^a\}$. We assume that future human pose is conditioned on the past observed and generated poses and predict each frame in an auto-regressive manner as formulated below:

$$p_\theta(\hat{\mathbf{Y}}^a) = \prod_{\delta=\tau+1}^{\tau+H} p_\theta(\hat{y}_\delta^a | \hat{y}_{\tau:\delta-1}^a, x_{1:\tau}^a, c_{1:\tau}) \tag{4.1}$$

While equation 4.1 provides a formulation to predicting the motion of a single agent, our ultimate goal is to predict the motion of multiple agents in team settings. Thus, we now work towards scaling equation 4.1 to incorporate multiple agents. We assume that there are K agents in the scene and the value of K is known a priori. The input to the model would then comprise the observed pose information of all the agents in the scene, spanning time $t = 1$ to τ: $\mathbf{X} = \{X^1, \ldots, X^K\} = \{x_1^{1:K}, x_2^{1:K}, \ldots, x_\tau^{1:K}\}$ and additional context information: $\mathbf{C} = \{c_1, \ldots, c_\tau\}$. The expected output of the model is the future trajectory frames

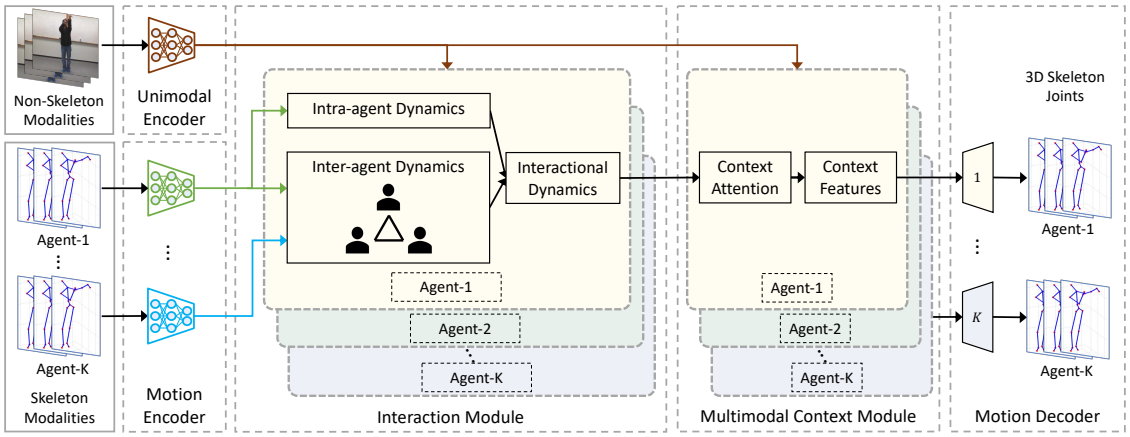


Figure 4.2: IMPRINT: Interactional Dynamics-aware Motion Prediction in Teams using Multimodal Context. IMPRINT consists of four module: Motion Encoder which is tasked to extract spatio-temporal representations for each agent, Interaction Module, which models the interactional dynamics among all the agents, Multimodal Context Module, which learns a complementary representation from other modalilites and Motion Decoder, which autogressively predicts the motion of all the agents. Our implementation is available on Github (https://github.com/MohammadYasar/IMPRINT).

over horizon H, i.e. the ground truth pose over the horizon $t = \tau + 1$ to $\tau + H$: $\mathbf{Y} = \{Y^1, \ldots, Y^K\} == \{y^{1:K}_{\tau+1}, y^{1:K}_{\tau+2}, \ldots, y^{1:K}_{\tau+H}\}$.

We assume that the future human pose of each agent is conditioned on the observed poses of all agents, plus additional contextual information, and predict each frame in an auto-regressive manner. Thus, the multi-agent motion prediction problem can be formulated as follows:

$$p_\theta(\hat{\mathbf{Y}}^a) = \prod_{\delta=\tau+1}^{\tau+H} p_\theta(\hat{y}^a_\delta | \hat{y}^a_{\tau:\delta-1}, x^{1:K}_{1:\tau}, c_{1:\tau}); \quad \forall a = 1, \ldots, K \tag{4.2}$$

4.4 Proposed Approach

We now introduce our proposed framework, IMPRINT: Interactional Dynamics-aware Motion Prediction in Teams using Multimodal Context. IMPRINT aims to accurately predicting human motion while being holistic to all the available modalities of data. The overall algorithm for our framework is provided in Algorithm. 1 and illustrated in Fig. 4.2. There are four modules of IMPRINT, which are as follows:

- **Motion Encoder:** The motion encoder is tasked with extracting spatio-temporal representations for each agent. The input to the encoder is the different components of motion. Unlike prior works, which use only the skeleton modality to model human motion, we extract motion information from skeleton modalities as well as from other data modalities, such as RGB.
- **Interaction Module:** The interaction module models the intra-agent and inter-agent dynamics among all the agents in the scene. It augments the representation of each agent by adaptively weighing the influence of other agents in the workspace.
- **Multimodal Context Module:** The context module aims to obtain a holistic representation of the environment by considering data from modalities on top of the existing skeleton modality. It is tasked to augment the representation for each agent by adaptively weighing the effects on the interaction of other available modalities, such as RGB, Depth, and fuse them in a complementary manner.
- **Motion Decoder:** The decoder is tasked to auto-regressively predict the motion of all the agents. It conditions the motion of each agent on the past generated motion of that agent as well as the past

representations of all the other agents.

We now describe each of IMPRINT's modules in detail in the following subsections.

4.4.1 Motion Encoder

The motion encoder module is tasked with extracting motion features from the past observed frames. Although we are predicting the *motion* of each agent, the input to the motion encoder is a series of *static* 3-D skeleton frames, with each frame representing joint poses, as well RGB/Depth images from other modalities (see Fig. 4.2). Thus, the first task of the motion encoder is to extract motion features from this static input sequence.

Towards achieving this, we extend the input stream from the skeleton modality, which initially comprised only position features, to contain velocity and acceleration data. These are calculated by considering the position values of each joint over time and performing a first and second differentiation over the temporal axis. Thus, the velocity and acceleration features are first and second-order derivatives of the position values of each joint. We take the derivative over each timestep to increase the granularity of the velocity and acceleration features.

We pass the position, velocity, and acceleration streams to separate encoders. Each encoder aims to learn a spatio-temporal representation over the past observation for a given agent. As we pose this as a sequence learning problem, we employ Recurrent Neural Networks (RNN), in particular, unidirectional Gated Recurrent Units (GRU) in the encoders, to extract temporal feature representations for each stream. GRUs represent one form of Recurrent Neural network architecture that are designed to capture and model sequential data by incorporating gating mechanisms. These mechanisms control the flow of information within the network, allowing it to selectively retain and update relevant information over time. Our choice of unidirectional GRUs over a bi-directional architecture is motivated by our need to predict human motion while minimizing the computational load on the robot. We choose GRUs over other RNNs such as LSTMs due to their comparative performance to LSTMs, while also being computationally more efficient. For each stream, the stream-specific GRU aims to encode the spatio-temporal information over the input sequence, which is formulated as:

$$h_{s,t} = GRU(h_{s,t-1}, x_{s,t}, \phi_s) \tag{4.3}$$

where s represents position, velocity, or acceleration. Here, $x_{s,t}$ represents the input to the GRU at time t and will take the value of $x_{pos,t}, x_{vel,t}, x_{acc,t}$ for position, velocity and acceleration at time t, respectively. $h_{s,t-1}$ represents the past hidden output at time $t-1$ and ϕ_s represents the stream-specific encoder weights for the GRU. The output from each encoder is passed to a keyless attention module [120]. The attention module is tasked to sparsely and adaptively extract the salient features from the three streams.

For non-skeleton modalities, we use separate modality-specific encoders to extract spatial features over the past trajectory frames. This is important for modalities such as RGB and Depth, which are unstructured and require spatial and temporal feature extraction. We use a modality-specific spatial feature encoder as it allows us to use pre-trained state-of-the-art feature encoders. Furthermore, we ensure that there is no interaction among the unimodal encoders at early layers, as such interaction may prevent the encoders from capturing modality-specific feature characteristics [76, 196, 163]. For extracting spatial features from the RGB frames, we use the ResNet-50 [197] architecture, which is a popular architecture for feature extraction in computer vision. One of the key benefits of the Residual Network (ResNet) family of Convolutional Neural Networks (CNNs) is their ability to effectively capture and represent high-level features from images. This motivated our choice of using the ResNet family of feature extractors to obtain spatial representation from images.

Once we have obtained the spatial features over the past trajectory frames, we pass the sequence of spatial features to a GRU. Similar to the skeleton encoders, the GRU is tasked with capturing the spatio-temporal representations from the sequence of spatial features. The operations can be summarized below:

$$
\begin{aligned}
X^u_{m,t} &= UFE_m(X^r_{m,t}) \\
v_{m,t} &= GRU(X^u_{m,t-1}, X^u_{m,t}, \phi_m)
\end{aligned}
\tag{4.4}
$$

where m represents the modality, which can be RGB, Depth, or any other non-skeleton modality. Here, X^r_m, t represents the raw unimodal data stream for modality m at time t. UFE_m represents a unimodal encoder which we use to extract spatial features, $X^u_{m,t}$. The extracted features are fed to the GRU to obtain a spatio-temporal representation $v_{m,t}$.

4.4.2 Interaction Module

To develop the Interaction module, we use the findings from the works on joint action [150, 169], which indicate that humans integrate self-behavior with a simultaneous prediction about other's behavior. As such, the interaction module is tasked with modeling the intra and inter-agent dynamics among all the agents in the scene. As we aim to predict the motion of all the agents, we introduce a novel mechanism to explicitly model the intra and inter-agent interaction and predict future trajectory conditioned on the interaction.

As illustrated in Fig. 4.2, the input to the module is the spatio-temporal representation obtained from the motion encoder. For each agent, we first model the *intra-agent dynamics*, where the goal is to identify which of the motion features provide the most information to predict the agent's future pose and obtain a salient representation by weighing them accordingly. The intra-agent dynamics are agent-specific and allow us to observe motion prediction at a higher granularity by exploring the impact of each of the different data streams on the agent's own motion. As each agent's representation comprises position, velocity, acceleration, and context representation, we aim to adaptively weigh the impact of each of these streams before fusing them to obtain a robust representation. This is performed using the attention mechanism [198], which is used to model the attention weights over the different representations. In our proposed approach, we adopted a lightweight attention mechanism similar to the Keyless attention approach proposed by Long et al. [120], instead of utilizing the resource-intensive self-attention approach [84] used by prior works [21, 76]. The overall operations are summarized below:

$$
\begin{aligned}
h^a_t &= [h^a_{pos,t}; h^a_{vel,t}; h^a_{acc,t}; v_{m,t}] \\
h^a_{intra,t} &= Attention^a(h^a_t)
\end{aligned}
\tag{4.5}
$$

where $h^a_{pos,t}, h^a_{vel,t}, h^a_{acc,t}$ represents the position, velocity and acceleration representations respectively for agent a. $v_{m,t}$ represents the spatio-temporal representation of non-skeletal modality which is encoded using a unimodal context encoder. $Attention^a$ represents the intra-agent interaction mechanism. For each agent, we have agent-specific attention module. $h^a_{intra,t}$ represents the weighted the output of the intra-agent interaction mechanism and is the weighted-sum of the different streams.

We use the attention mechanism to model agent dynamics as it provides an effective and lightweight way to model the dependencies while also reducing the likelihood of loss of information, unlike other aggregating strategies such as the pooling mechanisms. The attention weight is calculated in the following way: given a sequence of input representation vectors $(h^a_{pos,t}, h^a_{vel,t}, h^a_{acc,t}, v_{m,t})$ we seek to compute the weights over the provided input sequence. These weights are used to fuse the input representations. This is summarized below:

$$
h_{att} = \sum_{i=1}^{n} \lambda_i h_i \tag{4.6}
$$

Here h_{att} represents the fused representations over the input vectors h_i, and is the output of the attention mechanism. h_i can be any of the input vectors $h^a_{pos,t}, h^a_{vel,t}, h^a_{acc,t}, v_{m,t}$. λ_i represents the attention weights of each h_i and is computed as follows:

$$e_i = w^T h_i \tag{4.7}$$

$$\lambda_i = \frac{\exp(e_i)}{\sum_{j=1}^{n} \exp(e_j)} \tag{4.8}$$

where w^t is a learnable parameter. We utilize a lightweight 1D-CNN with a kernel size of 1 to calculate the attention weights λ_i.

Having obtained the intra-agent dynamics for all the agents in the scene, we next model the *inter-agent dynamics*. Here, we are interested in modeling how each agent's motion influences the motion of other agents in the team. This is in line with findings in joint action [150, 169], which states that the behavior of one agent is conditioned on their own behavior and the behavior of all the other agents in their vicinity. To perform this, we now consider the motion representations of all the agents to extract the weighted representations from them and re-use the attention mechanism $Attention^a$. For each agent, we concatenate the motion representations of all the agents and pass them to the agent-specific attention module, $Attention^a$. For each agent a, this is modeled as follows:

$$\begin{aligned} h_t &= [h_t^a; \ldots; h_t^K] \\ h_{inter,t}^a &= Attention^a(h_t) \end{aligned} \tag{4.9}$$

where h_t represents the extracted representations from multiple modalities of all the agents, $Attn^a$ now models the inter-agent interaction mechanism for agent a. $h_{inter,t}^a$ represents the weighted the output of the inter-agent interaction mechanism and is the weighted-sum of the representations of the different agents.

Finally, having modeled the intra and inter-agent interaction representations, we need to adaptively weigh these representations to obtain the interactional dynamics for each agent. This is performed by employing the attention mechanism, similar to the one used in prior steps. Here, the input for each agent now comprises the agent-specific representation and the inter-agent dynamics. The final output of the interaction module is the weighted sum of the intra-agent and inter-agent representations for each agent, and is as follows:

$$\begin{aligned} h_t^a &= [h_{intra_t}^a; h_{inter_t}^a] \\ h_{interact,t}^a &= Attention^a(h_t^a) \end{aligned} \tag{4.10}$$

Here, $h_{interact,t}^a$ denotes the *interactional dynamics*-aware representation for agent a and is computed for each agent.

4.4.3 Multimodal Context Module

As modern robots are equipped with multiple sensors such as RGB Camera, LiDAR, and Odometer, robot perception requires algorithms that can obtain a complimentary representation over the different data modalities. This would allow the robot to have a holistic representation of its environment, which is particularly important for close-proximity human-robot collaboration.

Along these lines, in IMPRINT, we propose a Multimodal Context Module that can obtain information beyond those available in the skeleton modality by fusing representations from multiple modalities of data in a complementary manner. Fusing heterogeneous data modalities is a non-trivial task as different modalities have disparate feature distributions, and data are captured using different sensors. The input to the multimodal context module is the agent-specific interactional dynamics representations and the representations from other modalities, as illustrated in Fig. 4.2. These representations are then passed to the context attention mechanism that extracts salient representations for each agent, conditioned on the skeleton and other contextual information. The operations of the context module can be formulated as follows:

$$\begin{aligned} h_{combined,t} &= [h_{interact,t}^a; \ldots; v_{m,t}] \\ h_{context,t}^a &= Context\_Attention^a(h_{combined,t}) \end{aligned} \tag{4.11}$$

Here, $h^a_{interact,t}$ represents the output of the interaction module, $v_{m,t}$ is the representation of the non-skeleton modalities and $h_{combined,t}$ corresponds to the concatenated representations of the different modalities. These representations are then passed to the agent-specific $Context\_Attention^a$ module. Furthermore, to stabilize training and mitigate the diminishing gradient problem due to multiple agents and multiple modalities, we add a residual connection between the output of the context module $h^a_{context,t}$ and $v_{m,t}$.

4.4.4 Motion Decoder

We developed an auto-regressive decoder, i.e., it uses the output of previous timesteps to predict the current pose and has only one stream as the output: 3-D skeletal joint position. We design agent-specific decoders to model the idiosyncrasies of each agent. The input to the decoder is the multimodal context representations, summarizing the interactional dynamics among all the agents and the context representations. In addition, the decoder has access to the last predicted frame. This is passed to a multi-head self-attention module, which learns the attention weights between the immediate agent-specific past output and the output of the context module that represents multi-agent and multimodal interactions.

The first part of the decoder is a GRU cell, that takes as input the output of the multi-head self-attention module as well as the output of the last timestep. This is followed by a fully connected layer. The operations at the decoder are formulated as follows:

$$\begin{aligned} p^a_t &= [h^a_{context,t}, h^a_{dec,t-1}] \\ p^a_{att,t} &= Att(p_t, \phi_{att}) \\ h^a_{dec,t} &= GRU(S^a_{t-1}, p^a_{att,t}, \phi_{pos}) \\ S^a_t &= \gamma(h^a_{dec,t}) \end{aligned} \tag{4.12}$$

where $h^a_{context,t}$ is the representation summarizing the interactional dynamics among all the agent and the context representations, $h^a_{dec,t-1}$ is the previous hidden output of the GRU. $p^a_{att,t}$ is the output of the attention mechanism in the decoder, which is passed to the GRU along with the previous GRU output S^a_{t-1}. ϕ_{att} and ϕ_{pos} represents the weights of the attention module and GRU cell respectively. γ represents the output layer of the decoder with $\mathbf{S_t}$ being the *predicted motion at time t*. We add a residual connection between decoder output at the previous and current timestep, which improves short-term prediction and results in a smoother output sequence [29].

4.5 Experimental Setup

4.5.1 Datasets

We extensively evaluated IMPRINT in team settings across two different scenarios. Our first scenario is in teams where all the members are humans. This provides a highly complex and contact rich environment where humans interact amongst themselves in goal-directed activities. Here, it is important to forecast the motion of each human conditioned on *observed* variables such as their motion and the motion of the others around them, as well as *unobserved* variables such as the interaction dynamics and the goal of the team. For our second scenario, we evaluated IMPRINT in a human-robot team setting. In this scenario, a human and a robot are in close-proximity and performing manipulation-oriented tasks which have tight coupling, such as hand-shake, hand-over. As such, it is important to model this *coupling* through modeling interaction dynamics and use it to forecast the motion of the human. In all scenarios, we predicted the motion of the humans, as we do not have access to their policy.

For the first scenario, in human-only teams, we evaluated the performance of our approach by applying it to two large and widely used multi-human datasets: NTU RGB+D 60 [55] and CMU Panoptic [56]. For NTU-RGB+D 60 dataset, we focused on the action classes involving more than one agent, resulting in 11 joint actions in total, ranging from hand-shaking to hugging, similar to previous work [28]. We used the cross-subject evaluation scheme [55], with 20 subjects for training and validation and a separate 20 for

Algorithm 1: IMPRINT

Input: Dataset D, Agents K, Modules: Motion Encoder p_θ, Motion Decoder q_ϕ, IntraAttention g_θ^{intra}, InterAttention g_θ^{inter}, InteractAttention $g_\theta^{interact}$, Context Attention z_θ

Output: Skeletal Joint Positions of all agents: S_t

for x_t, c_t, y_t *in* D **do**
  for a *in* K **do**
    # **Motion Encoder:**
    $h^a_{pos,t}, h^a_{vel,t}, h^a_{acc,t}, v^a_{m,t} \leftarrow p_\theta(x_t, c_t)$
  end
  # **Interaction Module:**
  $h^a_{intra,t} \leftarrow g_\theta^{intra}(h^a_{pos,t}, h^a_{vel,t}, h^a_{acc,t}, v^a_{m,t})$
  $h^a_{inter,t} \leftarrow g_\theta^{inter}(h^a_{pos,t}, h^a_{vel,t}, \ldots, h^K_{acc,t}, v^K_{m,t})$
  $h^a_{interact,t} \leftarrow g_\theta^{interact}(h^a_{intra,t}, h^a_{inter,t})$
  # **Context Module:**
  $h^a_{context,t} \leftarrow z_\theta(h^a_{interact,t}, v^m_t)$
  # **Decoder:**
  $S^a_t \leftarrow q_\theta(h^a_{context,t}, h^a_{dec,t-1})$
end
return S_t

testing. For the CMU Panoptic dataset, we focused on the Haggling action, which consisted of more than two agents and had a defined training and testing protocol. Similar to prior works on motion prediction [79, 35, 29, 37, 36, 21], we used the skeleton modality and modeled all provided joints of each agent in 3D Cartesian space across all methods. In addition, we also used other available modalities of data, such as RGB data, for learning the scene context and obtaining complementary representation that can improve the motion prediction.

For the second scenario, where there are human-robot teams, we evaluated the performance by applying it to the KTH-HRC dataset [38]. The dataset comprises tasks such as hand-shake and hand-wave between a human and a robot. Here, we only predict the motion of the human team member, conditioned on the past motion of both the robot and the human, as the robot policy is fully observable, unlike the human policy. In line with the original paper of the dataset [38], we represent the human by four joints "RightShoulder," "RightArm," "RightForeArm," and "RightHand". Similar to the first scenario, we model the human joints in 3D Cartesian space, resulting in a 12-dimensional vector. The robot is represented by a 7-dimensional vector, each indicating a joint angle. We perform training and validation on 80% of each trial and test on the held-out 20%.

For all three datasets, the skeleton poses were provided in the original release of the datasets. We used the default sampling rate for each dataset. For the NTU RGB+D, each task ranged from 2 to 3 seconds, which translated to 60 to 90 frames per demonstration. For the CMU Panoptic dataset, each task duration ranged from 5 to 15 minutes. For the KTH-HRC dataset, the original sampling rate was 100 fps, which was down-sampled to match the 40 fps of the robot recordings. As the average duration of each task ranged from 10 seconds to 15 seconds, this meant each trial contained 400 to 600 frames.

4.5.2 State-of-the-art methods

Multi-human teams: For evaluating our model on multi-agent settings, we compared the performance of IMPRINT against several state-of-the-art approaches: Joint Learning [28]S, Joint Learning + Social [28], Joint Learning + Social + Context [28] and Scalable + Interpertable [21]. All the Joint Learning models are based on the sequence-to-sequence architecture. The Joint Learning method assumes no interaction among the agents and predicts each agent's motion independently. For the case of theJoint Learning + Social method, a permutation invariant pooling mechanism is applied to pool social features across all agents [28]. For Joint Learning + Social + Context method, an additional context module is added in the form of a spatio-

temporal context CNN, which extracts RGB features from the scene. For the Scalable + Interpretable method [21], the authors proposed an encoder-decoder approach with adversarial regularization on the latent space. The encoder learns a representation of the past motion data, which is then passed to an attention module to disentangle and extract relevant multi-agent features from agent-specific representations while addressing the limitations of pooling (e.g., max, average), which tend to summarize and thereby lose valuable information [82, 81]. To ensure a fair comparison, we fine-tuned hyper-parameters for all the approaches.

Human-Robot teams: We further evaluated our approach in a human-robot collaboration scenario. For this case, we compared the performance of our method against state-of-the-art approaches: Seq2Seq [29], Seq2Seq-SPL [36] and our Scalable + Interpretable [21]. Furthermore, we also compared against the zero-velocity baseline [29]. The Seq2Seq approach is based on the original neural machine translation approach [199]. The authors use a GRU at the encoder and implement weight sharing between the encoder and decoder. Furthermore, the authors introduced a residual connection at the decoder, which has been effective in modeling both the position and velocity of the skeleton joints and was the first reported architecture to improve upon the zero-velocity baseline. The Seq2Seq-SPL approach introduced a structural prediction layer (SPL) at the decoder, which allows conditional prediction of the skeleton joints along a hierarchy.

4.5.3 Evaluation Metric

We evaluated the performance of all models using the Mean Squared Error (MSE), which is the l_2 distance between the ground-truth and predicted poses at each timestep, averaged over the number of joints and sequence length, similar to prior work [86, 37, 36, 28]. The MSE is calculated as:

$$\mathcal{L}(\mathcal{X}, \hat{\mathcal{X}}) = \frac{1}{T \times K \times J} \sum_{t=1}^{T} \sum_{a=1}^{K} \sum_{j=1}^{J} (x_{t,j}^{a} - \hat{x}_{t,j}^{a})^2 \tag{4.13}$$

where, T and J are the total number of frame and joints, respectively. K represents the total number of agents. The MSE jointly encodes global body motion and skeletal movements [28], making it an ideal metric.

4.5.4 Implementation Details

Input and Output Modalities

For the NTU RGB+D 60 and CMU Panoptic datasets, we used skeleton and RGB modalities as input. The output for both datasets is only skeleton modality, which is the future pose of all the humans. The length of the input and output sequence of the skeleton modality is 15 timesteps. The length of the input sequence for the RGB modality is downsampled to 5 timesteps.

For the KTH-HRC dataset, we use the skeleton and robot joint angles as input. The output is only the skeleton modality, which is the future human pose of the human collaborator. The input and output sequence length of all modalities is 40 timesteps.

Learning Architecture

Unimodal Encoder for Non-skeleton modality: For the RGB modality in NTU RGB+D 60 and CMU Panoptic datasets, we employed modality-specific spatial encoders to extract unimodal features from raw RGB input. We utilized the ResNet-50 architecture [197], which is pre-trained on ImageNet to extract RGB features. The extracted ResNet features are then passed through a fully connected neural network to produce feature embeddings of sizes 128 and 512 for the CMU Panoptic and NTU RGB+D 60 datasets, respectively.

Motion Encoder: Each agent's motion is modeled using a motion encoder. As such, we varied the number of motion encoders depending on the number of agents. For the motion encoder in NTU RGB+D 60 and CMU Panoptic datasets, we used four Gated Recurrent Units (GRUs) to model the position, velocity, acceleration, and RGB features. The hidden state dimension of the GRUs was 256 for NTU RGB+D 60 and

Table 4.1: MSE (in cm^2) comparison of different multi-agent methods on the NTU RGB+D dataset (Lower is better).

Approaches	Frames					
	2	4	8	10	13	15
Joint Learning [28]	9.68	15.84	29.88	37.52	49.55	57.93
Joint Learning + Social [28]	9.71	15.97	30.36	38.69	51.68	59.38
Joint Learning + Social + Context [28]	9.78	16.02	30.46	38.39	50.91	59.63
Scalable + Interpretable [21]	9.66	15.66	29.05	36.16	47.20	54.84
IMPRINT	**9.64**	**15.61**	**28.62**	**35.49**	**46.19**	**53.63**

128 for the CMU Panoptic dataset. We used dropout regularization for all GRUs with a dropout probability of 0.1. As each human in the joint settings of NTU RGB+D and the CMU Panoptic datasets had specific roles per task, we assigned separate motion encoders for each humans. This meant for the NTU RGB+D, where each task comprised two humans, we had two motion encoders to model the individual roles the humans were performing. For the CMU Panoptic, this meant scaling to 3 encoders as the haggling task involved three humans.

For the motion encoder in the KTH-HRC dataset, we used three GRUs to model the position, velocity, and acceleration features of the human and one GRU to model the robot joint angles. The hidden state dimension of all the GRUs was 32, with each GRU having dropout regularization with a dropout probability of 0.1.

Interaction module: The interaction module comprised three distinct attention mechanisms to model the intra-agent, inter-agent, and interactional dynamics. Each attention sub-module comprised a 1-D CNN followed by a softmax layer. We used a stride of 1 and kernel size of 1 for the CNN. We implemented a residual connection between the output and the input for each attention mechanism, which allows for more stable learning.

Multimodal Context module: The multimodal context module is used for the NTU RGB+D 60 and the CMU Panoptic datasets. The module comprised one GRU cell and one attention sub-module. The input to the GRU was the spatial RGB features that are extracted from the ResNet-50 architecture. The hidden size of the GRU was 256 and 128 for the NTU RGB+D 60 and the CMU Panoptic datasets, respectively. We employed a similar attention mechanism to the Interaction module for fusing the representations from the different modalities. The attention mechanism was composed of one 1-D CNN followed by a softmax layer.

Motion Decoder: Similar to the encoder, the number of decoders varies depending on the number of agents, with each agent having one decoder. The decoder comprised one GRU unit followed by a linear layer. We implemented weight sharing between the encoder and decoder GRUs for all scenarios. The hidden state of the GRUs was 256 for NTU RGB+D 60, 128 for the CMU Panoptic dataset, and 32 for the KTH-HRC datasets. The output dimension of the linear layer was 75 for NTU RGB+D 60, 57 for CMU Panoptic, and 12 for KTH-HRC datasets.

Training details:

All the experiments reported in this paper were run on PyTorch v1.6. We trained the overall architecture of IMPRINT in an end-to-end-manner, using the Adam [97] optimizer. We used an initial learning rate of $1e-4$ for experiments on the NTU RGB+D 60 dataset, $5e-4$ for the CMU Panoptic, and $1e-3$ for experiments on the KTH-HRC datasets. For all experiments, we used weight decay on plateau with a decay factor of 0.1 and early stopping on the validation set.

Table 4.2: MSE (in cm^2) comparison of different multi-agent methods on the CMU Panoptic Dataset (Lower is better).

Approaches	Frames					
	2	4	8	10	13	15
Joint Learning [28]	1.33	2.29	4.15	5.09	6.55	7.56
Joint Learning + Social [28]	1.39	2.39	4.35	5.35	6.87	7.90
Joint Learning + Social + Context [28]	1.40	2.44	4.59	5.71	7.42	8.58
Scalable + Interpretable [21]	1.33	2.22	3.94	4.79	6.07	6.94
IMPRINT	**1.28**	**2.14**	**3.75**	**4.50**	**5.76**	**6.59**

4.6 Results

4.6.1 Multi-human motion Prediction

NTU RGB+D 60 Dataset

Results: We first present the results of all the evaluated models on the NTU RGB+D 60 dataset in Table 4.1, which comprises two humans performing 11 different mutual actions. We report the results at distinct frame intervals instead of seconds to circumvent the problem of frame drops during data collection and subsequent evaluation. The frame intervals are selected to evaluate the performance for all the models across short-term horizons: 2 & 4 frames, mid-term horizons: 8 & 10 frames, and finally, long-term horizons: 13 & 15 frames. The results in Table 4.1 suggest that IMPRINT outperformed all other methods across all the evaluated horizons. Furthermore, the performance gains accumulate over time, with IMPRINT significantly outperforming the closest benchmarked approach (by up to 1.20 cm^2) for long-term horizons.

Discussion: IMPRINT outperformed all the evaluated models, suggesting improved representation learning and sequence modeling. One possible reason behind the superior performance is IMPRINT's ability to model the interaction dynamics by explicitly modeling the intra and inter-agent representations. Prior works have aimed at modeling inter-agent dynamics using social pooling [28] or attention approaches [21]. While these works have improved the state-of-the-art, they do not apply conditioning among the agent's past trajectories and the trajectories of other agents in the vicinity. On the contrary, IMPRINT introduced the Interaction module, which allows IMPRINT to disentangle the team dynamics (inter-agent) from self-dynamics (intra-agent). The Interaction module uses the Keyless Attention mechanism [120], which allows IMPRINT to effectively model the interactions without losing relevant information [82] while being light-weight compared to the original self-attention mechanism [84].

To complement the Interaction module, IMPRINT also introduced the Context module, which allows for modeling of representation that is not available in the skeleton modality. Thus, the combination of the Interaction module and the Context module provided a holistic and robust representation at the encoder, which resulted in significant improvements over prior state-of-the-art across all the horizons, particularly over the long term.

CMU Panoptic Dataset

Results: We report the results of all the multi-agent models on the CMU Panoptic dataset in Table 4.2. The frame intervals are similar to that of the NTU RGB+D 60 Dataset in Table 4.1. However, here we are dealing with three humans and model a different hierarchy and composition of skeletal joints. The results in Table 4.2 suggest that IMPRINT achieved superior performance in comparison with the evaluated techniques across all the reported horizons. The performance improvement of IMPRINT becomes more apparent over the long-term, with an improvement of up to $0.35cm^2$ over the next-best approach.

Table 4.3: MSE (in cm^2) comparison of different multi-agent motion prediction methods on the KTH-HRC Dataset (Lower is better).

Approaches	Frames					
	5	10	20	30	35	40
Zero-Velocity [29]	0.11	0.34	1.18	2.38	3.07	3.81
Seq2Seq [29]	0.14	0.36	1.09	2.17	2.81	3.49
Seq2Seq-SPL [36]	0.17	0.42	1.20	2.33	2.98	3.66
Scalable + Interpretable [21]	0.06	0.20	0.72	1.61	2.21	2.91
IMPRINT (without context module)	**0.06**	**0.18**	**0.63**	**1.36**	**1.85**	**2.42**

Discussion: The results in Table 4.2 further validates IMPRINT's superior sequence modeling and representation learning. The introduction of the Interaction module provides a mechanism to condition the motion of any agent on their own motion and the motion of the other agents. While having representations from more than two agents poses a non-trivial problem, the Interaction module is nevertheless able to learn a representation for each agent that provides a more robust representation. This representation is further enhanced by the Context module, which gleans complementary representation from the other modalities, thus providing a holistic representation upon which the decoder can condition its prediction, thereby achieving lower prediction errors.

We also observed a performance trend with attention mechanisms: IMPRINT and Scalable + Interpretable, which attained better performances than the pooling mechanisms: Joint Learning, Joint Learning + Social, and Joint Learning + Social + Context. While both mechanisms provide a tool to summarize information, they have very distinct methodologies. Attention mechanism such as self-attention [84] and Keyless attention [120] aims to learn a weight over the different input representation. The self-attention mechanism, as used by Scalable + Interpretable [21] first performs a linear projection of the input vectors to query, key, and value embeddings. The embeddings are then used to compute attention weights using the scaled-dot product softmax approach. On the other hand, in IMPRINT, we did not use the linear projection and instead aimed to directly learn an expectation over the input vectors. We then multiplied the learned weights with the input vector, similar to the dot-product attention. Thus, in both approaches, there is a mechanism to weigh different input vectors. On the other hand, Joint Learning, Joint Learning + Social ,and Joint Learning + Social + Context rely on pooling strategies, which runs the risk of losing valuable information in the input vectors. This leads to a less robust representation at the encoder, which in turn leads to less accurate prediction at the decoder.

Interestingly, the MSE reported in Table 4.2 is lower than the ones in Table 4.1, despite modeling a higher number of agents. One possible explanation can be the type of task in the CMU Panoptic Dataset, which is *haggling*, whereas the NTU RGB+D 60 Dataset has 11 different mutual actions, ranging from *walking apart*, *walking toward*, *to hugging*. The actions available in the NTU RGB+D Dataset involve a greater range of motion, both with respect to the joints for activities such as *hugging* and *kicking* and with respect to the global position for activities such as *walking toward* and *walking apart*. This leads to higher stochasticity and variability of motion compared to the *haggling* activity, making the prediction more challenging and resulting in relatively higher MSE.

4.6.2 Human-Robot Collaboration Experiments

Results: We next report the results of the Human-Robot Collaboration experiments on the KTH-HRC dataset in Table 4.3. Similar to the previous experiments, we report the performances of all models at distinct frame intervals, which are different in this case as the frame rates are different in this dataset: 5 & 10 for short-term, 20 & 30 for mid-term, and 35 & 40 for long-term horizons. As the sampling frequency is 40 frames-per-second, we forecast up to 1 second of human motion. As can be observed in the table, IMPRINT attained the best performances over each evaluated horizon.

Table 4.4: The results of the ablation experiments on the NTU RGB+D Dataset (Lower is better).

Approaches	Frames					
	2	4	8	10	13	15
IMPRINT without Context Module	9.68	15.82	29.53	36.87	48.35	56.30
IMPRINT without Interaction Module	**9.55**	**15.46**	**28.57**	35.59	46.59	54.27
IMPRINT without Skip Connections	9.62	15.58	28.72	35.72	46.66	54.30
IMPRINT with Teacher Forcing	9.71	15.89	29.73	37.11	48.57	56.46
IMPRINT	9.64	15.61	28.62	**35.49**	**46.19**	**53.63**

Discussion: The results in Table 4.3 underline IMPRINT superiority over other evaluated approaches as it again outperformed all other evaluated approaches over all the horizons, achieving an improvement of up to 0.49 cm^2 over the long-term horizons. The results provide more evidence of IMPRINT's superior representation learning and subsequent sequence modeling capabilities. Unlike the previous scenario, we do not have data from other modalities, thus only relying on skeleton data. As such, we do not utilize the Context module in these experiments, which is one of the benefits of the modularity of IMPRINT.

While all approaches model human skeleton data, IMPRINT also conditions its prediction on robot joint angle data. The raw robot joint data is passed through a motion encoder, similar to the one that is used for human motion, and the resulting representation is passed to the Interaction module. As observed in previous multi-human scenarios, human motion is highly dependent on the presence of other agents. Thus, in the human-robot team setting here, where the human and robot are collaborating in close-proximity, there is a strong likelihood that the robot would influence the human motion. This is corroborated with recent work showing that robots can also influence human actions [200]. To this end, IMPRINT's Interaction module is particularly crucial as it allows IMPRINT to model the intra-agent and the inter-agent dynamics. The introduction of the Interaction module provided a more robust representation for IMPRINT, which attained the best results across all the temporal horizons. One possible reasoning behind the superior performance is that the module provides IMPRINT a mechanism to explicitly condition the impact of the robot on the future pose of the human.

On the other hand, the other evaluated approaches did not model the interaction between the human and robot, instead only considering the motion of the human. Among the evaluated methods, we see that Scalable + Interpretable [21] outperformed other approaches. This can be due to the combination of the multi-stream encoder, which models velocity and acceleration in addition to the skeletal joint position, and the use of the self-attention mechanism to disentangle these different streams. We also observed that all of the evaluated approaches outperformed the zero-velocity baseline for the long-term horizon (35 & 40 frames). For the short-term (5 & 10 frames), only IMPRINTand Scalable + Interpretable can outperform the baseline.

4.6.3 Ablation Experiments

Results: We performed extensive ablation experiments on the NTU RGB+D 60 dataset. For the ablation experiments, we performed two architectural ablations: first, at a modular level, where we removed the Multimodal Context module and the Interaction Module. Next, we explored some of the additions at the architectural level, particularly the use of skip connections. Here, we removed all the skip connections of IMPRINT. Lastly, we investigated the teacher forcing (TF) technique [96], which is a common technique used in sequence learning to prevent error propagation, where the decoder has access to ground-truth data and can use that as an input during training time.

The results from the ablation experiments on the NTU RGB+D 60 datasets are reported in Table 4. The results suggest that IMPRINT outperforms IMPRINT without Context Module, IMPRINT without Interaction Module, IMPRINT without Skip Connections, and IMPRINT with Teacher Forcing. We observed a significant drop in performance for IMPRINT without Context Module and IMPRINT with Teacher Forcing. Interestingly, IMPRINT without Skip Connections led to improved performance over short-term horizons

but poorer performance over mid to long-term horizons. We saw the same trend for IMPRINT without Interaction module, where we observed that for short horizons, ablating the Interaction Module led to improved performance over short-term horizons. However, the performance dropped for mid to long-term horizons.

