Learning and simulation of sport strategies (boxing) for virtual reality training

Mohamed Younes

COLLEGE MATHS, TELECOMS
DOCTORAL INFORMATIQUE, SIGNAL
BRETAGNE SYSTEMES, ELECTRONIQUE

Université de Rennes

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES

ÉCOLE DOCTORALE N° 601
Mathématiques, Télécommunications, Informatique, Signal, Systèmes, Électronique
Spécialité : *Informatique*

Par

« Mohamed YOUNES »

« Apprentissage et Simulation des Stratégies de Sport (la Boxe) pour l'Entrainement en Réalité Virtuelle. »

Thèse présentée et soutenue à « L'université de Rennes », le « 24/05/2024 »
Unité de recherche : INRIA

Rapporteurs avant soutenance :

Mohamed DAOUDI Professeur, Institut Mines Telecom, Lille
Céline LOSCOS Principal research engineer, Huawei, France

Composition du Jury :

Présidente :	Maud MARCHAL	Professeure, INSA Rennes
Examinateurs :	Hubert SHUM	Associate Professor, Durham University, UK
	Libin LIU	Assistant Professor, Peking University, China
	Céline LOSCOS	Principal research engineer, Huawei, France
	Mohamed DAOUDI	Professeur, Institut Mines Telecom, Lille
Dir. de thèse :	Franck MULTON	Directeur de Recherche, Inria Rennes

Invité(s) :

Richard KULPA Professeur de l'Université de Rennes 2
Ewa KIJAK Maîtresse de conférences, Université de Rennes
Simon MALINOWSKI Maître de conférences, Université de Rennes

Résumé

Le monde du sport a toujours été un domaine d'étude fascinant en raison de sa nature complexe, en particulier lorsqu'il s'agit des stratégies employées par les athlètes pendant la compétition. Les stratégies sportives, en particulier celles exécutées dans les sports de combat ou les arts martiaux, comme la boxe, impliquent des interactions complexes entre les combattants qui nécessitent des capacités d'anticipation élevées et des mouvements rapides.

La boxe exige la maîtrise de plusieurs compétences secondaires, ce qui rend difficile un entraînement efficace sans une orientation et une pratique appropriées. Ces sous-compétences comprennent le jeu de jambes, l'équilibre, la vitesse, la puissance, la technique et la planification stratégique, qui doivent toutes être intégrées de manière transparente pendant les combats [82]. Le perfectionnement de ces sous-compétences par des séances d'entraînement et de sparring répétitives pour les athlètes d'élite les conduit souvent à se blesser, ce qui réduit considérablement leurs performances lors des combats officiels [82, 85, 32]. Ces questions ouvrent de nouvelles perspectives sur la création de nouvelles modalités d'analyse et d'entraînement des athlètes.

Ces dernières années, le développement d'environnements virtuels à des fins d'entraînement a suscité un intérêt croissant, notamment grâce à la technologie de la réalité virtuelle (RV). La technologie de la réalité virtuelle offre une approche révolutionnaire de l'entraînement sportif en fournissant un environnement contrôlé, où les stimuli peuvent être standardisés, ajustés et même dépasser les conditions de la vie réelle. Cette caractéristique unique permet aux entraîneurs et aux formateurs de concevoir des programmes d'entraînement personnalisés qui complètent les méthodes conventionnelles. Par exemple, les simulations de réalité virtuelle peuvent aider les athlètes à améliorer leurs compétences défensives sans les exposer à des blessures potentielles dues à des coups ou des impacts répétés [104]. Les avantages de l'utilisation de l'entraînement en réalité virtuelle vont bien au-delà de la simple prévention des dommages physiques. Au fur et à mesure que les athlètes progressent dans leur carrière, il arrive un moment où ils atteignent un plateau et ont besoin d'une amélioration continue pour rester compétitifs. Cependant, dépasser leurs limites physiques devient de plus en plus difficile. Grâce à la réalité virtuelle,

les athlètes peuvent s'engager dans de nouvelles modalités d'entraînement qui mettent à l'épreuve leurs capacités cognitives et leur perception, ce qui leur permet de conserver leur avantage même s'ils ont atteint leur apogée physique [61].

Dans le contexte de la boxe, l'une des principales préoccupations des entraîneurs et des officiels est le développement des capacités d'anticipation des boxeurs dans les situations défensives. Les méthodes d'entraînement traditionnelles impliquent souvent d'encaisser de nombreux coups de poing pour développer la résistance et les réflexes - une stratégie qui présente des risques inhérents pour la santé et l'état de l'athlète. Les solutions de réalité virtuelle répondent à ce problème en permettant aux boxeurs de se concentrer uniquement sur le traitement des données critiques concernant leurs adversaires, telles que le langage corporel, le jeu de jambes et les schémas d'attaque. Cette conscience accrue leur permettra en fin de compte de prédire avec précision les assauts à venir et d'y répondre par des contre-mesures efficaces. Ces objectifs constituent les fondements d'un certain nombre de projets de recherche sur l'entraînement et le sport, tels que le projet REVEA [61], mis en place dans la perspective des Jeux olympiques et paralympiques de Paris 2024, et coparrainé par le ministère français de l'enseignement supérieur, de la recherche et de l'innovation et le ministère de l'éducation nationale, de la jeunesse et des sports (appel à propositions PPR « Sport Haute Performance » du financement France 2030). Ce projet met en œuvre un système d'entraînement en réalité virtuelle pour aider les athlètes de boxe à améliorer des compétences secondaires telles que la vitesse, la coordination motrice et l'entraînement de la force. Ce paradigme implique principalement l'interaction avec un adversaire virtuel pour s'entraîner. Le comportement de cet adversaire virtuel est dicté par des scénarios fixes mis en œuvre à l'aide de moteurs de jeu commerciaux tels que Unity, où les mouvements des humains virtuels sont stockés dans une base de données construite à l'aide de la capture de mouvements, qui sont récupérés et joués selon le scénario choisi. Cependant, ce comportement statique de l'adversaire virtuel peut affecter l'immersion de l'athlète, car le mouvement rejoué non réactif peut souvent être perçu comme non réaliste. De plus, la prévisibilité du comportement statique de l'adversaire virtuel peut réduire les avantages de l'entraînement en réalité virtuelle, car l'athlète peut exploiter les faiblesses de l'adversaire virtuel après quelques interactions. Néanmoins, la conception de scénarios plus adaptatifs et diversifiés pour les interactions entre l'athlète et l'adversaire virtuel représente un défi et prend du temps.

Nous émettons l'hypothèse que la simulation de comportements interactifs réalistes peut grandement améliorer les performances et individualiser l'entraînement, en modé-

lisant les interactions entre les boxeurs à l'aide de techniques basées sur les données, et en tirant parti de l'animation de personnages basée sur la physique. Cela permettrait d'avoir un adversaire virtuel qui reproduit le comportement réactif d'adversaires réels, comme un futur adversaire, en utilisant des données antérieures le concernant, sous la forme d'enregistrements vidéo de combats précédents. Cela permettrait également d'atténuer les limites du système d'entraînement en réalité virtuelle évoqué plus haut, car les mouvements de l'adversaire virtuel seraient plus réalistes et réagiraient mieux aux actions de l'utilisateur. De plus, son comportement serait appris automatiquement à partir de données au lieu de dépendre d'un travail manuel.

Les Objectifs

L'objectif de cette thèse est d'apprendre et de simuler des stratégies d'arts martiaux et d'imiter les interactions de combat, afin d'entraîner les athlètes dans un environnement d'entraînement en réalité virtuelle. En concevant un modèle de stratégie pour la boxe, nous pouvons proposer un ensemble de réponses plausibles de la part d'un adversaire virtuel en temps réel, en fonction de l'action actuelle de l'utilisateur et de l'historique de ses mouvements. Il s'agit de modéliser les interactions entre les boxeurs sur la base de sources de données annotées ou non, qui peuvent prendre la forme d'enregistrements vidéo de combats antérieurs ou de données de capture de mouvement. Il s'agit également d'animer l'adversaire pour qu'il effectue des mouvements réalistes dans l'environnement virtuel. Par réaliste, on entend ici : « similaire à un mouvement et à une stratégie que le boxeur réel ciblé aurait effectués dans les mêmes conditions ». L'adversaire virtuel doit être intelligent dans ses décisions et contrôlable. Il permet donc aux utilisateurs de faire l'expérience d'une simulation réaliste d'entraînement à la boxe et d'améliorer leurs compétences, tout en les préparant à se défendre face à des stratégies et des styles de boxe spécifiques. En effet, l'objectif de l'imitation d'interaction n'est pas d'avoir un adversaire optimal dans le but de gagner contre l'utilisateur, mais un adversaire qui imite un adversaire éventuel et reproduit son comportement. De tels adversaires devraient non seulement imiter les actions humaines, mais aussi s'adapter dynamiquement aux données de l'utilisateur, créant ainsi des rencontres plus nuancées et stochastiques qui reflètent mieux les stratégies du monde réel induites par les vrais athlètes.

Vue d'ensemble

Dans cette thèse, nous explorons l'extraction et la simulation du comportement interactif des combattants à partir de données de mouvement. A cette fin, nous proposons d'utiliser des méthodes d'estimation du mouvement humain à partir de vidéos, l'apprentissage par renforcement basé sur l'imitation, et la simulation de personnages basée sur la physique. L'organisation de cette thèse est la suivante.

Tout d'abord, nous commençons par introduire plusieurs concepts utilisés dans cette thèse au chapitre 1 : à savoir l'apprentissage par renforcement, l'apprentissage par imitation et la simulation basée sur la physique.

Dans le chapitre 2, nous évaluons les capacités de différentes catégories de méthodes actuelles de pose humaine en 2D, et leur précision dans l'extraction des informations de pose à partir de vidéos RVB de boxeurs dans des conditions d'enregistrement difficiles, et de mouvements de boxe rapides.

Dans le chapitre 3, nous abordons le problème de la simulation plausible du comportement interactif des combattants à partir de données de mouvement. Nous proposons une approche pour imiter les interactions et les mouvements de plusieurs personnages basés sur la physique, à partir de données de mouvement non structurées. Nous nous sommes concentrés uniquement sur l'imitation de l'interaction à partir de données non structurées de deux combattants pratiquant le light shadow boxing avec un minimum de contacts physiques (présenté sous forme de poster à SIGGRAPH Asia 2022 [151]). Ensuite, nous avons étendu l'approche à d'autres données d'interaction impliquant la boxe avec contact physique, ainsi qu'à une autre activité de combat. Ce travail a été présenté au Symposium on Computer Animation (SCA'23) et publié dans le journal Proceedings of the ACM in Computer Graphics and Interactive Techniques (PACMCGIT) [152].

Enfin, nous concluons le chapitre 3.5 en rappelant les contributions et en identifiant certaines limites. Ensuite, nous avons proposé quelques extensions et perspectives potentielles à ce travail.

Analyse comparative des méthodes d'estimation de la pose en 2D pour les sports de combat

Dans ce chapitre, nous avons analysé les performances des méthodes HPE (Human Pose Estimation) dans un domaine sportif impliquant des mouvements spécifiques de boxe

rapide et nous avons proposé un protocole d'évaluation pour ce benchmark. Les travaux antérieurs ont évalué les HPE soit à l'aide de mesures globales, soit en les évaluant sur des activités humaines générales. Dans ce travail, nous avons présenté des comparaisons de performance plus fines appliquées aux sports de combat, et en particulier à la boxe.

L'un des aspects les plus pertinents de ce travail est l'évaluation complète du potentiel des méthodes HPE qui peuvent être utiles dans l'analyse des sports à rythme rapide et la reconnaissance des activités. Ce contexte a motivé notre choix d'analyser leurs performances par rapport à différents mouvements propres à la boxe. L'une des conclusions importantes est que certaines positions articulaires sont mieux estimées que d'autres, selon qu'elles font face à la caméra, qu'elles sont occultées par le corps du boxeur ou qu'elles sont impliquées dans des actions rapides.

La conclusion générale de ce travail est que les approches HPE descendantes (détection des personnes puis estimation individuelle des articulations) sont plus performantes que les approches HPE ascendantes (estimation globale des articulation puis association et identification des personnes), même en présence d'un arrière-plan complexe, d'auto-occlusions et d'occlusions entre deux adversaires. Les méthodes qui affinent leurs estimations à l'aide d'informations temporelles provenant d'images adjacentes sont les plus performantes dans ces scénarios. Par conséquent, nous suggérons que les futurs travaux liés à l'analyse et aux études sportives basées sur la vidéo utilisent des approches HPE descendantes pour l'extraction de mouvement, en particulier celles qui utilisent des fenêtres temporelles pour une estimation plus précise.

Imitation des interactions à partir de la capture de mouvements

Dans cette thèse, nous nous intéressons à la génération de comportements réactifs d'agents physiquement incarnés d'une manière guidée par les données en tirant parti de techniques d'apprentissage automatique et de simulations basées sur la physique.

Étant donné les mouvements extraits de paires de combattants, notre objectif est de simuler leur interaction d'une manière physiquement plausible, et de s'assurer que la réaction de l'agent simulé face à un adversaire est similaire à la réaction du combattant correspondant. Nous abordons ce problème comme un problème d'apprentissage par imitation à partir de la démonstration de plusieurs experts en interaction. Alors que les techniques d'apprentissage par imitation [95] ont été explorées pour imiter les mouvements

et le comportement de personnages basés sur la physique en utilisant des données de capture de mouvement comme démonstrations, elles n'ont pas été explorées pour simuler le comportement interactif de plusieurs personnages.

Dans ce chapitre, nous avons proposé un système antagoniste innovant conçu pour imiter les interactions de combat complexes entre plusieurs personnages basés sur la physique, en utilisant des clips de mouvement non structurés. S'appuyant sur les fondements du cadre d'apprentissage de l'imitation antagoniste générative multi-agents [120], notre approche incorpore des adaptations cruciales pour simuler efficacement les comportements de plusieurs personnages basés sur la physique. La première amélioration significative concerne la modélisation du comportement réactif, dans laquelle nous établissons une transition entre l'observation complète actuelle, qui comprend l'auto-observation de l'agent lui-même et l'observation actuelle de l'adversaire, et l'auto-observation suivante. Cette transition permet de saisir la nature dynamique des réponses des personnages à leur adversaire, ce qui se traduit par une simulation plus plausible. En outre, nous avons conçu une stratégie d'entraînement qui englobe à la fois les mouvements simples et les priorités d'interaction. Les séquences obtenues ne se contentent pas d'imiter les mouvements de référence avec le même ordre de clip, mais présentent des comportements interactifs similaires à ceux de l'ensemble des données d'interaction en maximisant les récompenses attribuées par chaque composante. Notre approche nous a donc permis d'imiter la réaction personnalisée de combattants aux styles spécifiques. Nous pouvons également donner aux utilisateurs un certain contrôle sur la simulation, en ajoutant des récompenses spécifiques à une tâche : suivre une direction donnée, minimiser les impacts reçus ou maximiser les dommages causés aux adversaires lors de la recherche de l'action suivante, tout en continuant à imiter le style de l'ensemble de données d'interaction. Nous pourrions imaginer d'autres récompenses, telles que viser des parties spécifiques du corps de l'adversaire. Les résultats montrent que, bien que l'ensemble de données d'interaction puisse être suffisant pour apprendre des politiques d'imitation de mouvement et d'interaction, l'association d'un mouvement unique complémentaire permet de généraliser à un plus grand nombre de situations avec des mouvements réalistes. D'autre part, notre méthode peut également être utilisée pour simuler de nouveaux styles individuels ou de nouvelles activités à plusieurs personnages (escrime, danse, travail collaboratif, etc.), en réentraînant le même système mais avec de nouveaux ensembles de données à un ou plusieurs personnages.