Discussion: The results in Table 4.4 justify the design choices of IMPRINT. This is further evidence of IMPRINT's efficacy in predicting human motion over prior unimodal approaches [21, 29, 36], and multimodal approaches [28]. We first notice a drop in performance for all evaluated horizons when we ablate the Multimodal Context Module. This implies the benefit of having the module which can extract and fuse data from different modalities in a complementary manner. In addition, we observed a drop in performance over the mid and long-term when we ablate the skip connection. This suggests the efficacy of the skip connection, which provides a shortcut for the gradient to propagate to the different encoders from the attention mechanisms, which in turn allows the encoders to learn better spatial-temporal representation.

Next, we ablate the Interaction Module to investigate its impact on the prediction. We observe that IMPRINT without Interaction Module performs comparatively over short horizons. This suggests that for shorter horizons, the motion encoder representation and the multimodal context representation provide relevant information for predicting multi-agent motion. However, for longer horizons, having the inter-agent and intra-agent dynamics from the Interaction module allows the IMPRINT to better predict the motion of all the agents, as observed by the improvement in performance for IMPRINT with the interaction module compared to IMPRINT without the interaction module. We observed a similar trend when we ablated the Skip Connections. This is because in IMPRINT's Interaction module, we model intra-agent and inter-agent dynamics. For intra-agent dynamics, the goal is to identify which motion features provide the most information to predict the agent's future pose and obtain a salient representation by weighing them accordingly. For inter-agent dynamics, the goal is to model how each agent's motion influences the motion of other agents in the team, which aligns with findings in joint action. Finally, having the intra-agent and inter-agent dynamics provides IMPRINT the flexibility to model and adaptively weigh these representations to obtain the interactional dynamics for each agent.

Next, we introduce teacher forcing as a way to improve the sequence prediction for IMPRINT. Interestingly, we see performance degradation across the board. We used the same teacher forcing method similar to the prior works [21], which attained improvements over a non-teacher forcing setup. One key difference between IMPRINT and the prior work is that the latter used a discriminator for distribution matching, which acted in an adversarial manner to the reconstruction loss. In such a situation, having TF may improve performance. However, in IMPRINT, there is no adversarial loss, allowing IMPRINT to condition its output on the multimodal encoder representation and the decoder's past prediction. As such, when we added TF, this led to IMPRINT being reliant on the ground-truth data during training, which ultimately led to poor generalization during test time. Hence, we did not use TF to train IMPRINT, but nevertheless, our results suggest superior performance over state-of-the-art methods, some of which used TF.

4.6.4 Significance Analysis

Deep representation learning models contain various parameters, and often the performance of these models depends on the initialization of these parameters. Thus, we intend to compare the performance of these models across multiple runs. For this purpose, we conducted a significance analysis by following the procedure proposed by Dror, Shlomov, and Reichart [4]. In this experimentation, we trained and tested three models (J+S+C: Joint Learning + Social + Context, S+I: Scalable + Interpretable, and IMPRINT) on NTU RGD+D dataset. We evaluated each model five times by initializing the models' parameters with a different random set of values. Finally, we conducted the significance analysis at level $\alpha = 0.05$. Additionally, we calculated the average and standard deviation of MSE from these five runs. We presented the experimental results in Table 4.5.

Results and Discussion: The experimental results in Table 4.5 suggest that IMPRINT attains the lowest average MSE. Moreover, IMPRINT attains the lowest standard deviation across multiple runs across all the prediction frame intervals except the frame intervals of 8. Most importantly, the significance analysis following the procedure proposed by Dror, Shlomov, and Reichart [4] suggest that IMPRINT significantly outperformed all the evaluated models across all the prediction frame intervals. The primary difference

Table 4.5: Significance analysis of different motion prediction models on the NTU RGB+D Dataset (Lower is better). J+S+C: Joint Learning + Social + Context, S+I: Scalable + Interpretable, IMPRINT. § We conducted significance analysis at level $\alpha = 0.05$ (Following the procedure proposed by Dror, Shlomov, and Reichart [4]).

Approaches	Frames						Significant over
	2	4	8	10	13	15	
J+S+C	9.80 ±0.02	16.08 ±0.07	30.67 ±0.20	38.69 ±0.28	51.36 ±0.37	60.21 ±0.43	-
S+I	9.69 ±0.03	15.69 ±0.02	29.10 ±0.02	36.22 ±0.10	47.31 ±0.13	54.99 ±0.17	J+S+C
IMPRINT	**9.64** ±0.01	**15.61** ±0.02	**28.61** ±0.09	**35.52** ±0.09	**46.28** ±0.12	**53.77** ±0.16	**J+S+C & S+I**

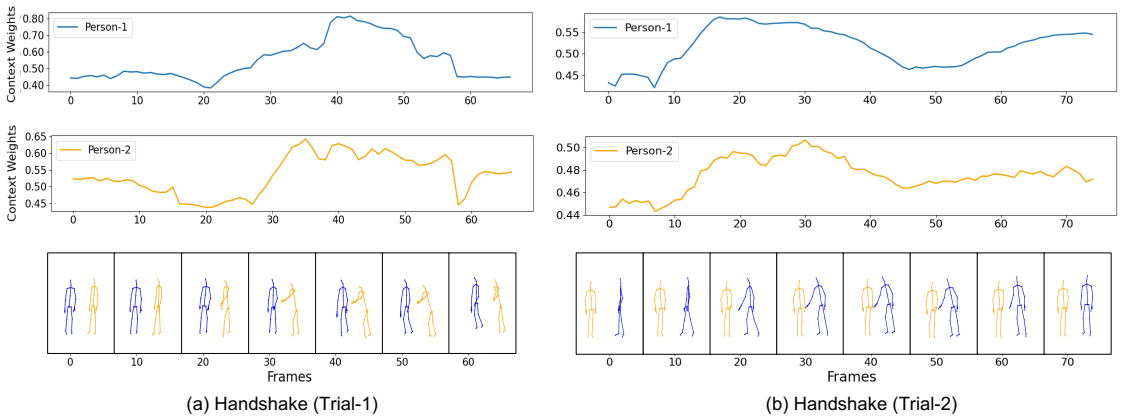


Figure 4.3: Change in the Multimodal Context Weights over time for each agent for the task of *handshake* (The skeleton for Person-1 is black and for Person-2 is orange).

between IMPRINT and other evaluated models is that IMPRINT uses inter-agent and intra-agent dynamics with multimodal context, whereas the other models use unimodal context without multi-agent dynamics to predict human motion. This significant performance improvement of IMPRINT further validates the importance of our proposed multimodal representation learning to improve motion prediction in teams.

4.6.5 Attention Weights Interpretation

Results: We visualized the learned Multimodal Context weights for each agent for the task of *handshake*. This analysis provides a tool for interpreting how our proposed framework, IMPRINT, weighs the different modules across time for a given task. For each temporal window of observations, IMPRINT extracts the motion features and context features from other modalities and then weighs the representations according to the current phase of a given task. We log the context weights for each input, given the current frame and task, and visualize the change in values over time.

Figures 4.3(a) and 4.3(b) illustrate the change in Multimodal Context weights over time for two different trials of the *Handshake* task in the NTU RGB+D 60 dataset. A higher value would suggest that IMPRINT provided more weight on the context representation obtained from other modalities of data, in this case, RGB. On the other hand, a lower value would indicate that IMPRINT relied more on the representation obtained from skeleton features. The weights are normalized to fall between 0 and 1. The two trials illustrated here are randomly sampled from all the demonstrations of the handshake task, where the phenomenon manifested consistently.

Discussion: The qualitative results from Figures 4.3(a) and 4.3(b) provide valuable insights on how IMPRINT uses context representations. From the figures, one can observe Agent-1 and Agent-2 having a similar change in their Context Weights over time for both the trials. In Figure 4.3(a), we see that the agents initiate and execute the *handshake* between the 30 to 40 frames horizons. Interestingly, this coincides with when there is an increase in the context weight for both the agents. We see a similar pattern in Figure 4.3(b),

where the agents perform the handshake between 10 to 30 frames, which coincides with an increase in the context weight during this time.

The findings suggest that IMPRINT places higher weight on context information when the agents are about to execute a highly coordinated joint activity. This phenomenon is also supported by prior works on multimodal representation learning [101], where information related to complex interactions cannot be solely captured by skeletal data and often requires information from other modalities such as RGB. On the other hand, IMPRINT can adjust the weights when there isn't a high degree of coordination, such as before and after the handshake. During those periods of the task, there is a lower weight on the context information.

4.7 Limitations

In this work, we have proposed a novel sequence-learning approach for predicting human motion in multi-agent scenarios. Our proposed approach, VADER, outperformed state-of-the-art approaches for human-human and human-robot teams. With VADER, we have introduced a novel Interaction Module, which learns to adaptively weight the intra-agent and inter-agent representations before fusing them to learn the interactional dynamics of all the agents in the team. The Interaction module is generalizable to human-human and human-robot teams, as justified by our experiments. Furthermore, to complement the skeleton modality, we have proposed the Multimodal Context module, which helps to obtain a robust representation by fusing the features from the skeleton and non-skeleton modalities.

While IMPRINT incorporates multimodal data (e.g., RGB, depth sensors, skeletal data) to enhance motion prediction, the fusion of these diverse data types can be challenging. The process of aligning and integrating different modalities effectively is complex and computationally intensive. Additionally, the model's performance may degrade if there are inconsistencies or noise in any of the modalities, which can lead to suboptimal fusion and less accurate predictions.

IMPRINT utilizes Gated Recurrent Units (GRUs) and Recurrent Neural Networks (RNNs) for sequence learning, which are known for their effectiveness in handling short to medium-term dependencies. However, GRUs and RNNs can struggle with long-horizon predictions due to their inherent limitations in maintaining and processing information over extended time sequences. This can result in diminished accuracy when the model is required to predict motion far into the future, particularly in complex scenarios where long-term dependencies play a crucial role in the interactions between multiple agents. Exploring alternative architectures, such as transformers, could potentially address this limitation by providing greater expressiveness and better handling of long-term dependencies.

Chapter 5

PoseTron: Enabling Close-Proximity Human-Robot Collaboration Through Multi-human Motion Prediction

As robots enter human workspaces, there is a crucial need for robots to understand and predict human motion to achieve safe and fluent human-robot collaboration (HRC). However, accurate prediction is challenging due to a lack of large-scale datasets for close-proximity HRC and the absence of generalizable algorithms. To overcome these challenges, we present INTERACT, a comprehensive multimodal dataset covering 3-D Skeleton, RGB+D, gaze, and robot joint data for human-human and human-robot collaboration. Additionally, we introduce PoseTron, a novel transformer-based architecture to address the gap in learning algorithms. PoseTron introduces a conditional attention mechanism in the encoder enabling efficient weighing of motion information from all agents to incorporate team dynamics. The decoder features a novel multimodal attention mechanism, which weights representations from different modalities and the encoder outputs to predict future motion. We extensively evaluated PoseTron by comparing its performance on the INTERACT dataset against state-of-the-art algorithms. The results suggest that PoseTron outperformed all other methods across all the scenarios, attaining lowest prediction errors. Furthermore, we conducted a comprehensive ablation study, emphasizing the importance of design choices, pointing towards a promising direction for integrating motion prediction with robot perception in safe and effective HRC.

5.1 Introduction

Collaborative robots (cobots) capable of safely operating in close-proximity to humans have the potential to significantly enhance efficiency and productivity across various industries, ranging from manufacturing to fulfillment [106]. At the core of achieving effective and fluent close-proximity human-robot collaboration (HRC) lies the crucial ability of robots to perceive and anticipate human intentions [12, 13, 14, 15, 16, 17, 18, 19]. This imperative need for anticipation and adaptability mirrors the fundamental aspects of human interactions. Tasks such as navigating through crowded environments or exchanging objects heavily rely on our innate capacity to observe and anticipate the actions of others [5, 6, 7, 8]. This anticipatory capability would empower robots to proactively adjust their actions, avoid collisions, and provide valuable assistance to humans in dynamic and often unpredictable environments, similar to how humans employ anticipatory and feedback mechanisms to develop suitable motor behaviors [201, 202].

The concept of anticipation has received extensive attention particularly in the social navigation and collaborative manipulation domains. The primary goal in the former is to navigate safely in the presence of humans, thus avoiding any potential interference [44, 106, 127, 203, 204]. On the other hand, anticipating the next human activity would enable robots to contribute to the task proactively, improving efficiency and

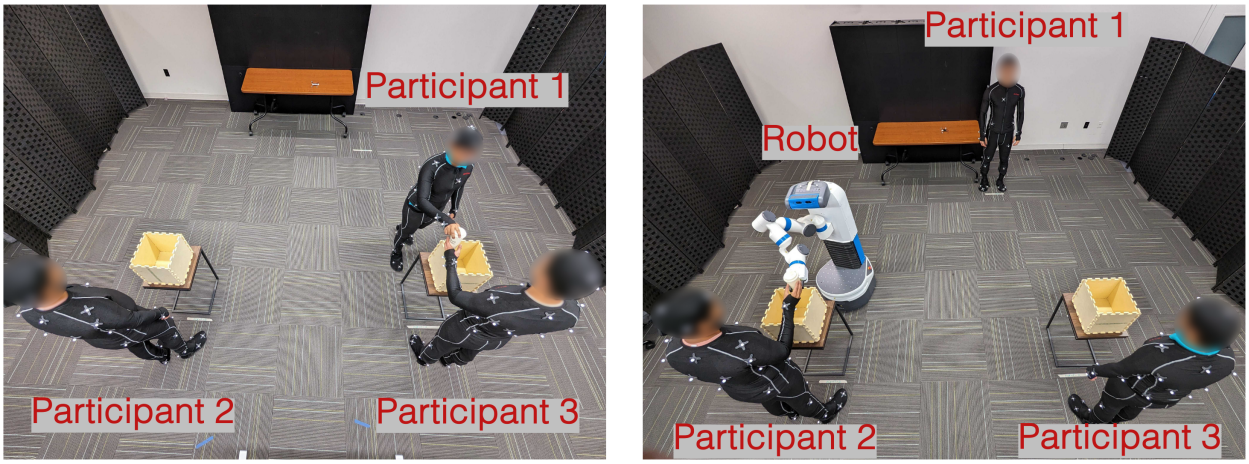


Figure 5.1: Samples of Close-proximity Human-Human and Human-Robot Collaboration from the INTERACT Dataset.

taking preemptive action to enhance safety [205, 206, 108, 72, 7]. However, with the introduction of cobots, which are expected to engage with humans over extended periods in close-proximity settings, there is a need to anticipate human motion at a higher spatial and temporal granularity [117, 20, 74]. Anticipation in this scenario would entail predicting future human motion conditioned on past motion, representing a shift from the 2-D global position predicted in social navigation or categorical human activities in collaborative manipulation to 3-D skeletal joint positions.

While the concept of anticipation is crucial for cobots, the current state-of-the-art in motion prediction needs to be revised when addressing the specific challenges in close-proximity HRC. These challenges are multifaceted, with one of the primary bottlenecks being the scarcity of comprehensive real-world datasets that feature scenarios involving robots collaborating with one or more humans [37, 8, 66]. The availability of such datasets is essential as it is a fundamental requirement for developing and validating algorithms capable of accurately predicting human motion and intention. Furthermore, there is a notable gap in learning algorithms capable of reliably predicting the motion of multiple humans. Existing research on human motion prediction predominantly focuses on dyadic scenarios – involving one human and one robot [66, 38], or, in some cases, excludes robots altogether [67]. This limitation constrains the robot's anticipation capabilities to dyadic collaboration, which may not always reflect the nature of real-world scenarios which may haave multiple humans and/or robots.

To address the aforementioned challenges, we introduce INTERACT, a comprehensive human-human and human-robot collaboration dataset. INTERACT stands out by featuring a large-scale collection of multimodal data encompassing both Human-Human Collaboration (HHC) and Human-Robot Collaboration (HRC). Each collaborative task contains at least three participants, presenting a shift from dyadic to team interaction. INTERACT comprises assembly tasks involving three participants in HHC scenarios and four participants in HRC scenarios, with one of the participants being a robot (see Fig. 5.1). The dataset comprises 3-D human Skeleton joint positions of 3 participants, RGB and depth data of the workspace and the interaction from two viewpoints, ego-view data from the two human participants, eye-tracking and gaze data from two of the humans, and robot joint data, all synchronized to provide an exhaustive picture of the collaboration task. As part of the data collection effort, we recruited 63 participants, which amounted to 21 groups of three human participants for HHC and three humans and a robot participant for HRC scenarios, amounting to approximately 1 M samples of synchronized multimodal data.

INTERACT has three novel characteristics that distinguish it from prior datasets such as Mogaze [67], Thor [16], and FACT-HRC[66] (see Tab. 5.1). First, INTERACT represents close-proximity collaboration

Table 5.1: Summary of Publicly Available Multimodal Human-Robot Interaction Datasets.

Datasets	Setting	# Agents	Sensor Modalities								Duration of Recordings
			# 3-D Skeletons	RGB	Multi-view	Depth	Multi-view	Ego Data	No. of Person	Robot Joint Positions	(Approximate)
HARMONIC[207]	HRI (Shared Autonomy)	1	1	✔	✗	✔	✗	✔	1	✔	5 hours
MHHRI [208]	HHI and HRI	2	1	✔	✔	✔	✔	✗	N/A	✗	7 hours
MoGaze [67]	HI	1	1	✗	✗	✗	✗	✔	1	✗	3 hours
UE-HRI [209]	HRI	1	1	✔	✔	✔	✗	✗	✗	✗	12 hours
FACT HRC [66]	HRC	1	1	✔	✔	✔	✔	✔	1	✔	20 hours
INTERACT (Ours)	HHC and HRC	3 (HHC) & 4 (HRC)	3	✔	✔	✔	✔	✔	2	✔	9 hours

in groups of three humans + one robot for the HRC scenarios, which provides a novel and highly interactive scenario for data collection, distinguishing it from other datasets [66, 67] which were limited to dyadic interaction. Second, we introduce a simple but effective setup for data collection where the robot was completely autonomous in close-proximity settings with humans, which is a shift from other datasets that were primarily tele-operated or featured a form of shared autonomy. Finally, to understand the present state of HRC, we collected data on the same task comprising only human participants. This allows us to investigate how HRC compared to HHC and provides a mechanism to systematically analyze how humans behave differently in the presence of a robot collaborator. In addition, it allows the research community to evaluate the generalizability of their algorithms by training them on data from HHC scenarios and testing in HRC scenarios.

To address the gap in learning algorithms that can accurately predict the motion of multiple agents, we propose a novel and efficient transformer architecture [84], PoseTron (*pronounced "pos·i·tron"*). PoseTron employs an encoder-decoder framework where the encoder is tasked to extract spatio-temporal representation in human motion, fuse representations from diverse modalities, and learn the interaction dynamics among all agents. Additionally, we introduce specialized attention modules to capture agent-specific motion patterns. We employ self-attention mechanisms for extracting spatio-temporal features within each agent's motion and conditional attention mechanisms that enable agents to incorporate team dynamics by querying another agent's representations. The encoder output comprises encoded representation from non-skeletal modalities and skeleton representation that encapsulates the complexity of individual human motion and team dynamics.

The output of the encoder, along with the last observed motion, is passed to the decoder. The decoder employs an auto-regressive mechanism for future motion prediction. Additionally, the decoder utilizes a conditional attention mechanism to incorporate salient representation from its generated output and the encoder's representations.

We conducted extensive experiments to assess the efficacy of PoseTron by deploying on the INTERACT dataset. Our experiments included evaluating the performance of PoseTron on i) HHC-train, HHC-test, ii) HRC-train, HRC-test, and iii) HHC-train, HRC-test setups. Our results suggest PoseTron consistently outperformed the state-of-the-art approaches over all three evaluation scenarios. Furthermore, we conducted a comprehensive ablation analysis of PoseTron's learning modules and the relevance of multimodal data in the INTERACT dataset. The results validate PoseTron's architectural choices and underscore its ability to leverage complementary information from diverse data sources. The outcomes of these experiments promise to narrow the gap in close-proximity HRC by providing a substantial dataset and valuable insights for advancing state-of-the-art anticipation techniques.

5.2 Related Work

5.2.1 Multimodal Datasets in HRI

Multimodal datasets have piqued the interest of diverse communities, spanning human–robot interaction [66, 16, 207], computer vision [210, 211], action recognition [55, 152, 212], and natural language processing [213, 214]. In the context of Human-Robot Interaction (HRI), the capture and analysis of data from various modalities are crucial, empowering robots to comprehend, anticipate, and coexist with humans in diverse environments. In alignment with this, recent datasets have emerged, providing multimodal data for shared-autonomy [207], social navigation [16], and dyadic Human-Robot Collaboration (HRC) [66].

Newman et al. [207] introduced the HARMONIC dataset, capturing multimodal data, including RGB, Gaze, and Robot information, during human-robot collaboration tasks. However, the dataset is confined to shared autonomy in a table-manipulation scenario. Celiktutan et al. [208] proposed the MHHRI dataset, encompassing dyadic (Human-Human) and triadic (Human-Human-Robot) interactions but is limited to human-side interactions, lacking human perspective (ego-centric) data crucial for understanding human intent [207]. Tian et al. [66] presented the FACT-HRC dataset, focusing on human-robot handover interactions in collaborative environments but restricted to dyadic (human-robot) scenarios.

While existing datasets have made strides in addressing various aspects of HRI, a noticeable gap persists in the availability of datasets tailored to close-proximity HRC scenarios involving multiple humans. Furthermore, many of these datasets rely on teleoperation within Wizard-of-Oz setups, which does not accurately represent how humans will naturally interact with an autonomous robotic agent. The challenges of recruiting human participants, physically co-locating robots and humans, and the imperative to uphold human privacy rights further compound the limitations of such datasets. As a result, most datasets feature a limited number of participants, which often fails to capture the complexity and diversity of behaviors encountered in real-world HRI settings.

5.2.2 Human Motion Prediction

Human motion prediction is widely considered one of the essential parts of robotic intelligence that would enhance robot perception and allow for rapid and high fidelity reactions towards complex environment changes [26, 27, 15, 29]. The notion of prediction has found application in diverse areas within HRI, spanning shared autonomy [207, 195], social navigation [126, 127], and autonomous vehicles [128, 48, 215]. Ngiam et al. [215] proposed a model for predicting joint trajectories of multiple agents, using a masking strategy and attention mechanisms. Tang et al. [26] illustrated the relevance of motion prediction in planning, computing conditional probability density for the trajectories of other agents based on a hypothetical rollout of the self-agent. Yasar et al. [1] proposed a multi-agent adversarial auto-encoder approach for predicting future human motion, with the authors using a self-attention mechanism to weigh the different agent representations before predicting future motion. Adeli et al. [28] proposed a social pooling mechanism on top of the seq2seq architecture for predicting multi-agent motion prediction.

Accurate human motion prediction is pivotal for ensuring the safety of HRI, particularly in close-proximity HRC. Despite recent improvements, there's a notable gap in their application to HRC scenarios [16]. The challenge lies in their training on datasets predominantly featuring single-agent human motion [67], often lacking the context of robots within the workspace. This limitation hinders their applicability to collaborative environments.

5.3 INTERACT: HHC and HRC Dataset

In this section, we present our first solution for enabling close-proximity HRC: the INTERACT dataset[1]. Our proposed dataset stands out from other datasets in close-proximity HHC and HRC scenarios by providing a large-scale collection of synchronized multimodal data, as illustrated in Fig 5.1 and summarized in Tab. 5.2. The dataset includes 3-D Skeletal joint data of human participants, RGB and depth data from two viewpoints in the workspace, ego-view, eye-tracking, and gaze positions data from the two human participants, and robot joint data. The comprehensive dataset can be leveraged for various tasks, including motion prediction, goal prediction, and imitation learning.

5.3.1 Study Apparatus and Implementation

The objective of collecting each of the modalities in INTERACT is to provide the robot with a comprehensive understanding of the interaction and its surrounding environment. For data collection in INTERACT, we utilized the OptiTrack Motion Capture system [216] for collecting 3-D Skeleton data, 2 ZED Cameras from StereoLabs [217] to capture RGB and depth from two different viewpoints of the workspace, and 2 eye-tracking devices by Pupil Labs [218] for collecting ego point of view data, eye-tracking and gaze data. For

[1]Dataset is available at https://bit.ly/posetron-interact.

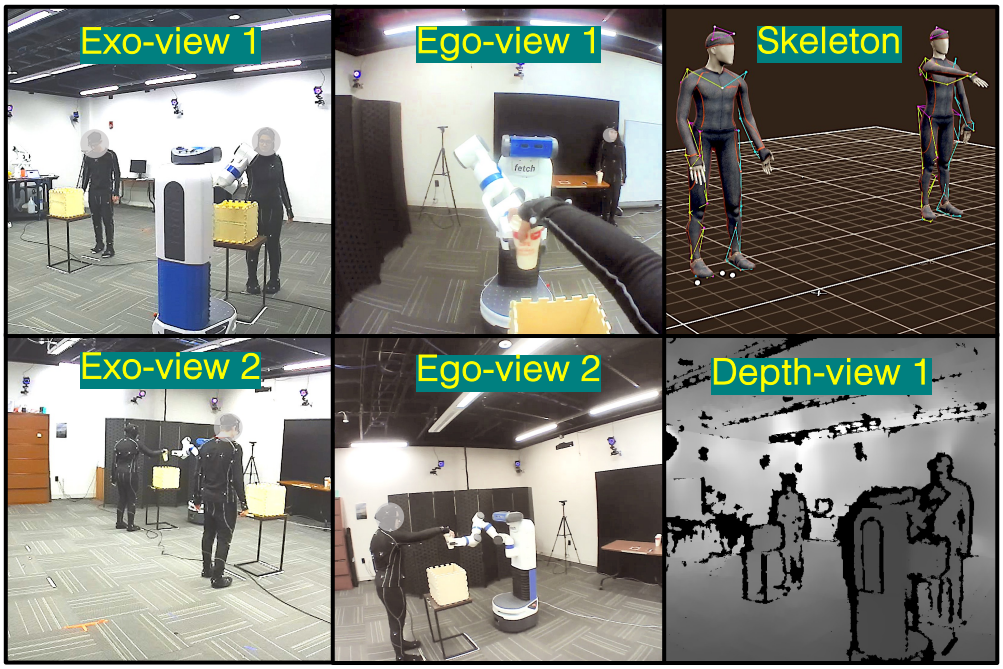


Figure 5.2: Human-Robot Collaboration samples from the INTERACT dataset. The dataset comprises 3-D skeletons from three participants, and RGB+D Camera views from two perspectives and Ego POV from two Participants.

HRC, we introduce a Fetch Robot [219], which is a mobile manipulator in the shared workspace to collaborate with the other participants.

Prior to data collection, all equipment were meticulously calibrated to ensure accurate data synchronization. For collecting 3-D Skeleton data, participants were equipped with a full-body motion tracking suit by Opti-Track, comprising 41 passive markers. The 3-D Skeleton poses represent a lightweight and accurate source of information for predicting human intent. In addition to 3-D Skeleton data, we incorporated RGB and depth data obtained from two cameras in opposing corners of the workspace, as illustrated in Fig. 5.2. This setup allowed us to maximize coverage and provide diverse perspectives on the collaboration. To further enhance the prediction of human intent and motion, we equipped two participants with eye-tracking devices, enabling the collection of valuable first-person viewpoints in addition to the third-person perspectives captured by other sensors, with prior work showing the benefit of eye-tracking and gaze information for predicting human intent [67, 66, 207]. Finally, in HRC scenarios, we deployed the Fetch robot and collected robot joint data.

5.3.2 Human Ethics

Our study protocols were reviewed and approved by the Institutional Review Board. All participants provided informed consent for participating in the study and having their data recorded as part of a public dataset for research purposes. At the end of the study, participants were compensated with a $30 gift card for approximately 2 hours of their time.

5.3.3 Participants

A total of 63 adults participated in the study (31.7% female ($n = 20$), 66.67% male ($n = 42$) and 1.58% non-binary ($n = 1$)). The mean age of the participants was 23.65 years ($SD = 3.91$). The participants were predominantly right-handed 84.1% ($n = 53$) and 15.9% left-handed ($n = 10$). Participants also recorded their experience with robots on a Likert scale from "no experience" (1) to "expert-level experience" (5), with the mean experience level at 2.21 ($SD = 1.19$).

Table 5.2: Summary Statistics of the INTERACT Dataset.

Scenario	Number of Participants	Variation	Average Duration (sec)	Total Timestamps	Total Multimodal Frames (Million)
HHC	3 H	w/o obstacle	104.5	201 K	1.20 M
		w obstacle	127.0	263 K	1.58 M
HRC	3 H + 1 R	w/o obstacle	140.9	300 K	1.80 M
		w obstacle	149.8	306 K	1.84 M
Total	-	-	-	1.07 M	6.42 M

5.3.4 Data Collection Procedure

All the tasks involved three human participants in both HHC and HRC scenarios. This arrangement led to 21 groups from the 63 recruited participants. Each group engaged in 12 collaborative assembly task sessions, with an equal distribution of 6 HHC and 6 HRC sessions. Within each scenario, two variations were introduced – one with obstacles in the workspace and one without.

Pre-Task Survey: Before beginning the study, participants were asked to review consent documents and task instructions. They then filled out a pre-task survey, which collected demographic information and their prior robot experience. Next, all three participants were equipped with Motion Capture suits to collect 3-D Skeleton Pose data. Two participants (Participants 2 and 3) also wore the eye-tracker, which would collect their ego point-of-view, eye-tracking, and gaze data during the task.

The scenarios were counterbalanced, and each group was assigned either HHC or HRC as their initial scenario. After completing all the sessions for their initial scenario, they switched to the other. They participated in six sessions within each scenario, three for each variation. In Variation-1, there were no obstacles in the workspace, whereas in Variation-2, there were obstacles that participants would have to move around. To minimize the learning effect, participants rotated roles after each session. For instance, if a person started as Participant 1 in the first session, they became Participant 2 in the second and Participant 3 in the third session. This rotation applied to all group members in both HHC and HRC scenarios, ensuring balanced role distribution across variations.

Human-Human Collaboration (HHC) Scenario: In this scenario, three human participants collaborated on an assembly task. In each session, Participant 1 transported cups to Workspace 2 and Workspace 3 three times each, while Participants 2 and 3 followed this workflow:

- Received a cup from Participant 1.
- Moved to Workspace 1.
- Extracted Lego pieces and instructions from the cup.
- Assembled Lego pieces as per instructions.
- Repeated steps 1-4 for three times.

The session was considered complete when Participants 2 and 3 assembled the Lego structure as specified in the instructions. We used different Lego structures for different variations to minimize the learning effect. After three sessions, obstacles were introduced to the workspace, or the scenario changed.

Human-Robot Collaboration (HRC) Scenario: In this scenario, a Fetch robot (as Participant 4) joined three human participants to complete a similar assembly task. The robot received cups from Participant 1 and transported them between Workspaces 2 and 3. Participants 2 and 3 followed a similar workflow to the HHC scenario.

Post-Session Survey: After each session, participants filled out a post-session survey containing questions that covered various aspects: participant-specific questions (e.g., "I needed to observe and anticipate the activities of group member-1/2/3"), group-specific questions (e.g., "Which group member had the greatest

impact on the coordination of the group?") and robot-specific assessments, rated on a Likert scale (e.g., "The robot was effective in coordinating the actions with both the group members").

5.4 Multi-agent Motion Prediction

Our objective is to improve the robot's perception by providing it with the capability to forecast the motion of all human collaborators in the team. Human motion prediction is formally described as the task of estimating the future human pose for a certain period, given their past pose. We will present the problem for single-agent motion prediction and later extend the formulation to multiple humans. We assume access to 3-D skeletal joint positions as the primary data source, along with additional modalities (e.g., RGB). Our notation consistently utilizes superscripts to indicate agents and subscripts to represent time across all formulations.

We begin by considering the scenario of an individual agent, denoted as agent i. The objective here is to predict the future trajectory of this agent's pose, given the observed pose trajectory spanning from time $t = 1$ to τ, represented as $\mathbf{X}^i = \{x_1^i, \ldots, x_\tau^i\}$, and any additional sensor data from other sources, referred to as $D = \{d_1, \ldots, d_\tau\}$. In this context, each pose frame $x_t^i \in \mathbb{R}^N$ represents the skeletal pose in an N-dimensional space. The dimensionality, N, is determined by the number of joints, indicated as J, in the skeleton and the dimension of each joint, with $N = 3 \times J$. The input frame from other sensors, $d_t \in \mathbb{R}^N$, comprises raw data from complementary modalities such as RGB and Gaze data.

The model's objective is to generate future trajectory frames within a time horizon H, denoted as $\mathbf{Y}^i = \{y_{\tau+1}^i, \ldots, y_{\tau+H}^i\}$. Our primary goal is to acquire the underlying representation that enables the model to accurately predict plausible future human poses, which are denoted as $\hat{\mathbf{Y}}^i = \{\hat{y}_{\tau+1}^i, \ldots, \hat{y}_{\tau+H}^i\}$. We work under the assumption that predicting future human poses relies on past observed and generated poses, and we predict each frame in an autoregressive manner, as described below:

$$p_\theta(\hat{\mathbf{Y}}^i) = \prod_{\delta=\tau+1}^{\tau+H} p_\theta(\hat{y}_\delta^i | \hat{y}_{\tau:\delta-1}^i, x_{1:\tau}^i, d_{1:T}) \tag{5.1}$$

In the context of multi-agent motion prediction, the input consists of the observed poses of all agents in the scene from time $t = 1$ to τ: $\mathbf{X} = \{X^1, \ldots, X^K\} = \{x_1^{1:K}, x_2^{1:K}, \ldots, x_\tau^{1:K}\}$ and additional multimodal input: $\mathbf{D} = \{d_1, \ldots, d_\tau\}$. The expected output of the model is the future trajectory frames over a horizon H, which represents the ground truth poses over the horizon $t = \tau + 1$ to $\tau + H$: $\mathbf{Y} = \{Y^1, \ldots, Y^K\} == \{y_{\tau+1}^{1:K}, y_{\tau+2}^{1:K}, \ldots, y_{\tau+H}^{1:K}\}$. Thus, the multi-agent motion prediction problem can be formulated as follows:

$$p_\theta(\hat{\mathbf{Y}}^i) = \prod_{\delta=\tau+1}^{\tau+H} p_\theta(\hat{y}_\delta^i | \hat{y}_{\tau:\delta-1}^i, x_{1:\tau}^{1:K}, d_{1:\tau}); \quad \forall i = 1, \ldots, K \tag{5.2}$$

5.5 PoseTron

We now introduce our proposed framework for multi-agent human motion prediction: PoseTron. PoseTron (see Fig. 5.3) is a multimodal sequence learning architecture that aims to accurately predict the future poses of all humans, irrespective of their number or their collaborative scenario (HHC/HRC). PoseTron comprises two specialized modules: the encoder (Sect. 5.5.1), which aims to encode the motion of all the agents and modalities of all the data streams, and the decoder (Sect. 5.5.2) which uses the encoded representation to forecast future human pose.

5.5.1 Multimodal Pose Encoder

The input to the Encoder is observed motion of all the agents, comprising agent-specific skeletal input $\mathbf{X} = \{X^1, \ldots, X^K\}$ and non-skeletal input $\mathbf{D} = \{d_1, \ldots, d_\tau\}$, as depicted in Fig. 5.3. For the skeleton sequence spanning T timesteps, we extend the 3-D joint position with time derivatives: velocity and acceleration. Thus, the original input, comprised of T tokens, is extended to $3T$ tokens.

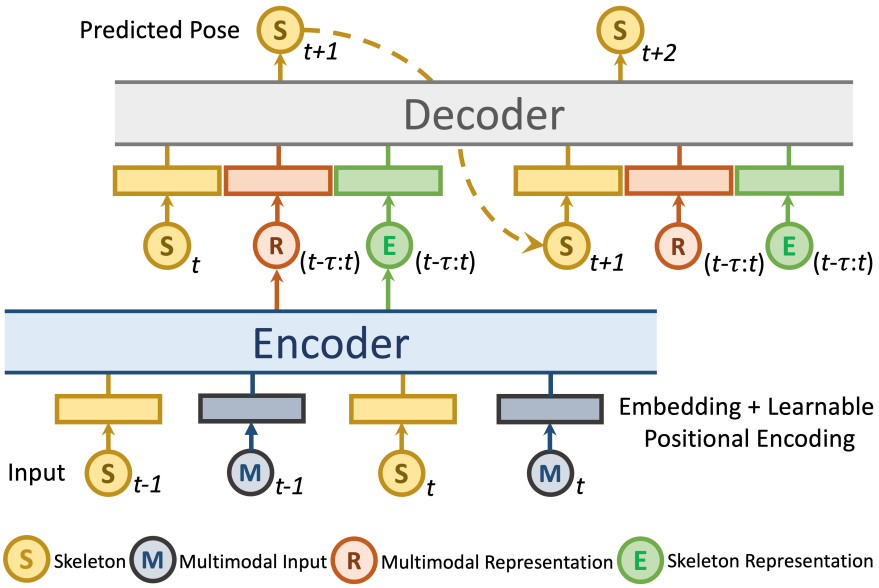


Figure 5.3: Overall Architecture of PoseTron. PoseTron consists of two modules: Multimodal Pose Encoder and Decoder. The Encoder encodes the motion of all the agents and modalities of all the data streams. The Decoder uses the encoded skeleton representation and the multimodal non-skeletal representation to forecast future pose.

Input Embedding

Skeleton modalities: We separately encode the skeletal input for each agent, $X^i_{input} \in \mathbf{X}$. The input sequence is first passed through an embedding layer to convert pose information into $d-$dimensional vectors. Next, we add positional encoding to each input frame. This is required as we are not using recurrent neural architectures, instead relying on a simple feedforward architecture, following the transformer implementation [84], which lacks the inherent notion of token order or position. We use a learnable positional encoding instead of the fixed sinusoidal positional encoding of the transformer architecture [84]. The operations can be formulated as follows:

$$\begin{aligned} X^i_{\text{token}} &= \text{E}(X^i_{input}); X^i_{\text{positional}} = \text{PE}(X^i_{\text{token}}) \\ X^i_{\text{embed}} &= X^i_{\text{token}} + X^i_{\text{positional}} \end{aligned} \tag{5.3}$$

Here, E represents the token embedding function that maps input tokens X^i_{input} to token embeddings X^i_{token}. PE represents the learnable positional embedding function that is required to inject information about the relative or absolute position of the tokens in the sequence. The operations in Eq. 5.3 are repeated for velocity and acceleration input.

Non-Skeleton modalities: For vision modalities such as RGB, the encoding process involves leveraging a feature extractor to obtain representation over a time horizon, followed by a temporal encoding of these features. We use a pre-trained SwinTransformer [220] architecture to extract features. This allows us to reduce the training footprint and leverage existing architectures for extracting rich representations. We pass the extracted representations, which have a dimension of $\mathbb{R}^{\mathbb{T}\times\mathbb{K}}$ with K and T being the feature dimension and timesteps, respectively, through the input embedding layers, using the same operations as Eq. 5.3 to add positional encoding. The overall operations are summarized as follows:

$$\begin{aligned} X_{features,\ m,\ t} &= \text{FE}(X_{m,\ t}) \\ X_{embed,\ m,\ t} &= \text{InputEmbeeding}(X_{features,\ m,\ t}) \end{aligned} \tag{5.4}$$

Here, m represents the modality, which can be one of RGB, Gaze, or any other available modality, and $X_{m,t}$ is the raw input of the modality. FE represents a pre-trained feature extractor, which is used to

extract representations $X_{features,m,t}$. The extracted representations are then passed to the input embedding function, previously defined in Eq. 5.3.