Outre les applications susmentionnées dans le domaine de l'entraînement sportif, l'approche proposée peut potentiellement être appliquée à divers autres domaines im-

pliquant des interactions multi-agents, tels que les jeux de réalité virtuelle, la robotique et l'animation. En simulant des interactions réalistes entre des personnages virtuels, notre système peut contribuer à créer des expériences plus engageantes et immersives pour les utilisateurs.

Acknowledgement

I am deeply appreciative of the countless individuals whose unwavering support and guidance have enabled me to complete this thesis, with a special acknowledgment to my parents. I offer my most profound gratitude to my thesis advisors. Your support and patience have illuminated my path throughout this hard yet rewarding journey. The insightful conversations and exchanges we have shared have been instrumental in shaping my research into its current form.

A warm thank you goes out to my beloved family, dearest colleagues, and cherished loved ones. Your unwavering faith in my abilities and ceaseless encouragement have served as an indomitable source of motivation and fortitude. It is with immense gratitude that I acknowledge the crucial role you have played in fostering my growth.

Finally, I express my heartfelt gratitude to those who have stood by me during moments of challenge and adversity. Your unwavering support and resilience have granted me the courage to surmount obstacles and press onward towards my goals. May your own aspirations come to fruition, and may the future bestow upon you boundless happiness and success.

In closing, I humbly express my deepest appreciation to all those who have contributed to my academic endeavors. Your unwavering belief in my potential and your selfless acts of kindness will never be forgotten.

Table of Contents

INTRODUCTION

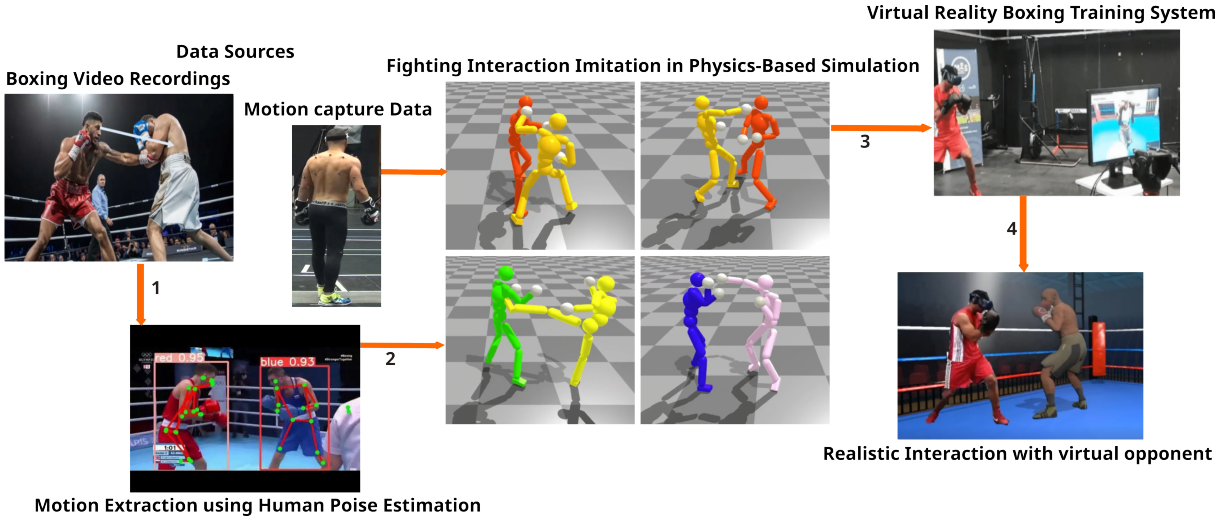


Figure 1 – Simulating interactive behavior of combat sports such as boxing (upper left) requires: 1. **extracting their motor behavior through motion estimation methods** from data sources such as video recordings (bottom left) or motion capture data, 2. modeling these interaction and **teaching physically simulated agents how to perform the highly dynamic motor skills of fighters as well as their interactive behavior in their decision learned making from demonstrations** (middle), then 3. transferring these learned agents to the virtual environment training system in order to 4. control personalised virtual opponents for realistic fighting against the user (right).

The world of sports has always been a fascinating field of study due to its complex nature, particularly when it comes to strategies employed by athletes during competition. Sports strategies, especially those executed in fighting sports or martial arts, such as boxing, involve complex interactions between fighters that require high anticipation skills and rapid movements.

Boxing requires mastery of various sub-skills, making it challenging to train effectively without proper guidance and practice. These sub-skills include footwork, balance, speed, power, technique, and strategic planning, all of which must be integrated seamlessly during fights [82]. Honing those sub-skills through repetitive sparring and training sessions for elite athlete often lead to them getting injured, therefore significantly reducing their

performance in official fights [82, 85, 32]. These issues open new perspectives on creating new modalities for analyzing and training athletes. In recent years, there has been growing interest in developing virtual environments for training purposes, specifically through Virtual Reality (VR) technology. Virtual Reality technology offers a revolutionary approach to sports training by providing a controlled environment, where stimuli can be standardized, adjusted, and even exceed real-life conditions. This unique feature enables coaches and trainers to design customized training programs that supplement conventional methods. For instance, virtual reality simulations can help athletes enhance their defensive skills without exposing them to potential injuries from repeated hits or impacts [104]. The advantages of utilizing virtual reality training extend far beyond just avoiding physical harm. As athletes progress in their careers, there comes a point when they hit a plateau and require continuous improvement to stay competitive. However, pushing past their physical limitations becomes increasingly challenging. With virtual reality, athletes can engage in novel training modalities that challenge their cognitive abilities and perception, enabling them to maintain their edge despite reaching their physical peak [61]. In the context of boxing, one major concern among trainers and officials is the development of boxers' anticipatory skills in defensive situations. Traditional training methods often involve enduring numerous punches to build up resistance and reflexes – a strategy that poses inherent risks to the athlete's health and condition. Virtual reality solutions address this issue by allowing boxers to concentrate solely on processing critical data about their opponents, such as body language, footwork, and attack patterns. This heightened awareness will ultimately empower them to accurately predict incoming assaults and respond with effective countermeasures. These objectives constitute the foundations of a number of training and sport research projects, such as the REVEA project [61], set up in the run-up to the Paris 2024 Olympic and Paralympic Games, and co-sponsored by the French Ministry of Higher Education, Research and Innovation and the Ministry of National Education, Youth and Sport (PPR "Sport Haute Performance" call of France 2030 funding). This project implements a virtual reality training system for helping boxing athletes to improve sub-skills such as speed, motor coordination and strength training. This paradigm involves mainly interacting with a virtual opponent for practicing. The behavior of this virtual opponent is dictated with fixed scenarios implemented using commercial game engines such as Unity, where the movements of the virtual humans are stored in a database constructed using motion capture, which are retrieved and played back following the chosen scenario. However, this static behavior of the virtual opponent

can affect the immersion of the athlete, as the non-responsive replayed motion can often be perceived as non realistic. Moreover, the predictability of the static behavior of the virtual opponent can reduce the benefits of the virtual reality training as the athlete can exploit the weaknesses of the virtual opponent after few interactions. Nevertheless, it is challenging and time-consuming to craft and design more adaptive and diverse scenarios for the interactions between the athlete and the virtual opponent.

We hypothesize that simulating realistic interactive behaviors can greatly improve performance and individualize training, by modeling the interactions between boxers using data-driven techniques, and leveraging physics-based character animation. This would allow having a virtual opponent that replicates the reactive behavior of real opponents, such as a future adversary, by using prior data about him, in the form of video recordings of previous fights. It would also mitigate the limitations of the virtual reality training system discussed earlier, as the virtual opponent's motion would be more realistic and responsive to the user's actions. Moreover, its behavior would be learned automatically from data instead of relying on manual labour.

Thus, the objective of this thesis is to learn and simulate martial arts strategies and imitate the fighting interactions, for training athletes in virtual reality training environment. By designing a strategy model for boxing, we can propose a set of plausible responses from a virtual opponent in real-time, depending on the user's current action and his moves history. This involves modeling interactions between boxers based on annotated, or non annotated data sources, that could take the form of prior fights video recordings or motion capture data. It also involves animating the opponent to perform realistic movements inside the virtual environment. Here, realistic means: "similar to a movement and strategy that the targeted real boxer would have done in the same condition". The virtual opponent should be intelligent in its decisions, and controllable. Hence, it allows users to experience a realistic boxing training simulation and improve their skills, while preparing them to defend in front of specific boxing strategies and styles. Indeed, the goal of interaction imitation is not to have an optimal opponent with the purpose of winning against the user, but an opponent that imitates an eventual opponent and replicates its behavior. Such opponents should not only mimic human actions but also adapt dynamically according to the user's own inputs, creating more nuanced and stochastic encounters that better mirror real-world strategies induced by real athletes.

When it comes to the use of virtual/augmented reality in assessing users in martial arts training, the authors of [38], proposed a martial arts system that utilizes real-time

image processing and computer vision to allow players to fight virtual enemies using kicks, punches, and acrobatic moves. This system used a profile view and two displays to provide an enhanced view of various martial arts techniques. The system however was limited in terms of the realism in interactions as it was in the form of a 2D game, and only covered a limited set of techniques. Therefore, it was limited in both the immersion and the behavior of the virtual enemies.

Another work [57] proposed a martial arts training system utilizing motion capture technology, allowing users to practice at home and receive feedback through a virtual coach. The system was able to distinguish skill levels and enhance performance. However, it was limited in implementing practical exercises like sparring that requires random movement and strategy simulation. It also assumed perfect feedback from the virtual coach. It also required that users are equipped with motion capture systems with multiple cameras and markers. The feedback from the virtual coach was not physically driven as it relies on replaying pre-recorded motion capture clips for generating the motions, therefore sharing the same limitation as the REVEA project in terms of the opponent's behavior.

The authors in [107] presented a framework for animating virtual humans that interact with real users in virtual reality in a Kung-Fu fighting example, using a combination of motion retrieval and motion adaptation techniques. Their system adapts a moderate-sized database of motions to the situation, considering the current pose of the character, the position of the target, and the type of motion ordered. They mainly addressed the differences between the virtual humans and the constraints of motion capture data. The behavioral model of this system however was also very limited as it relies on manually designed finite state machines and the coverage of the motion capture database. The work of [158] introduced an autonomous Karate Kumite character for VR-based training, accepting input from a tracked human athlete. Its behavior relies on analyzed Karate rules and motion capture data. Techniques like motion graphs, alignment, and blending were employed with custom enhancements for higher requirements. It had limitations in terms of the user action recognition and also relied on handcrafted rules for the reaction generation for the opponent.

Physics-Based Interaction Simulation with Deep Reinforcement Learning

In this thesis, differently from prior works, we are interested in generating reactive behavior of physically embodied agents in a data-driven manner by leveraging machine learning techniques and physics-based simulation. The benefits of this approach include:

— The physical correctness and validity of the generated motion, especially when using motion capture data as reference [63, 22, 23, 148, 119, 1].

— The possibility to control the generated motions through external control signals, therefore leveraging learned/designed controllers for generating the intended behavior. This has been explored in various domains, such as controlling bipedal characters for locomotion and other tasks [25, 74, 44, 73, 102].

— The possibility to generate dynamic interactions with direct feedback by applying external forces to physical entities. Therefore, it is possible to have real users interacting with the physics simulation. This has been demonstrated on different scenarios such as interacting with real users, objects in the simulation and with other characters [75, 147, 97, 87, 40, 55, 48, 137].

However, most of these works deal with locomotive tasks, and do not necessarily handle intricate and interactive motions involving physical contacts, such as in fighting between multiple characters. Some of these work also assume having access to motion capture data which is hard to obtain in competitive fighting sports. In our case, to analyse and simulate the motor skills and interactive behaviour of boxers, it will be necessary to utilize the available pre-recorded fighting videos of the fighters of interest, and integrate the behavior into the physically embodied agents. The problem of extracting human motion information from images/videos is referred to as Human motion Estimation [161, 156] where the goal is to predict joint position in 3D space. It enables to track a skeleton belonging to a person, and its temporal trajectory, given a video. There are many challenges related to human motion extraction from video, such as depth ambiguities, dealing with different types of occlusions, shadows, video stability and quality, physical realism of the extracted motions, ... [143, 128, 2, 71].

Some methods even attempted to directly imitate motions in videos to control physically simulated characters [135, 154, 103]. Most of these works are usually applied to simple motions, such as locomotion, periodic motions, and most importantly motions that only

involve one person. These works also use 2D human pose estimation as a backbone in their initialization stage. While there has been major developments and improvements on the accuracy of these 2D human pose estimators, it is not clear how they perform in videos of competitive sports, such as boxing, which is a crucial step for extracting motions and analyse the behavior of fighters from video recordings. To this end, the first step of this thesis was to explore the capabilities of 2D pose estimators in estimating motions of boxers from video recordings. It raises specific challenges related to fast paced nature of competitive sport activities, and multiple people occlusions.

Given extracted motions of pairs of fighters, our next objective was to simulate their interaction in a physically plausible manner, and to make sure that the reaction of the simulated agent facing an opponent is similar to the reaction of the correspondent fighter. We approach this as an imitation learning problem from demonstration of multiple experts in interaction. While imitation learning techniques [95] have been explored to imitate motions and behavior for simulating physics-based characters using motion capture data as demonstrations, they have not been explored for simulating the interactive behavior of multiple characters.

Thesis Overview

In this thesis, we explore the extraction and simulation of the interactive behaviour of fighters from motion data. To this end, we propose to use methods of human motion estimation from videos, reinforcement-learning based imitation learning, and physics-based character simulation. The organization of this thesis is as follows.

First, we start by introducing several concepts used in this thesis in chapter 1: namely reinforcement learning, imitation learning and physics-based simulation.

In Chapter 2 we evaluate the capabilities of different categories of current 2D human pose methods, and their accuracy in extracting pose information given RGB videos of boxers under challenging recording conditions, and rapid boxing motions.

In Chapter 3 we address the problem of simulating interactive behavior of fighters given motion data in a plausible manner. We propose an approach for imitating interactions and motions of multiple physics-based characters, from unstructured motion data. We focused solely on imitating interaction from unstructured data of two fighters performing light shadow boxing with minimal physical contacts (presented as a poster in SIGGRAPH Asia 2022 [151]). Then, we extended the approach to be applied on more interaction data

involving boxing with physical contact, as well as another fighting activity. This work was presented in The Symposium on Computer Animation (SCA'23) and published in the journal Proceedings of the ACM in Computer Graphics and Interactive Techniques (PACMCGIT) [152].

Lastly, we conclude in chapter 3.5 by recalling the contributions, and identifying some limitations. Then, we proposed some potential extensions and perspectives to this work.

Chapter 1

Background

In the second part of this thesis, we utilize reinforcement learning to develop control policies that empower physically simulated agents to execute intricate motor skills in interactive fighting situation from motion capture data. This section presents a comprehensive review of motor control in physics-based character simulation and fundamental concepts in reinforcement learning. We also introduce the notation employed throughout the document.

1.1 Physics-Based simulation

In this work, we simulate the interactive behavior of fighters in physics simulation. We mainly use the physics simulator IsaacGym [80]. In this section, we will describe the characters being controlled in the physics simulation: Physical properties, observation/state description, action and motor definitions, ...

1.1.1 Fighter body model

We represent each fighter as a 3D humanoid character, modeled as a set of articulated rigid bodies [33], where each rigid body link is attached to its parent link with spherical joints (spanning 3 degrees of freedom), with the exception of elbows and knees being attached with revolute joints (only 1 degree of freedom) and the hands being attached to the arm with a 0 degree-of-freedom fixed joint. The body parts' proportions are based on a human actor, as can be seen in figure 1.1. We use a similar character model to previous work [99, 100, 145] while adapting it to reflect a fighter by changing density and geometry of hands as well as some physical properties. This model is based on a mathematical representation of the character's body configuration, which is characterized by its pose and velocity:

— Pose ($\mathbf{q}$): The pose of the character is represented by a vector $\mathbf{q}$, which consists of

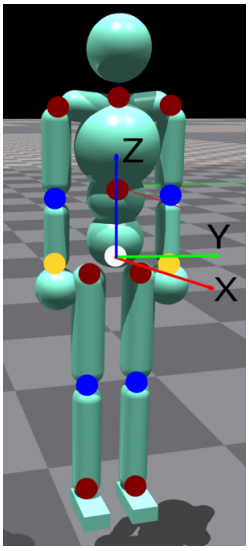


Figure 1.1 – The 3D humanoid model used in the physics-based fighting simulation. The humanoid is modeled as a set of articulated rigid bodies, where each rigid body link is attached to its parent link with spherical joints (in red), with the exception of elbows and knees being attached with revolute joints (in blue) and the hands being attached to the arm with a fixed joint (in yellow).

multiple components. These components describe the global position and rotation of the character's root (x_{root} and q_{root}) and the local rotations of each joint (q_j) in the character's local coordinate frame. The root refers to a reference point in the character's body, usually the pelvis or the hip joint. The joint rotations are represented in their local coordinate frames to simplify the animation process and ensure that the character's movements remain consistent regardless of the global position and orientation.

— Velocity ($\dot{\mathbf{q}}$): The velocity of the character is represented by a vector $\dot{\mathbf{q}}$, which also consists of multiple components. These components describe the root's linear velocity ($\dot{x}_{root}$), the root's angular velocity (ω_{root}), and the local angular velocity of each joint (ω_j). The linear velocity refers to the rate of change of the root's position, while the angular velocity represents the rate of change of the root's orientation. The local angular velocities of the joints are essential for simulating realistic and smooth character movements.

The character's movements are controlled by applying torque actuation to each joint. Torque, denoted by f, is a measure of the force applied to a joint, which causes it to rotate.

1.1.2 Actuation model

An actuation model plays a significant role in controlling the motion of a character's body model. At each timestep, a motion controller is responsible for specifying an actuation signal (action in the case of a policy controller) that influences the character's motion. This output signal is then utilized by an actuation model to calculate control forces for each joint in the character's body. Some works [126, 14, 11] simply use the identity function for the actuation model and use the policy action $a = (f_0, f_1, \ldots, f_n)$ as the control torques applied to each joint. This approach has the advantage of simplicity and flexibility, as it allows the policy to directly control the joint torques. However, it can be challenging to stabilize the character's motion, as the policy needs to explicitly compensate for the character's dynamics and interactions with the environment.

Another line of work [102, 99, 138] instead uses Proportional Derivative (PD) controllers [125] for the actuation model where the policy controller's output $\mathbf{a} = (\hat{q}_1, \hat{q}_2, \ldots, \hat{q}_n)$ specifies target rotations $\hat{q}_j$ for each joints. Each has its own advantages and downsides [98]. This approach has the advantage of providing a more stable and robust control, as the PD controller can automatically compensate for the character's dynamics and interactions with the environment. However, it requires careful tuning of the gain parameters k_p and k_d for each joint, which can be time-consuming and require significant expertise. Additionally, the policy is limited to specifying target joint angles rather than directly controlling the joint torques, which can reduce flexibility; we use the latter in this work. In the case of a 1D revolute joint for example, the torque applied to the joint is computed as follows:

for a target rotation $\hat{q} \in \mathbb{R}$, represented by a scalar rotation angle, the PD controller is modeled as an angular spring and damper system. The torque f is calculated using the following formula:

$$f = k_p(\hat{q} - q) - k_d\dot{q},$$

where q represents the current rotation of the joint, $\dot{q}$ represents its angular velocity, and k_p and k_d are manually specified gain parameters. This actuation model effectively applies torques to move the joint towards the desired rotation while mitigating excessive joint velocities through damping.

Similarly, torques applied to 3D spherical joints can be computed given the target rotation $\hat{q}$ represented by a quaternion:

$$f = k_p \exp\_\text{map}(\hat{q} - q) - k_d \dot{q},$$

where $\dot{q} \in \mathbb{R}^3$ denotes the 3D angular velocity of the joint. The quaternion difference between two rotations, $q_1 - q_2$, can be computed using quaternion multiplication between q_1 and the conjugate q_2^T, such that $q_1 - q_2 = q_2^T q_1$. To derive the torque from the resulting quaternion, the exponential map representation of the underlying rotation [36] is used by utilizing the function $\exp\_\text{map}(q)$ to convert a quaternion q into this format.

The exponential map of a rotation angle θ, expressed in radians, around axis $\mathbf{v}$ is obtained by scaling the axis by the rotation angle, resulting in $\mathbf{q} = \theta\mathbf{v}$. The rotation angle and rotation axis can be determined as follows:

$$\mathbf{v} = \frac{\mathbf{q}}{\|\mathbf{q}\|}, \quad \theta = \|\mathbf{q}\|.$$

1.2 Reinforcement Learning

Reinforcement Learning (RL) [123] is typically formalized as an entity, referred to as an 'agent', engaging in a series of interactions with an environment, which is represented as a Markov Decision Process (MDP). The primary goal of the agent is to optimize its anticipated cumulative reward, or 'return'.

For the sake of completeness, a Markov Decision Process (MDP) serves as a mathematical paradigm for representing decision-making processes characterized by uncertainties. It consists of the following essential components:

— **State Space** S: A set comprising all attainable states that the system might inhabit at any instance.

— **Action Space** A: A set describing every feasible operation the agent may perform while being in a given state.

— **Transition Probabilities** $p(s'|s, a)$: Quantify probabilities tied to transitioning to succeeding states, provided the current state and the selected action. These capture inherent dynamics of the environment.

— **Reward Function** $r(s, a, s')$: Describing a predefined function generating actual-valued scalar gratification obtained after transitioning from one state s to another s' owing to action a.

— **Discount Factor** γ: Determines the importance of future rewards, impacting the weight carried by eventual payoffs towards shaping the current actions. $\gamma \in [0, 1]$

and it diminishes future rewards' significance relative to immediate ones.

As the agent interacts with the environment during a single episode, it begins in an initial state $s_0 \in S$. At every discrete time step t, the agent receives the current state $s_t \in S$ and chooses an action $a_t \in A$ based on its policy $\pi(a_t|s_t)$. Subsequently, the agent implements the chosen action, leading to a next state $s_{t+1} \sim p(s_{t+1}|s_t, a_t)$ and a scalar reward $r_t = r(s_t, a_t, s_{t+1})$ that mirrors the desirability of the state transition. This iterative process persists until reaching a finite time horizon T or continuing indefinitely.

Throughout an episode, the sequence of states, actions, rewards, and subsequent states forms a trajectory denoted as $\tau = (s_0, a_0, r_0, s_1, ..., s_{T-1}, a_{T-1}, r_{T-1}, s_T)$. The ultimate objective of the agent involves discovering an optimal policy π^* that maximizes its anticipated discounted return $J(\pi)$:

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, J(\pi) \tag{1.1}$$

$$J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[\sum_{t=0}^{T-1} \gamma^t r_t \right] \tag{1.2}$$

Here, $p(\tau|\pi) = p(s_0) \prod_{t=0}^{T-1} p(s_{t+1}|s_t, a_t)\pi(a_t|s_t)$ symbolizes the probability distribution of trajectories induced by a policy π.

The Reinforcement Learning framework is depicted in Figure 1.2.

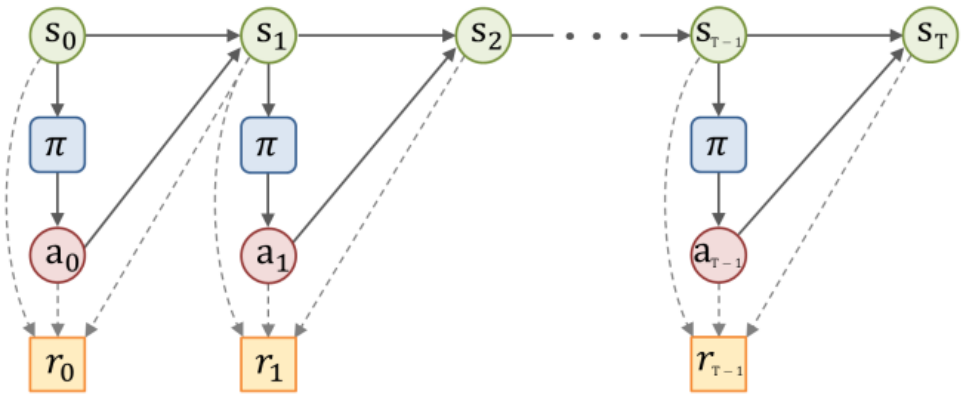


Figure 1.2 – Reinforcement Learning framework [96]

1.2.1 Value Function

In reinforcement learning, an important concept is the notion of **value function**, which serves to estimate the performance of a given policy. A value function provides an assessment of a policy's expected future returns, given a certain state or state-action pair [123].

Formally, let π denote a policy, i.e., a mapping from states to actions. The **state value function** of π, denoted as $V^\pi(s)$, offers an estimation of the expected return of an agent adhering to π starting from a specific state s:

$$V^\pi(s) = \mathbb{E}_\pi \left[G_t | S_t = s \right],$$

where G_t denotes the discounted sum of rewards obtained from time step t on-wards, i.e., $G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots$, where $\gamma \in [0, 1]$ is a discount factor. Therefore, we can interpret $V^\pi(s)$ as a measure of the desirability of the agent occupying a particular state s at timestep t.

On the other hand, the **state-action value function**, also known as the Q-function, represented as $Q^\pi(s, a)$, estimates the expected future return of executing action a at state s and subsequently following π for all remaining time steps:

$$\begin{aligned} Q^\pi(s, a) &= \mathbb{E}_\pi \left[G_t | S_t = s, A_t = a \right] \\ &= \sum_{r, s'} p(r, s' | s, a) \Big[r + \gamma V^\pi(s') \Big] \end{aligned}$$

$p(r, s'|s, a)$ represents the probability of obtaining reward r and transitioning to state s' after taking action a at state s. Consequently, $Q^\pi(s, a)$ captures the utility of performing action a when occupying state s.

It is common practice to omit this symbol from the notation, writing $V(s)$ instead of $V^\pi(s)$, and $Q(s, a)$ instead of $Q^\pi(s, a)$, when the dependency of value functions on π is clear from context.

The optimal policy's value function and value-state function are expressed as $V^*(s)$ and $Q^*(s, a)$. Notably, Q^* holds special significance since an optimal policy can be recovered by selecting, at every state s, the action associated with the highest Q^*-value:

$$\pi^*(s) = \operatorname*{argmax}_a Q^*(s, a).$$

The value functions and Q-functions admit recursive definitions through the Bellman optimality equation, facilitating efficient learning of these values for a specified policy π.

The Bellman equation for the state-action value function under policy π, Q^π, is given by:

$$\begin{aligned} Q^\pi(s,a) &= \mathbb{E}_{S_0 \sim p(\cdot|s,a), A_0 \sim \pi(\cdot|S_0)} \left[R(s,a,S_0) + \gamma Q^\pi(S_0, A_0) \right] \\ &= \sum_{s',r} p(s',r|s,a) \left[r + \gamma \sum_{a'} \pi(a'|s') Q^\pi(s',a') \right]. \end{aligned}$$

Using the definition of the state value function, V^π, as the expected return when following policy π from some initial state s, we have:

$$\begin{aligned} V^\pi(s) &= \mathbb{E}_{A \sim \pi(\cdot|s)} Q^\pi(s,A) \\ &= \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a) \left[r + \gamma \sum_{a'} \pi(a'|s') Q^\pi(s',a') \right] \\ &= \mathbb{E}_{S_0 \sim p(\cdot|s,a), A_0 \sim \pi(\cdot|S_0)} \left[R(s,A_0,S_0) + \gamma V^\pi(S_0) \right]. \end{aligned} \tag{1.3}$$

1.2.2 Policy Evaluation

Reinforcement Learning (RL) frequently entails estimating the value function or Q-function related to a specific policy π. An essential constituent in many RL algorithms is policy evaluation [123]; a dynamic programming methodology used to approximate a policy's value function via its recurrent definition provided in Equation 1.3. Through interaction with the environment, a dataset of trajectories $D = \{(s_i, a_i, r_i, s'_i)\}$ is gathered following the current policy π, enabling to learn an approximation of V^π using an iterative procedure starting with an arbitrary initial value function V_0. Within the k^{th} iteration, an updated value function V_k may be deduced by minimizing the subsequent objective function:

$$V_k = \underset{\mathrm{V}}{\operatorname{argmin}} \, \mathbb{E}_{(s_i, r_i, s'_i) \sim D} \left(y_i - V(s_i) \right)^2, \tag{1.4}$$

with the target value $y_i = r_i + \gamma V_{k-1}(s'_i)$ serving as an estimated forecast of cumulative rewards accumulated from state s_i, using a previous version of the value function from the previous iteration V_{k-1}. The technique used for calculating the target can employ single-step bootstrapping or multi-step bootstrapping such as temporal difference TD(λ) [123]. In tabular setting, it can be demonstrated mathematically that as $k \to \infty$, V_k eventually

converges to V^π. Nevertheless, practical implementations commonly generate acceptable approximations within just a few iterations.

Akin to learning the value function, learning the Q-function also ensues via a similar iterative scheme. During each iteration k, an estimate of Q^π is achieved by optimizing the objective:

$$Q_k = \underset{\mathrm{Q}}{\operatorname{argmin}}\ \mathbb{E}_{(s_i,a_i,r_i,s'_i,a'_i)\sim D}(y_i - Q(s_i,a_i))^2. \tag{1.5}$$

In this case, a target value $y_i = r_i + \gamma Q_{k-1}(s'_i, a'_i)$ represents the projected expectation of potential return achievable from pair (s_i, a_i), using the Q-function computed from the prior iteration Q_{k-1}.

Practical implementations of this procedure involves using a parametric function for modeling the value function, such as neural networks. Therefore the optimization can be done using gradient descent for a number of update steps.

1.2.3 Policy Gradient Methods

There exists a number of methods to optimize the objective in the RL frameworks [124, 157, 12]. Each applies to different situations depending on the characteristics of the given problem like the dimensionality of the state/action space, the nature of action: continuous or discrete, ... In the context of this thesis, describing human motion involves using continuous state and action spaces. Therefore, policy gradient methods [124] are the appropriate choice for this category of RL problems. Policy gradient methods are a suitable class of algorithms for tasks with continuous actions, as well as being compatible with neural network function approximators for modelling the policy and value functions. These algorithms iteratively update the policy parameters through gradient ascent using an empirical estimate of the policy's expected return gradient with respect to its parameters, denoted as $\nabla_\pi J(\pi)$.