Multi-Head Self-Attention

The self-attention module is crucial in establishing temporal connections among individual skeleton embeddings. These skeletal embeddings undergo a self-attention process, enabling our framework to assess the significance of various tokens within the input sequence while handling each token. In this process, every position in the input sequence is linked to a weighted sum of all positions, including itself. These weights are determined dynamically based on the similarity between positions. The mechanism used to calculate a weighed representation for each position is as follows:

$$\begin{aligned} Q = X_{embed}W^Q;\ K = X_{embed}W^K; V = X_{embed}W^V \\ \text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \end{aligned} \tag{5.5}$$

Here, Q represents the query matrix representing the queries for each token, K represents the key matrix denoting the keys for each token, and V represents the value matrix denoting the values for each token. For each token, we calculate the attention scores over itself and all other tokens using the softmax function. W^Q, W^K, W^V represent the linear projection weights and $\frac{1}{\sqrt{d_k}}$ is the scaling factor for calculating the attention weights.

We pass the token embeddings X^i_{embed} through the attention mechanism (Attention) to obtain X^i_{att}. We repeat the aforementioned operations in Eqs. 5.3, 5.5 for each agents, thus obtaining X^i_{att};
$\forall i = 1, \ldots, K$. The operations can be represented as follows:

$$X^i_{att} = \text{Self-Attention}(X^i_{embed}, X^i_{embed}, X^i_{embed}) \tag{5.6}$$

Multi-Head Cross-Attention

Having computed the attention weights over skeleton tokens for each agent, the next task is to incorporate team dynamics by learning the association between tokens of different agents. To achieve this, we compute cross-attention scores for each agent-specific token X^i_{att} in the following manner: For a given agent, denoted as i, we treat their tokens as queries and compute key and value matrices for the remaining two agents, j and k. We then determine attention weights and calculate their averages to derive the ultimate token representation for each agent. These operations can be concisely summarized as follows:

$$\begin{aligned} X^{i,j}_{cross-att} &= \text{Cross-Attention}(X^i_{att}, X^j_{att}, X^j_{att}), \\ X^{i,k}_{cross-att} &= \text{Cross-Attention}(X^i_{att}, X^k_{att}, X^k_{att}), \\ X^i_{cross-att} &= \text{Mean}(X^{i,j}, X^{i,k}). \end{aligned} \tag{5.7}$$

5.5.2 Multimodal Pose Decoder

The input to the decoder is the agent-specific representation of the past motion *and* the multimodal representation from other non-skeleton modalities (see Fig. 5.3). Unlike prior multimodal approaches, which fuse representations from different modalities at the encoder [221, 222, 223], we choose to fuse the multimodal representations at the decoder. Fusing the representation at the decoder allows the decoder to leverage the context and dependencies between different modalities, which could lead to a more accurate generation. The decoder is auto-regressive, meaning it predicts future poses one at a time, taking into account the previously generated poses. We use the same decoder to generate the agent-specific poses separately.

Input Embedding

Similar to the encoder, the decoder has a separate input embedding and positional encoding. However, it must accommodate variable input sizes based on the size of the generated poses. In the initial decoding step, we pass the last observed pose along with the encoded skeletal and non-skeletal multimodal representations.

We add each generated pose to the decoder's input for each subsequent step while keeping the encoded representations constant. The operations are similar to Eq. 5.3, and is summarized below:

$$X^i_{dec-embed,t} = \text{InputEmbedding}(X^i_{dec-input,t}) \tag{5.8}$$

Multimodal Attention

The embedding from the decoder input, denoted as X^i_{dec}, along with the output of the encoder representation, is passed to the encoder-decoder attention module, $X^i_{cross-att}$ and $X_{embed,\ m,\ t}$. We use a similar attention mechanism as previously mentioned in Eq. 5.5. Here, the query is the decoder input, and the value and key are the output of the encoder, $X^i_{cross-att}$ and $X_{embed,\ m,\ t}$. The operations can be summarized as follows:

$$\begin{aligned} X^i_{enc} &= \text{Concat}(X^i_{cross-att}, X_{embed,\ m,\ t}) \\ X^i_{dec-att} &= \text{Decoder-Attention}(X^i_{dec-embed,t}, X^i_{enc}, X^i_{enc}) \end{aligned} \tag{5.9}$$

Output Embedding

The output of the decoder attention module is finally passed through linear layers to generate the output pose. The operations can be formulated as follows:

$$X^i_{output,t} = \text{OE}(X^i_{dec-att}) \tag{5.10}$$

Here, $X^i_{output,t}$ represents the generated future pose at time t of agent i. OE represents the output embedding, which is a linear projection of the decoder attention output to the pose space.

5.6 Experiments

In this section, we present our experimental details and results. We introduce the dataset and evaluation metric in Sect. 5.6.1, the implementation details in Sect. 5.6.2, and the results and discussion in Sect. 5.6.3.

5.6.1 Experimental Setup and Evaluation Metric

INTERACT Dataset

The proposed INTERACT dataset comprises 252 sessions of multimodal close-proximity collaboration data, which are evenly split into 126 episodes of HHC and 126 episodes of HRC scenarios, providing a large-scale dataset of multimodal and multi-agent interaction. For all the evaluation scenarios, we adopt a cross-group evaluation strategy, where we train and test on separate groups. The training set comprises all the even-numbered groups from 1 to 21, and the testing set comprises all the odd-numbered groups from 1 to 21 (recall that the dataset contains 21 groups in total). We propose three evaluation setups:

- HHC-Train, HHC-Test: In this setup, we trained and tested all the evaluated approaches on only Human-Human Collaboration data, using the aforementioned train and test sets. This approach allows us to establish how these approaches perform for multi-agent human motion prediction.
- HRC-Train, HRC-Test: In this setup, we trained and tested all the evaluated approaches on only Human-Robot Collaboration data. The purpose of this setup is to investigate the extent to which performance is influenced by the presence of a robot.
- HHC Train, HRC Test: In this setup, we use the train set of the HHC setup and the test set of the HRC setup. Here, we investigate how models trained on HHC generalize to HRC.

Evaluation metric

We report the Mean Per Joint Position Error (MPJPE) on 3D joint coordinates, a widely used metric for evaluating pose prediction performance [29, 28, 36, 1]. Since our dataset includes multiple agents, we also compute the Per Agent-MPJPE (PA-MPJPE) by averaging this evaluation metric across all agents. PA-MPJPE quantifies the average $L2$-Norm differences between each agent's predictions and ground truth. For all evaluated models, the input and output sequences have 25 timesteps.

Table 5.3: PA-MPJPE of different methods for the evaluation setup: HHC Train, HHC Test.

Approaches	5	10	15	20	25
Joint Learning [28]	5.74	7.68	9.33	10.85	12.30
Joint Learning + Social [28]	8.29	9.61	10.87	12.06	13.21
MA-AAE [1]	3.27	5.06	6.75	8.39	10.01
PoseTron (Skeleton Only)	3.35	5.04	6.51	7.83	9.03
PoseTron (Multimodal)	**3.21**	**4.86**	**6.34**	**7.65**	**8.84**

5.6.2 Implementation Details

In all experiments, we employ 3-D Skeletons and RGB data from two camera views as input, with the output being all agents' future 3-D Skeleton poses. While PoseTron adopts a modular architecture capable of accommodating the additional modalities, we restricted to using only 3-D Skeleton and RGB data in this work. We will explore the other modalities as part of future work. The feature dimension for Skeleton joints is 3×51 for each agent, while the RGB data from the two ZED cameras is pre-processed to dimensions of $3 \times 224 \times 224$ for each view. Both input and output sequences have a length of 25.

All the experiments were conducted using PyTorch [224] 2.0.1 running on an NVIDIA A100 GPU. For all the evaluated methods, we utilized a batch size of 256 and fine-tuned hyperparameters for optimal results. For extracting RGB features, we use a pre-trained SwinTransformer [220]. For PoseTron, we configured the encoder and decoder to have 8 attention heads while keeping the feedforward layer dimension fixed at 256. We used two stacks of encoder and decoder. For training PoseTron, we use the AdamW [97] optimizer with cosine annealing and warm restarts [225] with an initial learning rate of 0.001. We trained each evaluated approach for a maximum of 150 epochs, with a training time of approx. 3 hours.

5.6.3 Results and Discussion

In this section, we compare our approach, PoseTron, with three state-of-the-art multi-agent motion prediction approaches: Joint Learning [28], Joint Learning + Social [28] and Multi-Agent Adversarial Auto-encoder (MA-AAE) [1]. Joint Learning and Joint Learning + Social represent sequence2sequence [199, 29] approaches for motion prediction using pooling mechanisms to obtain joint representations over all the agents. On the other hand, MA-AAE uses a multi-agent adversarial auto-encoder with a self-attention mechanism to obtain interaction dynamics between multiple agents. We report the PA-MPJPE at distinct frame intervals, 5, 10, 15, 20, and 25, to evaluate model performances over different horizons.

Human-Human Collaboration Scenarios

Results: In Table 5.3, we compare the performance of our method, PoseTron against state-of-the-art multi-agent motion prediction methods on the HHC Train, HHC-Test setup. We use the same training and testing strategy for all the evaluated methods. As can be observed in Table 5.3, PoseTron (Skeleton Only) and PoseTron (Multimodal) strongly outperformed all the state-of-the-art approaches on majority of frame intervals, attaining the lowest PA-MPJPE.

Discussion: The two variants of PoseTron outperformed all the state-of-the-art approaches, which underlines the architectural improvements over the other evaluated methods. While all the evaluated approaches use recurrent neural networks for their sequence learning backbone, PoseTron adopts the transformer approach, which allows it to consider all the frames at the encoder and decoder. This allows PoseTron to extract salient representations from all the available frames. PoseTron also differs in its approach to modeling the interaction between multiple agents. While Joint Learning + Social [28] uses social pooling, an approach that has also been used in social navigation [48, 30], and MA-AAE [1] uses the self-attention mechanism, PoseTron uses the conditional attention at the encoder where for a given query agent, it can separately attend

Table 5.4: PA-MPJPE of different multi-agent motion prediction methods for the evaluation setup: HRC Train, HRC Test.

Approaches	5	10	15	20	25
Joint Learning [28]	6.08	7.72	9.05	10.24	11.36
Joint Learning + Social [28]	7.37	8.73	9.95	11.06	13.26
MA-AAE [1]	3.34	5.08	6.77	8.44	10.03
PoseTron (Skeleton Only)	2.90	4.46	5.89	7.22	8.38
PoseTron (Multimodal)	**2.71**	**4.22**	**5.51**	**6.64**	**7.65**

and weigh over the different motion frames of the other agents. The combination of the conditioning and the feedforward architecture, which allows each frame token to attend to all other tokens, enables superior representation learning at the encoder.

Human-Robot Collaboration Scenarios

Results: In Table 5.4, we report the performance of PoseTron against state-of-the-art multi-agent motion prediction methods on the HRC evaluation setup. Similar to the HHC evaluation setup, we were consistent with the training and testing strategy for all the evaluated methods. The two variants of our approach PoseTron (i.e., Skeleton and Multimodal) again outperformed the state-of-the-art evaluated methods consistently over all the frame intervals.

Discussion: The results presented in Table 5.4 emphasize the superior performance of PoseTron when compared to the other evaluated methods. Similar to the evaluation for HHC in Table 5.3, we observed that both variants of PoseTron achieved superior performance. In addition to the encoder operations that provide PoseTron with superior representation and sequence modeling capabilities, it distinguishes itself from other approaches through its unique decoder strategies, which further enhance its performance. While all the approaches follow an auto-regressive approach, PoseTron stands out as the only one capable of attending to all generated and past frames. This capability is achieved without increasing complexity, as it reuses the attention mechanism within the decoder. The key and value matrices are iteratively updated as the number of generated poses increases. Moreover, PoseTron (Multimodal) leverages multimodal representations in the decoder, enabling it to query multimodal features during the generation of future poses. The combination of these operations contributes to its improved performance over PoseTron (Skeleton Only) and other approaches.

Training on HHC, Testing on HRC

Results: In Table 5.5, we present PoseTron's performance against state-of-the-art multi-agent motion prediction methods, with all methods being trained on the HHC training set and tested on the HRC test set, ensuring no group overlap. This allows us to assess the generalizability of existing approaches exclusively trained on HHC to HRC scenarios. The test set is the same as in Table 5.4 for direct comparison. Both PoseTron variants consistently outperform state-of-the-art methods across majority of frame intervals, demonstrating superior generalizability over the two scenarios.

Discussion: The results presented in Table 5.5 provide a strong indication of PoseTron's generalizability compared to other approaches. PoseTron's generalizability can be attributed to the attention mechanisms in both the encoder and decoder, which allows PoseTron to efficiently utilize the multiple streams of agent and multimodal data in the context of PoseTron(Multimodal). Compared to Table 5.4, we observed some interesting trends. Firstly, all the approaches had a performance drop when training on HHC and testing on HRC, compared to training and testing on HRC. As the test is the same, and only the training data is different, this provides the strongest signal on the importance of HRC data.

Table 5.5: PPA-MPJPE of different multi-agent motion prediction methods for the evaluation setup: HHC Train, HRC Test.

Approaches	5	10	15	20	25
Joint Learning [28]	6.29	8.19	9.78	11.23	12.57
Joint Learning + Social [28]	8.77	10.33	11.85	13.32	14.73
MA-AAE [1]	3.46	5.33	7.15	8.92	10.64
PoseTron (Skeleton Only)	3.69	5.53	7.12	8.52	9.76
PoseTron (Multimodal)	**2.82**	**4.61**	**6.25**	**7.80**	**9.25**

Ablation Results

In this section, we compare the performance of PoseTron with ablated versions of itself, firstly at an architectural level where we remove the learnable positional embedding: PoseTron w/o L.P.E (Multiple RGB View), and use the default non-linearity of the original transformer [84], instead of SwishGLU which was used in PoseTron. Next, for the same ablated version, we further remove the RGB modalities, instead using only Skeleton: PoseTron w/o L.P.E (Skeleton Only). Finally, we ablate the modalities, first keeping one RGB view: PoseTron (One RGB View) and then using only Skeletons: PoseTron (Skeleton Only).

Results: We report the results of all the ablation experiments in Table 5.6, where we trained and tested on the HRC scenario. The results suggest that PoseTron using Multiple RGB View and Learnable Positional Embedding attained the best performance. The next best performing architecture was PoseTron with one RGB view: PoseTron (One RGB View). This was followed by PoseTron w/o L.P.E (Multiple RGB View), with PoseTron w/o L.P.E (Skeleton Only) performing worst of all the ablated versions.

Discussion: The results in Table 5.6 validate the architectural decisions made in designing PoseTron. Notably, PoseTron (Multiple RGB View) demonstrated superior performance, underscoring the advantages of incorporating multiple camera views alongside 3-D Skeletons. Following closely in performance was PoseTron (One RGB View), emphasizing two key insights: i) The utilization of additional modalities can enhance performance, and ii) PoseTron's cross-attention mechanism in the decoder effectively leverages multimodal features to construct a comprehensive representation.

The next best performing architecture was PoseTron w/o L.P.E (One RGB View), which features a variant of PoseTron that uses fixed positional encoding [84]. In addition, we ablate the SwishGLU activation function and replace it with ReLU. As observed in Table 5.6, the removal of these architectural details resulted in a performance drop over all the horizons, justifying the design choices in PoseTron.

Finally, the two Skeleton Only variants had higher prediction errors compared to the multimodal variants. Here again, PoseTron (Skeleton Only) with learnable positional embedding and SwishGLU activation outperformed PoseTron (Skeleton Only) without these features. This further emphasizes the effectiveness of incorporating multimodal information into the architecture.

5.6.4 Overall Discussion

The experiments provide several key insights in the context of motion prediction in HRC. One of the consistent themes across all the evaluation setups is the superior performance of PoseTron compared to state-of-the-art approaches in the field. One of the key distinguishing factors of PoseTron is its encoding mechanism, whereby it uses learnable positional encoding to add the notion of sequence, unlike the other recurrent approaches. This allows PoseTron to exploit the full context of the input and the generated sequence for its prediction. Furthermore, PoseTron introduces a novel mechanism to model the interaction among multiple agents through conditional attention. This enables individualized attention to different motion frames of other agents. Combined with the self-attention mechanism, this design choice leads to superior representation learning at the encoder.

Table 5.6: Ablation Study: HRC Train, HRC Test.

Approaches	5	10	15	20	25
PoseTron w/o L.P.E (Skeleton Only)	3.11	4.76	6.17	7.46	8.64
PoseTron w/o L.P.E (One RGB View)	3.00	4.56	5.92	7.15	8.23
PoseTron (Skeleton Only)	2.90	4.46	5.89	7.22	8.38
PoseTron (One RGB View)	2.85	4.40	5.75	6.97	8.05
PoseTron (Multiple RGB View)	**2.71**	**4.22**	**5.51**	**6.64**	**7.65**

Another contributing factor to PoseTron's superior performance is its decoder strategy. PoseTron stands out from other approaches by attending to all generated and past frames. Additionally, it incorporates multimodal representations into the decoder, enabling it to query multimodal features when generating future poses. This enhanced approach, as demonstrated in all experiments (Tables 5.3, 5.4, 5.5) and the ablation study (Table 5.6), leads to more accurate pose predictions.

The experiments also highlight the significance of the data source in training motion prediction models. As observed in Tabs. 5.4, 5.5, the performance dropped when the models were trained in one scenario and tested in another scenario. This emphasizes the need for more specialized datasets catering to the HRC setups. Notably, even in this setup, we observe the PoseTron's strong generalizability, as it attained the best performance. This generalizability is crucial in real-world applications where the ability to adapt to different scenarios is essential. While PoseTron and all other approaches were trained on an NVIDIA A100 GPU for training efficiency, we successfully ran PoseTron on a consumer-grade GPU: the NVIDIA RTX 2080Ti. This provides a pathway for our future work, which will focus on deploying PoseTron in real-time human-robot collaboration scenarios as part of the robot's perception stack.

5.7 Interact Dataset Details

The INTERACT dataset (https://bit.ly/posetron-interact) is a comprehensive human-human and human-robot collaboration dataset. It stands out for featuring a large-scale dataset with multimodal information. Each collaborative task involves a minimum of three participants. In human-human collaboration (HHC) scenarios, three participants engage in assembly tasks, while in human-robot collaboration (HRC) scenarios, four participants collaborate, including one robot. Two variations exist within both HHC and HRC setups, one without obstacles and one with obstacles, as illustrated in Fig 5.4.

5.7.1 Participant Roles

Human-Human Collaboration: For Human-Human Collaboration, there were three participants, each with specific roles:

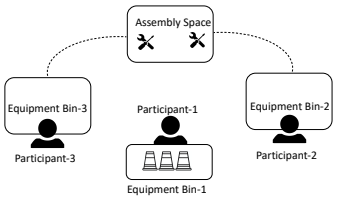
- **Participant-1 (P1):** P1's role is to pick cups containing Lego from Equipment Bin-1 and pass them to the other two participants at Equipment Bins 2 and 3. P1 can carry only one cup at any given time.
- **Participant-2 (P2):** P2's role is to receive the cup from P1 and move to the assembly space to build the Lego structure as per the instruction. P2 also needs to coordinate with P3 during the building process.
- **Participant-3 (P3):** Similar to P2, P3's role is to receive the cup from P1 and move to the assembly space to build the Lego structure as per the instruction. P3 also needs to coordinate with P2 during the building process.

Human-Robot Collaboration: For Human-Robot Collaboration, there were four participants, including one robot, each with specific roles:

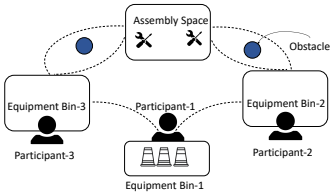
- **Participant-1 (P1):** P1's role is to pick up the cup from Equipment Bin-1 and hand it to the robot.

Table 5.7: Summary Statistics of the INTERACT Dataset.

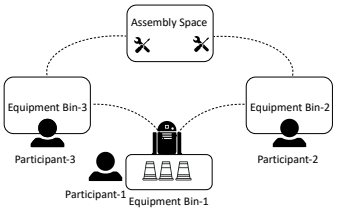
Scenario	Number of Participants	Variation	Average Duration (sec)	Max Duration (sec)	Min Duration (sec)
HHC	3 H	w/o obstacle	104.5 ± 27.7	216.0	60.9
		w obstacle	127.0 ± 49.9	322.6	50.9
HRC	3 H + 1 R	w/o obstacle	140.9 ± 22.2	226.3	88.6
		w obstacle	149.8 ± 40.6	337.5	70.5



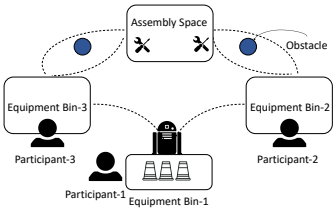
(a) HHC: Variation-1



(b) HHC: Variation-2



(c) HRC: Variation-1



(d) HRC: Variation-2

Figure 5.4: Setup for two scenarios: HHC and HRC, and thir corresponding variations.

- **Robot (R):** The robot is tasked with receiving the cup from P1 and moving to Equipment Bins 2 and 3 to provide the cup(s) to P2 and P3, respectively. The robot can carry only one cup at any given time.
- **Participant-2 (P2):** P2's role is to receive the cup from the robot and move to the assembly space to build the Lego structure as per the instruction. P2 also needs to coordinate with P3 during the building process.
- **Participant-3 (P3):** Similar to P2, P3's role is to receive the cup from the robot and move to the assembly space to build the Lego structure as per the instruction. P3 also needs to coordinate with P2 during the building process.

5.7.2 Dataset Details

Table 5.7 displays the average duration of each task and variation. For HHC, the mean, standard deviation, maximum, and minimum duration for Variations 1 and 2 are as follows: $(104.5, 27.7, 216.2,$
$60.9)$ and $(127.0, 49.9, 322.6, 50.9)$. For HRC, the corresponding statistics are $(140.9, 22.2, 226.3, 88.6)$ and $(149.8, 40.6, 337.5, 70.5)$ for Variations 1 and 2.

5.7.3 Dataset Collection and Synchronization

For 3D Skeleton Data, Motion Capture suits from the Optitrack Motion Capture System [216] were employed. Each suit provided real-time data of 51 skeleton-joint positions at an original sampling rate of 120 fps. To capture RGB+D data, two ZED Cameras from StereoLabs [217] were used, synchronized during data collection. RGB and depth data were collected from the left cameras of each ZED device, at a rate of 30 fps. For ego-view data of two participants, two eye-tracking devices from Pupil Labs [218] were utilized,

Table 5.8: Pre-task survey questionnaire

Question	Summary Response
Your Age	23.65(±3.91)
Your Gender	31.7% female, 66.67% male, 1.58% non-binary
What is your level of experience with robots?	2.21 ±1.19 [Scale: no experience" (1) to "expert-level experience" (5)]
Which hand do you primarily use to write?	right-handed 84.1% and 15.9% left-handed

Table 5.9: Post-task survey questionnaire for Participant 1 in HHC

Question	Summary Response
I verbally communicated with my teammates during the task.	Yes: 80.3%, No: 19.7%
I needed to observe and anticipate the activities of group member-2	Strongly Disagree: 8.8%, Disagree: 5.8%, Neutral: 7.3%, Agree: 21.2%, Strongly Agree: 56.9%
The activities of group member-2 had an impact on my performance.	Strongly Disagree: 18.2%, Disagree: 11.7%, Neutral: 18.2%, Agree: 13.9%, Strongly Agree: 38.0%
I needed to observe and anticipate the activities of group member-3.	Strongly Disagree: 8.8%, Disagree: 8.8%, Neutral: 6.6%, Agree: 19.0%, Strongly Agree: 56.9%
The activities of group member-3 had an impact on my performance.	Strongly Disagree: 19.0%, Disagree: 12.4%, Neutral: 15.3%, Agree: 17.5%, Strongly Agree: 35.8%
How will your rate the overall group coordination?	4.25 [Scale: no coordination" (1) to "high-coordination" (5)]

Table 5.10: Post-task survey questionnaire for Participant 2 in HHC

Question	Summary Response
I verbally communicated with my teammates during the task.	Yes: 51.7%, No: 48.3%
I needed to observe and anticipate the activities of group member-1	Strongly Disagree: 25.9%, Disagree: 19.7%, Neutral: 20.4%, Agree: 12.9%, Strongly Agree: 21.1%
The activities of group member-1 had an impact on my performance.	Strongly Disagree: 18.4%, Disagree: 17.7%, Neutral: 16.3%, Agree: 21.8%, Strongly Agree: 25.9%
I needed to coordinate my activities with activities of group member-3.	Strongly Disagree: 11.6%, Disagree: 11.6%, Neutral: 21.1%, Agree: 34.0%, Strongly Agree: 21.8%
How will your rate the overall group coordination?	4.47 [Scale: no coordination" (1) to "high-coordination" (5)]

providing ego-centric data at 30 fps. To mark the beginning of the task, a specific event (hand clap) visible to both the ZED cameras and the eye-tracking glasses was employed. Consequently, all recordings before the hand-clap were removed during post-processing. We utilized epoch timestamps to synchronize data across all modalities.

5.8 Surveys

We conducted Pre-task and Post-task surveys to gather participant information, perception and insights regarding the study.

Pre-task: For pre-task survey, we collected demographic information about the participants. Table 5.8 provides the aggregate information of all three participants. As reported in the table, the mean age of the participants was 34.65 ± 3.91, with 66.7% participants being male. The majority of participants were

Table 5.11: Post-task survey questionnaire for Participant 3 in HHC

Question	Summary Response
I verbally communicated with my teammates during the task.	Yes: 58.4%, No: 41.6%
I needed to observe and anticipate the activities of group member-1.	Strongly Disagree: 29.2%, Disagree: 18.2%, Neutral: 19.7%, Agree: 20.4%, Strongly Agree: 12.4%
The activities of group member-1 had an impact on my performance.	Strongly Disagree: 19.0%, Disagree: 19.0%, Neutral: 21.2%, Agree: 19.7%, Strongly Agree: 21.2%
I needed to coordinate my activities with activities of group member-2.	Strongly Disagree: 8.8%, Disagree: 16.1%, Neutral: 21.9%, Agree: 38.7%, Strongly Agree: 14.6%
How will your rate the overall group coordination?	4.34 [Scale: no coordination" (1) to "high-coordination" (5)]

Table 5.12: Post-task survey questionnaire for Participant 2 in HRC

Question	Summary Response
I verbally communicated with my teammates during the task.	Yes: 27.9%, No: 72.1%
I needed to observe and anticipate the activities of the robot	Strongly Disagree: 22.1%, Disagree: 15.6%, Neutral: 12.3%, Agree: 18.0%, Strongly Agree: 32.0%
The robot was effective in coordinating with both group members.	Strongly Disagree: 4.9%, Disagree: 10.7%, Neutral: 27.9%, Agree: 32.0%, Strongly Agree: 24.6%
The robot collaborator had an impact on my performance.	Strongly Disagree: 7.4%, Disagree: 7.4%, Neutral: 25.4%, Agree: 29.5%, Strongly Agree: 30.3%
The robot can replace the human collaborator.	Yes: 62.3%, No: 37.7%
I was uncomfortable around the robot.	Strongly Disagree: 52.5%, Disagree: 27.9%, Neutral: 12.3%, Agree: 2.5%, Strongly Agree: 4.9%
How will your rate the overall group coordination?	4.28 [Scale: no coordination" (1) to "high-coordination" (5)]

Table 5.13: Post-task survey questionnaire for Participant 3 in HRC

Question	Summary Response
I verbally communicated with my teammates during the task.	Yes: 29.1%, No: 70.9%
I needed to observe and anticipate the activities of the robot	Strongly Disagree: 17.2%, Disagree: 9.7%, Neutral: 19.4%, Agree: 26.1%, Strongly Agree: 27.6%
The robot was effective in coordinating with both group members.	Strongly Disagree: 1.5%, Disagree: 14.2%, Neutral: 21.6%, Agree: 30.6%, Strongly Agree: 32.1%
The robot collaborator had an impact on my performance.	Strongly Disagree: 6.7%, Disagree: 11.9%, Neutral: 23.1%, Agree: 29.9%, Strongly Agree: 28.4%
The robot can replace the human collaborator.	Yes: 66.4%, No: 34.3%
I was uncomfortable around the robot.	Strongly Disagree: 52.2%, Disagree: 26.1%, Neutral: 9.7%, Agree: 7.5%, Strongly Agree: 4.5%
How will your rate the overall group coordination?	4.24 [Scale: no coordination" (1) to "high-coordination" (5)]

right-handed at 84.1%. We also had an even spread in the participants' level of experience with robots with a mean of 2.21 ± 1.19.

Post-task: After each trial, participants were asked to complete a Post-task survey. Tables 5.9-5.11 depict the participants' response for HHC, while Tables 5.12-5.13 depict the participants' response for HRC. The

Table 5.14: List of common notations used

X	Observed Skeleton Pose
D	Data from non-skeleton modalities
$\hat{Y}$	Model's Prediction
Y	Ground-Truth Future Pose
FE	Unimodal Feature Extractor
OE	Output Embedding
HHC	Human-Human Collaboration
HRC	Human-Robot Collaboration
PA-MPJPE	Per Agent Mean Per Joint Position Error

tables provide some interesting insights on participants' perspectives, which are summarized below.

- **Anticipation in HHC:** For P1, 78.1% either agreed or strongly agreed on the need to observe and anticipate P2, while 75.9% either agreed or strongly agreed on the need to observe and anticipate P3, as reported in Table 5.9. For P2 and P3, this number was 34.0% and 32.8% respectively in Tables 5.10 and 5.11. This aligns with P1's role, as P1 needed to time their movement conditioned on P2 and P3's activities. For P2 and P3, their roles involved receiving the cup and assembling the Lego.
- **Anticipation in HRC:** In the HRC setting, the robot handed the cup to P2 and P3, with 50% of P2 (Table 5.12) and 53.7% P3 (Table 5.13) either agreeing or strongly agreeing that they needed to observe and anticipate the robot. Thus, introducing the robot altered participants' anticipation dynamics.
- **Comfort and Impact on Performance:** 80.4% of P2 and 78.3% of P3 either disagreed or strongly disagreed that they were uncomfortable around the robot. In addition, 59.8% of P2 and and 58.3% of P3 either agreed or strongly agreed that the robot impacted their performance.

5.9 Learning Architecture Details

Table 5.14 provide a list of all the major notations used in the main paper. As highlighted in the main paper's, PoseTron attained the lowest PA-MPJPE among the evaluted methods. In this section we list PoseTron's main architectural details. PoseTron comprised of several modules and sub-modules, which are listed below:

- **Positional Encoding:** We used Learnable Positional Encodings for the input skeleton and non-skeleton data at the encoder and decoder.
- **Pre-trained feature extractor:** We used the SwinTrnsformer [220] for extracting RGB features. We used pre-trained normalization and pre-processing for the input images following the original Swin-Transformer work.
- **Skeleton Encoder:** We used the same encoder to encode the skeleton information for each agent. To improve time complexity, we used 8 attention heads.
- **Skeleton Decoder:** We designed an auto-regressive decoder, which has shown to have better prediction performance than a non-autoregressive architecture. Similar to the encoder, we used 8 attention heads We used a dropout probability of 0.2 for all the linear layers.
- **Learning Rate Scheduler:** We use the AdamW optimizer with cosine annealing and warm restarts with an initial learning rate of 0.001. For the learning rate scheduler, the number of training steps was $10,000$, with warmup steps of 1000.

The various modules combine to enable highly accurate pose predictions for multiple agents in both Human-Human and Human-Robot scenarios.

Chapter 6

Improving Human Motion Prediction Through Continual Learning

Human motion prediction is an essential component for enabling closer human-robot collaboration. The task of accurately predicting human motion is non-trivial. It is compounded by the variability of human motion, both at a skeletal level due to the varying size of humans and at a motion level due to individual movement's idiosyncrasies. These variables make it challenging for learning algorithms to obtain a general representation that is robust to the diverse spatio-temporal patterns of human motion. In this work, we propose a modular sequence learning approach that allows end-to-end training while also having the flexibility of being fine-tuned. Our approach relies on the diversity of training samples to first learn a robust representation, which can then be fine-tuned in a continual learning setup to predict the motion of new subjects. We evaluated the proposed approach by comparing its performance against state-of-the-art baselines. The results suggest that our approach outperforms other methods over all the evaluated temporal horizons, using a small amount of data for fine-tuning. The improved performance of our approach opens up the possibility of using continual learning for personalized and reliable motion prediction.

6.1 Introduction

Human motion prediction involves forecasting future human poses given past motion. For enabling efficient Human-Robot Collaboration, a crucial aspect of robot perception is real-time anticipatory modeling of human motion [15, 20, 70, 74]. Fluid tasks such as collaborative assembly, handovers, and navigating through moving crowds require combining aspects of perception, representation, and motion analysis to accurately and timely predict probable human motion [226, 72, 141, 76, 78, 75, 73]. This would enable the robot to anticipate the human pose and intent and plan accordingly around the human partner without disturbing the natural flow of the human's motion. However, accurate and timely prediction of human motion remains a non-trivial problem due to the complex and interpersonal nature of human behavior [1, 227].

To address the aperiodic and stochastic nature of human motion, prior work has framed the problem of predicting future poses like that of sequence learning, modeling the spatio-temporal aspect of human motion using Recurrent Neural Networks [35, 29, 36, 1]. These approaches aim to learn a unified representation from training samples that are expected to generalize for test data. However, generalization comes at the cost of learning individual subtleties of motion, which is crucial for human-robot collaboration. When training these networks, the core assumption is that the given data points are realizations of independent and identically distributed (i.i.d) random variables. However, this assumption is often violated, e.g., when training and test data come from different distributions (dataset bias or domain shift) or the data points are highly interdependent (e.g., when the data exhibits temporal or spatial correlations) [228]. Both these cases are observed in human motion prediction, making it challenging to deploy models trained on benchmark models to the real world.

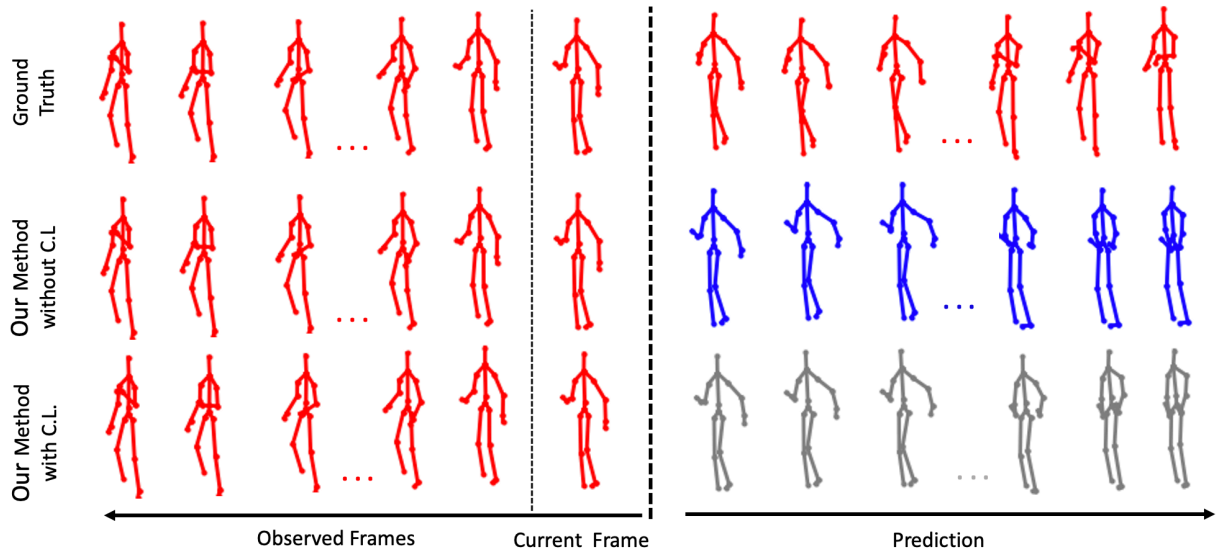


Figure 6.1: Qualitative performance of different motion prediction methods for walking on UTD-MHAD. The framework trained with Curriculum Learning (C.L.) have predictions that are closer to the ground-truth poses.

While generalization at the cost of learning individual preferences is sub-optimal, there is also a need to learn a robust representation over a diverse range of training samples. As such, training and generalizing over a benchmark dataset cannot be discarded and is, in fact, necessary as the first step to accurate motion prediction. Prior work on language modeling has demonstrated the benefit of learning a rich representation on a large training data followed by fine-tuning on a target task [171, 172, 24]. For human motion prediction, this can be posed as a continual learning problem whereby a motion prediction model acquires prior knowledge by observing a large range of human activities. This is followed by fine-tuning its parameters to accurately capture the subtleties of motion prediction for a particular individual. Such a learning setup, however, brings additional challenges to an already non-trivial problem, with prior work on continual learning demonstrating the risk of *catastrophic forgetting* [229, 230].

To address the challenges mentioned above, we propose a continual learning scheme that can improve human motion prediction accuracy while reducing the risk of catastrophic forgetting. Our framework is modular and is developed to acquire new knowledge and refine existing knowledge based on the new input. In line with prior work on computational neuroscience, which states that the brain must carry out two complementary tasks: generalize across experiences, and retain specific episodic-like events [231, 230]; we utilize a two-phase learning scheme. Our framework aims to learn a robust representation of past observations by training on a benchmark dataset in the first phase. This is achieved by using a modular encoder-decoder architecture with adversarial regularization [1], that has state-of-the-art performance on benchmark datasets. In the second phase, we use the representation learning aspect of the framework to condition future poses and fine-tune only the decoder module on new samples in a curriculum learning setup [232]. This mitigates the problem of training from scratch while also providing performance gains, both quantitatively over short, mid, and long-term horizons and qualitatively in terms of generating motion that is perceptibly similar to the ground-truth.