Deriving the policy gradient starts by reformulating the expected return of the policy $J(\pi)$ with respect to the policy's marginal state distribution $d_\pi(s)$ instead of its trajectory distribution $p(\tau|\pi)$:

$$J(\pi) = \mathbb{E}_{s\sim d_\pi(s)}\mathbb{E}_{a\sim\pi(a|s)}[Q_\pi(s,a)]$$

Where, $d_\pi(s) = \sum_{t=0}^{\infty}\gamma^t p(s_t = s|\pi)$ is defined as the unnormalized discounted state distribution induced by π, where $p(s_t = s|\pi)$ is the probability that the agent is in state s

at timestep t when following the policy π [124]. The policy gradient can then be derived by differentiating J with respect to its parameters:

$$\nabla_\pi J(\pi) = \nabla_\pi \mathbb{E}_{s\sim d_\pi(s)} \mathbb{E}_{a\sim\pi(a|s)}[Q_\pi(s,a)] = \mathbb{E}_{s\sim d_\pi(s)} \int_a \nabla_\pi \pi(a|s) Q_\pi(s,a) da.$$

Since the action space is large in most real application, the integral over actions is usually intractable. Therefore, the integral is replaced by an expectation over actions using the score function $\nabla_\pi \log \pi(a|s)$ so that it can transformed into a more tractable formulation that allows for computing the policy gradient:

$$\begin{aligned}
\nabla_\pi J(\pi) &= \mathbb{E}_{s\sim d_\pi(s)} \int_a \nabla_\pi \pi(a|s) Q_\pi(s,a) da \\
&= \int_a \nabla_\pi \pi(a|s) Q_\pi(s,a) da, \quad a \in A \\
&= \mathbb{E}_{s\sim d_\pi(s)} \mathbb{E}_{a\sim\pi(a|s)} \left[\nabla_\pi \log \pi(a|s) Q_\pi(s,a)\right] \\
&= \mathbb{E}_{s\sim d_\pi(s)} \mathbb{E}_{a\sim\pi(a|s)} \left[\frac{\nabla_\pi \pi(a|s)}{\pi(a|s)} Q_\pi(s,a)\right] \\
&= \mathbb{E}_{s\sim d_\pi(s)} \mathbb{E}_{a\sim\pi(a|s)} \left[\nabla_\pi \log \pi(a|s) Q_\pi(s,a)\right].
\end{aligned} \tag{1.6}$$

In practice, the procedure is initiated by first collecting trajectories induced by the policy interacting with the environment. At timestep t, the gradient of the log-likelihood of the action taken by the policy $\nabla_\pi \log \pi(a|s)$ is calculated and then scaled by the Q-function. Finally, estimates of the policy gradient are computed by averaging the scaled gradients across all timesteps of the collected trajectories.

A number of practical design choices is used to make this simple procedure effective in real world applications. The most important ones are the use of baselines when estimating the gradients and the estimation of the value function.

Variance reduction with baselines: When estimating the gradients in Equation 1.6, the estimator often exhibits high variance, which can result in slow and unstable learning. A prevalent variance reduction strategy involves introducing a baseline in the form of the value function, as expressed in Equation 1.7:

$$\begin{aligned}
\nabla_\pi J(\pi) &= \mathbb{E}_{s\sim d_\pi(s)} \mathbb{E}_{a\sim\pi(a|s)} \left[\nabla_\pi \log \pi(a|s)(Q_\pi(s,a) - V_\pi(s))\right] \\
&= \mathbb{E}_{s\sim d_\pi(s)} \mathbb{E}_{a\sim\pi(a|s)} \left[\nabla_\pi \log \pi(a|s) A_\pi(s,a)\right].
\end{aligned} \tag{1.7}$$

In this equation, $A_\pi(s,a) = Q_\pi(s,a) - V_\pi(s)$ denotes the advantage function, which quantifies the extent to which a specific action surpasses the average return at a given

state. The update described in Equation 1.7 can be understood as increasing the likelihood of actions that outperform the average return at each state and decreasing the likelihood of actions that fall short of the average. It has been demonstrated that the baseline does not impact the policy gradient [124].

Value function estimation: To estimate the Q-function and value function, various approximation techniques can be employed. One common approach involves utilizing data from the policy π to fit function approximators to Q_π and V_π: the value function V_π is approximated using a function approximator. The Q-function, Q_π(s, a), is directly estimated by approximating the empirical returns G_π(s, a), which represents the actual rewards obtained when following policy π. The empirical return G_π(s, a) for taking an action at a given state can be calculated by summing the discounted rewards along a trajectory that starts with state s and action a. This sum is typically represented as: $G_\pi(s, a) = \sum_{t=0}^{\infty}(\gamma^t * r_t)$, where γ is the discount factor, r_t is the reward received at time t, and the sum is taken over all time steps t in the trajectory, as depicted in Equation 1.2.1

1.2.4 Reinforcement Learning Training

For training policies in the work of this thesis, we use the Multi Agent Proximal Policy Optimization algorithm, a variant of the Proximal Policy Optimization (PPO) [115] in the multi-agent setting, which is a policy gradient algorithm with additional modifications to enhance stability and sample efficiency. Algorithm 1 outlines the training procedure that will be utilized for the RL part of this work. Both the policy and value function will be modeled using neural networks. At each update iteration, a batch of trajectories $\{\tau_i\}_{i=1}^{m}$ is collected from the current policy π. The data is then used to update the value function $V(s)$ with target values y_i computed using Temporal Difference TD(λ) [123]. The advantage $A_i \approx A_\pi(s_i, a_i)$ for each timestep is then estimated using GAE [114], which are subsequently employed to update the policy via PPO.

1.2.5 Stochastic Games and Multi-Agent Reinforcement Learning

Stochastic or Markov games [70] were proposed as the standard framework for modeling multiple adaptive agents with interacting or competing goals. Markov games still assume that state transitions are Markovian. However, the difference with the single-agent MDP framework is that each agent has its own set of actions. For n agents, the joint-action

Algorithm 1 RL Training Procedure

$\pi \leftarrow$ Initialize policy
$V \leftarrow$ Initialize value function
while not done **do**
    $D \leftarrow \emptyset$ Initialize dataset buffer
    for $i = 1, \ldots, m$ **do**
        $\tau_i \leftarrow (s_t, a_t, r_t, \ldots, s_T)$ Collect trajectory i with current policy π
        for $t = 0, \ldots, T-1$ **do**
            $y_t \leftarrow$ Calculate target value from τ_i with Temporal Difference TD(λ)
            Record y_t in τ_i
            $A_t \leftarrow$ Calculate advantage from τ_i with GAE(λ)
            Record A_t in τ_i
        end for
        Store τ_i in D
    end for
    Update V using samples (s_i, y_i) from D
    Update π using samples (s_i, a_i, A_i) from D with PPO
end while

space is $A = A_1 \times A_2 \times \cdots \times A_n$.

The state transition $T : S \times A_1 \times \cdots \times A_n \to p(S)$ and reward functions $R_i : S \times R_1 \times \cdots \times R_n \to R$ now depend on the joint action of all agents.

Similar to the MDP objective, each agent i is trying to maximize its expected reward under a joint policy $\pi = (\pi_1, \ldots, \pi_n)$, which assigns a policy π_i to each agent i:

$$V_{\pi_i}(s) = E\left[\sum_{j=0}^{\infty} \gamma^j r_{i,t+j}\right] \tag{1.8}$$

Contrary to the MDP case, there may not exist an optimal stationary deterministic policy, meaning that the optimal stationary policy is sometimes probabilistic; mapping states to discrete probability distributions over actions $\pi^* : S \to p(A)$. The best response and Nash equilibrium concepts [59] can be extended to such games: a policy π_i is the best response if there is no other policy for agent i that gives higher expected future reward when other agents keep their policies fixed.

If we assume only 1 agent, or the case where other agents play a fixed policy, the Markov game reduces to an MDP.

In contrast to the general RL methods, assumptions regarding the observations that agents make are made. Due to the dependency on other agents' strategy, most Multi

Agent Reinforcement Learning (MARL) methods assume that either actions of all participating agents or rewards (or both) are observed from all agents. Although this helps us to model opponents and learn over joint actions, it may be unrealistic to assume such common knowledge, especially in distributed systems where such information is usually not available.

A major division of algorithms for MARL is whether we are considering other agents as part of the environment (Independent Learning) or we explicitly try to model other agents (Joint Action Learning).

In this work, we use the former where each agent represents a specific fighter. Therefore, we approach the problem of simulating interactions between fighters as a Markov Game of two agents that are optimized with MARL Independent Learning.

1.3 Imitation Learning

Imitation learning [113, 7, 95] is a sub-field of machine learning concerned with enabling an autonomous agent to replicate the behaviors and skills exhibited by a proficient (or "expert") entity through observational data. This methodology enables machines to acquire new abilities without explicit programming or extensive trial-and-error interactions, thus accelerating the learning process. The expert can be human or artificial; for instance, it might involve watching videos of humans performing various tasks like driving a car or playing chess. In contrast, the imitator is the learner who seeks to reproduce observed actions based on these observations. A significant challenge in imitation learning lies in accurately interpreting and generalizing patterns from limited samples provided by the expert, as minor discrepancies between the expert and imitator could result in substantial performance differences. Additionally, the complexity of the environment, variability of the demonstrated skill, and availability of relevant information all impact the effectiveness of imitation learning techniques.

When training an agent via imitation learning, we do not focus on maximizing explicit task-related rewards as in conventional reinforcement learning methods. Instead, the approach relies solely on having access to expert demonstrations illustrating the desired way to accomplish a given task. By examining these samples, agents strive to learn policies that mimic the behavior exhibited in the showcased demonstration.

Demonstrations generally comprise records of the sequence of states visited along with the corresponding actions executed by the demonstrator throughout the process. Conse-

quently, such collections are referred to as expert demonstration trajectories, denoted as $\tau_e = \{(s_t, a_t)\}$, where s_t represents the state at time t, and a_t denotes the action chosen by the expert at that same time step.

The field of imitation learning encompasses numerous areas of study, but broadly speaking, it can be divided into two major branches: behavioral cloning (BC) [9, 111, 26] and inverse reinforcement learning (IRL) [112, 90]. Here, behavioral cloning deals with treating imitation learning as a supervised learning problem, striving to construct a relationship between input states and output actions guided by the available expert demonstrations. On the other hand, inverse reinforcement learning concentrates on deducing the latent reward function motivating the expert's choices, allowing to utilize standard reinforcement learning methodologies after obtaining said function estimate.

The imitation learning from observation [50, 10, 131] considers autonomous agent attempting to learn how to perform tasks by observing *state-only* demonstrations $\tau_e = \{o_t\}$ generated by an expert as opposed to classical imitation learning paradigms that assumes that experts' actions are also provided, which is not usually the case for motion data.

In the second part of this thesis, we focus on a scenario where a fighter engages in a martial arts fight with an opponent. Our goal is to develop a physically embodied agent that can imitate the fighter's motions and react to the opponent's actions in a realistic manner. To achieve this, we leverage expert demonstrations in the form of human pose trajectories extracted from video or motion capture data. Specifically, we formulate this problem as a multi-agent imitation learning from observation task, where the agent learns to mimic the fighter's actions and respond to the opponent's movements based on the provided demonstrations.

1.4 Physics-based character animation and Reinforcement Learning

The physically simulated embodied agent described earlier is controlled by a policy trained with reinforcement learning (RL). In this section we describe the observations and actions definition for linking the RL agent and the physically simulated character.

1.4.1 RL State representation

The motion of a character is described by a set of state features that capture the configuration of the character's body. These state features serve as input to the RL policy, which controls the character's movements. The following state features are typically considered:

— Height of the root off the ground h: This feature represents the height of the character's root, which is usually the pelvis or hip joint, relative to the ground. It is a 1-dimensional (1D) scalar value.

— Position of each link p: This feature describes the position of each link (i.e., body part) in the character's local coordinate frame. Since the character's local coordinate frame is 3-dimensional (3D), each link's position is represented by a 3D vector, resulting in a total of 3D x N links dimensions.

— Linear velocity of each link $\dot{p}$: This feature represents the linear velocity of each link in the character's local coordinate frame. Similar to the position feature, each link's linear velocity is a 3D vector, resulting in a total of 3D x N links dimensions.

— Rotation of each link using a tangent-normal encoding (u, v): This feature encodes the 3D rotation of each link in the character's local coordinate frame using a tangent-normal representation. The tangent u and normal v vectors are 3D vectors that correspond to the tangent and normal of the link's coordinate frame expressed in the character's local coordinate frame. This encoding provides a smooth and unique representation of a given rotation and results in a total of 6D x N links dimensions for both u and v vectors.

— Angular velocity of each link (ω): This feature describes the angular velocity of each link in the character's local coordinate frame. Each link's angular velocity is a 3D vector, resulting in a total of 3D x N links dimensions.

The character's local coordinate frame is defined with the origin located at the root (pelvis), the x-axis oriented along the root link's facing direction, and the z-axis aligned with the global up vector. This coordinate frame simplifies the animation process and ensures consistent character movements.

1.4.2 RL Action representation

One crucial aspect of controlling a character's motion is the selection of action parameterizations, which determines how the output of the control policy is represented and passed to the actuation model. Specifically, each action a generated by the policy specifies target rotations for the character's joints. However, the representation of these rotations requires careful consideration, particularly when employing neural networks.

There are several options for representing rotations, each with its own trade-offs. Quaternions, for example, provide a compact and singularity-free representation, but can be computationally expensive to work with. Euler angles, on the other hand, are more intuitive and easy to compute, but suffer from singularities and gimbal lock. Exponential maps offer a compromise between the two, providing a compact representation that uses 3 parameters without singularities and gimbal lock. [36] provides a comprehensive review of these and other rotation parameterizations, highlighting their advantages and disadvantages.

In this work, we follow recent advances in physics-based character animation [99, 100] for representing joint rotations: For spherical joints, we use exponential maps as a compact and singularity-free representation. For revolute joints, we represent the rotation with a single angle value θ, which provides a simple and intuitive representation. This choice of action parameterization allows us to effectively control the character's motion while avoiding the pitfalls of other representation methods.

Chapter 2

Benchmarking 2D pose estimation methods for fighting sports

Machine learning and computer vision techniques in sports have emerged as valuable tools for tailoring training regimens especially by evaluating the performance of sport athletes and analysis of upcoming adversaries, in particular the use of human motion estimation methods. For example, in team sports, this technique has been utilized to track players trajectories on the playing surface/playground [42, 21, 13]. This allows for the extraction of metrics like total running distance and average velocity, among others.

In combat sports, such as boxing, wrestling, and fencing however, monitoring an athlete's motion is crucial to evaluate their performance and identify areas for improvement [19, 58]. Posture and body movements play a significant role in these sports, and assessing them accurately can help coaches and athletes develop effective training strategies. Unlike traditional motion monitoring in other sports, which often focuses on global location and orientation across a terrain, combat sports require a more detailed analysis of body movements throughout the contest. This approach enables a deeper understanding of an athlete's performance, allowing for targeted improvements and enhanced competitive success.

In this type of sport, where athletes compete against each other, there are limitations to using sensors for monitoring motion; Unlike motion capture in laboratory settings, monitoring motion in real-world competitive conditions presents unique challenges. It is impractical to mount sensors onto the bodies of opponents who do not wish to have their movements captured for training purposes. Additionally, the use of sensors during competitions is often forbidden. This emphasize the need for alternative methods of motion analysis. As a result, analyzing prior contests without sensors becomes a crucial aspect of understanding and improving athletes' performance. This approach allows coaches and athletes to study body movements and posture from previous matches, enabling them to identify patterns, strengths, and weaknesses in their own performance and that of their

opponents. By focusing on these critical factors, coaches and athletes can develop effective training strategies that lead to enhanced competitive success, all without the need for invasive sensor technology.

Accessible resources comprise a compilation of videos, mostly acquired under challenging circumstances, such as television recordings, those supplied by event organizers, or produced by coaching teams from difficult view points. Estimating the poses of both rivals amidst the highly intricate and unregulated fighting situations presents significant hurdles. One must consider moving cameras, multiple perspectives, varying angles, and occasionally poor resolution and lighting.

Human pose estimation (HPE) constitutes an intensely researched area spanning numerous fields, including entertainment [49], healthcare [93], ergonomics [106] and sports [88, 134, 60]. HPE involves inferring the spatial configuration of major joints defining a person's posture using sensory inputs such as visual imagery or depth maps.

Recent advancements in deep-learning architectures have considerably improved 2D HPE accuracy within RGB frames over the past years. Numerous sizable datasets featuring annotated RGB photos have surfaced accordingly, offering network training and testing opportunities — examples include "MPII Human Pose" [5] and "MS COCO"[69]. These collections incorporate diverse situations and actions.

Previous studies assessed the effectiveness of HPE approaches in sports settings, notably regarding tennis serves [146], dance performances [5], and gymnastics routines [134]. In combative sports, primary obstacles consist of:

— Rapid body parts movements generating motion blur, which can hinder certain temporal window-based techniques.

— Partial self-occlusions resulting from limited viewing angles, where cameras might concentrate solely on particular sections of the athlete's body.

— Intermittent full occlusions with competitors, officials, and environmental elements obstructing direct lines of sight.

Gaining familiarity with limitations associated with different HPE strategies within demanding scenarios such as boxing helps inform data scientist in sports. Such insights could assist sport and data scientists in selecting optimal human action recognition tools. Ultimately, this level of understanding should empower researchers and data science firms to make informed decisions about human body pose tracking solutions.

In this chapter, we analyze the performance of state of the art body pose estimation algorithms in the specific case of boxing. To this end, we collected a test dataset of

professional boxing motions comprised of RGB videos and motion capture sequences to simulate the possible self-occlusion and rapid motions that occur in real boxing video. RGB videos were captured and synchronized with an opto-electronic commercial motion capture system, used as a reference, Qualisys [108]. The RGB camera was calibrated to find the correspondence between a pixel in the RGB image and in the 3D space in the Qualisys system. Six state-of-the-art human pose tracking methods were applied to the RGB video: OpenPose [16], Detectron2 [142], Cascaded pyramid network for multi-person pose estimation [17], AlphaPose [30], Deep Dual Consecutive Network for Human Pose Estimation [78] and SimCC [68].

In the following sections, we will first introduce related work on HPE and associated benchmarks as well as relevant surveys for sports applications 2.1. Next, we describe the publicly available datasets/benchmarks and evaluation metrics of HPE, highlighting their strengths and limitations in the context of sports: section 2.2. We then introduce the different HPE methods chosen for this study and their applicability to combat sports: section 2.3. Then, we will describe the protocol used for video and motion capture sequence acquisition, evaluate the different HPE methods and give a comparison and use-case analysis in section 2.5. Finally we discuss and summarize our findings to draw conclusions and insights about HPE in combat sports in section 2.6.

2.1 Related Work

In this section, we present the most popular and recent HPE and we address the problem of its application to sports, in particular their relevance for fighting sports analysis. We then describe existing standard benchmarks, and how previous works designed protocols to evaluate the accuracy of HPE in various application domains.

2.1.1 Human Pose Estimation and associated benchmarks

Following the development of camera-based 2D human-pose estimation, many survey papers about this area have been published in the recent decades. Before the adoption of deep neural network approaches in HPE, its framework was categorized into a pre-possessing stage, which includes feature extraction, camera calibration, body detection and foreground segmentation. Then a body parts' parsing stage, that includes feature extraction with handcrafted or learned features according to their representations: shape,

contour, geometric and appearance features [77]. Another way of defining the HPE framework is to break it down into three main components: feature description, human body models, and modeling methods. Feature description involves the use of low-level and high-level features, as well as motion features for motion-related methods. Human body models encompass kinematic, planar, and volumetric models, and may incorporate human pose priors. Modeling methods include discriminative and generative approaches, as well as bottom-up, top-down, and motion-based techniques [34].

Classical methods for HPE have been outperformed by deep neural network methods (DNN). DNN approaches, such as those using convolutional neural networks (CNN), offer several advantages over traditional methods. Firstly, CNNs learn and extract features from training on dataset images. This allows for combined part detection and accurate localization of human body parts. Secondly, DNN methods enable learning features and human body kinematics through graphical modeling. Lastly, these approaches can learn both features and body part locations. Challenges related to deep learning-based 2D and 3D HPE methods are the influence of body part occlusion, crowded people, network efficiency and adequate training data [18].

With the great progress of Deep Neural Networks and their application to HPE, new 2D HPE methods were proposed leading to a new families of approaches. Dang et al. categorized single person HPE into direct regression-based approaches and heatmap-based approaches [27]. The first category uses the output feature maps to directly regress keypoints. DeepPose [132] and SimDR [67] belongs to this type of approach. They have the advantage of being fast and direct and they can be applied to 3D data with slight changes. However, they suffer from difficulties to learn the mapping between the features and the regressed keypoints. The second category generates heatmaps focused on potential locations first, then predict keypoints based on the heatmaps. HR-Net [121] and DarkPose [155] applied this type of approach. They are easy to interpret by visualizing where their model is looking in the images to regress the decision. There are also easy to apply in complicated situations: lighting, occlusions, ... However, they require high memory consumption for high resolution heatmaps and are difficult to be extended for 3D estimation.

Multiple person HPE can be categorized into Top-down or bottom-up approaches. Top-down approaches have to detect all people from a given image, after which single-person approaches are performed in each detected bounding box. Examples of HPE methods in this category include AlphaPose [31], DCPose [78] and Cascaded Pyramid Network [17].

It has been shown that the human detection phase has no impact on the pose estimation performance, provided that it is accurate enough. Reversely, in the bottom-up approaches, all body parts (keypoints) are detected in the first stage, then they are associated to human instances, in the second stage. This approach is used in methods such as the popular OpenPose [16] and SPM [92].

When it comes to benchmarks of HPE methods, Chen et Al. [18] listed the PCKh@0.5 scores on the Max Planck Institute for Informatics (MPII) human pose testing set for regression and heatmap approaches before 2020, as well as the average precision score COCO test-dev set for top-down and bottom-up approaches before 2020. Zheng et Al. [162] also provided similar comparison including more recent approaches. These surveys however do not provide any extensive performance comparisons for the different methods that they evaluate and they are mainly based on public generic datasets that are not specialized in sports.

2.1.2 Surveys of HPE for sports

Badiola-Bengoa et Al. [8], performed a systematic review in which the objective was to provide an analysis of the application of HPE to the field of sport and physical exercise. The authors discussed the most used metrics and datasets. They introduced few 2D datasets for training HPE systems which were specialized for sports and physical exercise. Examples of these datasets include Leeds Sports Pose, Penn Action, and PoseTrack. They also introduced other datasets that are not specifically designed for this domain, but cover some sports or physical activities, such as Common Objects in Context (COCO), and (MPII). This survey however, did not provide numerical measurements.

2.2 Datasets and Benchmarks

In this section, we give details about a selection of the publicly available datasets related to HPE task introduced in the previous section. We also present the different benchmarks related to these datasets for evaluating the different HPE approaches.

2.2.1 Datasets

A number of researchers created datasets to evaluate their proposed HPE methods. The variety of the datasets in terms of the scenes and application field makes the fair

Dataset	**Size**	**Images/ Videos**	**#People**	**#Annotated joints**	**Targeted for sports**
LSP[51]	2k	images	1 each	14	multiple
LSP Extended[52]	10k	images	1 each	14	multiple
KTH Multiview Football II[54]	6k	images	2	14	football
Penn Action[160]	2.3k	videos	1 each	13	multiple
MPII[5]	25K	images	40k	16	with other human activities
COCO[69]	200k	images	250k	17	with other human activities
PoseTrack[6]	514	videos	multi	15	with other human activities

Table 2.1 – Summary of public HPE Evaluation Datasets

comparison of the different algorithms more difficult.

We summarize the most important and high quality publicly available datasets related to the application of HPE in the field of sport:

— LSP, Leeds Sports Pose [51]: This dataset contains 2000 annotated images from Flickr of 2000 people doing sport activities. The activities include athletics, badminton, baseball, gymnastics, parkour, soccer, tennis, or volleyball. The annotated 2D poses have 14 joints of the form (x, y, visibility) where (x, y) denote the pixel coordinates of the joints in the image and visibility represents if the keypoint is visible or not in the image. This dataset has been expanded to Leeds Sports Pose Extended in [52] to contains 10,000 annotated images from Flickr as well.

— KTH Multiview Football dataset II [54]: contains 5907 images of 3 different players during football match of the Allsvenskan league. The images are extracted from video sequences shot at 25Hz with a 1920x1080 pixels resolution for each player. Poses have 14 2D joints.

— Penn Action [160]: provided 2326 video sequences from a single view of different people collected from various video platforms, including YouTube. In each sequence, there is one person performing sport, including baseball swing, clean and jerk, jumping jacks, push up, strum guitar, bench press, golf swing, baseball pitch, sit up, tennis forehand, bowl, jump rope, pull up, squat and tennis serve. The poses have 13 2D joints.

In addition to these datasets in which the content is related to the field of sports, there are other datasets which are not developed specifically for this task, but which motion may be related to physical activity close to sports:

— MPII, Max Planck Institut Informatik dataset [5]: contains around 25,000 images spanning 491 different activities from 3913 videos downloaded from YouTube showcasing every day human activities. More than 40,000 individuals are presented in this database, with 16 annotated 2D body joints. The dataset has been used for a single person, multi-person pose estimation and action recognition models.

— COCO, Common Objects in Context [69]: commonly used for multi-person pose estimation models. It is a collection of a very large dataset with different types of annotations: object detection, keypoint detection, stuff segmentation, panoptic segmentation, and image captioning. It contains various human poses used in different body scales, including occlusion patterns. It contains a total of 200,000 images, with 250,000 people labeled with 17 2D joints.

— PoseTrack [6]: contains 514 videos including 66,374 frames: 300 videos training, 50 videos validation and 208 videos testing. These videos contain several people performing various activities. The videos come from MPII Human Pose dataset. This dataset is used for single-frame multi-person pose estimation, multi-person pose estimation in videos and multi-person articulated tracking. The poses are labeled with 15 2D joints.

These datasets do not include images or videos involving special fighting movements and only provide generic class labels for the sport activities. For example, various upper body attacks are labeled as punches or boxing.

In this work, we evaluate different state-of-the-art HPE methods on a collected dataset specialized in boxing, involving various classes of movements. A description of the evaluated methods is provided in 2.3 and the collected dataset is described in the section 2.4.1.

2.2.2 Metrics

Many evaluation metrics have been introduced in the literature to measure the performance of 2D HPE. Many factors should be considered when evaluating these methods: single/multi pose estimation, single images or video input, with/without camera motion. In this work, we mention the most commonly used ones:

— Percentage of Correct Parts (PCP): an evaluation metric that was introduced in early works of 2D HPE. In PCP, a limb (or body part) is considered as detected if the distance between the two predicted joint locations and the true limb joint locations is less than a fraction of the limb length (usually between 0.1 to 0.5) [132]. Hence, the higher the PCP, the better performance for the HPE. Recently, PCP has not been preferred as an evaluation metric because it penalizes shorter limbs [4].

— Percentage of Correct Keypoints (PCK): A detected joint is considered correct if the distance between the predicted and the true joint is within a certain threshold. A higher value of the PCK means a better model performance [129].

— Percentage of Detected Joints (PDJ): The percentage of predicted joint locations which are within a fraction of the bounding box's diagonal. It is similar to PCK for measuring the accuracy of localization of the keypoints but, instead of a fixed threshold, we use the diagonal of the bounding box of the person in each frame [132].

— Object Keypoint Similarity (OKS): A metric commonly used in the COCO Keypoint Challenge defined by the following formula:

$$OKS = \frac{\sum_i \exp\left(-d_i^2/2s^2k_i^2\right)\delta(v_i > 0)}{\sum_i \delta(v_i > 0)}, \tag{2.1}$$

where d_i is the Euclidean distance between the detected keypoint and the corresponding ground truth, v_i is the visibility flag of the ground truth, $\delta(v_i > 0)$ is referring to those samples that are labeled, s is the object's scale (square root of the object segment area), and k_i is a per-keypoint constant that controls falloff that was tuned such that the OKS is a perceptually meaningful and an easy to interpret similarity measure [69]. This metric plays the same role as Intersection over Union in object detection. It is calculated from the distance between predicted points and ground truth points normalized by the scale of the person. Typically, standard average precision and recall scores are reported in papers: AP^{50} (Average Precision at OKS = 0.50) AP^{75}, AP (the mean of AP scores at 10 positions, OKS = 0.50, 0.55..., 0.90, 0.95), AP^M for medium objects, AP^L for large objects, and Average recall (AR) at OKS = 0.50, 0.55..., 0.90, 0.95 [8].

— Mean Per Joint Position Error (MPJPE): the average Euclidean distance between predicted and ground truth joint locations [66]. Root Mean Squared Error (RMSE) can also be used in a similar way.

2.3 Choice of Human pose estimation methods

The aim of this work is to challenge recent (ate the beginning of the PhD) and popular HPE methods in the context of fighting sports. Among all the proposed methods, we have selected the ones with the highest reported performance (either most recent or most popular methods), among the main families of approaches. We also selected methods for which a code is available, to ensure using these methods as defined and tuned by the authors.

These methods all start from RGB images (either isolated images of sequences of images). For fast motion, such as boxing motion, it seems interesting to test the performance of sequence-based approaches, that consider dynamic properties of the motion. In fact, in fast and jerky motion, this type of approach may have some difficulties compared to traditional per-frame tracking approaches.

Recent works generally consider that the image is pre-segmented to isolate each person before pose estimation (named "top-down" approaches compared to "bottom-up" approaches which do not use this pre-segmentation). In the context of fighting sports, at least two opponents are in the image, with potential occlusions from one character by the other one. Hence, using or not pre-segmentation may have an influence on the HPE results in this difficult context.

Cascaded pyramid based approaches [17] have been designed to handle difficult occlusions, which occur frequently in fighting sports. It was consequently important to include a recent and popular method based on this approach.

Several previous works adapted neural network architectures designed for image segmentation, in order to track joints, such as Detectron2 [142]. Other methods replace the traditional heat maps by regressions on the positions.

Many other methods could have been tested in this benchmark, but we tried to address most of the main families of these methods, when this work was carried-out. A brief introduction of each selected method is presented in the following subsections.

Table 2.2 summaries the 6 different models tested in this study.

Method	**HPE Type**	**Approach**	**Input**
OpenPose	Bottom-Up	Part Affinity Fields for part association then confidence maps for part detection	Images
Detectron2	Top-Down	Person detection then modeling keypoint location as a one-hot mask then Mask R-CNN to predict K masks, one for each of K keypoint types	Images
SimDR	Top-Down	Disentangles the x- and y- coordinate of joint location into two independent 1D vectors, regarding the keypoint localization task as two sub-tasks of classification at horizontal and vertical directions	Images
CPN	Top-Down	Includes a GlobalNet based on the feature pyramid structure and a RefineNet which concatenates all the pyramid features as a context information. Online hard keypoint mining is integrated in RefineNet for the "hard" keypoints	Images
AlphaPose	Top-Down	Consists of three components: Symmetric Spatial Transformer Network, Parametric Pose Non-Maximum-Suppression, and Pose-Guided Proposals Generator	Images
DCPose	Top-Down	Pose Temporal Merger and a Pose Residual Fusion module allow abundant auxiliary information to be drawn from the adjacent frames, providing a localized and pose residual corrected search range for location keypoints. Pose Correction Network employs multiple effective receptive fields to refine pose estimation in this search range	Videos

Table 2.2 – Summary of the different HPE methods tested in this study.

2.3.1 OpenPose

OpenPose [16] is one of the most popular bottom-up methods for multi-person human pose estimation. It features real-time and multi-person pose estimation and uses multi-stage convolutional neural network. It iteratively predicts affinity fields to encode part-to-part association, while refining the predictions over successive stages. Then it predicts confidence maps over another set of stages.