6.2 Problem Formulation

Formally defined, human motion prediction is the problem of predicting the future human pose over a horizon, given their past pose and any additional contextual information. In this paper, we assume that

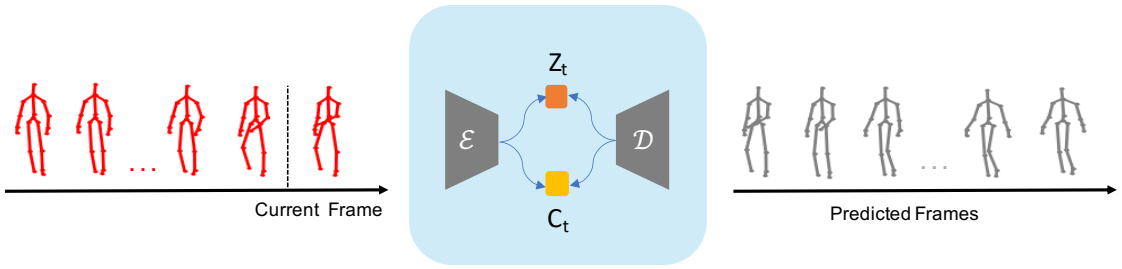


Figure 6.2: Motion prediction architecture

there is only one agent in the scene. For any particular scenario, the input to our model is the past or *observed* trajectory frames, spanning time $t = 1$ to τ,: $\mathbf{X} = \{x_1, \ldots, x_\tau\}$. Each frame $x_t \in \mathbb{R}^N$ denotes the N-dimensional body pose. N depends on the number of joints in the skeleton, J and the dimension of the joints D, where $N = J \times D$. The expected output of the model is the future trajectory frames over horizon H, i.e. the ground truth pose over the horizon $t = \tau + 1$ to $\tau + H$: $\mathbf{Y} = \{y_{\tau+1}, \ldots, y_{\tau+H}\}$.

Our first objective is to learn the underlying representation which would allow the model to generate feasible and accurate human poses $\hat{\mathbf{Y}} = \{\hat{y}_{\tau+1}, \ldots, \hat{y}_{\tau+H}\}$. We assume that future human pose is conditioned on the past observed or generated poses and predict each frame in an auto-regressive manner as formulated below:

$$p_\theta(\hat{\mathbf{Y}}) = \prod_{\delta=\tau+1}^{\tau+H} p_\theta(\hat{y}_\delta | \hat{y}_{\tau:\delta-1}, x_{1:\tau}) \tag{6.1}$$

where the joint distribution is parameterized by θ.

Next, we use these learned parameters to fine-tune for a specific agent who was not observed during the training phase, using a continual learning setup. Instead of updating all the model parameters, we update a specific module, say the decoder module, with corresponding parameters θ^*. We formulate this as follows, similar to prior work in continual learning [233]:

$$\log p\ (\theta^* | D) = \log p\ (D_B | \theta) + \log p\ (\theta | D_A) - \log p\ (D_B) \tag{6.2}$$

where D_A represents the first phase's training data, which involves learning a representation from the large data distribution. D_B represents the second phase's training data, whereby we aim to learn the parameters for a specific human. $logp(\theta|D_A)$ embeds all the prior information learned during the training phase.

6.3 Continual Learning for Human Motion Prediction

The collective goal of our approach is to accurately predict human motion while being flexible to parameter or architectural updates, given new data. Our overall framework is comprised of an encoder and decoder, trained end-to-end with adversarial regularization on the latent variables, building on top of our prior work [1]. The encoder aims to learn a rich representation over past trajectories, which the decoder can use to condition its prediction. To improve model stability and robustness of the latent space, we use adversarial regularization through discriminators. This acts as a regularizer during training and can improve the network's stability during parameter updates over new data. We will first describe the overall model for motion prediction and then discuss its flexibility for fine-tuning on a particular agent.

6.3.1 Overall model for motion prediction

Motion Encoder: The encoder learns a representation over the high-dimensional observed trajectory, projecting the input to a low-dimensional latent space. To obtain a rich and more robust representation over the past trajectories, we extract the past velocity and acceleration features along with the provided positional values, in line with prior work on motion prediction [1]. The velocity and acceleration features are first and second-order derivatives of the position values for each skeleton joint.

For encoding spatio-temporal representation from the position, velocity, and acceleration data, we employ Recurrent Neural Networks, in particular Gated Recurrent Units (GRU). We use unidirectional GRUs, as we wish to predict human motion in real-time. For each stream, the stream-specific GRU aims to extract the spatio-temporal representation that summarizes the input sequence, with the operation formulated as follows:

$$h_{s,t} = GRU(h_{s,t-1}, x_{s,t}, \phi_s) \tag{6.3}$$

where s represents the specific stream: position, velocity or acceleration, $x_{s,t}$ denotes the input to the GRU at t, $h_{s,t-1}$ corresponds to the past hidden state and ϕ_s represent the parameters of the GRU. The output from each GRU represents disparate information corresponding to the past trajectory and needs to be fused adaptively. As such, we use a multi-head self-attention mechanism [84] which is tasked to disentangle and extract relevant stream-specific representation.

$$\begin{aligned} h_t &= Concat(h_{pos,t}; h_{vel,t}; h_{acc,t}); \\ h_{att,t} &= Attention(h_t; \phi_{att}) \end{aligned} \tag{6.4}$$

where $h_{att,t}$ is the output of the attention mechanism, and ϕ_{att} represents the parameters. The output $h_{att,t}$ is used to obtain the latent representation, which is tasked to characterize the observed trajectory.

Latent Representation: The latent representation aims to capture relevant spatial and temporal semantics from the observed data, which can then be used to condition motion prediction. The latent representation is comprised of a continuous random variable and a categorical random variable. The motivation behind using both continuous and categorical variables is to jointly model the continuous aspect of human motion, such as the spatial semantics of a particular activity and the discrete characteristics of human motion such as the class activity or segment.

To obtain the continuous latent variable z_t, the output from the self-attention module is passed through a linear layer. In the case of the categorical latent variable c_t, the output from the self-attention module is passed through a linear layer followed by a softmax layer.

$$\begin{aligned} z_t &= Linear(h_{att,t}) \\ h_{c,t} &= Linear(h_{att,t}) \\ c_t &= softmax(h_{c,t}) \end{aligned} \tag{6.5}$$

Adversarial Regularization: To enforce a prior on the latent space, we use adversarial learning, similar to the Adversarial Autoencoders (AAE) [91] framework. This serves the purpose of a regularizer as there is a modification to the overall objective function: the objective function now consists of a reconstruction loss and an adversarial loss. We reason that this helps improve the stability of the overall framework for continual learning, as the parameters are updated based on two competing objectives: the reconstruction loss and the discriminator loss.

Similar to the GAN [93] and AAE [91] setups, the encoder aims to confuse the discriminators by trying to ensure that its output is similar to the aggregated prior. The discriminators are trained to distinguish the true samples generated using a given prior, from the latent space output of the encoder, thus establishing a min-max adversarial game between the networks [93, 91].

We use two discriminators, one for the continuous latent variable and the other for the categorical latent variable. The discriminators compute the probability that a point z_t or c_t is a sample from the prior distribution that we are trying to model (positive samples), or from the latent space (negative sample).

We use a Gaussian prior for continuous latent variables and a uniform distribution prior for categorical latent variables.

Decoder: The decoder uses the latent representation and the past generated pose to predict the future pose for each time step. It is auto-regressive, i.e., it uses the output of the previous timestep to predict the current pose and has only one stream: position as the expected output is future joint positions of the human.

The input to the decoder is the latent representation: z_t and c_t and the past generated pose, or the seed pose at time t if it is predicting the first time-step, $t + 1$. This is then passed to an attention mechanism

Table 6.1: MSE (in cm^2) comparison of fune-tuning vs no fine-tuning on UTD-MHAD for different test subjects) (Lower is better)

	Subject 2						Subject 4						Subject 6					
Frames	2	4	8	10	13	15	2	4	8	10	13	15	2	4	8	10	13	15
Zero-Velocity	11.20	27.37	66.85	86.23	112.39	127.64	13.11	32.33	77.22	97.80	123.99	138.53	10.37	25.56	64.62	85.29	115.33	135.17
Our method without Curriculum-Learning	**5.41**	**14.75**	33.87	41.89	51.33	56.12	7.78	18.26	41.66	51.46	62.67	68.1	6.99	16.18	38.51	49.40	63.7	71.68
Our method with Curriculum-Learning	7.62	16.17	**32.35**	**38.63**	**45.72**	**49.13**	**6.98**	**14.79**	**30.29**	**36.51**	**43.81**	**47.61**	**6.51**	**13.44**	**27.49**	**33.66**	**41.74**	**46.41**

that allows the decoder to adaptively condition its output on the latent variables that provide long-term information over the observed frames and the immediate generated frame. The output from this attention mechanism is next passed to a GRU cell, similar to the one at the encoder. This is followed by a Structured Prediction Layer (SPL) [36], which predicts each joint hierarchically following a skeleton tree, thus allowing the decoder to enforce structural prior on its final output. The operations at the decoder are formulated as follows:

$$\begin{aligned} p_t &= Concat(z_t, c_t, h_{dec,t-1}) \\ p_{att,t} &= Attention(p_t, \phi_{att}) \\ h_{dec,t} &= GRU(S_{t-1}, p_{att,t}, \phi_{pos}) \\ S_{t+1} &= \gamma(h_{dec,t}) \end{aligned} \tag{6.6}$$

6.3.2 Curriculum learning for the decoder:

The encoder-decoder architecture with adversarial regularization is trained to convergence on the training set. This training is followed by providing the overall architecture with unseen but small samples of motion data. This aims to relax the i.i.d assumption of the training procedure as our framework now has access to limited motion samples of the agent that it is trying to model.

Our choice of continual learning scheme is the curriculum learning setup [232], whereby we first train the network on a comparatively simpler task of representation learning, followed by a relatively difficult task of fine-tuning its parameters for a specific human subject. Our implementation is based on findings in connectionist models [234, 235], in particular self-organizing maps which reduce the levels of functional plasticity (i.e., ability to acquire knowledge in neural networks) through a two-phase training of the topographic neural map [236, 237]. The first phase is the organization phase, where the neural network is trained with a high learning rate and large spatial neighborhood size, allowing the network to reach an initial rough topological organization. The second phase is referred to as the tuning phase, where the learning rate and the neighborhood size are iteratively reduced for fine-tuning [230]. We aim to adopt these findings to a sequence learning framework.

Following prior work on developmental and curriculum learning [232, 238], we fine-tuned the architecture on the new data. We adopt techniques that will allow us to retain previous knowledge and avoid catastrophic forgetting during fine-tuning. In particular, we rely on *discriminative fine-tuning* [239], whereby we fine-tune only the decoder network at a different learning rate while freezing the encoder and the discriminator networks.

Fine-tuning the decoder: In line with equation 6.5, the input to the model is the sequence of observed poses: $\mathbf{X} = \{x_1, \ldots, x_\tau\}$, with the output being of the encoder being z_τ and c_τ. However, instead of imposing a prior on the latent space and training the encoder-decoder end-to-end, we only update the decoder's parameters. We also use a lower learning rate and rely on a small number of training samples to improve model stability and reduce the likelihood of catastrophic forgetting.

We leverage the representation learning capability of the framework that it attained when training on a large and diverse dataset. The encoder network is tasked to provide a representation summarizing the past observation that is used by the decoder to condition its prediction, similar to equation 6.6. The pre-trained weights from the training set are used to initialize the overall architecture and act as prior knowledge. The decoder weights are updated based on the reconstruction loss on the new data.

6.4 Experimental Setup

6.4.1 Dataset

We evaluated the performance of our approach on the widely used human-activity dataset: UTD-MHAD [85]. The dataset comprises 27 action classes covering activities from hand gestures to training exercises and daily activities, thus providing relevant activities for human-robot collaboration. Each activity was performed by 8 different subjects, with each subject repeating the activity 4 times. In our experiments, we use only Skeleton data for predicting human motion, following previous work in this domain [79, 35, 29, 37, 36], and considered each of the 20 provided joints. For all experiments, the model predicted output for the next 15 frames, using observation over the past 15 frames.

6.4.2 Generalized Representation Learning

We used the cross-subject evaluation scheme, training and validating on odd-numbered subjects for the first phase, thus providing the framework with a large training sample and maximizing the likelihood of encountering diverse demonstrations. To evaluate the performance, we hold out a section of the data for the validation set and early stopping. This reduces the likelihood of overfitting on the training data while also provide the mechanism for stopping the training procedure.

6.4.3 Curriculum Learning for a specific subject

Having learned a generalized representation, the second phase involved training the framework in a curriculum learning setup. Here, the experiments are conducted on a particular held-out even-numbered subject. We fine-tuned only the decoder using a reduced learning rate, with the encoder weights initialized from the first phase. As each subject has 4 trials, we trained on one trial and tested on the other 3 trials.

6.4.4 State-of-the-art method and baseline

For evaluating the efficacy of our curriculum learning setup, we compared against a non-curriculum learning framework, and the zero-velocity baseline [29]. The first benchmark [1] is comprised of an encoder-decoder framework, with adversarial regularization, but with no provision for curriculum learning. The zero-velocity baseline assumes that all the future predictions are identical to the last observed pose and is challenging to outperform for short-term prediction [29, 36]. It also allows us to gauge the movement dynamics, with a lower MSE for zero-velocity suggesting less movement and vice-versa for higher MSE.

6.4.5 Evaluation Metric

We evaluated the performance of all models using the Mean Squared Error (MSE), which is the l_2 distance between the ground-truth and the predicted poses at each timestep, averaged over the number of joints and sequence length, in line with prior work [86, 28, 38, 1]. The MSE is calculated as:

$$\mathcal{L}(\mathcal{X}, \hat{\mathcal{X}}) = \frac{1}{T.K} \sum_{t=1}^{T} \sum_{i=1}^{K} (x_t^i - \hat{x}_t^i)^2 \tag{6.7}$$

where, T and K are the total number of frame and joints respectively.

6.5 Results and Discussion

Results: We present the results of all approaches on the UTD-MHAD on table 6.1. We report the performance of all approaches at distinct frame intervals to circumvent the problem of frame drops during data collection and subsequent evaluation [1]. Our frame intervals aim to evaluate all models on short (2 & 4), mid (8 & 10), and long-term motion prediction (13 & 15). Table I depicts the performance of all approaches

with respect to the test subjects. The results in Table I suggest that fine-turning the framework allows it to outperforms all other methods and the zero-velocity baseline for short, mid, and long-term prediction.

Discussion: Our proposed approach outperformed the prior state-of-the-art approach and baseline both quantitatively by having lower MSE and qualitatively in terms of generating motion closer to the ground-truth pose (see Fig. 6.1). This shows the benefit of the curriculum learning approach while also suggesting that our overall framework is robust to catastrophic forgetting. The performance gain is especially significant over the mid and long-term, as the decoder is trained to learn the spatial-temporal movement pattern of a specific subject and can generate the future pose with higher accuracy.

Using a curriculum-learning setup, albeit on a small training sample, allows the framework to capture individual human motion subtleties, as seen by the lower MSE, particularly over the mid and long-term horizons. The approach is particularly useful when there is significant movement over the given horizon, as seen for Subjects 4 and 6 (table 6.1), who have higher MSE loss on the zero-velocity baseline. For Subject 2, there is overall less movement as seen by the zero-velocity MSE loss, and hence the performance gain is not significant over the mid and long-term and even worse over the short-term. Overall, the results are particularly promising as we did not fine-tune the encoder, instead only focusing on the decoder. Further improvement can be attained by fine-tuning the encoder.

Chapter 7

CoRaL: Continual Representation Learning for Overcoming Catastrophic Forgetting

7.1 Introduction

Intelligent agents have to adapt and interact with their environment using a continuous stream of observations, which requires that representations be learned in a continual manner [240]. However, continual learning does not suit current learning paradigms, which involve training Deep Neural Networks with the assumption that the training distribution is stationary and that the data samples are independent and identically distributed (i.i.d.) [229]. As such, current optimization strategies for training these networks focus on learning a representation from the existing data only and do not account explicitly for past observed data [241, 242]. As such, when these networks are tasked to learn from a sequential non-stationary data stream, they suffer from *catastrophic forgetting*, when the network forgets representation salient to the past task/data distribution [243].

Continual or Lifelong Learning approaches try to address the problem of catastrophic forgetting by acquiring new knowledge and refining existing representations from continuous non-stationary data such that the past knowledge is not completely overwritten [244, 245, 246, 247, 248]. Prior methods for addressing this problem can be grouped into three categories: *regularization-based*, *network expansion-based*, and *rehearsal-based* approaches. *Regularization-based* approaches induce a stability-plasticity trade-off in the network by penalizing the updates of specific parameters that are deemed important for past tasks [233, 249, 250]. *Network expansion-based* approaches instantiates new networks or modules for each new task [251, 252]. Lastly, *rehearsal-based* approaches mitigate forgetting by using a memory buffer of past data samples. These data are then replayed along with the samples of the current task to build optimization constraints during backpropagation [253, 254, 255, 256, 257, 258].

Although the previous works have all contributed to reducing catastrophic forgetting, their performances have yet to match offline learning. These methods predominantly focus on reducing the negative backward transfer of past tasks without explicitly improving the Representation Learning of the network, which is crucial for intelligent agents to generalize in incremental settings. However, we posit that the key to reducing backward transfer is to learn rich representations that can be shared among all tasks. Recently, work on self-supervised learning has shown promising results in learning robust representations using a pretext task [259, 260, 261, 262, 263]. While these methods can generate robust representations, they are prone to catastrophic forgetting when a new task is introduced.

To reduce catastrophic forgetting while maintaining performance, we propose CoRaL, a Continual Representation Learning approach for Overcoming Catastrophic Forgetting, that unifies Representation Learning

with Continual Learning (CL). Our approach tackles the problems of CL from two different aspects: learning effective representations that can be retained, refined, and transferred in incremental settings; and encouraging the model to retain its past responses. CoRaL introduces Representation Learning for non-stationary distributions to learn a robust representation. The Representation Learning module is a Siamese network setup [264, 259] comprising an encoder, projection, and predictor network. This is trained using the Cosine Similarity loss, which is used to minimize the distance between representations of the same class.

While learning transferable features can mitigate catastrophic forgetting, it may not explicitly direct the network to retain its past response to old training samples. To address this issue, we introduce a knowledge-distillation loss that compares the network's current output to its past output and penalizes divergence. The distillation loss imposes constraints on the parameter update, which prevents the network from forgetting the weights on the past samples. Thus, our overall framework unifies Representation Learning with Knowledge Distillation and is trained end-to-end with a novel objective function. The proposed objective function balances stability using the distillation loss and plasticity via the Representation Learning loss, which is now added to the existing Cross-Entropy loss. CoRaL is the first approach to efficiently combine Supervised Learning, Representation Learning, and Knowledge Distillation in an end-to-end manner through a novel objective function.

We performed extensive experiments to evaluate the efficacy of CoRaL, across three CL scenarios: incremental task, incremental class and incremental domain, on four widely used datasets in Continual Learning: permuted-MNIST [249], rotated-MNIST [254], Split-TinyImageNet [265] and split-CIFAR10 [249]. The results underline CoRaL's effectiveness in addressing catastrophic forgetting, as it outperformed all evaluated CL algorithms across all benchmarks attaining the highest accuracy with low standard deviation and the lowest forgetting. Furthermore, through extensive experiments, we demonstrated that these three objectives could be combined in a complementary manner for Continual Learning (CL). Finally, we conducted extensive ablation and stability-plasticity analyses to assess the efficacy of each of our modules across different scenarios and datasets. The ablation studies underline the importance of CoRaL's learning modules and provide empirical support for the objective function for Representation Learning. Our results provide promising direction for intelligent agents to learn continually.

7.2 Related Work

Continual Learning Strategies: Prior works in CL have commonly been evaluated in three scenarios: *incremental task*, *incremental class* and *incremental domain*. In *incremental task*, the output spaces (and task-learning layers) are disjoint, and task boundaries are explicitly stated [266]. Dissolving the class-boundaries leads to *incremental class*, where the model needs to infer both classes (new and old) and the shift in task. Finally, in *incremental domain*, the classes remain the same, but the inputs undergo a distribution shift.

Towards addressing the challenges brought about by these scenarios, recent work in CL can be grouped into *regularization-based*, *network expansion-based*, and *rehearsal-based* methods. *Regularization-based* approaches aim to address catastrophic forgetting by imposing constraints on the update of specific model parameters via additional regularization terms [233, 249, 250, 267, 268, 269]. For example, Elastic Weight Consolidation (EWC) identifies important parameters using the diagonal values of the Fisher information matrix, which are then regularized when learning on new tasks [233, 250]. Synaptic Intelligence takes a different approach to identify important parameters for each task, relying on loss sensitivity with respect to the particular parameters [249]. While regularization-based approaches have shown promising results, they are known to perform poorly when the number of tasks is high or in incremental-class settings.

While regularization approaches focus on constraining the updates of a fixed-capacity network, *network expansion-based* techniques add to the existing architecture every time there is a change in task [270, 251, 252, 271, 272]. For example, Progressive Neural Networks expand the architecture by allocating new sub-networks with fixed capacity for each new task while freezing previously trained networks [251]. Li et al. [252] proposed a learn to grow framework that employs a neural architecture search to find the optimal architecture for each sequential task. The key limitation with network expansion approaches is the increase

in computational overhead, and the added complexity of performing a hyper-parameter search for each new task.

Lastly, *rehearsal-based* methods use a memory buffer of past data which is replayed when learning new tasks [258, 257, 254, 253, 255, 256]. Buzzega et al. [258] proposed Dark Experience Replay, which added distillation loss and cross-entropy on previous task samples to reduce forgetting. Lopez-Paz et al. [254] proposed Gradient Episodic Memory, which uses past data to recall gradient directions and then project new gradients in a region that ensures that past representation is not over-written. These past data can also be used to add an additional objective term that can limit the forgetting on pivotal learned data points, as proposed in Hindsight Anchor Learning [273]. Sokar et al. [274] proposed a self-attention meta-learner, which incorporates an attention mechanism that learns to select particular representation for each task. Cha et al. [275] proposed Co^{2}L, that combines knowledge distillation with representation learning using the supervised Contrastive Learning objective [261]. The authors used a two-phase approach to train their framework, first for learning the representation and second for training the classifier.

Continual Learning in Intelligent Agents: For intelligent agents to become fully autonomous, they need to perceive and adapt to the changes in environmental dynamics [72, 1, 137, 103, 276]. Along this line, progress has been made in detecting changes and generalizing to new environments [155, 75, 76, 74, 101, 277, 278]. Recently, CL techniques have been introduced to applications ranging from object detection [279, 280, 281] to knowledge embeddings [282] to motion prediction [117, 283]. To mitigate catastrophic forgetting in object classification, Ayub et al. [279] proposed a centroid-based concept learning approach (CBCL), which uses a pre-trained feature extractor to obtain features for every input, on which an AggVar clustering algorithm is applied to generate centroids. Knoedler et al. [283] proposed a self-supervised approach to predicting pedestrian trajectories that uses online streams of data of pedestrian trajectories to continuously refine the model's prediction. Pellegrini et al. [284] proposed the use of latent replay, which combines with naive rehearsal, to classify objects on video benchmarks.

Although the aforementioned works have shown promising results for CL, learning effective representations that can be retained, refined, and transferred incrementally remains a long-standing challenge. Furthermore, certain approaches are only effective in specific scenarios, such as regularization-based approaches perform best on incremental-task and fail to achieve competitive results in other settings. Although recent works on improving representation learning have shown promising results [275, 285, 286], they require several changes which in turn relaxes the Continual Learning assumption, such as a two-phase training scheme for CL, class-balancing [275, 285]. To address these shortcomings, we propose a novel framework that unifies the Representation Learning for learning robust representation with a memory buffer that allows replaying of past samples and enables the network to optimize over a small set of past data.

7.3 Problem Formulation

Formally defined, a Continual Learning problem comprises a sequence of T distinct tasks containing non-overlapping input-output pairs. The overall goal for the agent is to accurately predict new classes as they appear without forgetting the discriminative ability of past classes. We use superscripts to represent the task and subscripts to represent the index in all our formulations.

Let us denote inputs as X and labels as Y. As such, (x_i^t, y_i^t) represent an input-label tuple for a given task t. For *incremental task* (IL-Task) scenarios, the output (or label) space is disjoint, i.e., $Y_i^n \neq Y_i^m$ for two different tasks m and n. The same also applies for *incremental class* (IL-Class) scenarios. In *incremental domain* (IL-Domain) settings, the output space remains the same but the input space changes with each domain, i.e., $Y_i^n = Y_i^m$ and $X_i^n \neq X_i^m$.

For each task, input-label $(x_i^t, y_i^t) \sim D^t$ pairs are independently drawn from some task-specific distribution D^t. The learner is tasked to learn a non-linear mapping function using an encoder f_θ and a g_θ, which would correctly predict the output label for the input. Here, θ represents the parameters of the non-linear functions. For IL-Task settings, the learner is trained using the following objective:

$$\mathcal{L}(\theta) := \sum_{t=1}^{T} \mathbf{E}_{D^t}[l(y^t, g_\theta(f_\theta(x^t, t)))] \tag{7.1}$$

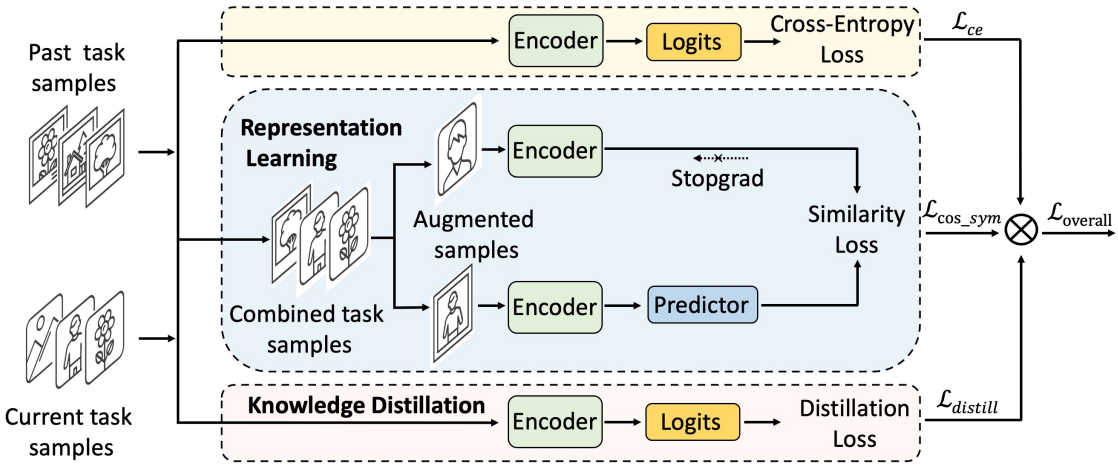


Figure 7.1: CoRaL: Continual Representation Learning for Overcoming Catastrophic Forgetting. The Representation Learning module maximizes the similarity between two augmented views of the input, which leads to more robust features. The Knowledge Distillation module distills the knowledge of previous tasks by buffering past input-output tuples of the network from memory. The two modules combine to reduce catastrophic forgetting.

Here, the learner has access to the task label and will have different task-learning layers per task. $l(\cdot,\cdot)$ represents the loss function that needs to be minimized. For IL-Class and IL-Domain, the learner has one task-specific layer and is trained as follows:

$$\mathcal{L}(\theta) := \sum_{t=1}^{T} \mathbf{E}_{D^t}[l(y^t, g_\theta(f_\theta(x^t)))] \tag{7.2}$$

7.4 CoRaL: Continual Representation Learning

We now introduce our proposed framework, CoRaL: Continual Representation Learning, an end-to-end representation learning framework to tackle catastrophic forgetting in CL. The overall algorithm for our framework is provided in Algo. 2 and illustrated in Fig. 7.1. There are two primary components of CoRaL, which work in tandem with the supervised learning objective : i) Representation Learning (Algo. 2, Lines 4-10), and, ii) Knowledge Distillation (Algo. 2, Lines 11-13).

The Representation Learning module is a Siamese network setup comprising an encoder, a projection, and a prediction network. The projection and prediction networks are MLPs, while the encoder consists of a backbone (e.g., ResNet). To aid the Representation Learning module, we propose a memory buffer that replays past input samples using reservoir sampling [258]. The input samples from the buffer undergo augmentations before being fed to the Representation Learning module, which is trained using the Cosine Similarity loss to encourage the encoder to minimize the embedding distance between similar inputs under changing distributions.

The memory buffer also stores the model's past output logits, which is used in the second part of the framework: the Knowledge Distillation module. The storing of past outputs ensures that even when the encoder learns robust representations, the task learning layer can map it to the correct class. In this module, past input samples are fed to the overall network (encoder + task-learning layer), and the output is compared to the past output from the buffer, with the objective being to penalize divergence between the two values. We will first describe the Representation Learning module of our framework, followed by the Knowledge Distillation module, and finally, the modified objective function.

Algorithm 2: CoRaL : Continual Representation Learning for Overcoming Catastrophic Forgetting

Input: Dataset D, Networks: Encoder f_θ, Predictor h_θ, Task Layer g_θ, Memory Buffer M, Scalars: α, β, Learning rate γ

for x^t, y^t, t *in* D^t **do**
- $x_1^t, x_2^t \leftarrow aug(x^t)$
- $\hat{y}^t \leftarrow g_\theta(f_\theta(x_1^t))$
- # **Representation Learning:**
  - $(x^\tau, \hat{y}^\tau) \leftarrow sample(M)$
  - $x_1^\tau, x_2^\tau \leftarrow aug(x^\tau)$
  - $x^{t,\tau} \leftarrow cat([x_1^t, x_2^t], [x_1^\tau, x_2^\tau])$
  - $z_1, z_2 \leftarrow f_\theta(x^{t,\tau})$
  - $p_1, p_2 \leftarrow h_\theta(z_1), h_\theta(z_2)$
  - $\mathcal{L}_{cos\_sym} \leftarrow \frac{1}{2}\mathcal{L}_{cos}(p_1, sg(z_2)) + \frac{1}{2}\mathcal{L}_{cos}(p_2, sg(z_1))$
- # **Knowledge Distillation:**
  - $y^\tau \leftarrow g_\theta(f_\theta(x_1^\tau))$
  - $\mathcal{L}_{distill} \leftarrow ||y^\tau - \hat{y}^\tau||_2^2$
- $\mathcal{L}_{overall} \leftarrow \mathcal{L}_{ce}(\hat{y}^t, y^t) + \alpha \cdot \mathcal{L}_{cos\_sym} + \beta \cdot \mathcal{L}_{distill}$
- $\theta \leftarrow \theta + \gamma \nabla_\theta \mathcal{L}$
- $M \leftarrow reservoir(M, (x^t, \hat{y}^t))$

end

7.4.1 Representation Learning

Objective Function

Contrastive learning [259, 260] have proven to be an effective technique for learning instance discrimination without labels. The core idea behind these works is the following: for every input, minimize the distance between the positive sample pairs and maximize the distance between the negative sample pairs. The positive sample pairs are the embeddings of the two augmented versions of the input, while all other embeddings are considered negative. Let (i, j) be the positive pairs. The contrastive learning objective can be defined using the InfoNCE loss:

$$\mathcal{L}_{i,j} = -\log \frac{\exp(sim(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} 1_{[k \neq i]} \exp(sim(z_i, z_k)/\tau)} \tag{7.3}$$

Here, z_i, z_j are the embeddings of the positive sample pairs, $1_{k \neq i}$ is an indicator function which is 1 for the $2N - 1$ negative sample pairs, i.e., when $k \neq i$. τ represents the temperature parameter used to scale the gradient. sim represents the dot product between the l_2 normalized embeddings.

While this formulation has proven effective in learning instance discrimination in the absence of labels, methods based on this contrastive formulation are sensitive to the choice of data augmentations [287]. This motivates the need to develop techniques that are robust to data augmentations and distribution shifts and is a key component for Continual Representation Learning. Here, due to the non-stationary data distribution, the encoder output for the positive samples is continuously changing, *along with the negative samples*, making the objective function in Eq. 7.3 challenging to optimize. Furthermore, such contrastive learning methods rely on a large number of negative samples, which require a large batch size, making their adoption intractable for a CL setup.

As our primary objective is to learn effective and robust representations under non-stationary settings, we introduce the modified Cosine Similarity loss [263] for CL, which only relies on the positive samples. The task then reduces to maximizing the similarity between two augmented versions of the same input, say z_1, z_2. As such, our objective function is:

$$\mathcal{L}_{cos}(p_1, z_2) = -\frac{p_1}{||p_1||_2} \cdot \frac{z_2}{||z_2||_2} \tag{7.4}$$

Here, p_1 is a non-linear transformation of z_1, which is processed through a predictor network.

Siamese Network Setup

We now describe our proposed representation learning module, which is trained using the modified Cosine Similarity loss (Eq. 7.4.) The input to the module is the augmented image samples. We first apply data augmentation on each input sample, in line with prior works [259, 260, 288, 287]. This augmentation effectively doubles the input sample size. The augmented samples are then passed to the Representation Learning module.

The Representation Learning module is a Siamese Network setup, comprised of one encoder f_θ, one projection network d_θ, and a prediction network h_θ. Inspired from BYOL [287] and SimSiam [263], our Siamese Network setup has one online network and one target network. The target network (f_θ & d_θ) provides the regression targets to train the online network, which is then used to update the gradients of the online network (f_θ, d_θ & h_θ). This is then followed by swapping the roles, i.e., the previously online network is now the target network and has to provide the regression targets, leading to two passes of optimization.

We use the same encoder and projection network for both online and target networks. For a given task t, our architecture takes as input two randomly augmented views from an image x_1^t, x_2^t, which is processed by the encoder network f_θ, followed by the projection network d_θ to get two embeddings z_1, z_2. As CoRaL is a rehearsal-based approach, the image can come from the stream of the current task t or from the input buffer τ. For a given pass, the prediction network h_θ processes one of the embeddings, for example, z_1, to output p_1 and matches it to the other embedding, z_2.

The outputs, p_1 and z_2, are normalized before calculating the Cosine Similarity loss (Eq. 7.4). Our framework uses symmetric loss, whereby we update the gradient of one network in one pass and then update the gradient of the other network in the next pass. In either pass, only the online network (one with the predictor) is updated ***end-to-end*** using backpropagation. The target network is not updated and is tasked to provide regression targets. The symmetric loss is an extension of Eq. 7.4 and is formulated below:

$$\mathcal{L}_{cos\_sym} = \frac{1}{2}\mathcal{L}_{cos}(p_1, sg(z_2)) + \frac{1}{2}\mathcal{L}_{cos}(p_2, sg(z_1)) \tag{7.5}$$

This is performed for all input samples, and the total loss is averaged. Here, sg refers to the stopgrad function and is used to prevent the target network from getting updated [263], meaning that the encoder on x_2 receives no gradients from z_2 and the encoder on x_1 receives no gradients from z_1.

Prior works on representation learning have leveraged self- supervised learning, with the frameworks trained in two phases: representation learning and task learning [259, 260, 287]. In the representation learning phase, the network has no access to labels and relies on a *pretext* task to distinguish between the unlabelled classes. In the task learning phase, the network has access to a small number of labels. Recent works on CL have also used a similar approach by using a two-phase training scheme and relaxing the non-i.i.d. assumption by introducing class-balancing strategies when training the classifier. A key difference between our work and prior works [275, 285] is that we do not decouple the representation learning from the task learning and in fact, unify the two objective functions, i.e., the Cross-Entropy loss and the Cosine Similarity loss. This approach has the benefit of learning robust features while also mapping the representations to their respective labels.

7.4.2 Knowledge Distillation

The Representation Learning module for CoRaL has the explicit objective of improving the robustness of the learned features at the encoder using the Cosine Similarity loss. However, standard backpropagation with the dual objective of Cross-Entropy and Cosine Similarity may not prevent catastrophic forgetting. As such, even if the representations are robust to changing distributions, the weights and output of the task-learning network may be prone to changes. To address this challenge, we introduce a Knowledge Distillation module to our framework.

Although knowledge distillation [290] has been mostly deployed in a teacher-student setting, where the teacher network distills its knowledge to a student network, in this work, we rely on self-distillation in CL settings [253, 240]. To perform self-distillation, we store the final network response along with the

Table 7.1: Performance comparison (averaged across 10 runs) of various CL methods on different scenarios (Accuracy in %)

Approach	Method	IL-Task		IL-Class		IL-Domain	
		S-CIFAR10	S-Tiny-ImageNet	S-CIFAR10	S-Tiny-ImageNet	P-MNIST	R-MNIST
Non-CL	JOINT	98.31 ± 0.12	82.04 ± 0.10	92.20 ± 0.15	59.99 ± 0.19	94.33 ± 0.17	95.76 ± 0.04
	SGD	61.02 ± 3.33	18.31 ± 0.68	19.62 ± 0.05	7.92 ± 0.26	40.70 ± 2.33	67.66 ± 8.53
Architectural	PNN [251]	95.13 ± 0.72	67.84 ± 0.29	-	-	-	-
Regularization	oEWC [250]	68.29 ± 3.92	19.20 ± 0.31	19.49 ± 0.12	7.58 ± 0.10	75.79 ± 2.25	77.35 ± 5.77
	SI [249]	68.05 ± 5.91	36.32 ± 0.13	19.48 ± 0.17	6.58 ± 0.31	65.86 ± 1.57	71.91 ± 5.83
	LwF [240]	63.29 ± 2.35	15.85 ± 0.58	19.62 ± 0.05	8.46 ± 0.22	-	-
Rehearsal	ER [256]	91.19 ± 0.94	38.17 ± 2.00	44.79 ± 1.86	8.49 ± 0.6	72.37 ± 0.87	85.01 ± 1.90
	GEM [254]	90.44 ± 0.94	-	25.54 ± 0.76	-	66.93 ± 1.25	80.80 ± 1.15
	A-GEM [257]	83.88 ± 1.49	22.77 ± 0.03	20.04 ± 0.34	8.07 ± 0.08	66.42 ± 4.00	81.91 ± 0.76
	iCARL [253]	88.99 ± 2.13	28.19 ± 1.47	49.02 ± 3.20	7.53 ± 0.79	-	-
	FDR [255]	91.01 ± 0.68	40.36 ± 0.68	30.91 ± 2.74	8.70 ± 0.19	74.77 ± 0.83	85.22 ± 3.35
	GSS [289]	88.80 ± 2.89	-	39.07 ± 5.59	-	63.72 ± 0.70	79.50 ± 0.41
	HAL [273]	82.51 ± 3.20	-	32.36 ± 2.70	-	74.15 ± 1.65	84.02 ± 0.98
	DER [258]	91.40 ± 0.92	40.22 ± 0.67	61.93 ± 1.79	11.87 ± 0.78	81.74 ± 1.07	90.04 ± 2.61
	DER++ [258]	91.92 ± 0.60	40.87 ± 1.16	64.88 ± 1.17	10.96 ± 1.17	83.58 ± 0.59	90.43 ± 1.87
	CoRaL (Ours)	**92.01 ± 0.32**	**41.37 ± 0.91**	**65.24 ± 1.09**	**14.06 ± 0.57**	**84.60 ± 0.48**	**91.79 ± 0.92**

corresponding inputs in the memory buffer using reservoir sampling. This retention of past input and network response allows the network to have similar outputs, even if there is a shift in representation.

For every data x^τ sampled from the memory buffer, we forward propagate the sample through the current network to obtain the final output before computing the softmax probability, y^τ. This output is then compared with the network's past response, $\hat{y}^\tau$, which is obtained from the memory buffer. Unlike prior distillation-based approaches [258] which have shown to benefit from storing both the network's output logits and the class label, we only store the network's output logits, thus simplifying the objective function. As we are computing the loss on pre-softmax outputs, we use the Mean Square Error between the logits of the current model and the past model. The overall operations in this module can be formulated as:

$$\mathcal{L}_{distill} = ||y^\tau - \hat{y}^\tau||_2^2 \tag{7.6}$$

7.4.3 Overall Objective for End-to-End Learning

In CoRaL, we introduce a new approach to combine two different modules for end-to-end training. Moreover, these modules are used to overcome catastrophic forgetting on past task samples. Learning a mapping between the inputs and the labels for the current task is done using the *Cross-Entropy* loss function for each mini-batch that is sampled from the current distribution.

Overall, CoRaL is comprised of three different loss functions that is trained end-to-end: the standard Cross-Entropy loss for supervised learning, the modified Cosine Similarity loss from Eq. 7.5, and the Distillation loss from Eq. 7.6 for the distillation learning. For the initial task, the framework uses only the Cross-Entropy loss. For every incremental task/class/domain that follows, the model is trained using the following objective function:

$$\mathcal{L}_{overall} = L_{ce} + \alpha \cdot \mathcal{L}_{cos\_sym} + \beta \cdot \mathcal{L}_{distill} \tag{7.7}$$

Here, α, β are hyper-parameters for the different losses.

7.5 Experimental Setup

7.5.1 Datasets

We evaluated our approach by comparing its performance to several state-of-the-art CL methods on four widely benchmarked datasets: Rotated MNIST (R-MNIST) [254], Permuted MNIST (P-MNIST) [233] which are variants of the MNIST dataset, Split CIFAR-10 (S-CIFAR-10) [249] which is a variant of the CIFAR10

Table 7.2: Backward Transfer (BWT) comparison (averaged across 10 runs) on **Incremental Domain** (in %).

Approach	Method	P-MNIST	R-MNIST
Non-CL	SGD	-57.65 ± 4.32	-20.34 ± 2.50
Architectural	PNN [251]	-	-
Regularization	oEWC [250]	-36.69 ± 2.34	-24.59 ± 5.37
	SI [249]	-27.91 ± 0.31	-22.91 ± 0.26
	LwF [240]	-	-
Rehearsal	ER [256]	-22.54 ± 0.95	-8.24 ± 1.56
	GEM [254]	-29.38 ± 2.56	-11.51 ± 4.75
	A-GEM [257]	-31.69 ± 3.92	-19.32 ± 1.17
	FDR [255]	-20.62 ± 0.65	-13.31 ± 2.60
	GSS [289]	-47.85 ± 1.82	-20.19 ± 6.45
	HAL [273]	-15.24 ± 1.33	-11.71 ± 0.26
	DER [258]	-13.79 ± 0.80	-5.99 ± 0.46
	DER++ [258]	-11.47 ± 0.33	-5.27 ± 0.26
	CoRaL (Ours)	**-9.92 ± 0.51**	**-4.65 ± 0.86**

[291] and Split TinyImageNet (S-Tiny-ImageNet) [265]. Please check the supplementary materials for more details on the datasets.

7.5.2 Continual Learning Scenarios

We consider three challenging CL scenarios for conducting evaluation inline with prior works [266, 249]. For all scenarios, the original dataset is split into separate tasks. For S-CIFAR-10, the original dataset is split into five 2-way classification tasks, whereas for S-Tiny-ImageNet, the original dataset is split into ten 20-category classification tasks. For P-MNIST and R-MNIST, the image pixels in the original dataset are permuted or rotated for 20 rounds, resulting in a shift in input while the classes remain unchanged.

For *incremental task* (IL-Task), models have access to the task label, and as a result, they are trained with task-specific components. For *incremental class* (IL-Class), models need to perform both classification of new samples as they arrive and infer the change in task. Lastly, for *incremental domain* (IL-Domain), models do not have access to task labels and need to only perform the classification of the input images, which may undergo perturbations.

7.5.3 Architectures

We use different encoders depending on the complexity of the dataset. On R-MNIST and P-MNIST, we use a fully connected neural network with two hidden layers of 100 ReLU units, following prior works [254]. On the CIFAR-10 and TinyImageNet, we use ResNet18. For implementation details, please look at the supplementary.

7.6 Training Details

There are two primary components of CoRaL, both of which work in tandem with the supervised learning objective: i) Representation Learning, and, ii) Knowledge Distillation.

7.6.1 Augmentation

CoRaL's Representation Learning module relies on applying random augmentations to create two views of the same image. For one of the views, we apply the standard data augmentations, thereby ensuring fair evaluation with prior Continual Learning algorithms [258, 251, 249]. This view is also used for the Knowledge Distillation module and standard training using Cross-Entropy. For the other view, we apply augmentations

in line with prior works on self-supervised learning [259, 263, 260]. The augmentations were mostly applied to the S-CIFAR-10 and S-TinyImageNet datasets. They are as follows:

- **RandomResizedCrop**: We use standard random cropping [292]. For both S-CIFAR-10 and S-TinyImageNet, we used a scale of [0.2, 1.0]. For R-MNIST, we used a scale of [0.7, 1.0].
- **RandomHorizontalFlip**: We applied random horizon flip for the S-CIFAR-10 and S-TinyImageNet datasets.
- **ColorJitter**: We distorted images for S-CIFAR-10 and S-TinyImageNet by color jittering and color dropping. We used a probability of 0.8 for color jittering on any image, with the coefficients of brightness, contrast, saturation, hue being 0.4, 0.4, 0.1, 0.1 respectively.
- **RandomGrayScale**: We used a probability of 0.2 to apply grayscaling to images for S-CIFAR-10 and S-TinyImageNet datasets.
- **GaussianBlur**: We applied Gaussian blur using a kernel size of 7x7 on the image samples of S-TinyImageNet.

7.6.2 Input to the encoder

In the Representation Learning module, we use the same encoder for both the online and target network. The input can be formulated as follows: for a given sample x, we create two augmented versions, x_1 and x_2, and feed one to the online network and the other to the target network. As such, there is variation between the inputs to the two networks.

7.6.3 Learning Architecture Details

CoRaL with Cosine Similarity: For S-CIFAR-10 and S-TinyImageNet, we used the ResNet18 architecture as the encoder. In addition, we introduced two MLP heads for projection and prediction networks respectively, only for the Representation Learning module. We used a 3-layer MLP with BatchNorm and ReLU activations for the projection network. The hidden units were [512, 512, 64]. We used a 2-layer MLP with BatchNorm and ReLU activation for the prediction network. The hidden units were [32, 64].

For P-MNIST and R-MNIST, we used a 2-layer MLP with 100 units each and ReLU activations as the encoder. For the prediction network, we used a 2-layer MLP with ReLU activations. The hidden units were [100, 50]. We used a 2-layer MLP with hidden units [25, 50] for the projection network.

In all the experiments, the encoders used were in line with the evaluated state-of-the-art Continual Learning algorithms [258, 233, 257, 254]. The only distinction is the addition of the prediction and projection networks, which were used to apply non-linear projection over the feature embeddings for the Representation Learning module.

CoRaL with Contrastive Learning: We used the same ResNet18 and 2-layer MLP encoders for our experiments with Contrastive Learning loss. We updated one encoder using the standard backpropagation and the other encoder using momentum, inspired by prior works [260, 288]. Furthermore, in these experiments, we used two projection heads for the two encoders, with both being a 2-layer MLP with ReLU activation. The hidden units for both were [64, 64].

We used the InfoNCE loss as the objective function. The temperature τ was 0.1 for the Representation Learning module. As previously mentioned, one of the encoder is updated ***end-to-end*** using backpropagation. The other encoder is updated more smoothly using the following formulation:

$$\theta_k \leftarrow m\theta_k + (1-m)\theta_q, \quad m \in [0,1] \tag{7.8}$$

Here, θ_q and θ_k represent the parameters of the standard and momentum encoder respectively. m represents the momentum hyper-parameter and controls the rate of update for the momentum encoder. A higher value of m would make the rate of update slower, while a lower value of m would increase the rate of update. The value of m used was 0.99. In addition, we maintained a queue of negative samples, with the feature bank size, F being 4096.

Table 7.3: List of common notations used

T	Tasks
X	Inputs
Y	Labels
D	Dataset
m	Momentum
F	Feature bank
τ	Temperature
$\mathcal{L}_{ce}$	Cross-Entropy Loss
$\mathcal{L}^{\tau}{}_{ce}$	Task specific Cross-Entropy Loss
$\mathcal{L}_{InfoNCE}$	InfoNCE Loss
$\mathcal{L}_{distill}$	Distillation Loss
α	Scalar multiplier for $\mathcal{L}_{cos\_sym}$ in Eq. 7
β	Scalar multiplier for $\mathcal{L}_{distill}$ in Eq. 7
Acc.	Accuracy
BWT	Backward Transfer

Table 7.4: Impact of different Cosine Similarity loss on S-CIFAR-10, averaged over 10 runs (in %).

Method	IL-Task	IL-Class
CoRaL with standard Cosine Similarity	90.55 ± 0.54	56.49 ± 1.42
CoRaL with modified Cosine Similarity	**92.01 ± 0.32**	**65.24 ± 1.09**

7.7 Sampling strategy for rehearsal

We use reservoir sampling to select past task samples. Our approach does not rely on task boundary and selects N random samples from the input stream S, guaranteeing that the samples have the same probability (N/S) of being stored in the buffer until the buffer size K is exceeded. Once K is exceeded, the samples are added by generating a random number, I, between 1 and J, where J is the index of the input sample. If $I < K$, the sample is added to the buffer.

7.8 Efficacy of the modified cosine similarity

In CoRaL, we proposed the use of the modified cosine similarity (Eq. 7.9, 7.10) for Continual learning. Our objective function is an extension to the standard cosine similarity (Eq. 7.9), as we introduced a predictor and also froze the representation for one of the networks at any one time. Thus, we evaluate the efficacy of our modified cosine similarity by removing the use of the predictor network and using a standard cosine similarity objective.

Results: Tab. 7.4 reports the accuracy averaged over 10 runs for both IL-Task and IL-Class on the S-CIFAR-10 dataset. The results suggest that our CoRaL with modified Cosine Similarity outperforms CoRaL with standard Cosine Similarity.

Discussion: The results in Tab. 7.4 suggest that CoRaL with modified Cosine Similarity achieved significant performance improvements over the standard Cosine similarity, especially in IL-Class settings. The use of the online network (with the predictor) and target network that are trained using the modified cosine similarity loss, allowed CoRaL to learn representations that are more robust to changing distributions.

Table 7.5: Hyper-parameters for CoRaL

Dataset	Parameter	Values
S-CIFAR-10	lr	0.05
	α	0.40
	β	0.10
S-TinyImageNet	lr	0.03
	α	0.50
	β	0.50
P-MNIST	lr	0.10
	α	0.10
	β	2.00
R-MNIST	lr	0.20
	α	0.50
	β	0.50

$$\mathcal{L}_{cos}(p_1, z_2) = -\frac{p_1}{||p_1||_2} \cdot \frac{z_2}{||z_2||_2} \tag{7.9}$$

$$\mathcal{L}_{cos\_sym} = \frac{1}{2}\mathcal{L}_{cos}(p_1, sg(z_2)) + \frac{1}{2}\mathcal{L}_{cos}(p_2, sg(z_1)) \tag{7.10}$$

7.9 Details of the Continual Learning Scenarios

We considered three challenging scenarios for Continual Learning, which are illustrated in Fig. 7.2.

- **Incremental Task (IL-Task)**: In this scenario, the classes of the original dataset are divided into equal splits for each task. The network has access to the task label t. This implies that for every sample the network can observe the (input, label, task) tuple: (x,y,t), as illustrated in Fig. 7.2. A typical network architecture used in this scenario has a "multi-headed" output layer, where each task has its output units. However, the network backbone is shared across all tasks.
- **Incremental Class (IL-Class)**: Similar to IL-Task, the original dataset is divided into equal splits. However, models do not have access to the task label and need to perform both the classification of new samples as they arrive and infer the change in task. For each sample, the network can observe the (input, label) tuple: (x,y). A typical network architecture used in this scenario has a "single-headed" output layer and a shared backbone as shown in Fig. 7.2. The output logit can potentially be expanded upon the arrival of new classes.
- **Incremental Domain (IL-Domain)**: In this scenario, models do not have access to the task label and need to classify the input images, which may undergo perturbations. A typical network architecture used in this scenario has a "single-headed" output and a shared backbone. As depicted in Fig. 7.2, the number of classes remain the same, but the input distribution, and hence the task, is changing.

7.10 Hyper-parameters in CoRaL

We report the list of common notations used in our formulation in Tab. 7.3. We also report the hyper-parameters that provided the best accuracies across different datasets in Tab. 7.5.

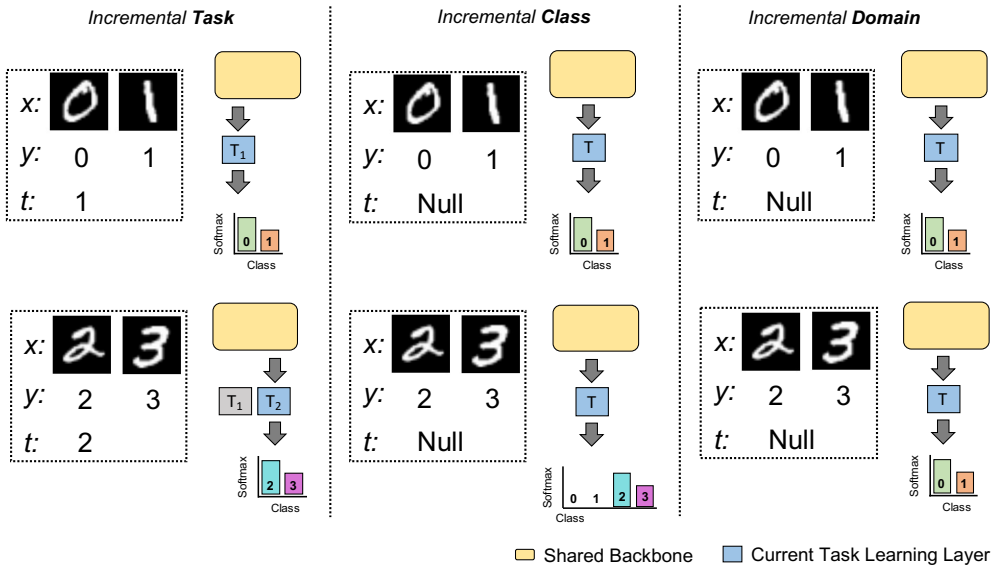


Figure 7.2: The three types of Continual Learning scenarios that were considered in our experiments. For Incremental Task models have access to task labels and hence can use task-learning layers. For Incremental Class, models do not have access to task labels and have one common task learning layer. For Incremental Domain, models also have the one task learning layer. The number of classes does not increase for Incremental Domain, but the inputs undergo perturbation.

7.11 Training Environment

All experiments were conducted using Pytorch-1.6, running on Nvidia RTX-6000 GPUs and Cuda 10.2. We used SGD optimizer for training CoRaL across all scenarios and datasets, with the learning rate reported in Tab. 7.5 for each dataset.

7.11.1 Evaluation Protocol

To ensure fair evaluation, we used a similar learning schedule for all evaluated methods and conducted a hyper-parameter search to ensure the best average accuracy. We compared CoRaL with state-of-the-art approaches that use a similar end-to-end training scheme. As such, we did not evaluate against techniques that require two-phases of training or relax the non-i.i.d assumption of CL by using class-balancing strategies[275, 286] . For the S-MNIST and R-MNIST datasets, we trained all methods for one epoch per task, using a mini-batch size of 128 following prior work [258]. For the S-CIFAR-10 and S-Tiny-ImageNet datasets, we used a mini-batch size of 32 and trained for 50 epochs per task, following prior works [249, 258]. We used a memory buffer of 200 samples using reservoir sampling. For all scenarios, the evaluation metric is the test-set accuracy after being trained on all the tasks (Acc.), averaged over ten independent runs.

7.12 Results and Discussion

7.12.1 Incremental Task

Results: We present the average accuracy over ten independent runs of all frameworks on IL-Task settings for the Split-CIFAR-10 (S-CIFAR-10) and Split-TinyImageNet (S-Tiny-ImageNet) datasets in Tab. 7.1. The results suggest that CoRaL outperformed all other methods on both the evaluated datasets. CoRaL achieved the highest average accuracy of 92.01% and 41.37% on S-CIFAR-10 and S-Tiny-ImageNet, respectively, while having low standard deviation.

Discussion: The results in Tab. 7.1 suggest the efficacy of the Representation Learning module in learning robust representation. The Representation Learning module increases the similarity between the positive samples, whereas the Knowledge Distillation module replays past samples and allows the network to optimize over them simultaneously. CoRaL achieved a performance improvement of 0.09% and 0.50% on S-CIFAR-10 and S-Tiny-ImageNet, while having a relatively low standard deviation, suggesting consistency in the results and the stability of the objective function in Eq. 7.7. Although we report methods that have an architectural expansion, such as PNN [251], it is not a fair comparison as PNN progressively adds a new learning network for each task, incurring significant memory overhead. In contrast, our work does not require a new network for each task and maintains the same buffer size as other approaches.

We also observed in Tab. 7.1 the effectiveness of rehearsal-based approaches (FDR,ER, DER++, CoRaL) compared to regularization-based approaches (oEWC, SI, LwF) over both the datasets. This is due to the network having access to past data samples and optimizing over them as well as the current data samples, providing a more effective way of recalling past representations. Moreover, regularization-based approaches add a penalty to parameter updates, which constrains the network from learning new tasks. As a result, the number of unregularized parameters decreases with each task, which leads to relatively low average accuracy.

7.12.2 Incremental Class

Results: We present the average accuracy over ten runs of all methods on IL-Class settings for the S-CIFAR10 and S-Tiny-ImageNet datasets in Tab. 7.1. The results suggest that CoRaL outperformed all other methods, further highlighting CoRaL's ability to mitigate catastrophic forgetting.

Discussion: The results underline the generalizability of CoRaL to different scenarios and posit a strong case for Representation Learning frameworks in CL. CoRaL's improved representation learning allows the encoder to learn a more robust representation, which along with the distillation loss, allows it to attain the best performance. As observed in Tab. 7.1, CoRaL significantly outperformed all other approaches by 2.19% in terms of average accuracy on the S-Tiny-ImageNet dataset. This provides empirical evidence of the benefit of the Siamese Network setup, which leads to a more robust Representation Learning under non-stationary distributions. Furthermore, the low standard deviation leads to more consistent results.

We observed that IL-Class presents a more significant challenge for all CL frameworks with a performance drop compared to IL-Task. This is because there are no task boundaries for IL-Class, leading to only one task-learning layer. This means that models need to *infer* the current task in addition to classifying the inputs, making it more challenging to recall knowledge over past *inferred* tasks.

7.12.3 Incremental Domain

Results: We present the average accuracy over ten runs of all models on IL-Domain settings for the P-MNIST and R-MNIST datasets in Tab. 7.1. Consistent with previous CL-scenarios, the results suggest that CoRaL achieved the highest average accuracy on both the datasets, further highlighting its superiority for addressing catastrophic forgetting in IL-Domain settings. On average, CoRaL outperformed all other approaches by 1.02% on the P-MNIST dataset and 1.36% on the R-MNIST dataset.

Discussion: The results reinforce the benefits of the Representation Learning module in CoRaL. As IL-Domain introduces input perturbation, the addition of Representation Learning is particularly effective as it explicitly directs the model to reduce the distance between samples that have undergone different perturbations but belong to the same class. This is not available in other evaluated approaches, which try to distinguish between these perturbations using the Cross-Entropy or other distillation losses, which are not explicitly targeted toward learning robust representations.

7.12.4 Backward Transfer

Results: Lastly, we present the average Backward Transfer (BWT) of all models, which is calculated in line with prior works [254, 258]. BWT is expected to increase with new tasks as the network no longer has access to all the data samples of the past tasks and is a good estimator for *catastrophic forgetting*. As such, we chose the CL scenario with the highest number of tasks: IL-Domain. We present the results for

Table 7.6: Impact of Representation Learning techniques.

Approach	IL-Task		IL-Class	
	S-CIFAR-10	S-Tiny-ImageNet	S-CIFAR-10	S-Tiny-ImageNet
CoRaL with CrL (MoCo)	90.52 ± 0.51	35.88 ± 1.61	61.20 ± 1.02	11.16 ± 1.07
CoRaL with CSL (SimSiam)	**92.01 ± 0.32**	**41.37 ± 0.91**	**65.24 ± 1.09**	**14.06 ± 0.57**

all the methods in Tab. 7.2 for the P-MNIST and R-MNIST datasets. For BWT, a negative value indicates forgetting, and as such, a lower negative value is desirable. As can be observed, CoRaL attained the lowest BWT, with an average BWT of -9.92% and -4.65% on P-MNIST and R-MNIST, respectively.

Discussion: The results in Tab. 7.2 highlight that CoRaL attained the lowest BWT over all approaches on both datasets. For the P-MNIST dataset, CoRaL outperformed all baselines by 1.55%, and for R-MNIST, CoRaL attained the lowest BWT, outperforming all approaches by 0.62% on average. This suggests that the augmentations introduced for the Representation Learning module allow CoRaL to learn more robust features resulting in less forgetting.

7.13 Ablation Study

7.13.1 Analyzing Different Representation Learning Approaches

Continual Learning requires frameworks to strike the right blend of stability and plasticity when learning on continuous data streams. Such frameworks strive to be *stable* to changing data distributions, retaining information on past tasks while exuding the requisite plasticity to learn new tasks efficiently. In this work, we presented a general framework for investigating the effectiveness of Representation Learning frameworks under non-stationary distributions. To assess the applicability of current Representation Learning frameworks, we compare two popular approaches: MoCo [260] and SimSiam[263], while fixing the parameters of the Knowledge Distillation module.

Results: Tab. 7.6 reports the average accuracy after ten runs on the S-CIFAR-10 and S-Tiny-ImageNet datasets. We conducted our analysis for both IL-Task and IL-Class scenarios for extensive evaluation. We compared two conceptually different approaches, *SimSiam* [263] and *MoCo* [260]. SimSiam is a Siamese Network setup trained using the negative symmetric cosine similarity. On the other hand, MoCo is also a Siamese setup, where one of the encoders is a momentum encoder, and the other is a standard encoder. MoCo is trained using the Contrastive loss.

Discussion: The results in Tab. 7.6 suggest that SimSiam, which is the approach used in this paper, outperformed MoCo for all the datasets and scenarios. The improvement is especially significant for IL-Class, with 4.04% and 2.90% gain for S-CIFAR-10 and S-Tiny-ImageNet, respectively. We posit that this is due to the Cosine Similarity objective, which does not rely on negative samples but tries to maximize the similarity between two augmentations. On the other hand, MoCo uses a momentum encoder along with a feature queue to maintain a consistent queue of negative samples, which is optimized using the InfoNCE loss (Eq. 7.3). As is the case with Continual Learning, the distribution for the negative samples keeps changing, making it challenging for the network to learn stable representations. The results justify the use of the Cosine Similarity loss for Representation Learning in CL, which provides CoRaL with the ideal blend of stability and plasticity when learning new tasks.

7.13.2 Impact of CoRaL's Learning Modules

Results: We extensively experimented across different scenarios and datasets to assess the importance of the two primary learning modules of CoRaL. Tab. 7.7 presents the accuracy while ablating a specific module, given the same encoder network and learning protocols.

Discussion: First, we ablate the Representation Learning module, i.e., we no longer have a Siamese Network setup. The Cosine Similarity loss is removed from the objective function, which is now comprised of only

Table 7.7: Ablation results (top-1) over different learning modules.

Method	IL-Domain		IL-Task	IL-Class
	P-MNIST	R-MNIST	S-CIFAR-10	S-CIFAR-10
CoRaLw/o R.L.	83.45	90.19	91.23	59.43
CoRaLw/o K.D.	41.99	69.45	72.92	19.67
CoRaL	**85.60**	**92.89**	**92.49**	**66.97**

Table 7.8: Effect of varying the plasticity parameter (α) on the average accuracy (after 5 independent runs) for S-CIFAR-10.

α	β	IL-Task	IL-Class
0.1	0.1	90.33 ± 1.22	63.80 ± 1.20
0.2	0.1	90.69 ± 0.91	64.19 ± 0.14
0.3	0.1	91.72 ± 0.29	65.31 ± 1.24
0.4	0.1	**92.10 ± 0.12**	**66.05 ± 0.71**
0.5	0.1	91.25 ± 0.79	63.01 ± 1.10
1.0	0.1	90.63 ± 0.89	62.69 ± 0.77

the Cross-Entropy and the distillation learning loss. The results in Tab. 7.7 suggest that the absence of the Representation Learning module results in a drop in accuracy across all scenarios and datasets. The drop is most significant in IL-Class scenarios (7.54% for S-CIFAR-10), which is the most challenging of all CL scenarios, asserting the importance of learning robust representations that are transferable.

We next ablate the Knowledge Distillation module, removing the distillation loss from the objective function. The results suggest that removing the Knowledge Distillation module has a significant impact on overall performance. There is a significant drop in accuracy for all scenarios and datasets, with the framework suffering most in IL-Class. This reinforces the importance of using a memory buffer which allows the network to retain its knowledge over past tasks. Our results also highlight that combining the Representation Learning and Knowledge Distillation modules provide the right balance between stability and plasticity, with the combination attaining the best performance.

7.14 Analysis of the Stability-Plasticity

We conducted extensive experiments to evaluate the effect of stability and plasticity on the average accuracy. We achieved this by varying the values of α and β of the objective function in Eq. 7.7.

7.14.1 Effect of Varying the Plasticity on the Accuracy

Results: We varied the weight of the Cosine Similarity Loss, α, keeping the weight for the Knowledge Distillation loss, β fixed at 0.1. Varying the α provides us the flexibility of increasing or decreasing the plasticity of CoRaL and allows us to assess the subsequent impact on the forgetting. Tab. 7.8 presents the average accuracy after all five tasks on the S-CIFAR-10 for the IL-Task and IL-Class scenarios. In addition, we also tracked the average accuracy after learning each new task, as shown in Fig. 7.3 for different values of α.

Discussion: The results in Tab. 7.8 suggest that increasing the plasticity of the framework initially allows CoRaL to learn robust representations with the highest accuracy for IL-Task at $\alpha = 0.4$. However, with a further increase in the plasticity, the accuracy drops, suggesting the need to constrain the plasticity of the framework in order to improve the stability of the learned representations. The trend is also similar for IL-Class, with the accuracy increasing initially, with the best value at $\alpha = 0.4$.

When we track the average accuracy after each task in Fig. 7.3, we see that increasing the plasticity results in the network attaining higher average accuracy for the initial tasks, especially for Task 1 and 2, where higher values of ($\alpha = 0.7 \sim 1.0$) led to higher accuracy for both IL-Task and IL-Class. However, as Continual

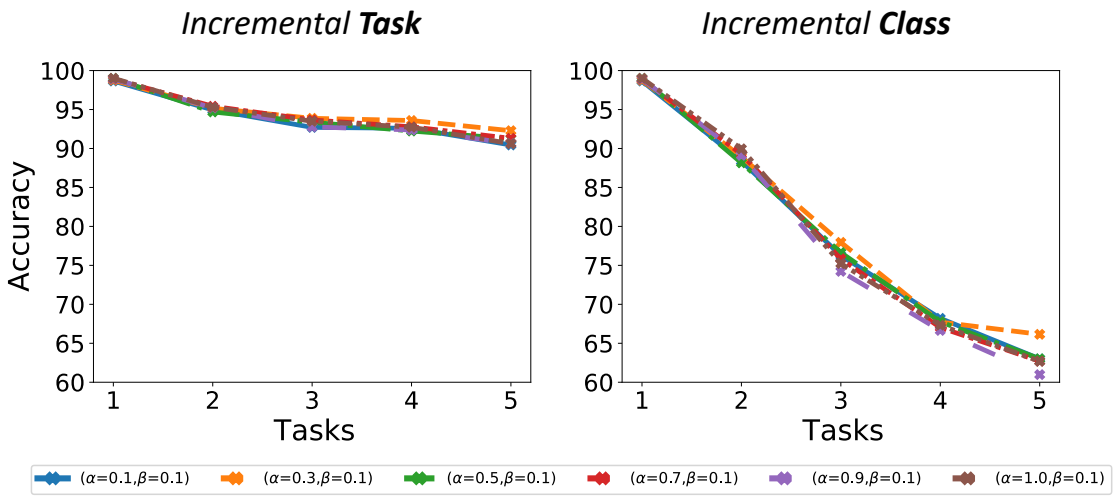


Figure 7.3: Plasticity Analysis on the S-CIFAR-10 dataset. We varied the weight (α) for the Representation Learning module

Learning requires the network to retain past knowledge, a high plasticity coefficient may result in higher forgetting. As seen in Fig. 7.3, the average accuracy for later tasks decreases at a faster rate, for high values of α. As such, there needs to be a trade-off between high plasticity, which may provide better accuracy initially, and moderate plasticity, which may provide better accuracy at later stages.

7.14.2 Effect of Varying the Stability on the Accuracy

Results: We varied the weight of the Knowledge Distillation Loss, β, keeping the weight for the Representation Learning loss, α fixed at 0.1. This provided us with a mechanism to tune the stability of our framework, with a higher value of β resulting in stronger optimization constraints when updating the network parameters. Tab. 7.9 presents the final accuracy after all five tasks on the S-CIFAR-10 for the IL-Task and IL-Class scenarios. In addition, we also tracked the average accuracy after learning each new task, as depicted in Fig. 7.4 for different values of β.

Discussion: The results in Tab. 7.9 suggest that increasing the stability parameter β initially allows CoRaL to put optimization constraints when learning new tasks and results in improved knowledge retention over past tasks. The Distillation Loss $\mathcal{L}_{distill}$ acts as a regularizer during the parameter update while replaying the past samples also relaxes the non-i.i.d assumption. The highest accuracy for IL-Task was at $\beta = 0.3$, whereas the highest accuracy for IL-Class was at $\beta = 0.2$. However, with a further increase in the stability ($\beta > 0.3$), the accuracy drops for both IL-Task and IL-Class, suggesting over-regularization.

When we track the average accuracy after each task in Fig. 7.4, we see that increasing the stability ($\beta = 0.1 \sim 0.3$) leads to the network attaining higher average accuracy. However, a further increase sin β results in over-constraining the network and leads to lower accuracy for all the tasks, as observed for $\beta > 0.3$. The lowest average accuracy after each task and after all five tasks was for $\beta = 1.0$. Interestingly, we also observed that the drop in accuracy after each new task is also lowest for $\beta = 1.0$. The combination of the network attaining low accuracy and low forgetting implies over-regularization, whereby the network is too stable to learn efficiently.

7.14.3 Discussion on the Stability-Plasticity Trade-off

Our experimental results in (Tabs. 7.8, 7.9 and Figs. 7.3, 7.4) underline the challenges of finding the right blend of stability and plasticity to mitigate catastrophic forgetting. An increase in plasticity, by increasing the weight of the Representation Learning loss, $\mathcal{L}_{cos\_sym}$ leads to the network learning transferable features, which results in higher average accuracy over the next task. However, a further increase in plasticity may

Table 7.9: Effect of varying the stability parameter (β) on the average accuracy (after 5 independent runs) for S-CIFAR-10.

α	β	IL-Task	IL-Class
0.1	0.1	90.33 ± 1.22	63.80 ± 1.20
0.1	0.2	90.72 ± 0.72	**65.18** ± 1.09
0.1	0.3	**90.94** ± 0.60	64.77 ± 0.72
0.1	0.4	90.74 ± 1.18	62.11 ± 2.22
0.1	0.5	89.23 ± 1.44	61.88 ± 0.39
0.1	1.0	86.77 ± 0.82	51.18 ± 5.14

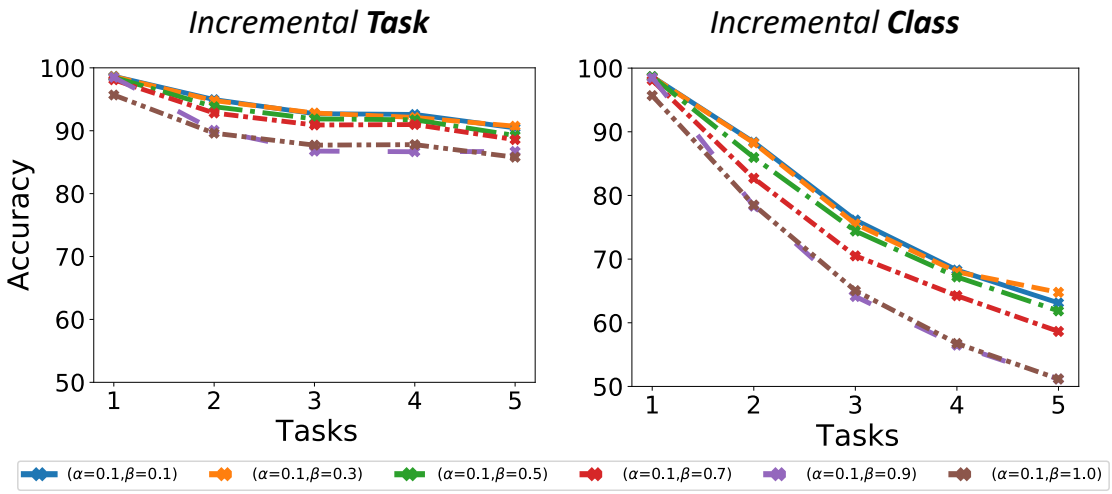


Figure 7.4: Stability Analysis on the S-CIFAR-10 dataset. We varied the weight (β) for the Knowledge Distillation module

lead to drops in accuracy. Similarly, an increase in the stability, by increasing the weight of the Knowledge Distillation loss, $\mathcal{L}_{distill}$, can improve the knowledge retention of the network by acting as a regularizer, up to a certain value. Further increase in the stability parameter might constrain the network from updating its weights, resulting in lower average accuracy.

7.15 Conclusion

In this work, we introduced CoRaL, a novel Continual Learning framework for addressing catastrophic forgetting. Our framework provides the right blend of stability in CL scenarios through the Knowledge Distillation module and plasticity via the Representation Learning module, thus providing a promising approach for intelligent agents to learn continually. CoRaL is trained end-to-end with a novel objective function that comprises the modified Cosine Similarity loss and the Distillation loss on top of the Cross-Entropy loss. Our results across three scenarios and four datasets suggest the efficacy of CoRaL, with our proposed approach outperforming all other techniques on all evaluated benchmarks. The ablation studies further validates the relevance of the Cosine Similarity loss for Continual Representation Learning and CoRaL's two proposed modules.

Chapter 8

LASSO: Learning Policies via State Space Modeling

The ability to perceive and react to changes in dynamic settings is a crucial step for robots to achieve autonomy. However, this remains an open challenge due to the difficulty of learning a robust representation of the environment, especially if the dynamics are non-stationary. To address the challenge of learning a robust policy for dynamic conditions, we propose LASSO, a novel algorithm that focuses on learning a representation that can capture the stochasticity of the environment, which is then used to learn a policy. We propose a Representation Learning module that uses an encoder-decoder architecture to learn a latent space that can encode the future states of the environment. The encoder-decoder framework functions in tandem with a Siamese Learning framework that measures the similarity between the current state and the desired goal state and works on maximizing this similarity. These representations are then passed to a Policy Learning module, which aims to learn a policy conditioned on these features. We evaluated LASSO by comparing its performance against state-of-the-art methods in static and dynamic-goal environments. The results suggest that LASSO outperformed all evaluated methods in all environments tested, achieving the highest success rate. Finally, we conducted extensive ablation studies, and the results underscore the importance of our proposed learning modules.

8.1 Introduction

Recent advances in machine intelligence and learning have significantly enhanced robot perception and decision-making, enabling their adoption across a variety of applications from healthcare and manufacturing settings to autonomous vehicles [89, 72, 293, 70, 137]. Despite significant advances in robot learning, robots' capabilities and actions are often limited in scope and require strong assumptions about the environment. As such, robots are mostly confined to their proverbial cage, limited by their inability to model the stochastic nature of dynamic environments [21]. For robots to become fully autonomous and reliable, they need to perceive and adapt to the changes in environmental dynamics [102, 103, 15]. Along this line, progress has been made in robot perception in detecting changes and generalizing to new environments [74, 78, 75, 76]. However, adapting to dynamic environments remains an open challenge for robot decision-making, and closed-loop control [294].

For example, consider a scenario where a robot needs to pick up a container from a conveyor belt and place it at a predefined location. For the robot to perform this pick and place task reliably, it needs to obtain a robust representation of the underlying dynamics of the environment (see Fig. 8.1). The robot needs to anticipate the position of the container before executing any pick action, given that it is constantly moving on the conveyor belt. At the same time, the robot needs to obtain a measure of the distance between its current state and desired final state in the representation space to accurately plan a course of action to reach the end goal. This knowledge about the environmental dynamics and the final state would allow the robot

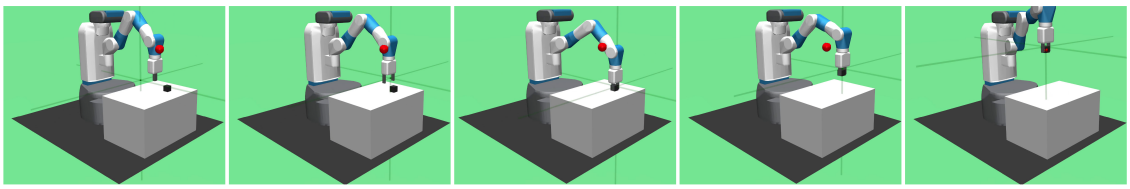


Figure 8.1: A Dynamic-goal Environment. In this environment, the final goal position (marked in red) is constantly being updated, which requires constant re-planning for the agent in order to reach the goal position successfully.

to achieve the desired task in a consistent and reliable manner.

One possible mechanism to achieve closed-loop decision-making is learning from experience through reinforcement. Prior works on Reinforcement Learning (RL) have significantly furthered the state-of-the-art decision-making in autonomous agents [295, 296], and notably improved robot decision-making on complex tasks ranging from navigation to manipulation [297, 298, 133, 299]. However, these algorithms have been developed and trained in environments where the goal state is static, and only the robot itself can bring changes into the environment, which is not always an accurate depiction of the real world environments [300]. This has motivated an array of techniques such as domain randomization [301, 300] and state abstraction [302] to overcome the simulator-to-real (sim2real) gap in robotics, where algorithms that are trained in simulation do not generalize to the real world. Although these techniques have all contributed to improving the decision-making of autonomous agents and reducing the sim2real gap, they fail to explicitly consider the non-stationary dynamics of the environment [302].