OpenPose use a multi-stage Convolutional Neural Network (CNN) architecture for real-time articulated human pose and motion estimation. The system initializes the CNN architecture with the initial 10 layers of the pre-trained VGG-19 model, followed by fine-tuning to adapt the network to the specific task. The input image undergoes analysis through this initialized CNN, resulting in a set of feature maps that serve as input to the first stage.

The first stage of OpenPose's pipeline focuses on estimating Part Affinity Fields (PAFs). This stage adopts an iterative approach where the predicted PAFs are progressively concatenated with the original image features. This strategy facilitates refining the predictions throughout successive stages, enhancing overall accuracy and robustness. Notably, PAFs prove instrumental in associating body parts across multiple frames or individuals within complex scenes.

Following the first stage, the second stage enters the scene, dedicated to predicting confidence maps — a crucial component responsible for detecting individual body parts. Similar to the first stage, it embraces the same iterative methodology, continually integrating prior knowledge from preceding steps to generate increasingly accurate confidences maps. These enhanced confidence maps facilitate precise localization and identification of various body parts present in the input images.

Each stage comprises numerous convolution blocks, acting as building blocks for constructing these neural networks. In turn, each convolution block consists of three 3 x 3 convolutional kernels arranged in a sequential manner. By concatenating these filters, computational overhead can be effectively reduced without compromising performance, ultimately contributing to efficient processing and rapid inference times inherent to OpenPose.

This method was chosen for this study because of its popularity and as being the default method for a lot of works that involves action recognition using skeleton data. It is also used as a candidate for bottom-up approaches in multi-person HPE.

2.3.2 Detectron2

Detectron2 [142] is a Meta AI Research's library that provides state-of-the-art detection and segmentation algorithms. It allows to detect person keypoints (eyes, ears, and main joints) for human pose estimation.

The person keypoints estimation is done on individual images. It gives the bounding box of the human and their keypoint estimations using the available COCO Person Keypoint Detection model with Keypoint R-CNN. The model is based on Mask R-CNN [43], which is flexible enough to extend it to human pose estimation. The keypoint's location is modelled as a one-hot mask, and Mask R-CNN is adopted to predict K masks, one for each of K keypoint types (e.g., left shoulder, right elbow). It is a top-down method where we first detect humans and then estimate keypoints within each bounding box independently.

We chose this method because of its popularity as well as for evaluating the flexibility of the Mask R-CNN model.

2.3.3 SimDR

The "SIMple yet promising Disentangled Representation" (SimDR) for keypoint coordinate [67] tries to address the shortcomings of 2D heatmap representation used in the majority of HPE methods. Their approach reformulates human keypoint localization as a classification task.

They proposed to disentangle the representation of horizontal and vertical coordinates for keypoint location, leading to a more efficient scheme without extra up-sampling and refinement. This allows to directly remove the time-consuming up-sampling module of some HPE methods, which may induce lightweight architectures for HPE.

This method was chosen for being recent and for the fact that it does not use heatmap representation as the other methods.

2.3.4 Cascaded Pyramid Network

Cascaded Pyramid Network (CPN) [17] is a multi-person pose estimation model that aims to tackle the problem of "hard" keypoints estimation in challenging cases, such as occluded keypoints, invisible keypoints and complex background.

The approach consists of two stages: GlobalNet and RefineNet. The GlobalNet is a feature pyramid network based on ResNet backbone that can successfully localize "simple"

keypoints like eyes and hands, but may fail to precisely estimate the occluded or invisible keypoints due to its reliance on context rather than the appearance feature nearby. The RefineNet explicitly handles the "hard" keypoints by integrating all levels of feature representations generated by GlobalNet, and tweaking the loss propagation in an online manner to focus on hard keypoints [17].

The two stages process, together with the tackling of hard keypoints in occlusions and complex background, were the two main motivations to consider this method in our study.

2.3.5 Alphapose

Alphapose [31] is a Regional Multi-Person Pose Estimation. It is a popular top-down method for pose estimation. AlphaPose architecture is applicable for detecting both single and multi-person poses in images or video fields. It consists of three components: Symmetric Spatial Transformer Network (SSTN) with parallel single-person pose estimator (SPPE), Parametric Pose Non-Maximum-Suppression, and Pose-Guided Proposals Generator (PGPG). In particular, PGPG is a method used for data augmentation, so that the SSTN+SPPE module adapts to the 'imperfect' human proposals generated by the human detector. The SPPE becomes able to handle human localization errors due to the utilization of symmetric STN that selects region of interests automatically. Conversely, parallel SPPE helps the STN to focus on the correct area and extract high quality human-dominant regions. Finally, the parametric pose NMS can be used to reduce redundant detections.

This method was chosen for its popularity and its unique way for handling person detection issues in a top-down approach.

2.3.6 DCPose

Deep Dual Consecutive Network for Human Pose Estimation (DCPose) [78] is a video-based human pose estimation framework, using temporal information from adjacent frames, after generating pose heatmaps to facilitate keypoint detection.

Three modular components are designed in the framework:

— A Pose Temporal Merger that encodes keypoint spatial context based on initial estimation to generate effective searching scopes to correct pose prediction contained in a specific range.

— A Pose Temporal Merger that encodes keypoint spatial context based on initial estimation. It aims to generate effective searching scopes to correct pose prediction contained in a specific range.

— Pose Correction Network aims at refining efficiently the pose estimations.

We chose this method since it considers temporal dependencies of poses between adjacent frames which is more practical for HPE from videos.

2.4 Comparative Methodology

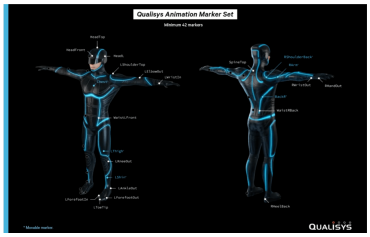
In order to compare the different HPE methods, we collected sequences of boxing, associated with a ground truth skeleton. The sequences were acquired with a RGB camera and an optical marker-based Qualisys motion capture system. The Qualisys system reconstructs the 3D positions of the markers placed on a fighter. The ground truth 2D skeletons used for the comparison to 2D HPE methods, is obtained by projecting the 3D skeleton on the 2D image plane. To this end, we used the intrinsic and extrinsic parameters of the cameras used to collect the 2D videos. Thus, it requires a calibration step between the Qualisys 3D space and the RGB camera 2D space, as well as a synchronisation of the sequences recorded by the two devices. Finally, as the Qualisys skeleton model is defined differently from that used by the HPE methods, a mapping step between the two models is necessary to obtain a representation against which to compare the methods' results.

In this section, we describe the different steps used to prepare the boxing dataset. Section 2.4.1 details the material, the different processing steps and the protocol used to acquire the test sequences. Section 2.4.1 explains how the 2D skeleton data provided by the HPE methods based, and the reference skeleton are compared.

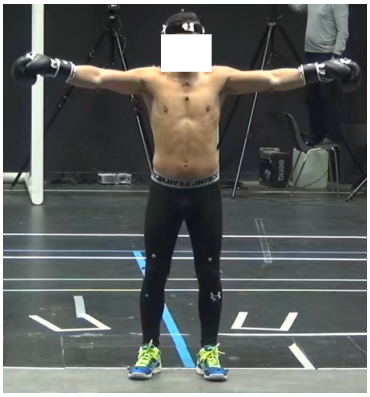
2.4.1 Sequences' acquisition

Different systems for recording motion capture are available. Inertial systems track the acceleration and orientation of sensors attached to participants/objects in three dimensions, but the accuracy is lower than when using opto-electronic systems.Thus, we preferred to use an optical marker-based system composed of reflective markers and infrared cameras. We used the Qualisys motion capture system as a reference system, which is commonly used in clinical and sport movement analysis studies. Using linear transformation, the system acquires the exact position and orientation of each camera, with respect

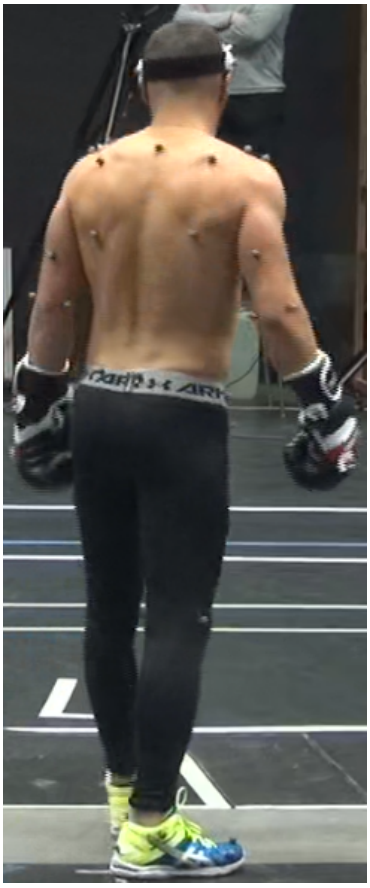
to the others, to be able to create the three-dimensional representation of the capture space and triangulate the marker positions[109]. This system is capable of tracking individual markers with sub-millimeter accuracy. The Qualisys system used was composed of 22 Oqus cameras with 200Hz sampling frequency. 46 markers were placed on the boxer's body according to the Qualisys animation marker set guidelines for motion capture as shown in Fig. 2.1. The motion capture sequences were recorded using the Qualisys Track Manager [108] (QTM) Software to reconstruct, label, and track a calibrated skeleton. Motion capture skeleton data files were exported in TSV format (Qualisys) and we used the resulted skeleton as a ground-truth for HPE methods.



(a) Qualisys animation marker set



(b) Marker placement on the boxer's body from front



(c) Marker placement on the boxer's body from behind

Figure 2.1 – Marker placement on the boxer's body according to Qualisys animation marker set guidelines.

During the motion capture, an RGB camera was located to the right hand side of the

boxer while making sure that they were completely in the field-of-view of the camera. This way we simulated similar side viewing conditions found in television broadcasting of boxing fights, as shown in Fig. 2.2. The used RGB camera was a Sony Exmor R HDR-GW66, and the video footage of all sequences was taken in 1080p resolution at 25 Hz, with low distortion.

Figure 2.2 – RGB camera recording located on the right side of the boxer.

Synchronization and system calibration

To compare the video (HPE) and motion capture outputs, we has to deal with temporal and spatial alignment between both records. The two main ways of synchronizing multiple cameras are 1) hardware synchronization (that requires particular equipment and camera support) and 2) software synchronization (that involves manual video alignment after recording). As the Qualisys System allows for hardware synchronization for up to 2 attached RGB cameras, we selected this option, for better accuracy.

Since the RGB and infra-red cameras are sampling at different rates, motion capture data was down-sampled from 200 downto 25 Hz in order to match the frame rate of the RGB camera.

For spatial calibration, we chose Zhang method [159]. This method is based on moving a chessboard in the filed of the cameras. The details of the calibration process is found in appendix 2.7.1.

After camera calibration, we get approximations for the camera intrinsic and extrinsic parameters. We can then project the 3D skeleton that we get from Qualisys into the same

image plane, for each camera. Hence, we can compare the joints locations (supposed to be the ground truth) to those obtained with various HPE methods in the camera local image frame (see Fig. 2.3).

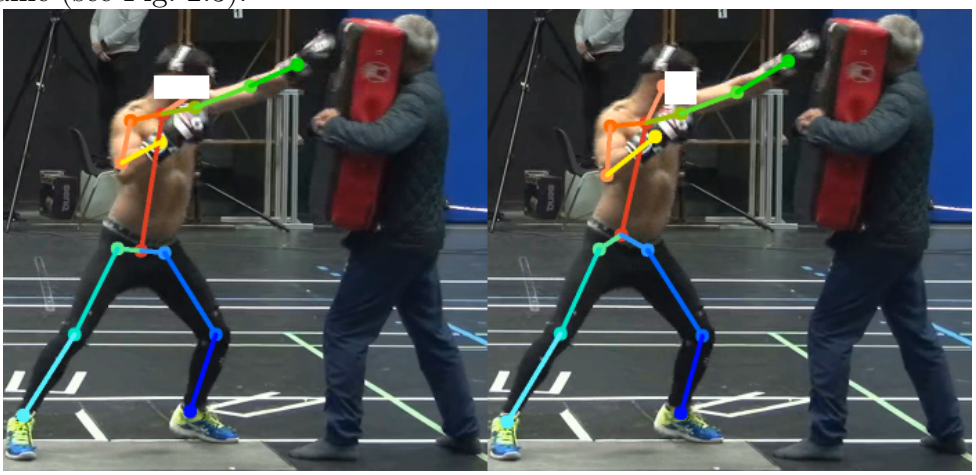


Figure 2.3 – Right: projection of 3D skeleton to the RGB camera image plane, using the camera parameters obtained after calibration. Left: an example of HPE result applied to the same video.

Recording session protocol

The recording session was conducted according to the guidelines of the Declaration of Helsinki, and approved by the Institutional Review Board (or Ethics Committee) of Inria (protocol code 2021-17 May 21st 2021). The volunteer was a professional boxer (27 years old, super bantamweight category, 57.2 kg, European Boxing Union, Pro Boxing Record: 16-0-0 (Win-Loss-Draw)) at the time. They were accompanied by their coach who played the role of opponent and was equipped with protections, as depicted in Fig. 2.3.

After being equipped with the 46 reflective markers, the subject performed a warming-up, which also enabled him to get used of wearing the markers. Then, the subject was instructed to perform a series of boxing actions that involved offensive actions (Jab, Cross, Hook, Uppercut), as well as defensive actions (external and chasing parry, retreat and dodging). Each sequence contains a repetition of an action or a combination of actions for a number of times. In total, the boxer performed 25 sequences, each sequence lasting in average 23 seconds. The action sequences are detailed in Table 2.9 in appendix 2.7.2.

Data processing

After the motion capture session, we get initial data from the Qualisys Motion Capture system. The raw data consisted of markers relative positions to the stage's origin. A labeling process is then done by assigning each marker to a body part according to Qualisys Animation Marker Set, and corresponding skeleton. Afterwards, Skeleton Solving is used to calculate skeleton data, i.e. 3D joints locations, and joint angles. We exported skeleton data in TSV format for each action sequence. Each file contains for each frame the 3D position of 24 skeleton joints defined by Qualisys system in Cartesian coordinates.

All tested HPE methods were trained originally on the COCO dataset and offer pre-trained models that use 17 COCO keypoints. Therefore, matching keypoints from COCO and Qualisys skeletons is required in order to compare them respectively. Keypoints used by COCO are shown in table 2.3

The skeleton data for all sequences were preprocessed to keep only 3D positions of joints that are used in COCO skeleton keypoints. In fact, the definition of joints positions between Qualisys and the COCO model are slightly different. Joint location is more precise for Qualisys since it uses a well-admitted biomechanical model with accurate joint center estimation based on external markers. The COCO model is less accurate, and some keypoints may be located outside of the body. This is especially true for the root joint that is considered to be on the middle of hips for COCO definition while being on the pelvis for Qualisys. Therefore only keypoints that are similarly placed in the two models were considered for the evaluation. Hence, we have excluded the head, the neck and the root joints, which differ between the two systems. The number of skeleton keypoints was reduced from 24 down to 15 as shown in Fig. 2.4.

When reducing skeleton keypoints, we kept the head, neck, and pelvis for the sake of visualisation. They are not considered for the evaluation, as explained in a later section. The COCO model also includes some keypoints that are not relevant for action analysis, such as ears and eyes keypoints. Those were also not considered for the evaluation. Table 2.3 shows keypoint naming used by the two skeleton representations and their correspondences: 2 keypoints on the same row are considered to be equivalent.