To address the challenges of adapting to non-stationary dynamical environments, we propose LASSO: a novel algorithm that aims to learn a robust representation of the underlying dynamics of the environment, which can then be reliably used for policy learning. LASSO comprises a Representation Learning module and a Policy Learning module. We decouple Policy Learning from Representation Learning, thus allowing for a modular architecture, which is trained with specific objective functions. This modular architecture provides the flexibility of learning a representation that is geared toward obtaining a model of the environment dynamics. The learned representation can then be used by the Policy Learning module as a reliable approximation of the state. LASSO can be viewed as lightweight extension to model-free approaches by learning a robust representation from the experience buffer, similar to prior representation learning approaches such as CURL [63]. Unlike model-based approaches [64, 65] which perform planning on the world model and as such may not require a policy network, LASSO does not propagate stochastic gradients of returns to its learned representation, instead only focusing on the representation to forecast future states.

The Representation Learning module comprises a State-space Forecasting framework and a Siamese Learning framework. The State-space Forecasting framework is an Encoder-Decoder architecture that is trained to predict future states, conditioned on past states. This creates an information bottleneck, which allows the overall framework to learn the features that are most representative of the future states in the form of the latent space. While learning latent features that can summarize future states is essential, estimating the distance between the current state and the final goal state is equally important. To achieve this, we introduce a Cosine Similarity objective to minimize the distance between the current latent representation and the final goal state. This is done by using Siamese Network [259, 260, 288, 287], whereby we feed the current state to one of the networks and the final goal state to the other network. The learned latent representation is next passed to the Policy Learning module.

The Policy learning module comprises an *actor* network, which uses the latent space to learn a policy, and a *critic* network, which evaluates the actor network's policy, thus creating a policy iteration-evaluation loop [58]. We train the Policy Learning module using the maximum entropy reinforcement learning objective [298], which encourages the networks to maximize the expected return and entropy. Thus, there are three specialized objective functions for LASSO: the reconstruction loss, the modified cosine similarity loss, and the maximum entropy reinforcement learning loss.

We evaluated the performance of our approach on the OpenAI Fetch Robotics Environments [297], which provides several challenging manipulation tasks. The original tasks are static with respect to the final goal, i.e., the final goal state does not change. This has prompted us to develop new dynamic goal environments where the final goal state is constantly being updated. We have also evaluated our algorithm on these dynamic goal environments. The results of our experiments suggest that our proposed approach outperformed state-of-the-art reinforcement learning methods over all the evaluated environments for static and dynamic goal settings. Finally, we provide extensive ablation studies which support the benefit of our proposed learning modules.

8.2 Related Work

Representation Learning: Learning rich representations of high dimensional data remains one of the long-standing challenges of machine perception [154, 101]. This is especially important for RL, where sample efficiency is considered one of the biggest barriers to widespread adoption of RL for autonomous agents [294]. Prior works have proposed the use of auxiliary tasks such as forecasting future states or introducing additional reward or control objectives [64, 65, 303], to improve the sample efficiency. Recently, Laskin et al. have proposed CURL [63], which introduced contrastive learning for learning robust representation from pixel data, which can later be utilized for downstream policy learning.

Policy Learning: Recent advances in deep RL have enabled AI agents to achieve remarkable performance on a variety of tasks ranging from chess and go to robotic locomotion and manipulation [296, 304, 305, 10, 133]. Prior works in RL can be broadly categorized to model-free [306, 307, 308, 298, 211] approaches, where there is no assumption about the environment, and model-based approaches [10, 65, 309, 64], where there are some assumptions about the environment.

Model-free RL has been widely applied in robotics due to its simplicity in implementation and a relatively lower load of assumptions compared to model-based RL. As model-free approaches do not involve explicit learning of the transition dynamics and are comparatively sample inefficient compared to model-based algorithms. However, recent works, such as soft actor-critic [298], have proposed the use of maximum entropy reinforcement learning [310] as the training objective, thus encoding the exploration as part of the policy search. This approach has the benefit of improving the agent's exploration and hence sample efficiency. As model-free approaches do not have an explicit model of the environment, they may be prone to overestimating their expected return. To mitigate the over-estimation bias of actor-critic methods, Kuznetsov et al. [211] proposed Truncated Quantile Critics, which truncates the return distribution and uses an ensemble of critic networks to estimate the return.

Although the aforementioned methods have improved the state-of-the-art in robot learning, there remains a gap in generalizing these techniques to dynamic environments, as the majority of these algorithms are model-free and, as such, do not explicitly rely on learning a model of the environment. Furthermore, prior works on representation learning for RL have primarily focused on image-based static environments, which are themselves highly data-intensive, and are not goal-conditioned, thereby not learning the distance between the current observation and goal state. As such, there remains a gap in both representation learning and policy learning, which is manifested every time there is a change in environment dynamics or when these algorithms are introduced to a real-world setting with unknown dynamics [311].

8.3 Problem Formulation

Our goal is to teach the robot a control policy that can generalize to static environments, where the goal position does not change, as well as to dynamic environments, where the goal position is constantly updated. This requires the robot to perceive its environment, plan, and act continuously in order to perform the task successfully. Instead of specifying behaviors that may not generalize if there is a change in the environment, we let the robot learn the behavior on its own. This is achieved using a reward signal that encourages the robot to go to states which will provide a higher cumulative reward.

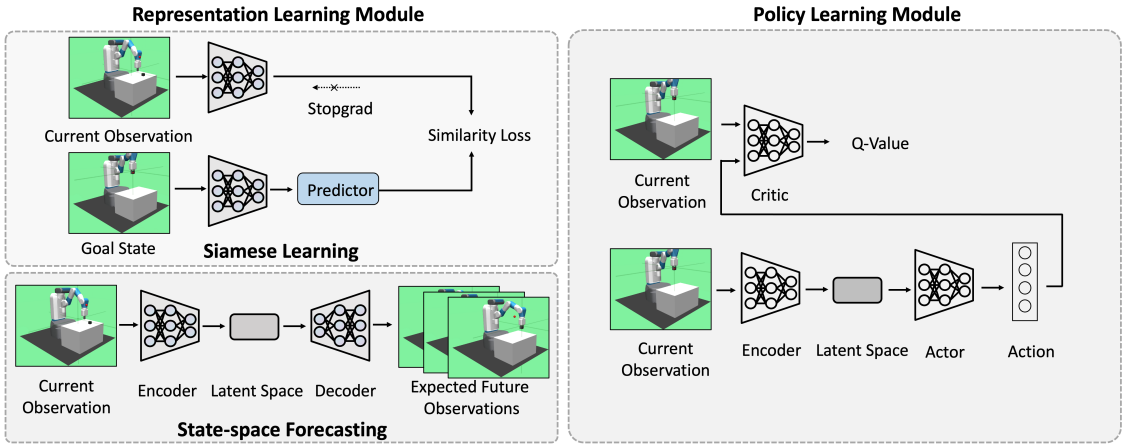


Figure 8.2: LASSO: Learning Policies via State Space Modeling. The Representation learning Module performs two objectives: 1) Learning a representation about the future states using the Encoder-Decoder Framework and 2) Minimizing the distance between the current state and the final goal state using the Siamese Framework. The Policy Learning Module comprises the actor network, which uses the latent representation to learn a policy, and the critic network, which evaluates the Value of this policy.

We formulate the problem as a Markov Decision Process (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma)$. Here, $\mathcal{S}$ is the state space, $\mathcal{A}$ represents the action, $\mathcal{R}$ denotes the reward, $\mathcal{P}$ denotes the probability density of the next state $S_{t+1} \in \mathcal{S}$, given the current state $S_t \in \mathcal{S}$ and action $a_t \in \mathcal{A}$. $\gamma \in [0, 1]$ represents the discount factor. Due to the continuous nature of the problem, the state space $\mathcal{S}$ and action space $\mathcal{A}$ are continuous. The environment provides a bounded reward signal, $r : \mathcal{S} \times \mathcal{A} \rightarrow [r_{min}, r_{max}]$ on each transition.

At each time step t, the agent perceives the state s_t, takes an action $a_t \in A$ that is drawn from a policy $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$, and with probability $p(s_{t+1}|s_t, a_t)$ enters a new state s_{t+1}, receiving reward $r(s_t, a_t)$ from the environment. Here π represents the mapping between the states s_t to a probability distribution, which emits the action a_t. The overall goal can then be formulated as finding the optimal policy π^* that achieves the maximum expected return.

8.4 LASSO: Learning Latent Policies via State Space Modeling

LASSO comprises a Representation Learning module and a Policy Learning module (Fig. 8.2). The Representation Learning module aims to learn a prior about the environment. It then uses this prior and information about the goal to obtain a robust representation that can be used for learning a policy. The Policy Learning module aims to learn an optimal behavior that maximizes the cumulative reward and entropy.

8.4.1 Representation Learning Module

State-space Forecasting

In order to develop a prior about the environment, we propose an encoder-decoder architecture, which is tasked to forecast the future states, given the observed state and the goal. The goal may remain static for static environments or may change in dynamic environments. The input to the encoder-decoder architecture is the state and the goal, with the expected output being the forecasted states. This can be formulated as follows:

$$p_\theta(\hat{\mathbf{Y}}) = \prod_{\delta=\tau+1}^{\tau+H} p_\theta(\hat{y}_\delta | \hat{y}_{\tau:\delta-1}, x_{1:\tau}) \tag{8.1}$$

Here, $\hat{Y}$ represents the predicted future (or forecasted) states, $x \in X$ represents the past observed state(s).

Encoder: The encoder aims to learn a representation over the current state and goal. As we pose this as a sequence learning problem, we employ Recurrent Neural Networks, in particular, unidirectional Gated Recurrent Units (GRU) in the encoder. Our choice of unidirectional GRUs over a bi-directional architecture is motivated by the need to predict future states in real time. The functions at the encoder can be formulated as:

$$h_{s,t} = GRU(h_{s,t-1}, x_{s,t}, \phi_{enc}) \tag{8.2}$$

where s represents the current state and goal space. $h_{s,t-1}$ represents the past hidden output and ϕ_{enc} represents the encoder weights for the GRU.

Latent Space: The output of the encoder is passed through a linear layer to obtain the latent space. This latent representation is used as the input for several components of the overall architecture: a) input to the decoder, b) input to the predictor network for Siamese Learning, and c) input to the actor-network for Policy Learning. We did not enforce a prior on the latent space, as that may interfere with learning.

Decoder: The decoder is auto-regressive, i.e., it uses the output of previous timesteps to predict the current state. The input to the decoder is the latent space, summarizing the past state and goal, and the last predicted state.

The first part of the decoder is a GRU cell that takes as input the latent representation as well as the output of the last timestep. This is followed by a fully connected layer. The operations at the decoder are formulated as follows:

$$\begin{aligned} h_{dec,t} &= GRU(S_{t-1}, h_{latent,t}, \phi_{dec}) \\ S_t &= \gamma(h_{dec,t}) \end{aligned} \tag{8.3}$$

where $h_{latent,t}$ is the latent representation summarizing the past state(s) and the goal, S_{t-1} is the previous output of the decoder. γ represents the output layer of the decoder with $\mathbf{S_t}$ being the *predicted state at time* t. ϕ_{dec} represents the decoder weights. We add a residual connection between the output of the previous and current timesteps to improve the smoothness of the output sequence.

Siamese Learning

While having knowledge about the forecasted states, conditioned on the current state and goal, can aid in learning a robust policy, there is also a need to learn the distance between the current state and the desired goal state. To obtain this distance in representation space, we introduce a Siamese Learning setup, where we have two versions of the network. For both versions, we re-use the exact same encoder as the previous step. One of the networks is fed the current state and achieved goal, whereas the other network is fed the current state and desired goal. If the achieved goal and desired goal represent the same state, the representation distance between the two should be zero, or the similarity of the representation should be 1.

Contrastive learning [259, 260] have proven to be an effective technique for learning instance discrimination without labels [312, 313, 314, 315]. The core idea behind these works is the following: for every input, minimize the distance between the positive sample pairs and maximize the distance between the negative sample pairs. The positive sample pairs are the embeddings of the two augmented versions of the input, while all other embeddings are considered negative.

While this formulation has proven to be effective in learning instance discrimination in the absence of labels, methods based on this contrastive formulation are sensitive to the choice of data augmentations [287] and can be prone to distribution shifts. As the robot observes new states using the exploration-exploitation trade-off [58], the distribution of states is non-stationary, which makes the traditional objective function of contrastive learning ill-posed for our problem. Thus we introduce the modified Cosine Similarity loss [263], which only relies on the positive samples. The task then reduces to maximizing the similarity between the current state and achieved goal, and the final goal state that the robot needs to reach. As such, our objective function is:

$$\mathcal{L}_{cos}(p_1, z_2) = -\frac{p_1}{||p_1||_2} \cdot \frac{z_2}{||z_2||_2} \tag{8.4}$$

Here, z_1, z_2 are the embeddings of the encoder. p_1, p_2 are non-linear transformations of z_1, z_2.

Siamese Network Setup: Here, we describe our proposed Siamese Network setup, which is trained using the modified Cosine Similarity loss, as described in Eq. 8.4. The Representation Learning module is a Siamese Network setup, comprised of one encoder f_θ, one projection network d_θ, and a prediction network

h_θ, in line with prior work on representation learning [287, 263]. Inspired from BYOL [287] and SimSiam [263], our Siamese Network setup has one online network and one target network. The target network (f_θ & d_θ) provides the regression targets to train the online network, which is then used to update the gradients of the online network (f_θ, d_θ, & h_θ). This update is then followed by swapping the roles, i.e., the previously online network is now the target network and has to provide the regression targets, leading to two passes of optimization.

The input to the module are two views of the state: current state and achieved goal x_1, and current state and the final goal x_2. We use the same encoder and projection network for both online, and target networks, which is similar to SimSiam [263] and SimCLRv2[259]. For a given task t, the input are the two views x_1, x_2, which is processed by the encoder network f_θ, followed by the projection network d_θ to get two embeddings z_1, z_2. For a given pass, the prediction network h_θ processes one of the embeddings, for example, z_1, to output p_1 and matches it to the other embedding, z_2.

The outputs, p_1 and z_2, are normalized before calculating the Cosine Similarity loss (Eq. 8.4). Our framework uses symmetric loss, whereby we update the gradient of one network in one pass and then update the gradient of the other network in the next pass. In either pass, only the online network (one with the predictor) is updated end-to-end using backpropagation. The target network is not updated and is tasked to provide regression targets. The symmetric loss is an extension of Eq. 8.4 and can be formulated as follows:

$$\mathcal{L}_{cos\_sym} = \frac{1}{2}\mathcal{L}_{cos}(p_1, sg(z_2)) + \frac{1}{2}\mathcal{L}_{cos}(p_2, sg(z_1)) \tag{8.5}$$

Here, sg refers to the stopgrad function and is used to prevent the target network from getting updated [263], meaning that the encoder on x_2 receives no gradients from z_2 in the first term and the encoder on x_1 receives no gradients from z_1. This provides the encoder a prior of the final goal position, which can be static or dynamic, depending on the task.

8.4.2 Policy Learning

Having learned a robust representation of the environment, the agent now needs to use the representation to learn an optimal policy, that would provide maximum cumulative reward, while also maximizing the entropy. The representation can be considered a prior, as it contains information about the future state due to the state-forecasting module. In addition, it also learns an implicit distance metric between the current state and the goal state, due to the Siamese Network setup. We aim to use this prior to exploit any dynamics of the environment, which may provide better performance.

For learning a policy, we use the soft-actor-critic framework [298] that comprises an actor network [58] that is tasked to output an action, and a critic network that evaluates the action of the actor by computing the value function. The learning of the actor is based on the policy gradient approach, which aims to directly learn a policy from the observed states.

Critic Network: The critic network is tasked to predict the *Q-Value*, conditioned on the current state, and goal. As such, the input of the critic is the state, goal and action. The output is the expected Q-value, which is learned by iteratively minimizing a sequence of loss functions, where for the i^{th} step, the loss function can be defined as:

$$\mathcal{L}_i(\theta) = \mathbb{E}(r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) - Q(s, a; \theta)^2) \tag{8.6}$$

Here, θ represents the parameters of the critic, s, a represents the current state and action tuple, whereas s', a' represents the state, action tuple of the next timestep.

Actor Network: The actor network aims to learn a policy π_ϕ, conditioned on the latent representation. We use a stochastic actor, which samples actions from policy π_ϕ and is trained using the maximum-entropy RL objective [298]:

$$\mathcal{L}_i(\phi) = \mathbb{E}_{a\sim\pi}[Q^\pi(s, a) - \alpha \log \pi_\phi(a|s)] \tag{8.7}$$

where the actions a are sampled stochastically from the policy $a_\phi(0, \eta) \sim \tanh(\mu_\phi(s) + \sigma_\phi(s) \odot \eta)$ and $\eta \sim \mathcal{N}(0, I)$ is a Gaussian distribution, with 0 mean and I variance.

Table 8.1: Success Rate on static environments (Higher is better).

Approaches	S-Reach	S-PnP	S-Push	S-Slide
DDPG	1.00	0.93	1.00	0.75
SAC	1.00	1.00	1.00	0.82
REDQ	1.00	0.11	0.40	0.05
CURL	0.30	0.30	0.40	0.10
LASSO (Ours)	**1.00**	**1.00**	**1.00**	**0.91**

8.5 Experimental Setup

8.5.1 Evaluation

We evaluated the performance of our approach, LASSO, and baselines on the OpenAI Fetch Robotics environments [297]. While these environments provide a benchmark for evaluating the algorithms in static-goal settings, they lack the stochasticity of dynamic-goal environments, where the final goal position may constantly be moving. As such, we augmented the static environments and created dynamic environments by constantly moving the goal positions. For all the tasks, we measured the performance as the number of times the algorithm has performed the task successfully, i.e., the success rate. In addition, we measure the sample efficiency of each algorithm which is defined as the number of epochs within which the algorithms converge.

8.5.2 Environments

We extensively tested all the evaluated algorithms on the static environments, where the final goal state remain unchanged, and the newly proposed dynamic environments, where the final goal state is constantly moving.

Static environments

In static environments, the goal positions do not move.

Static Reach (S-Reach): In this task, the robot has to reach a goal position, which is randomly generated and can assume any position of the x-y-z plane.

Static Slide (S-Slide): In this task, a puck is placed on a table, and the goal position is outside of the end-effector's reach. The robot needs to hit the puck with the appropriate force and direction for it to reach the goal position.

Static Push (S-Push): In this task, a cube is placed on a table, and the goal position is within end-effector's reach. The robot has to push the cube towards the goal position.

Static PickAndPlace (S-PnP): In the S-PnP task, the robot needs to pick a cube and then place it to a goal position.

Dynamic environments

In these environments, the goal position is constantly moving. We build these environments on top of the static ones. Similar to the static environments, the initial goal position is randomly sampled. However, we introduce stochasticity in the final goal position by moving it at each timestep through a linear translation in the x-y axes. We also put an additional constraint to ensure that the goal position does not move towards the robot.

Dynamic Reach (D-Reach): Similar to the static reach task, the robot needs to reach the goal position. However, the goal position is constantly moving. We update the goal position using a translation function in the x-y axis.

Dynamic Slide (D-Slide): In this task, the robot has to hit the puck toward the goal position. However, the goal position is moving and similar to the static case, it is out of reach of the robot's end-effector.

Table 8.2: Success Rate on dynamic environments (Higher is better).

Approaches	D-Reach	D-PnP	D-Slide
DDPG	1.00	0.93	0.09
SAC	1.00	0.95	0.70
REDQ	1.00	0.11	0.04
CURL	0.10	0.30	0.10
LASSO (Ours)	**1.00**	**0.98**	**0.71**

Dynamic PickAndPlace (D-PnP): In this task, the robot has to grasp the cube and place it toward the goal position, which is constantly moving along the x-y axes.

8.5.3 State-of-the-Art Methods

For evaluating LASSO, we compared against four state-of-the-art approaches: Deep Deterministic Policy Gradient (DDPG) [297], Soft Actor-Critic (SAC) [316], Randomized Ensembled Double Q-Learning (REDQ) [317] and Contrastive Unsupervised Representations for Reinforcement Learning (CURL) [63]. All of the approaches use the actor-critic architecture with hindsight experience replay. In the case of DDPG, the actor network is deterministic and the objective function is to maximize the expected return. For SAC, CURL and REDQ, the actor network is stochastic and the objective function is based on the maximum entropy reinforcement learning function. As CURL uses Contrastive Learning and was primarily proposed and evaluated for raw pixel-based inputs, we had to adopt it for our state-space based modeling. This involved modifying the image-based encoder for state-space input, and removing the image-based augmentation techniques that CURL used.

8.5.4 Learning Architecture Details

Our approach is divided into two modules: Representation Learning and Policy Learning. In the Representation Learning module, we use the same encoder for both the State-space Forecasting and the Siamese Learning framework. The output of the Representation Learning module is used as input by the Policy Learning.

The encoder comprises a GRU cell and a Linear layer. The hidden state dimension of the GRU cell is 128 and the hidden size for linear layer is 64. We use dropout regularization for the GRU cell with dropout probability of 0.1. For the case of the decoder, we reuse the GRU cell at the encoder. For the case of the Policy Learning module, we used a hidden state dimension of 256 for both the actor and critic network. We have provided further details about the learning architecture in the supporting video.

8.6 Results and Discussion

8.6.1 Static Environments

Results: We present the success rate of all approaches on static environments in Table 8.1. For the task of S-Reach, we tested each approach on 20 separate runs. For S-Push and S-Slide, we tested each approach on 80 runs. Finally, for S-PnP, which contains the highest possible variations due to the number of pick and place positions, we tested each approach 120 times. We report the average success rate of all the runs in Table 8.1. The results in Table 8.1 suggest that our proposed approach attained the best performance across all static environments. Furthermore, we illustrate how the success rate of each of the evaluated algorithms changes over time in Fig 8.4. The results in Fig 8.4 suggest that LASSO had the best overall sample complexity in the S-PnP environment Fig 8.4 (Left), and consistently outperformed other algorithms for the S-Slide environment Fig 8.4 (Right).

Discussion: Our proposed method obtained state-of-the-art results across the evaluated static environments. The results highlight the benefit of LASSO's Representation Learning module, which outputs the latent space that can be used to learn a robust policy. For instance, in the S-Slide environment, having the

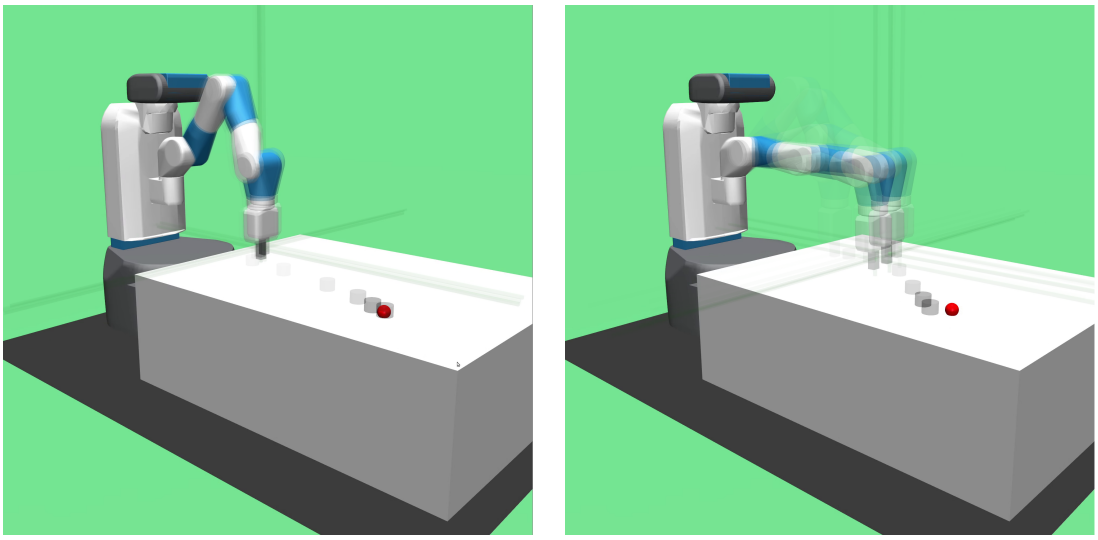


Figure 8.3: Qualitative comparison between LASSO (left) and SAC (right) on Static Slide. LASSO can successfully complete the task due to its ability to model the environmental dynamics.

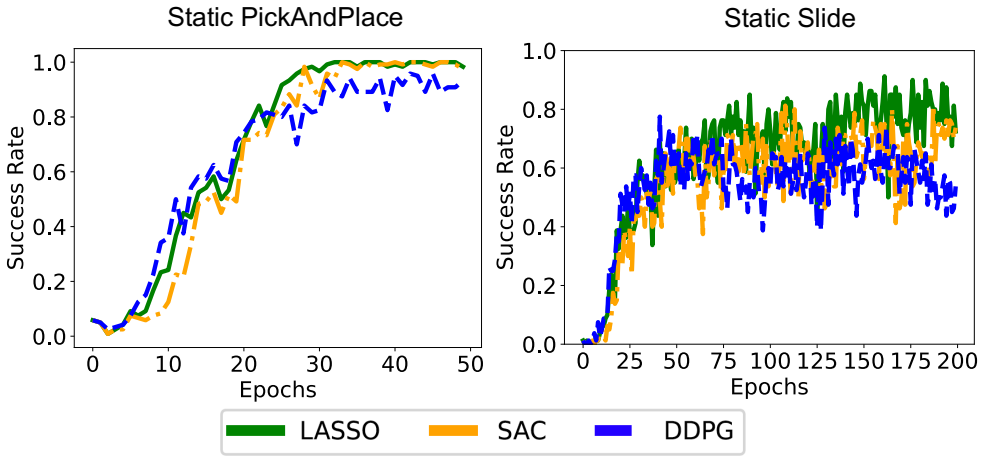


Figure 8.4: Performance comparison of all the evaluated benchmarks

knowledge of the future states allows LASSO to predict the required force and direction that needs to be applied to the puck by the robot's end-effector to reach the goal. This is complemented by the Siamese Learning, which provides our framework with a measure of distance between the current state and the final goal state. The improved representation learning allowed LASSO to attain state-of-the-art performance across all the evaluated tasks while also achieving quicker convergence, as observed in Fig. 8.4, where LASSO converged earliest for S-PnP at 31 epochs compared to 33 for SAC Fig. 8.4(Left) and achieved the best performance for S-Slide at 0.91 Fig. 8.4(Right) where it consistently outperformed all other approaches. For S-Slide, where accurate physics model of environment is required, LASSO can perform the task more successfully than other approaches as illustrated in Fig. 8.3

8.6.2 Dynamic Environments

Results: We present the performance (success rate) of all approaches on dynamic environments in Table 8.2. Similar to the static case, we tested each model on 20 separate runs for D-Reach, and on 80 and 120 runs for D-Slide and D-PnP, respectively. The results in Table 8.2 suggest that LASSO attained the best performance across all environments, especially on challenging tasks such as D-Slide and D-PnP.

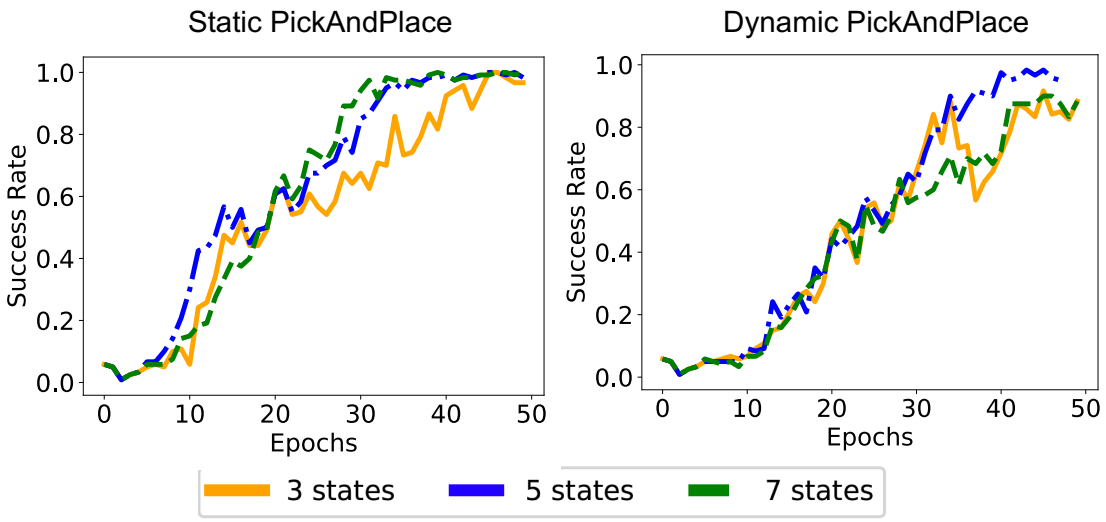


Figure 8.5: Impact of forecasted states on the overall performance

Discussion: Our proposed method obtained state-of-the-art results across all the evaluated environments. The results highlight the benefit of improved representation learning, especially for tasks that require long-horizon planning, such as D-Slide. The proposed State-space Forecasting module and the goal-conditioned distance similarity (Siamese Learning) module combine to provide a more robust representation for the downstream Policy Learning module. The forecasting module provides the framework with the ability to predict future states and the future goal position for dynamic environments. This is particularly significant as it allows the actor network to learn about the environment dynamics. Furthermore, having the Siamese Network provides essential information on the cosine distance between the current state and the final goal state.

We also observed the advantage of LASSO's Representation Learning method over Contrastive Learning approaches such as CURL. One possible explanation for this is the use of a generative modeling framework in LASSO compared to the Contrastive approach taken by CURL. Contrastive Learning requires augmenting the input to create two separate views. This has proven effective for vision-based approaches where augmenting the input does not necessarily result in a loss of semantics. However, this may not naturally apply for state-space modeling as the input dimension is limited and translations or rotations to the input results in a change in semantics. LASSO's representation learning is further complemented by the Siamese Network Setup, which used the modified cosine similarity objective to minimize the distance between the current state and the goal state.

8.6.3 Ablation study

Ablating different learning modules of LASSO

Results: We conducted an ablation study on static environments to evaluate the importance of various learning modules in our approach. Table 8.3 shows the impact of specific learning modules while keeping the Policy Learning module constant. We report the Success Rate (SR) along with the number of epochs required to converge. The results suggest that having the Siamese Learning and State-space Forecasting modules provided the best results across all the evaluated environments.

Discussion: The results in Table 8.3 suggest that LASSO with Siamese + Forecasting outperformed its ablated variants. Interestingly, we observed that ablating just the State-space Forecasting or Siamese Learning did not prevent these models from converging. However, it delayed the convergence for S-Reach, where it

Table 8.3: Ablation of LASSO's learning modules

Approaches	S-Reach		S-PnP		S-Push		S-Slide	
	SR	Epoch	SR	Epoch	SR	Epoch	SR	Epoch
No Siamese Learning	1.0	8	1.0	40	1.0	46	0.83	151
No State-space Forecasting	1.0	7	1.0	31	1.0	29	0.81	178
Siamese + Forecasting	**1.0**	4	**1.0**	32	**1.0**	30	**0.91**	161

converged at 8 epochs without the Siamese, 7 epochs without the Forecasting, and 4 epochs when both were present. For S-PnP, the convergence was reached at 40 epochs without the Siamese Learning compared to 32 epochs with both. LASSO with Siamese Learning + State-space Forecasting obtained the best overall results across all the environments, which suggests that these modules can be trained together to obtain complementary representation, even though they were trained separately with different objective functions.

Impact of forecasted states on the overall performance

Results: As our Representation Learning module consisted of a forecasting mechanism, we experimented with different numbers of forecasted states and their subsequent impact on the overall performance. A higher number of forecasted states may provide a strong prior about the future. However, it may also result in error propagation by way of the recursive decoder. As such, we empirically evaluated the impact of the different forecasted states on the overall performance in Fig. 8.5, for the task of S-PnP and D-PnP, which are long-horizon tasks. The results suggest that forecasting 7 future states provided the best performance in terms of success rate and convergence for S-PnP, whereas for D-PnP, forecasting 5 future states provided the best performance in terms of success rate and convergence.

Discussion: The results in Fig. 8.5 underline the impact of the number of forecasted states on the performance of the overall framework. For S-PnP (Fig. 8.5 (Left)), we observed that the best performance was obtained with 7 forecasted states. On the other hand, for D-PnP (Fig. 8.5 (Right)), we observed that forecasting 5 states provides the best results. This difference in performance provides insights into the challenges of modeling environment state space. When there is no significant change in the environment, as is the case for S-PnP, having a higher number of forecasted states provides improved performance, as well as quicker convergence. As observed in Fig. 8.5 (Left), LASSO with 7 states converged after 31 epochs compare to LASSO with 3 states, which converged after 45 epochs.

However, in dynamic environments, there are uncertainties involved in future state estimation. Thus, having a higher number of states, e.g., 7, leads to worse performance than 5, as the prediction may become less accurate beyond a certain number of states. The results also indicate that it is still necessary to forecast future states, as is shown by the superior performance of the model when it forecasts 5 states compared to 3 states.

8.7 Conclusion

In this work, we introduced LASSO, a novel algorithm that aims to learn a robust latent representation of the environment, which is then used to learn a policy. LASSO decoupled Representation Learning from Policy Learning. The Representation Learning module comprised an Encoder-Decoder framework which enabled learning of salient features about the future states by means of an information bottleneck. The Siamese Learning framework then used this latent space to minimize the distance between this representation and the final goal. The learned representation was then used by the actor network to learn a robust policy. Our results on static and dynamic environments suggest that LASSO outperformed all evaluated algorithms. Future work will focus on generalizing our algorithms to multi-agent scenarios, incorporating language and additional context. This will be a pretext to having collaborative policies in the real-world in the presence of humans.

Chapter 9

CollabPolicies: Policy Learning for Collaborative Systems

Building on LASSO, we next explore policy learning to encompass multi-agent systems, where coordination and collaboration between agents are essential. These systems require sophisticated methods for synchronization, and cooperation to achieve shared objectives. The following chapter will explore collaborative policy learning, addressing the unique challenges posed by environments that demand joint decision-making and integrated control strategies. Here, we extend the concepts of latent representation and policy learning to scenarios where multiple agents must work together, leveraging the strengths of each to reach collective goals.

In response to these limitations, we propose a CollabPolicy framework focused on collaborative policy learning. This work aims to establish a testing ground for evaluating foundation models' abilities to develop collaborative strategies, tackling critical challenges such as spatial reasoning, object localization, physical reasoning, and collaborative learning. Through this framework, we seek to advance the capabilities of foundation models in more complex, real-world scenarios.

9.1 Introduction

In recent years, the challenge of policy learning in multi-agent systems has become increasingly prominent, particularly as we aim to translate actions from pixels to decisions in complex environments. The process of policy learning—where agents learn to make decisions based on visual inputs—becomes exponentially more difficult as the number of agents increases. Each agent must not only interpret its own sensory data but also coordinate and communicate with others, leading to a combinatorial explosion of possible states and actions [318, 319]. This complexity is further amplified when agents operate in dynamic, open-world environments, where the need for collaboration and synchronization is crucial for success.

The complexity of policy learning in multi-agent systems arises from the need to process high-dimensional visual data, extract meaningful features, and then translate these features into actionable policies. Each agent's actions influence not only its own state but also the states of other agents, leading to a combinatorial explosion of possible interactions and outcomes. This complexity is further compounded in open-world environments, where the number of possible states and actions is vast, and where agents must deal with uncertainties, partial observability, and dynamic changes [320].

As the number of agents increases, so does the need for sophisticated coordination mechanisms. Agents must learn to share information efficiently, synchronize their actions, and adapt to the behaviors of others. This requires not just individual intelligence, but collective intelligence, where the group as a whole is capable of achieving goals that would be impossible for any single agent acting alone. In real-world applications, such

as autonomous driving, collaborative robotics, and disaster response, the ability of multiple agents to work together seamlessly is crucial for success [321, 322].

To address these challenges, we have developed a new dataset and simulation environment specifically designed for multi-agent systems. Our dataset is unique in that it includes language instructions, RGB-D (Red, Green, Blue, Depth) data from multiple viewpoints, and scenarios involving multiple robots performing a variety of open-world household tasks. These tasks are representative of real-world challenges where multiple agents must collaborate to complete complex objectives. For example, scenarios in our dataset include tasks such as coordinating to clean a room, jointly moving heavy objects, or preparing a meal, where success depends on the ability of the agents to communicate and work together effectively [323].

The simulator we have created is designed to model realistic household environments, providing a rich and varied set of challenges for multi-agent systems. It supports multiple viewpoints, allowing agents to perceive the environment from different perspectives, and it incorporates language instructions to guide the agents' actions. This combination of visual, linguistic, and spatial data makes our dataset and simulator ideal for testing the limits of current multi-agent learning algorithms.

To evaluate the effectiveness of our dataset and simulator, we conducted extensive benchmarking using several state-of-the-art models, including advanced Vision-Language Models (VLMs) and other multi-agent learning frameworks. These models have been recognized for their ability to process language and visual data independently, but when faced with our collaborative multi-agent scenarios, they revealed significant limitations. While these models performed well in tasks involving single agents or isolated tasks, their ability to coordinate, communicate, and adapt in real-time to the actions of other agents was limited [295, 324].

Our benchmarking results highlight the critical need for new architectures and algorithms that can better handle the complexities of multi-agent collaboration. The models we tested struggled with tasks that required coordinated effort and dynamic role allocation, emphasizing the importance of further research in this area. The challenges encountered by these models in our collaborative tasks underscore the potential impact of our dataset and simulator in driving the development of the next generation of intelligent, multi-agent systems.

By introducing this dataset and simulator, we aim to provide the research community with a valuable resource for developing and testing collaborative policies in multi-agent systems. Our work not only sheds light on the limitations of current state-of-the-art models but also offers a pathway for advancing the capabilities of AI in real-world, multi-agent scenarios. The insights gained from our benchmarks will serve as a foundation for future research, promoting the development of more sophisticated approaches to multi-agent policy learning [318, 319, 295].

9.2 Related Work

Multi-agent systems: Some of the earlier works at the intersection of Deep RL and MARL have proposed variants of the deep deterministic policy gradients [307]. Lowe et al. [318] proposed MADDPG, where the authors show the challenges of using traditional RL algorithms in the multi-agent case: Q-learning suffers due to the inherent non-stationarity of the environment, while policy gradient suffers from a variance that increases as the number of agents grows. The authors use the actor-critic framework, with centralized training in the form of a single critic and decentralized execution, in the form of multiple actors. The actors have access to local information, whereas the centralized critic has access to the policies of all the actors. R-MADDPG [325] extends the MADDPG framework with recurrent policies for the actors and critics in a partially observable setting with communication.

Lee et al. [326] proposed a modular framework that first individually trains each agent over a diverse set of primitive skills independently, followed by using a meta-controller to coordinate the appropriate primitive skills. The authors take a different approach to the DDPG variants by using a modular and hierarchical architecture to train the agents. Peng et al. [327] proposed FACMAC, which learns a centralized, factored critic, which factors joint action-value function into per-agent utilities. This differs from the original MADDPG variants, which use a monolithic critic, making FACMAC more suited to scaling in the presence of a larger number of agents. Moreover, the agent's policies are trained as a single joint-action policy, which

allows the learning of more coordinated behavior over decentralized policy learning. Pan et al. [328] proposed Offline Multi-Agent RL with Actor Rectification (OMAR), which combines zeroth-order optimization methods [329, 330] with first-order policy gradient methods. The zeroth-order optimization maintains an iteratively updated and refined sampling distribution to find better actions based on Q-values. The authors also demonstrate the limitations of strong offline RL baselines for multi-agent settings.

Policy Learning Under Partial Observability: Policy learning in multi-agent systems under partial observability is a critical challenge, as agents must operate based on incomplete and often noisy information. In such environments, the effectiveness of a policy is determined not only by the agent's ability to interpret its local observations but also by its capacity to communicate and collaborate with other agents. This challenge is compounded by the need to maintain a coherent memory of past events, enabling agents to infer the state of the environment and make decisions that contribute to the team's overall objective [331].

Research in this area has focused on developing algorithms that enhance the robustness of policy learning under partial observability. Recurrent neural networks (RNNs) and Long Short-Term Memory (LSTM) networks have been widely adopted to address this challenge, as they allow agents to retain information over time and use it to inform future actions [332]. For instance, the use of recurrent policies in R-MADDPG [325] demonstrates the effectiveness of incorporating memory into the decision-making process, particularly in environments where agents have limited access to the full state of the environment.

Moreover, attention mechanisms have been explored as a means to improve the efficiency of communication between agents, allowing them to focus on the most relevant pieces of information when making decisions. By selectively attending to important signals from the environment or other agents, these mechanisms help to reduce the complexity of the decision-making process, enabling more effective policy learning even in highly dynamic and partially observable settings [333].

Recent works have also investigated the use of multi-agent communication protocols, where agents share information explicitly to mitigate the effects of partial observability [334, 335]. These protocols can range from simple message passing to more sophisticated approaches where agents negotiate and coordinate their actions in real time. The development of these communication strategies is crucial for enabling robust multi-agent collaboration, especially in scenarios where agents must work together to achieve common goals despite having access to only a subset of the relevant information.

9.3 Problem Formulation

The overarching goal on this work is to enable collaborative policy learning among multiple agents, which can be a precursor for human-robot collaboration. In this work, we extend the Embodied Question Answering (EQA) framework to address the challenge of collaborative policy learning among multiple agents. The problem can be formally defined as follows:

We formulate our problem as a Markov Decision Process, defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \rho, \gamma)$ of states, actions, transition probability, reward, initial state distribution and discount factor. In our formulation, we assume the environment includes N agents. We first formulate the problem of performing long-horizon tasks from the perspective of a single agent and discuss how it can be scaled to multiple agents. To promote consistency in our terminology, we use superscripts to denote the index of agent and subscripts to denote time or primitive skill index.

9.3.1 Single-Agent RL

A reinforcement learning agent is modeled to perform sequential decision-making by interacting with the environment. The environment is usually formulated as an infinite horizon discounted Markov decision process (MDP), which is denoted by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \rho, \gamma)$.

At each time t, the agent receives an observation from the environment, s_t and chooses to execute an action a_t, which causes the system to transition to $s_{t+1} \sim P(.|s_t, a_t)$. The agent receives an instantaneous reward $R(s_t, a_t, s_{t+1})$. The goal of solving the MDP is thus to find a policy $\pi : S \rightarrow \Delta(A)$, which represents a

mapping from the state space S to the distribution over the action space A, so that $a_t \sim \pi(.|s_t)$ maximizes the expected cumulative reward:

$$\mathbf{E}[\sum_{t\geq 0} \gamma^t R(s_t, a_t, s_{t+1})|a_t \sim \pi(.|s_t), s_0] \tag{9.1}$$

Here, γ represents the discount factor. The reward is discounted to model uncertainty of future actions and state.

9.3.2 Multi-Agent RL

In multi-agent settings, the sequential decision-making problem is now augmented to involve multiple agents. As a result, there is an augmentation of the environment state space and the reward. Each agent has its own long-term reward to optimize, which now becomes a function of the policies for all the agents. For multi-agent settings, the problem is often formulated as a Markov Game (MG), which is defined by the tuple $(\mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i\in\mathcal{N}}, P, \{R^i\}, \gamma)$, where $\mathcal{N} = 1, \ldots, N$ denotes the number of agents, $\mathcal{S}$ denotes the joint observation space of all the agents, $\mathcal{A}^i$ denotes the action space of agent i.

At each timestep, t, each agent $i \in \mathcal{N}$ performs an action a^i, conditioned on the observation s^i. This leads to a state transition to a new observation s_{t+1}, with each agent getting a reward $R^i(s_t, a_t, st_{t+1})$. The goal of each agent is to optimize its own long-term reward by finding the policy $\pi^i : \mathcal{S} \to \Delta A^i$such that $a^i_t \sim \pi^i(.|s_t)$. As a result, the value-function $V^i : S \to R$ of agent i becomes a function of he joint policy $\pi : S \to \Delta(A)$ defined as $\pi(a|s) := \prod_{i\in N} \pi^i(a^i|s)$. This can be represented as follows:

$$V^i_{\pi^i, \pi^{-i}}(s) := E\Big[\sum_{t\geq 0} \gamma^t (s_t, a_t, s_{t+1})|a^i_t \sim \pi^i(.|s_t), s_0 = s\Big] \tag{9.2}$$

where $-i$ represents the indices of all agents in $\mathcal{N}$ except agent i. This demonstrates how the actions of other agents impact the reward function of any given agent.

9.4 CollabPolicy

In this section, we introduce our proposed dataset and the suite of multi-agent tasks designed to advance the state of collaborative policy learning. Unlike existing benchmarks that primarily focus on single-agent scenarios or limited multi-agent interactions, our work addresses the complexities of multi-agent collaboration, task execution, and object grounding. The dataset is specifically curated to challenge and enhance the coordination and communication capabilities of multiple agents operating in a shared environment. Additionally, we provide a comprehensive simulator that enables the evaluation of agent policies in a controlled yet realistic setting.

Our dataset consists of ten carefully crafted tasks, each designed to test various aspects of multi-agent collaboration. These tasks not only require agents to perform individual actions but also to coordinate their efforts in real time, adapt to dynamic changes in the environment, and communicate effectively to achieve common goals. Below, we discuss each task in detail:

- **Multi-Agent Stack the Blocks:** In this task, multiple agents are required to work together to stack a set of blocks in a predefined sequence to create a stable structure. The task tests the agents' ability to coordinate their movements, manage shared resources, and ensure that the blocks are placed in the correct order. The challenge lies in the need for precise timing and spatial reasoning, as improper placement by one agent can destabilize the entire structure, requiring the agents to restart or adjust their strategy.
- **Multi-Agent Put Hanger on the Rack:** This task involves agents collaborating to place hangers onto a clothing rack in an orderly fashion. The agents must navigate the environment, grasp hangers, and place them on the rack without obstructing each other's movements. The task emphasizes the importance of spatial awareness and role differentiation, where one agent might be responsible for retrieving hangers while another focuses on placement.

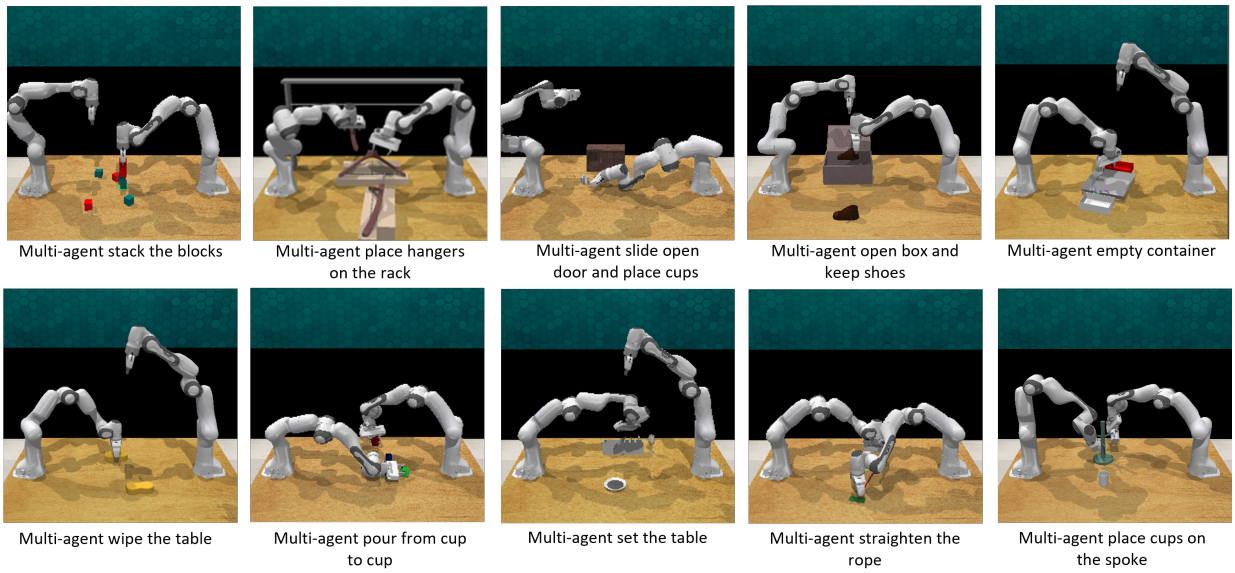


Figure 9.1: Collaborative Policy Learning Environments

- **Multi-Agent Slide Open Door and Place Cups:** In this scenario, agents must first slide open a door and then place cups inside a designated area. The task is divided into subtasks that require sequential execution, where one agent might be responsible for opening the door while another handles the cups. The challenge here is twofold: coordinating the timing of these actions and ensuring that the cups are placed securely within the designated area.
- **Multi-Agent Open Box and Keep Shoes:** This task involves multiple agents working together to open a box and place shoes inside it. The agents must coordinate their efforts to open the box efficiently and then carefully place the shoes inside without causing the box to close prematurely. This task tests the agents' ability to manage objects of varying shapes and sizes and to synchronize their actions to prevent mishaps.
- **Multi-Agent Empty Container:** In this task, agents need to collaborate to empty the contents of a container. This could involve one agent holding the container steady while another tilts it, or multiple agents working together to remove different items from the container. The task highlights the importance of role allocation and the ability to adapt to the weight and distribution of the container's contents.
- **Multi-Agent Wipe the Table:** Agents are tasked with collaboratively cleaning a table by wiping it down. Each agent may be responsible for a different section of the table, requiring them to coordinate their movements to ensure complete coverage without overlapping efforts. The task emphasizes the need for smooth, synchronized motions and the ability to divide the task effectively among the agents.
- **Multi-Agent Pour from Cup to Cup:** In this task, agents need to pour liquid from one cup to another without spilling. The task requires precise coordination between agents, especially if one agent is responsible for holding the receiving cup while another pours. The challenge lies in maintaining a steady hand, controlling the pour speed, and adjusting to any movements by the other agent.
- **Multi-Agent Set the Table:** This task involves agents working together to set a table with plates, utensils, and glasses. Each agent must navigate the environment to retrieve items and place them in the correct positions on the table. The task requires careful coordination to avoid collisions and to ensure that the table is set according to a predefined arrangement, testing the agents' ability to follow complex instructions and work in unison.
- **Multi-Agent Straighten the Rope:** In this task, agents must work together to straighten a coiled

or tangled rope. The task tests the agents' ability to manipulate flexible objects, requiring them to coordinate their pulls and adjustments to avoid creating new tangles. Communication is key, as the agents must continuously adjust their strategies based on the rope's current state.

- **Multi-Agent Place Cups on the Spoke:** This task requires agents to place cups onto the spokes of a rotating or stationary structure. The agents must coordinate their timing and placement to ensure that the cups are securely positioned on the spokes. The task challenges the agents' ability to synchronize their actions with the movement of the structure and to adapt to changes in its speed or direction.

9.5 Experimental Setup

9.5.1 Evaluated Algorithms

We evaluated on state-of-the-art imitation learning algorithms that were specifically designed for 3D object manipulation including PerAct [336] and RVT [337]. We specifically evaluate the ability of these algorithms to learn a policy in multi-agent scenario, where they need to predict the action of one of the agents, under partial observability.

PerAct [336]: PerAct encodes language goals and RGB-D voxel observations with a Perceiver Transformer [338], and outputs the optimal voxel for the next action. Unlike frameworks that rely on 2D images, PerAct's use of voxelized observation and action spaces offers a strong structural advantage for efficiently learning 6-DoF policies.

RVT [339]: To predict the key-frame pose, RVT first reconstructs a point cloud of the scene from input RGB-D images and renders it from five fixed virtual views—top, front, left, back, and right—around the robot. These views are processed by a multi-view transformer model, which predicts heatmaps for each view. The heatmap scores are then back-projected into 3D, with each point receiving an averaged score from its 2D projections. The 3D point with the highest score indicates the predicted gripper location. Additionally, RVT uses global features from these views to predict the gripper's rotation and state (open or closed).

9.6 Results and Discussion

The performance of state-of-the-art approaches on our multi-agent benchmark is summarized in Table 9.1. We evaluated the methods based on three key metrics: Translation Loss, Gripper Loss, and Collision Loss, which collectively provide insights into the effectiveness of the models in handling multi-agent tasks. Notably, PerAct employs a 3D voxelization approach, while RVT handles multi-view images, each bringing unique strengths to the table.

9.6.1 Translation Loss

Translation loss measures the accuracy of the agents' movements and positioning within the environment. This metric is particularly critical for tasks requiring precise spatial coordination, such as placing objects in specific locations or navigating through confined spaces. In this benchmark, PerAct outperformed RVT, achieving a significantly lower translation loss of 2.51 compared to RVT's 6.65. PerAct's use of 3D voxelization allows for a detailed and structured representation of the environment, enabling it to better capture spatial relationships and make more accurate movement decisions. On the other hand, RVT, which relies on multi-view images, may encounter challenges in integrating and interpreting these views, potentially leading to higher translation loss due to difficulties in spatial reasoning across multiple perspectives.

9.6.2 Gripper Loss

Gripper loss assesses the accuracy and reliability of the agents' manipulation of objects using their grippers. This metric is crucial for tasks involving object handling, where secure grasping and precise movement are essential. PerAct achieved a lower gripper loss of 3.47 compared to RVT's 5.44. The 3D voxelization

technique used by PerAct contributes to its superior performance by providing a more detailed understanding of object geometry and spatial relationships, facilitating better gripper control. In contrast, RVT's multi-view image handling, while effective in capturing different perspectives, may struggle with consistent object manipulation due to the complexity of fusing these views into a coherent representation for gripper actions.

9.6.3 Collision Loss

Collision loss measures how often agents collide with objects or other agents during task execution. Lower collision loss indicates better path planning and spatial awareness, critical for maintaining task efficiency and safety in multi-agent environments. PerAct demonstrated superior performance with a collision loss of 0.47, slightly better than RVT's 0.54. The advantage of 3D voxelization in PerAct allows for more precise spatial modeling, aiding in effective collision avoidance. While RVT's use of multi-view images provides comprehensive environmental coverage, the challenge of integrating these views could lead to occasional misinterpretations of spatial relationships, resulting in slightly higher collision rates.

9.6.4 Discussion

The results indicate that PerAct consistently outperforms RVT across all three metrics: translation, gripper, and collision losses. PerAct's reliance on 3D voxelization appears to provide a robust framework for detailed spatial understanding and manipulation, which is particularly advantageous in tasks that require precise object placement and avoidance of collisions. The voxel-based approach allows PerAct to model the environment with a high degree of accuracy, enabling more reliable decision-making in complex, multi-agent scenarios.

RVT, on the other hand, handles multi-view images to form its understanding of the environment. While this approach offers the advantage of capturing diverse perspectives, it introduces challenges in fusing these views into a cohesive spatial model. The increased complexity in processing and integrating multiple views likely contributes to the higher losses observed in translation and gripper tasks. However, RVT's approach is still highly valuable in scenarios where comprehensive environmental coverage is necessary, even though it may need further refinement to match the precision offered by voxel-based methods.

The slight difference in collision loss between the two methods is not surprising, as both the approaches use the same low-level controller- RRT. However, PerAct's consistent performance across all metrics underscores the benefits of 3D voxelization in maintaining a high level of spatial accuracy and control in multi-agent environments.

Overall, the results highlight the importance of the underlying representation technique in multi-agent systems. PerAct's 3D voxelization provides a strong foundation for precise and effective multi-agent collaboration, while RVT's multi-view image handling offers valuable insights.

9.7 Future Directions

Based on the findings from our preliminary experiments, several avenues for future work has emerged in the space of collaborative policy.

Refinement of Multi-View Image Integration: While RVT's multi-view image handling offers the advantage of capturing diverse perspectives, the challenges in fusing these views into a cohesive spatial model suggest an area for improvement. Furthermore, RVT may have been optimized for single-agent scenarios, unlike this setup. Future research could focus on developing more sophisticated fusion techniques that can better integrate multi-view data in collaborative setting, reducing complexity and improving precision in translation and gripper tasks.

Predicting Other Agents' Actions: In multi-agent systems, the ability to predict the actions of other agents is crucial for effective collaboration and conflict avoidance. Future research could focus on developing predictive models that allow agents to anticipate the movements and decisions of their peers, thereby enabling

Table 9.1: Performance of state-of-the-art approaches on multi-agent benchmark

Method	Translation Loss	Gripper Loss	Collision Loss
RVT	6.65	5.44	0.54
PerAct	**2.51**	**3.47**	**0.47**

more synchronized and cooperative behavior. This could involve leveraging recurrent neural networks, game-theoretic approaches, or reinforcement learning techniques that account for the intentions and strategies of other agents in the environment.

Exploration of Alternative Representation Techniques: While voxelization and multi-view images have proven effective, there may be other representation techniques that could offer advantages in specific multi-agent contexts. Future work could explore alternative methods, such as point cloud-based representations or neural implicit representations, to determine if they can provide superior performance in certain tasks or environments.

9.8 Conclusion

In this work, we introduced a dataset and simulator designed to advance the capabilities of intelligent, multi-agent systems, driving progress in areas such as cooperative robotics, autonomous vehicles, and beyond. The insights gained from our benchmarks on state-of-the-art models provide a solid foundation for future research, encouraging the development of more sophisticated approaches to multi-agent policy learning. Future work could build on this by refining multi-view image integration, combining voxelization with other techniques and improving the prediction of other agents' actions.

Chapter 10

Conclusion

In my dissertation, I have focused on investigating some of the open challenges in Human-Robot Interaction initially in isolation by modeling human behavor, modeling robot's decision making separately. Next, I focused on bridging the space by exploring the joint interaction between humans and robots.

10.1 Summary of Contributions

10.1.1 Modeling Human Motion

We developed advanced architectural frameworks, such as PoseTron and IMPRINT, which significantly enhance the prediction of human motion in both single-agent and multi-agent settings. These frameworks introduce novel attention mechanisms, including conditional attention and multimodal attention modules, to accurately weigh and integrate diverse inputs, resulting in more robust and contextually aware motion predictions. The introduction of the INTERACT dataset provided a comprehensive and multimodal source of data, specifically designed to capture the complexities of human-human and human-robot collaboration, filling a critical gap in existing datasets.

10.1.2 Robot Control

In the domain of robotic control, we introduced the LASSO algorithm, which addresses the challenge of integrating planning and execution in dynamic and uncertain environments. By decoupling representation learning from policy learning, LASSO enables robots to adaptively and autonomously update their plans based on real-time environmental changes, significantly enhancing their flexibility and effectiveness in human-robot interaction (HRI) scenarios.

10.1.3 Joint Human-Robot Interaction

The dissertation also tackled the intricate dynamics of joint human-robot interactions, where the actions of both humans and robots influence each other. The IMPRINT framework was designed to model these bidirectional interactions, incorporating multimodal data to provide a more accurate and holistic understanding of team dynamics. Additionally, the development of the CollabPolicy Benchmark facilitated the learning of collaborative policies, allowing robots to better integrate into human-robot teams and improve overall team performance.

10.2 Lessons Learned

10.2.1 Importance of Multimodal Data

The integration of multimodal data is vital for accurate motion prediction and robust robot control. However, the complexity of fusing diverse modalities, such as RGB, depth, skeletal data, and gaze, presents significant challenges. The success of models like IMPRINT and PoseTron underscores the need for advanced attention mechanisms and robust architectures capable of handling this complexity.

10.2.2 Scalability and Generalization

While the models developed in this dissertation demonstrated superior performance in controlled environments or static datasets [21, 2, 54, 22], the scalability of these approaches to larger teams and their generalization to diverse real-world scenarios remain areas for further exploration. The ability of a model to maintain performance across different contexts and with varying numbers of agents is crucial for its practical applicability in HRI.

10.2.3 Balancing Real-Time Performance and Model Complexity

The need for real-time processing in HRI applications requires careful balancing of model complexity and computational efficiency. While GRUs and RNNs offer advantages in short-term sequence modeling, their limitations over long horizons highlight the importance of exploring alternative architectures, such as transformers, which can better maintain long-range dependencies.

10.2.4 Addressing Long-Horizon Predictions

The challenge of making accurate long-horizon predictions remains a significant hurdle. The reliance on RNN-based architectures in earlier models was a limiting factor, prompting the exploration of transformer-based solutions that can more effectively capture and process extended sequences of data.

10.3 Future Work

10.3.1 Expanding Multimodal Fusion Techniques

Future research could delve deeper into the exploration of advanced multimodal fusion techniques to further enhance the performance of human-robot interaction (HRI) systems. Current models often rely on relatively simple fusion methods, such as early fusion (concatenating features at the input level) or late fusion (combining features at the decision level). However, these approaches can be limited in their ability to capture complex interdependencies between different modalities, such as visual, auditory, tactile, and proprioceptive data. Advanced techniques, such as dynamic fusion networks or attention-based fusion mechanisms, could be explored to allow for more adaptive and context-sensitive integration of multimodal inputs.

Additionally, there is a need to incorporate new sensor data into these models to broaden the scope of HRI applications. For example, integrating data from thermal cameras, LiDAR, or even biosensors could provide additional context that enhances a robot's ability to understand and predict human actions. Moreover, improving the robustness of these models in the face of noisy or incomplete data remains a critical challenge. Future work could focus on developing algorithms that can dynamically weigh the reliability of each input modality and adjust the fusion process accordingly, ensuring that the models remain effective even in less-than-ideal conditions.

10.3.2 Enhancing Scalability for Large Teams

Scalability is a significant concern when applying HRI models to large teams of humans and robots, especially in dynamic and complex environments. The models developed in this dissertation have shown promise in

controlled settings with a limited number of agents. However, real-world applications often involve larger teams, which introduce new challenges related to coordination, communication, and resource management.

Future research could focus on optimizing the underlying architectures of models like IMPRINT and VADER to better handle the increased computational demands of larger teams. This could involve parallelization techniques, where tasks are distributed across multiple processing units, or the development of more efficient algorithms that reduce the computational burden without sacrificing accuracy. Additionally, exploring decentralized approaches, where each agent processes and shares information locally with its neighbors rather than relying on a central controller, could allow for more efficient and scalable solutions in large-scale environments.

Another avenue for enhancing scalability is the use of hierarchical models, where agents are organized into sub-teams or layers, each responsible for different aspects of the overall task. This approach could help manage the complexity of large teams by breaking down tasks into smaller, more manageable components, while still maintaining overall coordination and coherence.

10.3.3 Real-World Application and Generalization

While the proposed models have demonstrated strong performance in controlled research environments, their generalization to a wide range of real-world scenarios remains an open challenge. Real-world environments are often far more complex and unpredictable than those typically used in academic research, with factors such as environmental noise, variable lighting conditions, and unpredictable human behavior all influencing the effectiveness of HRI systems.

To address this, future work should focus on deploying these models in diverse real-world settings, such as industrial automation, healthcare, and public service environments. This would provide valuable insights into the robustness and adaptability of the models, allowing researchers to identify and address any shortcomings. Additionally, real-world deployment can help uncover edge cases or rare events that are difficult to simulate in a lab environment, but crucial for ensuring the reliability and safety of HRI systems in practice.

Another key aspect of generalization is the ability of models to transfer knowledge from one domain to another. For instance, a model trained in a manufacturing setting should ideally be able to adapt to a healthcare environment with minimal retraining. Future research could explore transfer learning techniques or domain adaptation strategies that enable models to generalize more effectively across different tasks and environments, thereby broadening their applicability and reducing the need for extensive retraining.

10.3.4 Improving Long-Horizon Predictions

One of the key challenges identified in this dissertation is the difficulty of making accurate long-horizon predictions, particularly in the context of complex, multi-step tasks that require sustained attention and decision-making over extended periods. Traditional RNN-based architectures, such as GRUs and LSTMs, have shown limitations in this regard, often struggling to maintain performance over long sequences due to issues like vanishing gradients and limited memory capacity.

To address this, future work could focus on further developing transformer-based models or other novel architectures that are better suited to long-horizon predictions. Transformers, with their ability to capture long-range dependencies through self-attention mechanisms, offer a promising alternative to RNNs, particularly in tasks that involve planning, prediction, and decision-making over extended timeframes, as shown in our work [54]. Additionally, research could explore hybrid models that combine the strengths of different sequence learning approaches, such as integrating transformers with RNNs or using hierarchical models that operate at different timescales.

Another promising direction is the development of hierarchical or multi-scale models that can operate at different levels of abstraction. For example, a model could use a high-level planner to predict the global motion over a horizon, while lower-level models handle the detailed execution of individual joints. This approach could help mitigate the challenges of long-horizon prediction by breaking down complex tasks into more manageable components, each with its own focus and time scale.

10.3.5 Developing Explainable AI in HRI

As HRI models become increasingly complex, there is a growing need for transparency and interpretability to ensure that these systems are trustworthy and understandable to human collaborators. Explainable AI (XAI) in the context of HRI involves not only making the decision-making processes of robots more transparent but also ensuring that humans can understand and predict the behavior of these systems, especially in collaborative tasks.

Future research could focus on developing methods to better understand and visualize how these models make predictions, particularly in the context of multimodal fusion and interaction dynamics, which we briefly explored in this thesis [1, 2]. This might involve the use of attention visualization tools that show which parts of the input data the model is focusing on when making decisions, or the development of interpretable models that can explain their reasoning in human-readable terms.

10.4 Conclusion

In conclusion, this dissertation has made substantial progress in advancing Human-Robot Interaction (HRI) by introducing novel frameworks, methodologies, and insights that enhance the ability of robots to understand human behavior and make timely decisions. The work addresses complex challenges in modeling and predicting human-robot interactions, particularly in dynamic and collaborative environments, by integrating multi-modal data and advanced policy learning algorithms. Additionally, the introduction of comprehensive open-source datasets and benchmarks sets a new standard for evaluating and improving HRI systems, promoting open-source and replicable research. These contributions lay a strong foundation for future advancements, particularly in areas such as multimodal fusion, scalability, real-world applicability, long-horizon predictions, and explainable AI, all of which are essential for pushing the boundaries of HRI. As robotics continues to evolve, the frameworks and insights from this dissertation will be crucial enabling safer, more effective, and intuitive interactions across various real-world applications, from healthcare and industrial automation to public services and beyond.

Bibliography

[1] M. S. Yasar and T. Iqbal, "A scalable approach to predict multi-agent motion for human-robot collaboration," in *IEEE RA-L*, 2021.

[2] M. S. Yasar, M. M. Islam, and T. Iqbal, "Imprint: Interactional dynamics-aware motion prediction using multimodal context," *Transactions in Human-Robot Interaction (THRI)*, 2023.

[3] M. S. Yasar and T. Iqbal, "Vader: Vector-quantized generative adversarial network for motion prediction," *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[4] R. Dror, S. Shlomov, and R. Reichart, "Deep dominance-how to properly compare deep neural models," in *ACL*, pp. 2773–2785, 2019.

[5] Z. Wang, K. Mülling, M. P. Deisenroth, H. Ben Amor, D. Vogt, B. Schölkopf, and J. Peters, "Probabilistic movement modeling for intention inference in human–robot interaction," *IJRR*, 2013.

[6] A. M. Williams, P. Ward, J. M. Knowles, and N. J. Smeeton, "Anticipation skill in a real-world task: measurement, training, and transfer in tennis.," *Journal of Experimental Psychology: Applied*, 2002.

[7] M. Fiore, A. Clodic, and R. Alami, "On planning and task achievement modalities for human-robot collaboration," in *Experimental Robotics: The 14th International Symposium on Experimental Robotics*, pp. 293–306, Springer, 2016.

[8] A. Clodic, R. Alami, R. Chatila, and L. Clodic, "Key challenges and open questions for software architecture and framework of human-robot interaction applications," *Autonomous Robots*, vol. 42, no. 5, pp. 1151–1160, 2017.

[9] N. Sebanz, H. Bekkering, and G. Knoblich, "Joint action: bodies and minds moving together," *Trends in cognitive sciences*, vol. 10, no. 2, pp. 70–76, 2006.

[10] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[11] S. Thrun, W. Burgard, and D. Fox, "Probabilistic robotics (intelligent robotics and autonomous agents)," 2005.

[12] X. Yu, M. Hoggenmueller, and M. Tomitsch, "Your way or my way: Improving human-robot co-navigation through robot intent and pedestrian prediction visualisations," in *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 211–221, 2023.

[13] S. Lemaignan, M. Warnier, E. A. Sisbot, A. Clodic, and R. Alami, "Artificial cognition for social human–robot interaction: An implementation," *Artificial Intelligence*, vol. 247, pp. 45–69, 2017.

[14] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726–1743, 2013.

[15] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 895–935, 2020.

[16] A. Rudenko, T. P. Kucner, C. S. Swaminathan, R. T. Chadalavada, K. O. Arras, and A. J. Lilienthal, "Thör: Human-robot navigation data collection and accurate motion trajectories dataset," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 676–682, 2020.

[17] S. Haddadin, A. De Luca, and A. Albu-Schäffer, "Robot collisions: A survey on detection, isolation, and identification," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1292–1312, 2017.

[18] S. Haddadin and E. Croft, "Physical human–robot interaction," *Springer handbook of robotics*, pp. 1835–1874, 2016.

[19] S. El Zaatari, M. Marei, W. Li, and Z. Usman, "Cobot programming for collaborative industrial tasks: An overview," *Robotics and Autonomous Systems*, vol. 116, pp. 162–180, 2019.

[20] G. Hoffman, "Evaluating fluency in human–robot collaboration," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 3, pp. 209–218, 2019.

[21] M. S. Yasar and T. Iqbal, "A scalable approach to predict multi-agent motion for human-robot collaboration," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1686–1693, 2021.

[22] M. S. Yasar, T. Vitchutripop, and T. Iqbal, "Lasso: Learning latent policies via state space modeling," *IEEE ICRA, Under review*, 2023.

[23] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, *et al.*, "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[24] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 1877–1901, 2020.

[25] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[26] Y. Tang, L. Ma, W. Liu, and W.-S. Zheng, "Long-term human motion prediction by modeling motion context and enhancing motion dynamic," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, p. 935–941, AAAI Press, 2018.

[27] K. Lyu, H. Chen, Z. Liu, B. Zhang, and R. Wang, "3d human motion prediction: A survey," *Neurocomputing*, vol. 489, pp. 345–365, 2022.

[28] V. Adeli, E. Adeli, I. Reid, J. C. Niebles, and S. H. Rezatofighi, "Socially and contextually aware human motion and pose forecasting," *IEEE RA-L*, 2020.

[29] J. Martinez, M. J. Black, and J. Romero, "On human motion prediction using recurrent neural networks," in *IEEE CVPR*, 2017.

[30] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *IEEE CVPR*, 2016.

[31] J. Zaki and K. N. Ochsner, "The neuroscience of empathy: progress, pitfalls and promise," *Nature neuroscience*, vol. 15, no. 5, pp. 675–680, 2012.

[32] A. M. Lehrmann, P. V. Gehler, and S. Nowozin, "Efficient nonlinear markov models for human motion," in *CVPR*, 2014.

[33] G. W. Taylor, G. E. Hinton, and S. Roweis, "Modeling human motion using binary latent variables," *NeurIPS*, 2006.

[34] J. Wang, A. Hertzmann, and D. J. Fleet, "Gaussian process dynamical models," in *Advances in Neural Information Processing Systems* (Y. Weiss, B. Schölkopf, and J. Platt, eds.), vol. 18, MIT Press, 2005.

[35] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, "Recurrent network models for human dynamics," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4346–4354, IEEE, 2015.

[36] E. Aksan, M. Kaufmann, and O. Hilliges, "Structured prediction helps 3d human motion modelling," in *IEEE ICCV*, 2019.

[37] J. Bütepage, H. Kjellström, and D. Kragic, "Anticipating many futures: Online human motion prediction and generation for human-robot interaction," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[38] J. Bütepage, A. Ghadirzadeh, Ö. Ö. Karadag, M. Björkman, and D. Kragic, "Imitating by generating: Deep generative models for imitation of interactive tasks," *Frontiers in Robotics and AI*, 2020.

[39] S. Toyer, A. Cherian, T. Han, and S. Gould, "Human pose forecasting via deep markov models," in *International DICTA*, 2017.

[40] G. Hoffman and G. Weinberg, "Synchronization in human-robot musicianship," in *19th International Symposium in Robot and Human Interactive Communication*, pp. 718–724, IEEE, 2010.

[41] H. S. Koppula, A. Jain, and A. Saxena, "Anticipatory planning for human-robot teams," in *Experimental robotics*, Springer, 2016.

[42] H. B. Amor, G. Neumann, S. Kamthe, O. Kroemer, and J. Peters, "Interaction primitives for human-robot cooperation tasks," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[43] S. Nikolaidis, R. Ramakrishnan, K. Gu, and J. Shah, "Efficient model learning from joint-action demonstrations for human-robot collaborative tasks," in *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, HRI '15, (New York, NY, USA), p. 189–196, Association for Computing Machinery, 2015.

[44] J. Mainprice and D. Berenson, "Human-robot collaborative manipulation planning using early prediction of human motion," in *IROS*, IEEE, 2013.

[45] T. Iqbal, M. J. Gonzales, and L. D. Riek, "Joint action perception to enable fluent human-robot teamwork," in *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 400–406, IEEE, 2015.

[46] S. Sebo, B. Stoll, B. Scassellati, and M. F. Jung, "Robots in groups and teams: a literature review," *Proceedings of the ACM on Human-Computer Interaction*, vol. 4, no. CSCW2, pp. 1–36, 2020.

[47] T. Iqbal and L. D. Riek, "Temporal anticipation and adaptation methods for fluent human-robot teaming," in *IEEE ICRA*, 2021.

[48] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2255–2264, 2018.

[49] E. Schmerling, K. Leung, W. Vollprecht, and M. Pavone, "Multimodal probabilistic model-based planning for human-robot interaction," in *IEEE ICRA*, 2018.

[50] S. H. Park, G. Lee, M. Bhat, J. Seo, M. Kang, J. Francis, A. R. Jadhav, P. P. Liang, and L.-P. Morency, "Diverse and admissible trajectory forecasting through multimodal context understanding," *ECCV*, 2020.

[51] M. A. Graziano, *The Spaces Between Us: A Story of Neuroscience, Evolution, and Human Nature*. Oxford University Press, 2017.

[52] F. Rossi, M. Scheutz, G. A. Kaminka, J. Sapience, R. Alami, and D. Ramachandran, "Decision making under uncertainty in human-robot interaction: A survey," in *Human-Robot Interaction*, pp. 147–178, Springer, 2020.

[53] E. Pignat and S. Calinon, "Learning adaptive dressing assistance from human demonstration," *Robotics and Autonomous Systems*, vol. 93, pp. 61–75, 2017.

[54] M. S. Yasar, M. M. Islam, and T. Iqbal, "Posetron: Enabling close-proximity human-robot collaboration through multi-human motion prediction," in *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 830–839, 2024.

[55] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+ d: A large scale dataset for 3d human activity analysis," in *IEEE CVPR*, 2016.

[56] H. Joo, H. Liu, L. Tan, L. Gui, B. Nabbe, I. Matthews, T. Kanade, S. Nbuhara, and Y. Sheikh, "Panoptic studio: A massively multiview system for social motion capture," in *IEEE ICCV*, 2015.

[57] T. Iqbal and L. D. Riek, "Coordination dynamics in multihuman multirobot teams," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1712–1717, 2017.

[58] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[59] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[60] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009.

[61] R. Saxe and A. Wexler, "The effects of time pressure and uncertainty in human-robot interaction," *Journal of Experimental Psychology: Applied*, vol. 11, no. 3, pp. 185–193, 2005.

[62] S. Chernova and A. L. Thomaz, "Teaching robot behavior using human instruction," in *AAAI Fall Symposium Series*, 2014.

[63] M. Laskin, A. Srinivas, and P. Abbeel, "Curl: Contrastive unsupervised representations for reinforcement learning," in *International Conference on Machine Learning*, pp. 5639–5650, PMLR, 2020.

[64] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," *ICML*, 2019.

[65] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," in *International conference on machine learning*, pp. 2555–2565, PMLR, 2019.

[66] L. Tian, K. He, S. Xu, A. Cosgun, and D. Kulic, "Crafting with a robot assistant: Use social cues to inform adaptive handovers in human-robot collaboration," in *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 252–260, 2023.

[67] P. Kratzer, S. Bihlmaier, N. B. Midlagajni, R. Prakash, M. Toussaint, and J. Mainprice, "Mogaze: A dataset of full-body motions that includes workspace geometry and eye-gaze," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 367–373, 2020.

[68] M. A. Goodrich and A. C. Schultz, "Human-robot interaction: A survey," *Foundations and Trends in Human-Computer Interaction*, vol. 1, no. 3, pp. 203–275, 2007.

[69] S. Nikolaidis, Y. X. Zhu, D. Hsu, and S. Srinivasa, "Human-robot mutual adaptation in shared autonomy," in *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 294–302, 2017.

[70] T. Iqbal and L. D. Riek, "Human robot teaming: Approaches from joint action and dynamical systems," *Humanoid Robotics: A Reference, Springer*, 2017.

[71] B. H. Repp and Y.-H. Su, "Sensorimotor synchronization: a review of recent research (2006–2012)," *Psychonomic bulletin & review*, 2013.

[72] T. Iqbal, S. Li, C. Fourie, B. Hayes, and J. A. Shah, "Fast online segmentation of activities from partial trajectories," in *ICRA*, 2019.

[73] T. Iqbal and L. D. Riek, "A Method for Automatic Detection of Psychomotor Entrainment," *IEEE T-AC*, 2016.

[74] T. Iqbal, S. Rack, and L. D. Riek, "Movement coordination in human-robot teams: A dynamical systems approach," *IEEE T-RO*, 2016.

[75] T. Iqbal and L. D. Riek, "Coordination dynamics in multi-human multi-robot teams," *IEEE RA-L*, 2017.

[76] M. M. Islam and T. Iqbal, "Hamlet: A hierarchical multimodal attention-based human activity recognition algorithm," in *IROS*, 2020.

[77] G. Hoffman and C. Breazeal, "Cost-Based Anticipatory Action Selection for Human–Robot Fluency," *IEEE T-RO*, 2007.

[78] M. M. Islam and T. Iqbal, "Multi-gat: A graphical attention-based hierarchical multimodal representation learning approach for human activity recognition," in *IEEE RA-L*, 2021.