After down-sampling and reducing the number of keypoints, 3D skeleton data were projected to 2D images for each point of view, using the corresponding camera projection matrix. The resulting projected 2D points served as a ground truth to assess the quality of 2D pose estimation. This comparison was carried-out on the 16,462 frames collected along the motion capture session.

COCO	**Qualisys**
Nose - 0	*Head - 5*
	Neck - 4
Right Shoulder - 1	RightShoulder - 11
	RightArm - 12
Right Elbow - 2	RightForeArm - 13
	RightForeArmRoll - 14
Right Wrist - 3	RightHand - 15
Left Shoulder - 4	LeftShoulder - 6
	LeftArm - 7
Left Elbow - 5	LeftForeArm - 8
	LeftForeArmRoll - 9
Left Wrist - 6	LeftHand - 10
	Spine2 - 3
	Spine1 - 2
	Spine - 1
	Hips - 0
Right Hip - 7	RightUpLeg - 20
Right Knee - 8	RightLeg - 21
Right Ankle - 9	RightFoot - 22
	RightToeBase - 23
Left Hip - 10	LeftUpLeg - 16
Left Knee - 11	LeftLeg - 17
Left Ankle - 12	LeftFoot - 18
	LeftToeBase - 19
Right Eye - 13	
Left Eye - 14	
Right Ear - 15	
Left Ear - 16	

Table 2.3 – Comparison of the different keypoints used in COCO and Qualisys skeleton representations. COCO keypoints in italic are not considered for the evaluation.

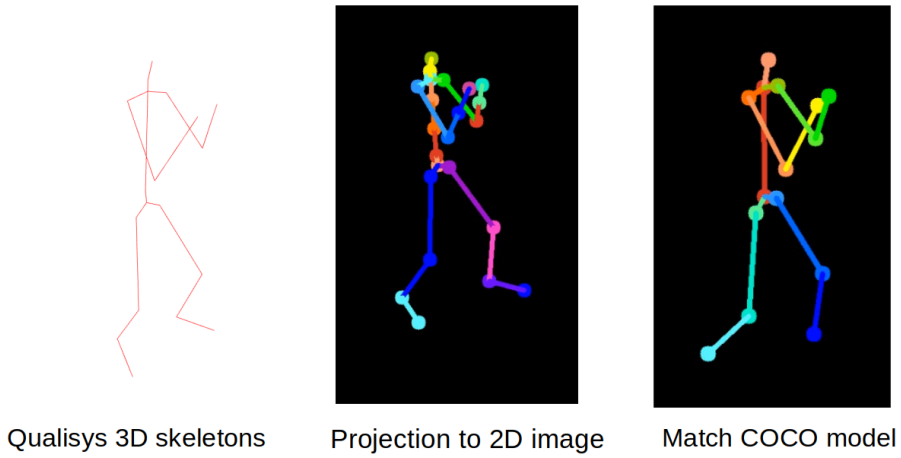


Figure 2.4 – Projection of Qualisys 3D Skeleton (left) to a 24-joint skeleton in the 2D image (middle), and reduction to the 15 joints that match COCO keypoints model (right).

2.4.2 2D pose estimation using RGB data

The boxing video sequences are used as inputs to the 6 HPE methods described in section 2.3. We give here some details about the configuration that we used for the inference for each of those methods. One has to notice that we used the same person detector Mask-RCNN Resnet50 FPN for top-down methods, when generating bounding boxes.

— OpenPose: we used OpenPose version 1.7.0 [94]. We performed inference using two of their pretrained models: COCO model and the body_25 model. The last one covers COCO keypoints extended with foot keypoints, and gives better results than COCO model [16].

— Detectron2: we used the Detectron2 pipeline [29], which is Modular image processing pipeline using OpenCV and Python generators powered by Detectron2. We used the COCO Person Keypoint Detection Baselines with Keypoint R-CNN R50-FPN 3x [142].

— SimDR: we used HRNet-W48 as a backbone for the SimDR representation [118].

— CPN: we used the Pytorch implementation of this method and the pretrained COCO.res101.384x288.CPN model [24].

— AlphaPose: we used their model Fast Pose (DUC) with ResNet152 as a backbone [3].

— DCPose: we used their pretrained model on the COCO dataset [28].

The inference of these methods returned the estimation of 17 2D keypoint locations, based on the COCO skeleton model, for each detected person in each video frame.

Since we had multiple people on the stage, they were detected by all HPE methods. Therefore, a tracking heuristic was applied in order to keep track of the subject of interest, and we eliminated the other skeletons. The tracking method uses the position of the target skeleton. Its position is calculated as the geometric center of all detected keypoints, or the center of the bounding box of each skeleton. Then we computed the distance d_s between the target skeleton's position at the current frame, and all skeleton's positions in the next frame. We assumed that the skeletons with the minimal distance between two successive frames belong to the same target. In order to prevent tracking failure related to occlusions, another distance d_c based on dominant color of upper and lower body parts is linearly combined with the skeleton's position distance: $\alpha d_s + (1 - \alpha) d_c$. This ensures that given two people wearing clothes with different colors, even if they occlude each other, the tracking would still keep following the right skeleton. The heuristic was implemented in a graphical user interface to allow human interaction while the tracking is done in real time, in order to allow for manual re-targeting in exceptional miss-tracking cases.

2.4.3 Evaluation metrics

In this study, we consider three main metrics for comparing predicted and ground truth joint locations:

— Percentage of Correct Keypoints with 150 mm as a threshold (PCK@150mm),

— Percentage of Detected Joints at 5% of the bounding box diagonal (PDJ@5%),

— and RMSE in mm between estimated and ground truth joint locations for every frame.

The conversion from distances of frame pixels to millimeters (mm) was done using the ratio of boxer's torso real length (439.56 mm) and projected skeleton spine length in the image. We did not consider the average precision of OKS because it uses per-keypoint constants that were fine-tuned for COCO dataset, while we do not consider exactly the same keypoints.

Each metric can be calculated with or without first aligning the predicted joints to the ground truth by considering relative positions to pelvis joint or middle of the hips. As mentioned before, since the obtained skeleton joint positions of Qualisys system differs slightly from COCO skeleton model, the 12 joints that are considered for the evaluation are: shoulders, elbows, wrists, hips, knees and ankles, right and left each. Figure 2.5 shows

the COCO and Qualisys skeletons used in visualization. Keypoints in blue are considered for evaluation.

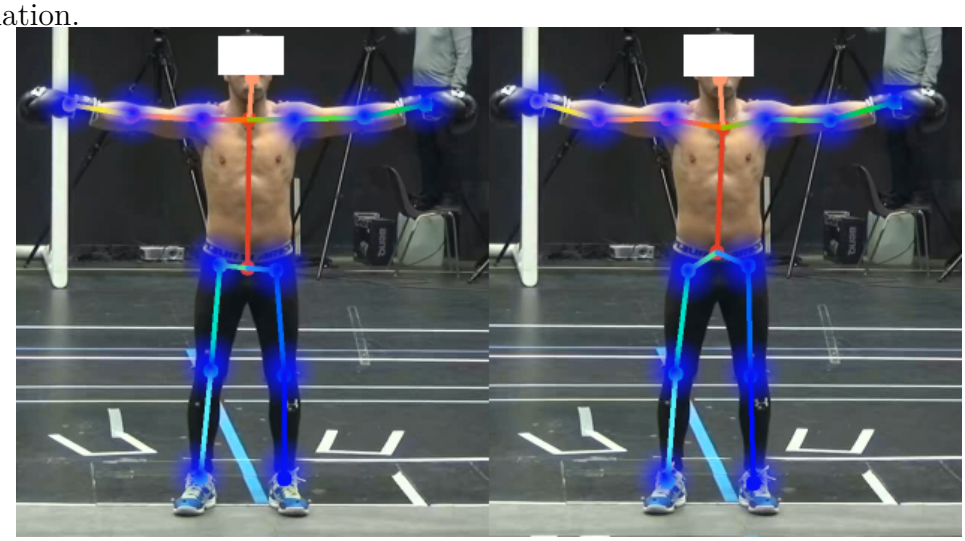


Figure 2.5 – Qualisys Skeleton (right) after reducing the number of keypoints to 15 and COCO skeleton (left). Keypoints in blue are considered for evaluation.

Since the 3D skeleton obtained from Qualysis can be ill-projected for some frames due to numerical errors, we used an alignment where the median of hips is the origin of the relative positions. Using alignment typically gives a better indication of the accuracy of the local configuration of joints, eliminating errors related to translation.

2.5 Results

We ran the 6 selected HPE methods on our entire boxing dataset which represents 16,462 images in total. Global results as well as analysis per joint and per actions are given in this section.

2.5.1 Global measures

As explained in the previous section, since skeleton definition between Qualisys and COCO models are slightly different, estimation errors can be calculated in 3 different ways: by considering absolute positions, relative positions to skeleton pelvis joint (root), or relative positions to skeleton hips's median.

We compute RMSE using these 3 different origins when applying OpenPose on each frame of the first boxing sequence. In this first sequence, the boxer was instructed to perform 5 jabs with the left hand. We can see in the Fig. 2.6 that there are spikes in RMSE

occurring in 5 regions of the graph which shows how Openpose struggled to estimate pose for this fast action.We can also see that the shape of the graph does not change much between the 3 coordinate origins. However, the RMSE calculated by using joint positions relative to Qualisys hips median is the lowest one. Similar results were obtained from the other HPE methods on the rest of the sequences.

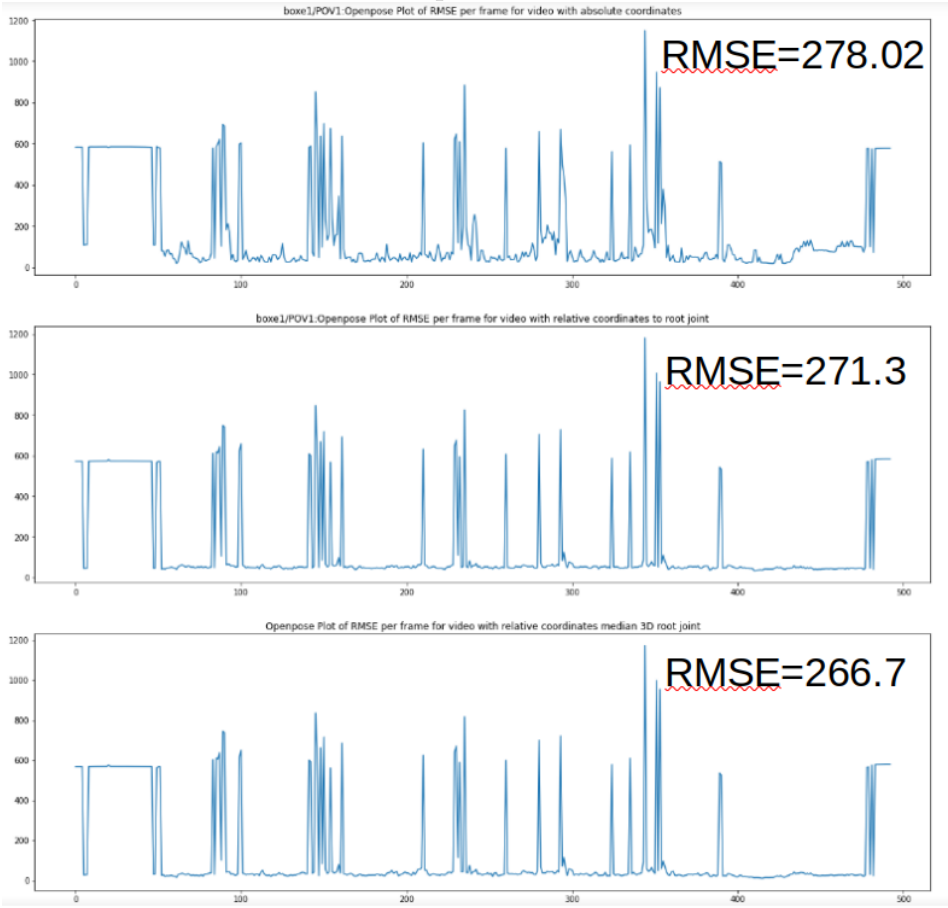


Figure 2.6 – RMSE in mm per frame computed on the sequence #1 for the OpenPose method with 3 different coordinate origins: (top) absolute coordinates, (middle) relative to skeleton root, (bottom) relative to hips median. Spikes in the graph correspond to fast actions (jabs).

In the following, we thus calculate all metrics on the HPE results by considering joint positions relative to skeleton hips's median for COCO and Qualisys skeletons, so that we get spatial alignment of the two skeletons.

Table 2.4 summarizes the global RMSE obtained on the results of HPE methods applied to all the boxing sequences. When applying the bottom-up method OpenPose on the boxing sequences, we notices that in some frames OpenPose returns zero predictions

for joints that are hard to estimate. This leads to having larger errors for this method compared to the other top-down methods, thus affecting the comparison. To show the impact of miss-detections, we calculate RMSE for this method in two ways: by considering miss-detected joint positions and by excluding them. By performing these two calculations, we aim at assessing the impact of rejecting keypoints with low confidence scores, as done by OpenPose. Table 2.4 provides the percentage of miss-detection in both models of OpenPose.

Method	**RMSE Relative to hips median (mm)**		**Missdetection %**
OpenPose Body25	265.7	36.19	1.28
OpenPose COCO	415.6	38.44	3.30
Detectron2	39.84		_
SimDR	41.58		_
CPN	36.87		_
AlphaPose	36.98		_
DCPose	**33.03**		_

Table 2.4 – Global RMSE in mm obtained from HPE methods applied to all sequences including (right) and excluding (left) OpenPose miss-detections.

For the bottom-up method OpenPose, the pretrained COCO model performed way worse than body_25, as reported by the authors in their work. However, when excluding miss-detections, we noticed that results of the two models become very similar. This shows that the body_25 model, in addition to using keypoints on the legs, also improves the detection of joints compared to their pretrained COCO model. But it still performs worse than the other top-down approaches, especially with 1.28% rate for joint miss-detection for the body_25 model.

The top-down approaches Detectron2, Cascaded Pyramid Network and AlphaPose, achieved closely similar results for the global RMSE. SimDR being a top-down approach that does not use 2D heatmap representation, it had better results than OpenPose in our testing, but had higher error than the other top-down methods. This implies that heatmap-based approaches are more suitable in situations where accuracy matters, despite higher execution time. DCPose achieved the lowest RMSE error in our dataset compared to the rest of the methods. This clearly shows the impact of using auxiliary features from

adjacent frames, which can help in refining the joints estimation.

Table 2.5 reports results based on the Percentage of Detected Joints (PDJ), at 5% of the boxer bounding box's diagonal. It also provides the Percentage of Correct Keypoints (PCK) using 150 mm as a threshold. Results are similar to the RMSE: OpenPose has the lowest detection rate for both their models. DCPose scores the highest results, and the other top-down approaches competing with close results. It is worth noticing that SimDR, an approach that does not use heatmaps, outperformed AlphaPose in PDJ@5% and Detectron2 in PCK@150mm.