[79] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-rnn: Deep learning on spatio-temporal graphs," in *IEEE CVPR*, 2016.

[80] N. Rhinehart, R. McAllister, K. Kitani, and S. Levine, "Precog: Prediction conditioned on goals in visual multi-agent settings," in *IEEE ICCV*, 2019.

[81] D. Yu, H. Wang, P. Chen, and Z. Wei, "Mixed pooling for convolutional neural networks," in *RSKT*, Springer, 2014.

[82] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *NIPS*, 2017.

[83] E. Barsoum, J. Kender, and Z. Liu, "Hp-gan: Probabilistic 3d human motion prediction via gan," in *IEEE CVPRW*, 2018.

[84] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[85] C. Chen, R. Jafari, and N. Kehtarnavaz, "Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor," in *IEEE ICIP*, 2015.

[86] J. Butepage, M. J. Black, D. Kragic, and H. Kjellstrom, "Deep representation learning for human motion prediction and classification," in *IEEE CVPR*, 2017.

[87] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, "Desire: Distant future prediction in dynamic scenes with interacting agents," in *IEEE CVPR*, 2017.

[88] G. J. Maeda, G. Neumann, M. Ewerton, R. Lioutikov, O. Kroemer, and J. Peters, "Probabilistic movement primitives for coordination of multiple human–robot collaborative tasks," *Autonomous Robots*, 2017.

[89] V. V. Unhelkar, P. A. Lasota, Q. Tyroller, R.-D. Buhai, L. Marceau, B. Deml, and J. A. Shah, "Human-aware robotic assistant for collaborative assembly: Integrating human motion prediction with planning in time," *IEEE RA-L*, 2018.

[90] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[91] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," *arXiv preprint arXiv:1511.05644*, 2015.

[92] E. Dupont, "Learning disentangled joint continuous and discrete representations," in *NIPS*, 2018.

[93] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[94] D. P. Losey, K. Srinivasan, A. Mandlekar, A. Garg, and D. Sadigh, "Controlling assistive robots with learned latent actions," in *IEEE ICRA*, 2020.

[95] M. Suguitan, R. Gomez, and G. Hoffman, "Moveae: Modifying affective robot movements using classifying variational autoencoders," in *ACM/IEEE HRI*, 2020.

[96] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural computation*, 1989.

[97] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, (San Diega, CA, USA), 2015.

[98] L. Maaten and G. Hinton, "Visualizing data using t-sne," *JMLR*, 2008.

[99] M. Knudsen and J. Kaivo-Oja, "Collaborative robots: Frontiers of current literature," *Journal of Intelligent Systems: Theory and Applications*, vol. 3, no. 2, pp. 13–20, 2020.

[100] A. M. Djuric, R. Urbanic, and J. Rickli, "A framework for collaborative robot (cobot) integration in advanced manufacturing systems," *SAE International Journal of Materials and Manufacturing*, 2016.

[101] M. M. Islam and T. Iqbal, "MuMu: Cooperative multitask learning-based guided multimodal fusion," in *AAAI*, 2022.

[102] C. Finn, A. Rajeswaran, S. Kakade, and S. Levine, "Online meta-learning," in *International Conference on Machine Learning*, 2019.

[103] M. S. Yasar and T. Iqbal, "Robots that can anticipate and learn in human-robot teams," in *Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction*, 2022.

[104] M. S. Yasar and T. Iqbal, "CoRaL: Continual representation learning for overcoming catastrophic forgetting," in *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pp. 1969–1978, 2023.

[105] M. M. Islam, M. S. Yasar, and T. Iqbal, "Maven: A memory augmented recurrent approach for multimodal fusion," *IEEE Transactions on Multimedia*, pp. 1–1, 2022.

[106] L. Sanneman, C. Fourie, J. A. Shah, *et al.*, "The state of industrial robotics: Emerging technologies, challenges, and key research directions," *Foundations and Trends® in Robotics*, 2021.

[107] G. Hoffman and C. Breazeal, "Effects of anticipatory action on human-robot teamwork," *Robotics and Autonomous Systems*, vol. 56, no. 8, pp. 701–717, 2007.

[108] P. A. Lasota, G. F. Rossano, and J. A. Shah, "Toward safe close-proximity human-robot interaction with standard industrial robots," in *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 339–344, IEEE, 2014.

[109] R. Freedman and S. Zilberstein, "Integration of planning with recognition for responsive interaction using classical planners," in *AAAI*, 2017.

[110] A. Van Den Oord, O. Vinyals, *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, 2017.

[111] M. Li, S. Chen, Y. Zhao, Y. Zhang, Y. Wang, and Q. Tian, "Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 214–223, 2020.

[112] W. Mao, M. Liu, and M. Salzmann, "History repeats itself: Human motion prediction via motion attention," in *European Conference on Computer Vision*, pp. 474–489, Springer, 2020.

[113] E. Aksan, M. Kaufmann, P. Cao, and O. Hilliges, "A spatio-temporal transformer for 3d human motion prediction," in *2021 International Conference on 3D Vision (3DV)*, pp. 565–574, IEEE, 2021.

[114] Z. Liu, S. Wu, S. Jin, S. Ji, Q. Liu, S. Lu, and L. Cheng, "Investigating pose representations and motion contexts modeling for 3d motion prediction," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 1, pp. 681–697, 2022.

[115] L.-Y. Gui, Y.-X. Wang, X. Liang, and J. M. Moura, "Adversarial geometry-aware human motion prediction," in *Proceedings of the european conference on computer vision (ECCV)*, pp. 786–803, 2018.

[116] J. N. Kundu, M. Gor, and R. V. Babu, "Bihmp-gan: Bidirectional 3d human motion prediction gan," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, pp. 8553–8560, 2019.

[117] M. S. Yasar and T. Iqbal, "Improving human motion prediction through continual learning," *ACM/IEEE Int. Conf. on Human-Robot Interaction (HRI), LEAP-HRI Workshop*, 2021.

[118] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[119] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[120] X. Long, C. Gan, G. Melo, X. Liu, Y. Li, F. Li, and S. Wen, "Multimodal keyless attention fusion for video classification," in *AAAI*, 2018.

[121] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, 2020.

[122] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks: A unified approach to action segmentation," in *Computer Vision–ECCV 2016 Workshops*, Springer, 2016.

[123] E. S. Short, M. L. Chang, and A. Thomaz, "Detecting contingency for hri in open-world environments," in *2018 13th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 425–433, IEEE, 2018.

[124] M. Zurek, A. Bobu, D. S. Brown, and A. D. Dragan, "Situational confidence assistance for lifelong shared autonomy," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2783–2789, IEEE, 2021.

[125] A. Bobu, A. Bajcsy, J. F. Fisac, S. Deglurkar, and A. D. Dragan, "Quantifying hypothesis space misspecification in learning from human–robot demonstrations and physical corrections," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 835–854, 2020.

[126] C. I. Mavrogiannis, W. B. Thomason, and R. A. Knepper, "Social momentum: A framework for legible navigation in dynamic multi-agent environments," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 361–369, 2018.

[127] C. I. Mavrogiannis and R. A. Knepper, "Decentralized multi-agent navigation planning with braids," in *Algorithmic foundations of robotics XII*, pp. 880–895, Springer, 2020.

[128] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *European Conference on Computer Vision*, pp. 683–700, Springer, 2020.

[129] H. Kivrak, F. Cakmak, H. Kose, and S. Yavuz, "Social navigation framework for assistive robots in human inhabited unknown environments," *Engineering Science and Technology, an International Journal*, vol. 24, no. 2, pp. 284–298, 2021.

[130] S. Javdani, H. Admoni, S. Pellegrinelli, S. S. Srinivasa, and J. A. Bagnell, "Shared autonomy via hindsight optimization for teleoperation and teaming," *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 717–742, 2018.

[131] A. Adu-Bredu, Z. Zeng, N. Pusalkar, and O. C. Jenkins, "Elephants don't pack groceries: Robot task planning for low entropy belief states," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 25–32, 2021.

[132] A. Adu-Bredu, N. Devraj, P.-H. Lin, Z. Zeng, and O. C. Jenkins, "Probabilistic inference in planning for partially observable long horizon problems," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3154–3161, IEEE, 2021.

[133] C. Finn and S. Levine, "Deep visual foresight for planning robot motion," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2786–2793, IEEE, 2017.

[134] P. Tisnikar, L. Wachowiak, G. Canal, A. Coles, M. Leonetti, and O. Celiktutan, "Towards autonomous collaborative robots that adapt and explain," in *IEEE ICRA 2022 Workshop on Prediction and Anticipation Reasoning in Human Robot Interaction*, 2022.

[135] T. Iqbal, S. Rack, and L. D. Riek, "Movement coordination in human–robot teams: a dynamical systems approach," *IEEE Transactions on Robotics*, vol. 32, no. 4, pp. 909–919, 2016.

[136] H. Green, M. Islam, S. Ali, and T. Iqbal, "ispy a humorous robot: Evaluating the perceptions of humor types in a robot partner," in *Proceedings of AAAI Spring Symposium on Putting AI in the Critical Loop: Assured Trust and Autonomy in Human-Machine Teams*, 2022.

[137] H. N. Green, M. M. Islam, S. Ali, and T. Iqbal, "Who's laughing nao? examining perceptions of failure in a humorous robot partner," in *ACM/IEEE HRI*, 2022.

[138] T. Iqbal and L. D. Riek, "Human-robot teaming: Approaches from joint action and dynamical systems," *Humanoid robotics: A reference*, pp. 2293–2312, 2017.

[139] M. Sabokrou, M. PourReza, M. Fayyaz, R. Entezari, M. Fathy, J. Gall, and E. Adeli, "Avid: Adversarial visual irregularity detection," in *Asian Conference on Computer Vision*, pp. 488–505, 2019.

[140] L. D. Riek, "Healthcare robotics," *Communications of the ACM*, vol. 60, no. 11, pp. 68–78, 2017.

[141] A. Kubota, T. Iqbal, J. A. Shah, and L. D. Riek, "Activity recognition in manufacturing: The roles of motion capture and semg+ inertial wearables in detecting fine vs. gross motion," in *2019 ICRA*, pp. 6533–6539, IEEE, 2019.

[142] G. Knoblich and J. S. Jordan, "Action coordination in groups and individuals: learning anticipatory control.," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 29, no. 5, p. 1006, 2003.

[143] G. Knoblich, S. Butterfill, and N. Sebanz, "Psychological research on joint action: theory and data," *Psychology of learning and motivation*, vol. 54, pp. 59–101, 2011.

[144] G. Hoffman and C. Breazeal, "Anticipatory perceptual simulation for human-robot joint practice: Theory and application study.," in *AAAI*, pp. 1357–1362, 2008.

[145] G. Hoffman and C. Breazeal, "Cost-based anticipatory action selection for human–robot fluency," *IEEE transactions on robotics*, vol. 23, no. 5, pp. 952–961, 2007.

[146] T. Iqbal, M. J. Gonzales, and L. D. Riek, "A model for time-synchronized sensing and motion to support human-robot fluency," in *ACM/IEEE International Conference on Human-Robot Interaction (HRI), Workshop on Timing in HRI*, pp. 1–6, 2014.

[147] T. Iqbal and L. D. Riek, "A method for automatic detection of psychomotor entrainment," *IEEE Transactions on Affective Computing*, vol. 7, no. 1, pp. 3–16, 2016.

[148] N. Sebanz and G. Knoblich, "Prediction in joint action: What, when, and where," *Topics in cognitive science*, vol. 1, no. 2, pp. 353–367, 2009.

[149] C. Vesper, S. Butterfill, G. Knoblich, and N. Sebanz, "A minimal architecture for joint action," *Neural Networks*, vol. 23, no. 8-9, pp. 998–1003, 2010.

[150] G. Novembre, L. F. Ticini, S. Schütz-Bosbach, and P. E. Keller, "Motor simulation and the coordination of self and other in real-time joint action," *Social cognitive and affective neuroscience*, vol. 9, no. 8, pp. 1062–1068, 2014.

[151] R. D. Newman-Norlund, M. L. Noordzij, R. G. Meulenbroek, and H. Bekkering, "Exploring the brain basis of joint action: co-ordination of actions, goals and intentions," *Social Neuroscience*, vol. 2, no. 1, pp. 48–65, 2007.

[152] Q. Kong, Z. Wu, Z. Deng, M. Klinkigt, B. Tong, and T. Murakami, "MMAct: A large-scale dataset for cross modal human action understanding," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 8658–8667, 2019.

[153] J. Liu, A. Shahroudy, M. L. Perez, G. Wang, L.-Y. Duan, and A. K. Chichung, "Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding," *IEEE transactions on PAMI*, 2019.

[154] M. M. Islam, M. S. Yasar, and T. Iqbal, "MAVEN: A memory augmented recurrent approach for multimodal fusion," in *IEEE Transaction on Multimedia*, 2022.

[155] M. M. Islam and T. Iqbal, "Multi-gat: A graphical attention-based hierarchical multimodal representation learning approach for human activity recognition," in *IEEE RA-L*, 2021.

[156] C. Chen, R. Jafari, and N. Kehtarnavaz, "Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor," in *2015 IEEE ICIP*, pp. 168–172, Sep. 2015.

[157] S. Samyoun*, M. M. Islam*, T. Iqbal, and J. Stankovic, "M3Sense: Affect-agnostic multitask representation learning using multimodal wearable sensors," in *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 2022.

[158] E. Sheppard and K. S. Lohan, "Multimodal representation learning for human robot interaction," in *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 445–446, 2020.

[159] Y. Chen, Q. Li, D. Kong, Y. L. Kei, S.-C. Zhu, T. Gao, Y. Zhu, and S. Huang, "Yourefit: Embodied reference understanding with language and gesture," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1385–1395, 2021.

[160] H. Tan and M. Bansal, "LXMERT: Learning cross-modality encoder representations from transformers," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, (Hong Kong, China), pp. 5100–5111, Association for Computational Linguistics, Nov. 2019.

[161] M. M. Islam, R. Mirzaiee, A. Gladstone, H. Green, and T. Iqbal, "CAESAR: A multimodal simulator for generating embodied relationship grounding dataset," in *NeurIPS [Under-Review]*, 2022.

[162] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," in *Advances in Neural Information Processing Systems*, 2019.

[163] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, "Visualbert: A simple and performant baseline for vision and language," *arXiv preprint arXiv:1908.03557*, 2019.

[164] M. A. Meredith and B. E. Stein, "Visual, auditory, and somatosensory convergence on cells in superior colliculus results in multisensory integration," *Journal of Neurophysiology*, vol. 56, no. 3, pp. 640–662, 1986.

[165] M. T. Wallace and B. E. Stein, "Development of multisensory neurons and multisensory integration in cat superior colliculus," *Journal of Neuroscience*, vol. 17, no. 7, pp. 2429–2444, 1997.

[166] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *ICML*, 2011.

[167] C. Spence, *Multisensory Perception*, pp. 1–56. American Cancer Society, 2018.

[168] W. Guo, J. Wang, and S. Wang, "Deep multimodal representation learning: A survey," *IEEE Access*, vol. 7, pp. 63373–63394, 2019.

[169] D. R. Forsyth, *Group dynamics*. Cengage Learning, 2018.

[170] J. Y. Zhang, P. Felsen, A. Kanazawa, and J. Malik, "Predicting 3d human dynamics from video," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7114–7123, 2019.

[171] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *North American Chapter of the Association for Computational Linguistics*, 2019.

[172] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *arXiv preprint arXiv:1906.08237*, 2019.

[173] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[174] F. Xiao, Y. J. Lee, K. Grauman, J. Malik, and C. Feichtenhofer, "Audiovisual slowfast networks for video recognition," *arXiv preprint arXiv:2001.08740*, 2020.

[175] E. Kazakos, A. Nagrani, A. Zisserman, and D. Damen, "Epic-fusion: Audio-visual temporal binding for egocentric action recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5492–5501, 2019.

[176] H. Alwassel, D. Mahajan, B. Korbar, L. Torresani, B. Ghanem, and D. Tran, "Self-supervised learning by cross-modal audio-video clustering," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[177] S. Münzner, P. Schmidt, A. Reiss, M. Hanselmann, R. Stiefelhagen, and R. Dürichen, "Cnn-based sensor fusion techniques for multimodal human activity recognition," in *Proceedings of the 2017 ACM ISWC*, p. 158–165, 2017.

[178] D. Ramachandram and G. W. Taylor, "Deep multimodal learning: A survey on recent advances and trends," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 96–108, 2017.

[179] H. R. V. Joze, A. Shaban, M. L. Iuzzolino, and K. Koishida, "MMTM: Multimodal transfer module for cnn fusion," in *CVPR*, 2020.

[180] T. Baltrušaitis, C. Ahuja, and L. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 423–443, 2019.

[181] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Spatiotemporal residual networks for video action recognition," in *Proceedings of the 30th NeurIPS'16*, (Red Hook, NY, USA), p. 3476–3484, Curran Associates Inc., 2016.

[182] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *NeurIPS*, pp. 568–576, 2014.

[183] J.-M. Perez-Rua, V. Vielzeuf, S. Pateux, M. Baccouche, and F. Jurie, "Mfas: Multimodal fusion architecture search," in *CVPR*, June 2019.

[184] A. Roitberg, N. Somani, A. Perzylo, M. Rickert, and A. Knoll, "Multimodal human activity recognition for industrial manufacturing processes in robotic workcells," in *ICMI*, 2015.

[185] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Spatiotemporal multiplier networks for video action recognition," in *CVPR*, pp. 4768–4777, 2017.

[186] S. Zhang, Y. Yang, J. Xiao, X. Liu, Y. Yang, D. Xie, and Y. Zhuang, "Fusing geometric features for skeleton-based action recognition using multilayer lstm networks," *IEEE Transactions on Multimedia*, vol. 20, no. 9, pp. 2330–2343, 2018.

[187] G. Liu, J. Qian, F. Wen, X. Zhu, R. Ying, and P. Liu, "Action recognition based on 3d skeleton and rgb frame fusion," in *2019 IEEE/RSJ IROS*, pp. 258–264, Nov 2019.

[188] M. A. Lee, Y. Zhu, P. Zachares, M. Tan, K. Srinivasan, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, "Making sense of vision and touch: Learning multimodal representations for contact-rich tasks," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 582–596, 2020.

[189] Y. Bekiroglu, R. Detry, and D. Kragic, "Learning tactile characterizations of object- and pose-specific grasps," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1554–1560, 2011.

[190] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine, "More than a feeling: Learning to grasp and regrasp using vision and touch," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3300–3307, 2018.

[191] Y. Gao, L. A. Hendricks, K. J. Kuchenbecker, and T. Darrell, "Deep learning for tactile understanding from visual and haptic data," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 536–543, IEEE, 2016.

[192] J. Sinapov, C. Schenck, and A. Stoytchev, "Learning relational object categories using behavioral exploration and multimodal perception," in *2014 IEEE international conference on robotics and automation (ICRA)*, pp. 5691–5698, IEEE, 2014.

[193] L. Xia, C. Chen, and J. Aggarwal, "View invariant human action recognition using histograms of 3d joints," in *CVPRW*, pp. 20–27, IEEE, 2012.

[194] R. M. Aronson and H. Admoni, "Gaze complements control input for goal prediction during assisted teleoperation," in *Robotics science and systems*, 2022.

[195] C. Z. Qiao, M. Sakr, K. Muelling, and H. Admoni, "Learning from demonstration for real-time user goal prediction and shared assistive control," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3270–3275, IEEE, 2021.

[196] H. R. V. Joze, A. Shaban, M. L. Iuzzolino, and K. Koishida, "Mmtm: Multimodal transfer module for cnn fusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13289–13299, 2020.

[197] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[198] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[199] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems* (Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, eds.), vol. 27, Curran Associates, Inc., 2014.

[200] A. Xie, D. Losey, R. Tolsma, C. Finn, and D. Sadigh, "Learning latent representations to influence multi-agent interaction," in *Proceedings of the 2020 Conference on Robot Learning* (J. Kober, F. Ramos, and C. Tomlin, eds.), vol. 155 of *Proceedings of Machine Learning Research*, pp. 575–588, PMLR, 16–18 Nov 2021.

[201] R. Shadmehr and F. A. Mussa-Ivaldi, "Adaptive representation of dynamics during learning of a motor task," *Journal of neuroscience*, vol. 14, no. 5, pp. 3208–3224, 1994.

[202] E. Todorov and M. I. Jordan, "Optimal feedback control as a theory of motor coordination," *Nature neuroscience*, vol. 5, no. 11, pp. 1226–1235, 2002.

[203] C. Mavrogiannis, K. Balasubramanian, S. Poddar, A. Gandra, and S. S. Srinivasa, "Winding through: Crowd navigation via topological invariance," *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 121–128, 2022.

[204] C. I. Mavrogiannis, V. Blukis, and R. A. Knepper, "Socially competent navigation planning by deep learning of multi-agent path topologies," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6817–6824, IEEE, 2017.

[205] V. V. Unhelkar, S. Li, and J. A. Shah, "Decision-making for bidirectional communication in sequential human-robot collaborative tasks," in *2020 15th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 329–341, IEEE, 2020.

[206] V. V. Unhelkar and J. A. Shah, "Learning models of sequential decision-making without complete state specification using bayesian nonparametric inference and active querying," 2018.

[207] B. A. Newman, R. M. Aronson, S. S. Srinivasa, K. Kitani, and H. Admoni, "Harmonic: A multimodal dataset of assistive human–robot collaboration," *The International Journal of Robotics Research*, vol. 41, no. 1, pp. 3–11, 2022.

[208] O. Celiktutan, E. Skordos, and H. Gunes, "Multimodal human-human-robot interactions (mhhri) dataset for studying personality and engagement," *IEEE Transactions on Affective Computing*, vol. 10, no. 4, pp. 484–497, 2017.

[209] A. Ben-Youssef, C. Clavel, S. Essid, M. Bilac, M. Chamoux, and A. Lim, "Ue-hri: a new dataset for the study of user engagement in spontaneous human-robot interactions," in *Proceedings of the 19th ACM international conference on multimodal interaction*, pp. 464–472, 2017.

[210] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[211] A. Kuznetsov, P. Shvechikov, A. Grishin, and D. Vetrov, "Controlling overestimation bias with truncated mixture of continuous distributional quantile critics," in *International Conference on Machine Learning*, pp. 5556–5566, PMLR, 2020.

[212] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6299–6308, 2017.

[213] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International journal of computer vision*, vol. 123, pp. 32–73, 2017.

[214] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755, Springer, 2014.

[215] J. Ngiam, V. Vasudevan, B. Caine, Z. Zhang, H.-T. L. Chiang, J. Ling, R. Roelofs, A. Bewley, C. Liu, A. Venugopal, D. J. Weiss, B. Sapp, Z. Chen, and J. Shlens, "Scene transformer: A unified architecture for predicting future trajectories of multiple agents," in *International Conference on Learning Representations*, 2022.

[216] N. Inc, "Optitrack." https://optitrack.com/, 2023. [Online; accessed 29 September 2023].

[217] S. Inc, "Stereolabs." https://www.stereolabs.com/, 2023. [Online; accessed 29 September 2023].

[218] P. L. GmbH, "Pupil labs." https://pupil-labs.com/, 2023. [Online; accessed 29 September 2023].

[219] I. Fetch Robotics, "Fetch robotics." https://fetchrobotics.com/, 2023. [Online; accessed 29 September 2023].

[220] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

[221] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021.

[222] Y.-H. H. Tsai, S. Bai, P. P. Liang, J. Z. Kolter, L.-P. Morency, and R. Salakhutdinov, "Multimodal transformer for unaligned multimodal language sequences," in *Proceedings of the conference. Association for Computational Linguistics. Meeting*, vol. 2019, p. 6558, NIH Public Access, 2019.

[223] Y.-C. Chen, L. Li, L. Yu, A. El Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu, "Uniter: Universal image-text representation learning," in *European conference on computer vision*, pp. 104–120, Springer, 2020.

[224] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[225] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *International Conference on Learning Representations*, 2017.

[226] A. Biswas, H. Admoni, and A. Steinfeld, "Fast on-board 3d torso pose recovery and forecasting," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, IEEE, 2019.

[227] T. Iqbal and L. D. Riek, "Role distribution in synchronous human-robot joint action," in *Proc. of IEEE Symp. on Robot and Human Interactive Communication, Towards a Framework for Joint Action*, 2014.

[228] T. Darrell, M. Kloft, M. Pontil, G. Rätsch, and E. Rodner, "Machine learning with interdependent and non-identically distributed data (dagstuhl seminar 15152)," in *Dagstuhl Reports*, vol. 5, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015.

[229] T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, and N. Díaz-Rodríguez, "Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges," *Information fusion*, vol. 58, pp. 52–68, 2020.

[230] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[231] D. Kumaran, D. Hassabis, and J. L. McClelland, "What learning systems do intelligent agents need? complementary learning systems theory updated," *Trends in cognitive sciences*, vol. 20, no. 7, pp. 512–534, 2016.

[232] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

[233] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[234] F. M. Richardson and M. S. Thomas, "Critical periods and catastrophic interference effects in the development of self-organizing feature maps," *Developmental science*, vol. 11, no. 3, pp. 371–389, 2008.

[235] M. S. Thomas and M. H. Johnson, "The computational modeling of sensitive periods," *Developmental Psychobiology*, vol. 48, no. 4, p. 337, 2006.

[236] T. Kohonen, "Exploration of very large databases by self-organizing maps," in *Proceedings of international conference on neural networks (icnn'97)*, vol. 1, pp. PL1–PL6, IEEE, 1997.

[237] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.

[238] A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, S. G. Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou, *et al.*, "Hybrid computing using a neural network with dynamic external memory," *Nature*, vol. 538, no. 7626, pp. 471–476, 2016.

[239] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," *arXiv preprint arXiv:1801.06146*, 2018.

[240] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE T-PAMI*, vol. 40, no. 12, pp. 2935–2947, 2017.

[241] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[242] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[243] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*, vol. 24, pp. 109–165, Elsevier, 1989.

[244] J. C. Schlimmer and D. Fisher, "A case study of incremental concept induction," in *AAAI*, vol. 86, pp. 496–501, 1986.

[245] S. Thrun, "Lifelong learning algorithms," in *Learning to learn*, pp. 181–209, Springer, 1998.

[246] R. S. Sutton, S. D. Whitehead, *et al.*, "Online learning with random representations," in *Proceedings of the Tenth International Conference on Machine Learning*, pp. 314–321, 2014.

[247] D. Hassabis, D. Kumaran, C. Summerfield, and M. Botvinick, "Neuroscience-inspired artificial intelligence," *Neuron*, vol. 95, no. 2, pp. 245–258, 2017.

[248] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[249] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *International Conference on Machine Learning*, pp. 3987–3995, PMLR, 2017.

[250] J. Schwarz, W. Czarnecki, J. Luketina, A. Grabska-Barwinska, Y. W. Teh, R. Pascanu, and R. Hadsell, "Progress & compress: A scalable framework for continual learning," in *International Conference on Machine Learning*, pp. 4528–4537, PMLR, 2018.

[251] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," *arXiv preprint arXiv:1606.04671*, 2016.

[252] X. Li, Y. Zhou, T. Wu, R. Socher, and C. Xiong, "Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting," in *International Conference on Machine Learning*, pp. 3925–3934, PMLR, 2019.

[253] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

[254] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," *Advances in neural information processing systems*, vol. 30, pp. 6467–6476, 2017.

[255] A. Benjamin, D. Rolnick, and K. Kording, "Measuring and regularizing networks in function space," in *International Conference on Learning Representations*, 2018.

[256] M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, and G. Tesauro, "Learning to learn without forgetting by maximizing transfer and minimizing interference," in *In International Conference on Learning Representations (ICLR)*, 2019.

[257] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with a-gem," in *International Conference on Learning Representations*, 2018.

[258] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara, "Dark experience for general continual learning: a strong, simple baseline," in *NeurIPS*, 2020.

[259] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, pp. 1597–1607, PMLR, 2020.

[260] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[261] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[262] Z. Dai, B. Cai, Y. Lin, and J. Chen, "Unimoco: Unsupervised, semi-supervised and full-supervised visual representation learning," *arXiv preprint arXiv:2103.10773*, 2021.

[263] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

[264] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 7, no. 04, pp. 669–688, 1993.

[265] P. Chrabaszcz, I. Loshchilov, and F. Hutter, "A downsampled variant of imagenet as an alternative to the cifar datasets," *arXiv preprint arXiv:1707.08819*, 2017.

[266] Y.-C. Hsu, Y.-C. Liu, A. Ramasamy, and Z. Kira, "Re-evaluating continual learning scenarios: A categorization and case for strong baselines," in *NeurIPS Continual learning Workshop*, 2018.

[267] M. Farajtabar, N. Azizan, A. Mott, and A. Li, "Orthogonal gradient descent for continual learning," in *International Conference on Artificial Intelligence and Statistics*, pp. 3762–3773, PMLR, 2020.

[268] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 139–154, 2018.

[269] G. Saha, I. Garg, A. Ankit, and K. Roy, "Space: Structured compression and sharing of representational space for continual learning," *IEEE Access*, vol. 9, pp. 150480–150494, 2021.

[270] S. S. Sarwar, A. Ankit, and K. Roy, "Incremental learning in deep convolutional neural networks using partial network sharing," *IEEE Access*, vol. 8, pp. 4615–4628, 2019.

[271] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, "Lifelong learning with dynamically expandable networks," in *Sixth International Conference on Learning Representations*, ICLR, 2018.

[272] J. Yoon, S. Kim, E. Yang, and S. J. Hwang, "Scalable and order-robust continual learning with additive parameter decomposition," in *International Conference on Learning Representations*, 2019.

[273] A. Chaudhry, A. Gordo, P. Dokania, P. Torr, and D. Lopez-Paz, "Using hindsight to anchor past knowledge in continual learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 6993–7001, 2021.

[274] G. Sokar, D. C. Mocanu, and M. Pechenizkiy, "Self-attention meta-learner for continual learning," in *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 1658–1660, 2021.

[275] C. et al., "Co^{2}L: Contrastive continual learning," in *ICCV*, 2021.

[276] C. M. de Melo, S. Marsella, and J. Gratch, "Increasing fairness by delegating decisions to autonomous agents.," in *AAMAS*, pp. 419–425, 2017.

[277] E. Aghapour and N. Ayanian, "Double meta-learning for data efficient policy optimization in non-stationary environments," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9935–9942, IEEE, 2021.

[278] H. Ma, W. Hönig, T. S. Kumar, N. Ayanian, and S. Koenig, "Lifelong path planning with kinematic constraints for multi-agent pickup and delivery," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 7651–7658, 2019.

[279] A. Ayub and A. R. Wagner, "Tell me what this is: few-shot incremental object learning by a robot," in *IEEE/RSJ IROS*, 2020.

[280] A. Ayub and A. R. Wagner, "Continual learning of visual concepts for robots through limited supervision," in *Companion of the 2021 ACM/IEEE HRI*, 2021.

[281] N. Churamani, S. Kalkan, and H. Gunes, "Continual learning for affective robotics: Why, what and how?," in *29th IEEE RO-MAN*, 2020.

[282] A. Daruna, M. Gupta, M. Sridharan, and S. Chernova, "Continual learning of knowledge graph embeddings," *IEEE RA-L*, 2021.

[283] L. Knoedler, C. Salmi, H. Zhu, B. Brito, and J. Alonso-Mora, "Improving pedestrian prediction models with self-supervised continual learning," *IEEE RA-L*, 2022.

[284] L. Pellegrini, G. Graffieti, V. Lomonaco, and D. Maltoni, "Latent replay for real-time continual learning," in *IEEE/RSJ IROS*, 2020.

[285] J. Gallardo, T. L. Hayes, C. Kanan, and C. Tech, "Self-supervised training enhances online continual learning," 2021.

[286] Q. Pham, C. Liu, and S. Hoi, "Dualnet: Continual learning, fast and slow," *NeurIPS*, 2021.

[287] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, B. Piot, k. kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent - a new approach to self-supervised learning," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 21271–21284, Curran Associates, Inc., 2020.

[288] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *arXiv preprint arXiv:2003.04297*, 2020.

[289] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio, "Gradient based sample selection for online continual learning.," *NeurIPS*, pp. 11816–11825, 2019.

[290] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[291] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[292] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[293] V. V. Unhelkar, S. Dörr, A. Bubeck, P. A. Lasota, J. Perez, H. C. Siu, J. C. Boerkoel, Q. Tyroller, J. Bix, S. Bartscher, *et al.*, "Mobile robots for moving-floor assembly lines: Design, evaluation, and deployment," *IEEE Robotics & Automation Magazine*, vol. 25, no. 2, pp. 72–81, 2018.

[294] J. Ibarz, J. Tan, C. Finn, M. Kalakrishnan, P. Pastor, and S. Levine, "How to train your robot with deep reinforcement learning: lessons we have learned," *The International Journal of Robotics Research*, vol. 40, no. 4-5, pp. 698–721, 2021.

[295] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, *et al.*, "Grandmaster level in starcraft ii using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[296] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[297] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. Pieter Abbeel, and W. Zaremba, "Hindsight experience replay," *Advances in neural information processing systems*, vol. 30, 2017.

[298] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*, pp. 1861–1870, PMLR, 2018.

[299] C. Finn, S. Levine, and P. Abbeel, "Guided cost learning: Deep inverse optimal control via policy optimization," in *International conference on machine learning*, pp. 49–58, PMLR, 2016.

[300] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac gym: High performance GPU based physics simulation for robot learning," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[301] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 3803–3810, IEEE, 2018.

[302] A. Xie, J. Harrison, and C. Finn, "Deep reinforcement learning amidst continual structured non-stationarity," in *Proceedings of the 38th International Conference on Machine Learning* (M. Meila and T. Zhang, eds.), vol. 139 of *Proceedings of Machine Learning Research*, pp. 11393–11403, PMLR, 18–24 Jul 2021.

[303] M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Silver, and K. Kavukcuoglu, "Reinforcement learning with unsupervised auxiliary tasks," in *International Conference on Learning Representations*, 2017.

[304] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, *et al.*, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[305] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International journal of robotics research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[306] T. Haarnoja, V. Pong, A. Zhou, M. Dalal, P. Abbeel, and S. Levine, "Composable deep reinforcement learning for robotic manipulation," in *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 6244–6251, IEEE, 2018.

[307] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *ICLR*, 2016.

[308] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 3389–3396, IEEE, 2017.

[309] M. Janner, J. Fu, M. Zhang, and S. Levine, "When to trust your model: Model-based policy optimization," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[310] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey, *et al.*, "Maximum entropy inverse reinforcement learning.," in *Aaai*, vol. 8, pp. 1433–1438, Chicago, IL, USA, 2008.

[311] S.-Y. Chen, Y. Yu, Q. Da, J. Tan, H.-K. Huang, and H.-H. Tang, "Stabilizing reinforcement learning in dynamic environment with application to online recommendation," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1187–1196, 2018.

[312] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3733–3742, 2018.

[313] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," in *International Conference on Learning Representations*, 2018.

[314] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," *Advances in Neural Information Processing Systems*, vol. 32, pp. 15535–15545, 2019.

[315] I. Misra and L. v. d. Maaten, "Self-supervised learning of pretext-invariant representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6707–6717, 2020.

[316] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, "Reinforcement learning with deep energy-based policies," in *International conference on machine learning*, pp. 1352–1361, PMLR, 2017.

[317] X. Chen, C. Wang, Z. Zhou, and K. W. Ross, "Randomized ensembled double q-learning: Learning fast without a model," in *International Conference on Learning Representations*, 2021.

[318] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in neural information processing systems*, vol. 30, 2017.

[319] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.

[320] P. Stone and M. Veloso, "Multiagent systems: A survey from a machine learning perspective," *Autonomous Robots*, vol. 8, pp. 345–383, 2000.

[321] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, 2008.

[322] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *Handbook of reinforcement learning and control*, pp. 321–384, 2021.

[323] M. Samvelyan, T. Rashid, C. S. De Witt, G. Farquhar, N. Nardelli, T. G. Rudner, C.-M. Hung, P. H. Torr, J. Foerster, and S. Whiteson, "The starcraft multi-agent challenge," *arXiv preprint arXiv:1902.04043*, 2019.

[324] B. Baker, I. Kanitscheider, T. Markov, Y. Wu, G. Powell, B. McGrew, and I. Mordatch, "Emergent tool use from multi-agent autocurricula," *arXiv preprint arXiv:1909.07528*, 2019.

[325] R. E. Wang, M. Everett, and J. P. How, "R-maddpg for partially observable environments and limited communication," *arXiv preprint arXiv:2002.06684*, 2020.

[326] Y. Lee, J. Yang, and J. J. Lim, "Learning to coordinate manipulation skills via skill behavior diversification," in *International conference on learning representations*, 2019.

[327] B. Peng, T. Rashid, C. Schroeder de Witt, P.-A. Kamienny, P. Torr, W. Böhmer, and S. Whiteson, "Facmac: Factored multi-agent centralised policy gradients," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12208–12221, 2021.

[328] L. Pan, L. Huang, T. Ma, and H. Xu, "Plan better amid conservatism: Offline multi-agent reinforcement learning with actor rectification," in *International Conference on Machine Learning*, pp. 17221–17237, PMLR, 2022.

[329] F. P. Such, V. Madhavan, E. Conti, J. Lehman, K. O. Stanley, and J. Clune, "Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning," *arXiv preprint arXiv:1712.06567*, 2017.

[330] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, "Evolution strategies as a scalable alternative to reinforcement learning," *arXiv preprint arXiv:1703.03864*, 2017.

[331] F. A. Oliehoek, C. Amato, *et al.*, *A concise introduction to decentralized POMDPs*, vol. 1. Springer, 2016.

[332] M. Hausknecht and P. Stone, "Deep recurrent q-learning for partially observable mdps," in *2015 aaai fall symposium series*, 2015.

[333] S. Iqbal and F. Sha, "Actor-attention-critic for multi-agent reinforcement learning," in *International conference on machine learning*, pp. 2961–2970, PMLR, 2019.

[334] J. Foerster, I. A. Assael, N. De Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," *Advances in neural information processing systems*, vol. 29, 2016.

[335] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, *et al.*, "Value-decomposition networks for cooperative multi-agent learning," *arXiv preprint arXiv:1706.05296*, 2017.

[336] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Conference on Robot Learning*, pp. 785–799, PMLR, 2023.

[337] P. Goyal, S. Niekum, and R. J. Mooney, "Using natural language for reward shaping in reinforcement learning," *arXiv preprint arXiv:1903.02020*, 2019.

[338] A. Jaegle, F. Gimeno, A. Brock, O. Vinyals, A. Zisserman, and J. Carreira, "Perceiver: General perception with iterative attention," in *International conference on machine learning*, pp. 4651–4664, PMLR, 2021.

[339] A. Goyal, J. Xu, Y. Guo, V. Blukis, Y.-W. Chao, and D. Fox, "Rvt: Robotic view transformer for 3d object manipulation," in *Conference on Robot Learning*, pp. 694–710, PMLR, 2023.