Method	**PDJ@5%**	#	**PCK@150mm**	#
OpenPose Body25	0.82	6	0.90	6
OpenPose COCO	0.80	7	0.88	7
Detectron2	0.839	2	0.910	5
SimDR	0.8315	4	0.911	4
CPN	0.832	3	0.9173	2
AlphaPose	0.8312	5	0.912	3
DCPose	**0.864**	1	**0.9174**	1

Table 2.5 – Global PDJ and PCK comparison

2.5.2 Per-joint analysis

In order to evaluate the impact of view point and occlusions on the estimation of joint positions, we calculated the global RMSE in mm per joint. We especially focused on joints that were mostly relevant for the evaluation of this type of boxing movements. Results are given in Table 2.6. We found that the RMSE for joints that were facing the camera across all sequences (joints on the right side of the boxer's body) had lower errors compared to the ones that were occluded. This result was expected, as the models had to estimate joint positions based on their prior of the human skeleton model. The table 2.6 also shows the impact of joints involved in boxing actions on the errors: joints that are less involved in boxing actions such as hips and knees have lower errors compared to others that are actively involved such as wrists and elbows. This implies that the different HPE methods struggle with estimating positions of joints in fast motion, especially captured at

low refresh rate. The level of struggling depends on the used approach as well. Although, some methods have similar results in global metrics, we can see some differences in the results per joint. Hence, CPN scored the best for the left elbow that was completely occluded in most of the sequences. This shows the impact of their RefineNet network for tackling the hard keypoints. Detectron2 also had the lowest error for the knees. Methods that interpolate joint positions and have components doing post-processing based on adjacent frames, such as DCPose, achieved generally better results in this area than other methods that uses no temporal dependencies.

Mthd.	OpenPose Body25	OpenPose COCO	Detectron2	SimDR	CPN	AlphaPose	DCPose
Right Shlder	39.09	64.38	33.67	43.19	37.85	34.26	**30.99**
Left Shlder	46.84	85.71	26.96	39.19	31.35	26.82	**25.36**
Right Elbow	88.97	177.53	33.01	41	34.86	32.97	**27.06**
Left Elbow	353.64	466.25	75.38	66.40	**58.21**	64.66	58.22
Right Wrist	182.89	349.05	40.78	61.01	43.39	43.36	**37.87**
Left Wrist	764.59	1220.87	73.59	84.32	71.99	70.80	**58.55**
Right Hip	12.08	46.06	10.62	12.07	10.74	**10.14**	11.07
Left Hip	18.54	46.06	10.62	12.07	10.74	**10.14**	11.07
Right Knee	24.32	69.92	**23.29**	31.26	27.20	24.56	27.78
Left Knee	46.85	68.94	**25.43**	39.87	29.32	28.90	25.85
Right Ankle	71.23	90.72	36.64	51.09	39	34.81	**28.73**
Left Ankle	74.31	89.65	44.23	55.34	44.11	40.66	**34.46**

Table 2.6 – RMSE in mm by joint obtained from HPE methods applied to all sequences. Left joints are more likely to be occluded.

2.5.3 Per-action type analysis

Since our dataset consists of sequences of boxing actions performed by a professional boxer, we segment and annotate boxing actions in each sequence. The classes that we considered are: uppercut, engagement, hook, jab, parry/block, retreat/evade and cross.

We calculated the RMSE in mm and PCK@50mm for frames involving these actions. The results in tables 2.7 and 2.8 show how each HPE method performs for different actions. Table 2.7 shows that actions performed at high speed (uppercut, hook, jab, cross, and evade) mostly have the highest errors, for all methods, compared to boxer's engagement and parrying/blocking. The cross has the highest error, and it also corresponds to the faster motion. Retreating and evading the opponent require from the boxer to move their upper-body, or their whole body, to step back, or lean down/sideways. This type of motion may lead to upper-body self-occlusions, which is a complex problem to solve for HPE. We reported lower RMSE for CPN compared to DCPose for Retreat/Evade action.

Method	**Cross**	**Hook**	**Uppercut**	**Jab**	**Retreat/ Evade**	**Parry/ Block**	**Engagement**
OpenPose Body 25	432.75	236.97	188.90	183.84	167.17	93.34	103.80
OpenPose COCO	638.60	442.93	479.48	459.64	348.73	283.05	309.71
Detectron2	50.87	46.67	43.77	49.20	38.66	38.45	35.57
SimDR	55.28	46.11	45.98	41.22	44.69	35.48	30.15
CPN	50.84	42.49	40.10	40.66	**36.57**	34.85	29.09
AlphaPose	44.42086	46.35	40.28	35.80	41.10	34.19	32.80
DCPose	**44.42087**	**42.40**	**39.61**	**35.00**	37.16	**32.00**	**27.88**

Table 2.7 – Comparison of RMSE in mm for different boxing actions. Cross, hook, uppercut, jab and evade are fast actions.

Table 2.8 reports the PCK values (with a threshold of 50 mm) for the various HPE methods, for each action class. We chose a smaller threshold than previous evaluations to penalize more miss-detections. We especially noticed that OpenPose gets higher score than most Top-down methods for most actions. When combining these results with RMSE reported in table 2.7, we can conclude that bottom-up methods obtain low RMSE results, mainly because of miss-detections. Detectron2 also scored the best for uppercut and retreat/evade categories.

Method	Cross	Hook	Uppercut	Jab	Retreat/ Evade	Parry/ Block	Engagement
OpenPose Body25	0.589	0.643	0.624	0.677	0.635	0.717	0.743
OpenPose COCO	0.570	0.631	0.621	0.666	0.643	0.709	0.728
Detectron2	0.586	0.673	**0.693**	0.675	**0.689**	0.674	0.748
SimDR	0.588	0.647	0.659	0.686	0.634	0.690	0.752
CPN	0.622	0.665	0.648	0.698	0.657	0.682	0.735
AlphaPose	0.609	0.637	0.646	0.681	0.635	0.692	0.723
DCPose	**0.642**	**0.677**	0.687	**0.722**	0.672	**0.757**	**0.788**

Table 2.8 – Comparison of PCK at 50 mm for different boxing actions. cross, hook, uppercut, jab and evade are considered as fast actions.

2.6 Conclusions

In this work, we analyzed how HPE methods perform in a sport domain involving specific fast boxing motions and we proposed an evaluation protocol for this benchmark. Previous works have evaluated HPE either through global metrics or by evaluating them on general human activities. In this work, we presented a more fine-grained performance comparisons applied to fighting sports, and especially to boxing.

One of the most relevant aspects of this work is to fully evaluate the potential of HPE methods that can be useful in fast paced sport analysis, and activity recognition. This context motivated our choice to analyze their performance with respect to different boxing relevant motions. One important finding was that some joint locations were better estimated than others based on whether they were facing the camera, occluded by the boxer's body, or involved in fast actions. All tested HPE methods were equally struggling to estimate boxer's elbows and wrist, even when facing the camera, as was shown in table 2.6. This shows the difficulty of estimating joints at high speed, and also those covered under special clothing, such as the boxing gloves. A solution to this might be to fine-tune the pre-trained models on small annotated dataset of the sport of interest or to up-sample the video's frame rate using advanced frame interpolation techniques before applying HPE methods. This assumption is supported by current works, such as Kitamura et al.

(2022) [56] who obtained significant improvement of OpenPose for acrobatic movements estimation, after leveraging the training process with new gymnastics and simulated data.

The videos in our experiments included mostly 2 people, therefore the performance of this category of methods might get even worse in real world situations where more people are included (referee and spectators in boxing for example). All used Top-down image-based approaches performed similarly in global results as well as per joint/ action results.

We did not consider computation time in this study, since the application context is offline sport analysis and activity recognition. In general, OpenPose was the fastest for human poses estimation since it does not go through human detection process. SimDR was the second fastest method because of the reduced complexity on the last layer of their model, that does not use heatmaps.

The overall finding of this work was that top-down HPE approaches perform well even with complex background, self-occlusions and occlusions between two opponents compared to bottom-up HPE approaches. Methods that refine their estimations using temporal information from adjacent frames perform the best in these scenarios. Therefore, we suggest that future work related to video-based sport analysis and studies should utilize top-down HPE approaches for motion extraction, especially ones that operate on temporal windows for more accurate estimation.

2.7 Appendix

2.7.1 Camera Calibration

A calibration routine was used to estimate intrinsic and extrinsic parameters of the camera. Intrinsic camera parameters describe internal properties of a camera (focal length (f_x, f_y) and image center (x_{c0}, y_{c0})). Extrinsic camera parameters describe its orientation and position in world space (which we consider to be relative to Qualisys world space for 3D coordinates), in terms of rotation R and translation t. The mapping formula from 3D world space points (X_w, Y_w, Z_w) to 2D image points (x_c, y_c) is given in Fig. 2.7.

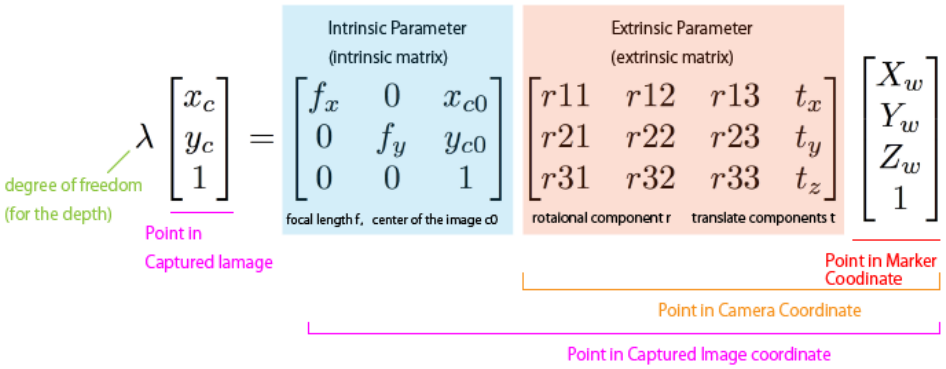


Figure 2.7 – Pinhole Camera Projection formula (from [105]): intrinsic and extrinsic parameters are used to project 3D points (X_w, Y_w, Z, w) to 2D image points (x_c, y_c)

A chessboard was moved around the stage, including deliberate twisting and tilting motions. The used chessboard had 18x12 internal corners. All squares have identical length of 5 cm for good visibility from the cameras. The chessboard was printed on A0 format then glued to a wooden board to ensure co-planarity. Care was taken to keep the front of the calibration board within the field of view during the calibration of the camera as shown in Fig. 2.8. Six markers were placed at different locations on the board to make correspondences between 2D positions in images and 3D positions in the Qualisys coordinate system; This help to estimate the extrinsic parameters with respect to the motion capture reference system.

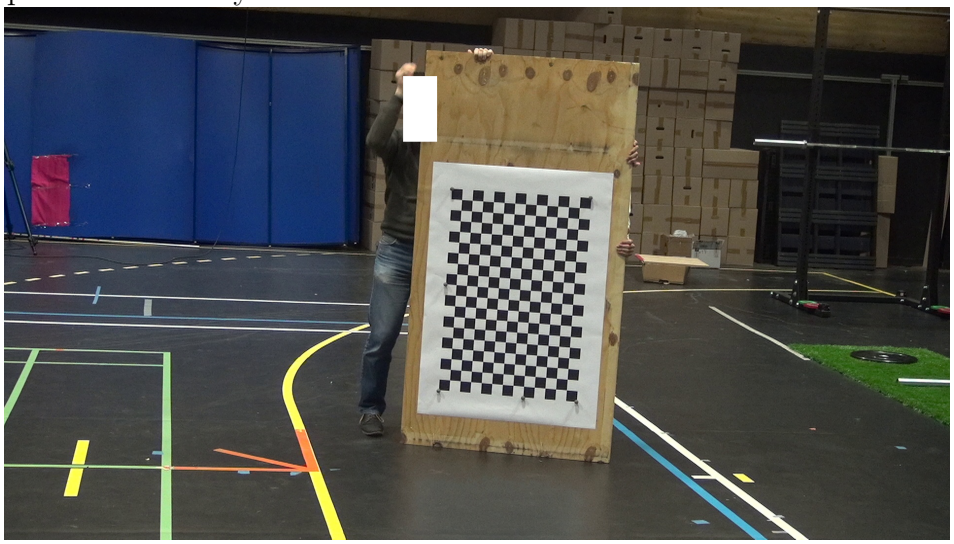


Figure 2.8 – The used calibration chessboard

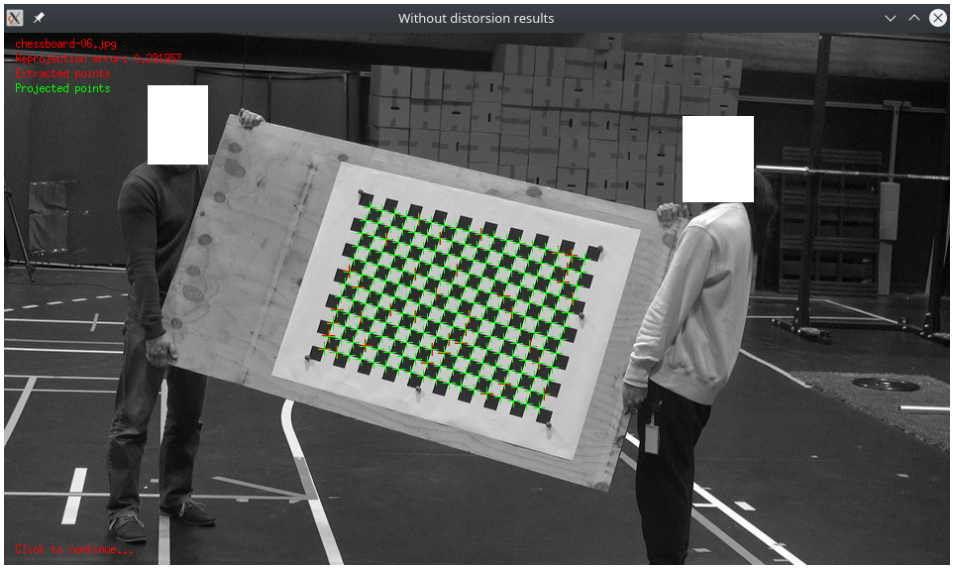


Figure 2.9 – Camera calibration process using OpenCV for intrinsic parameters: Red points are chessboard corners detected by OpenCV in the pose image. Green points are re-projection of the corners from camera reference to image reference using the resulting intrinsic parameters.

For each frame of the calibration sequence, the internal corners of the calibration pattern were detected using the OpenCV library. The detections were then processed by OpenCV to approximate intrinsic matrices for the camera, then its distortion as indicated in Fig. 2.9. The extrinsic parameters were estimated using EPNP (Efficient Perspective-n-Point) [64] implementation of OpenCV.

The global re-projection error from intrinsic camera calibration with distortion was of 0.38 px using 17 different chessboard poses.

2.7.2 Boxing Sequences

Table 2.9 – Summary of the different boxing actions performed by the boxer in each sequence.

Sequence	**Type**	**Number of hits**	**Repetition**	**Actions**
#1	Attack	1	5	Jab Left
#2	Attack	1	5	Cross Right
#3	Attack	2	5	Cross Right Jab Left

#4	Attack	3	5	Cross Right Jab Left Cross Right
#6	Attack	2	6	Jab Left Jab Left
#7	Attack	3	6	Jab Left Cross Right Jab Left
#8	Attack	3	4	Jab Left Jab Left Cross Right
#9	Attack	2	4	Cross Jab Hook Left
#10	Attack	2	5	Cross Jab Hook Left
#11	Attack	3	5	Jab Cross Hook Left Body
#12	Attack	3	5	Jab Left Hook Right Body Hook Right
#13	Attack	3	5	Cross Right Hook Left Body Hook Left
#14	Attack	2	5	Hook Right Hook Left Body
#15	Attack	2	6	Hook Right Body Hook Left
#16	Attack	2	6	Uppercut Right Hook Left

#17	Attack	2	4	Uppercut Left Hook Right
#18	Attack	3	5	Jab Left Cross Right DisplacementToRight Hook Left
#19	Attack	2	9	Cross Right Jab Left DisplacementToLeft
#20x2	Defense	1	23	External Parry
#21	Defense	1	27	Chasing Parry
#22	Defense	1	15	Retreat
#23	Defense	1	17	Side Dodge
#24	Defense	1	20	Down Dodge

2.7.3 Video Tracking Tool for HPE

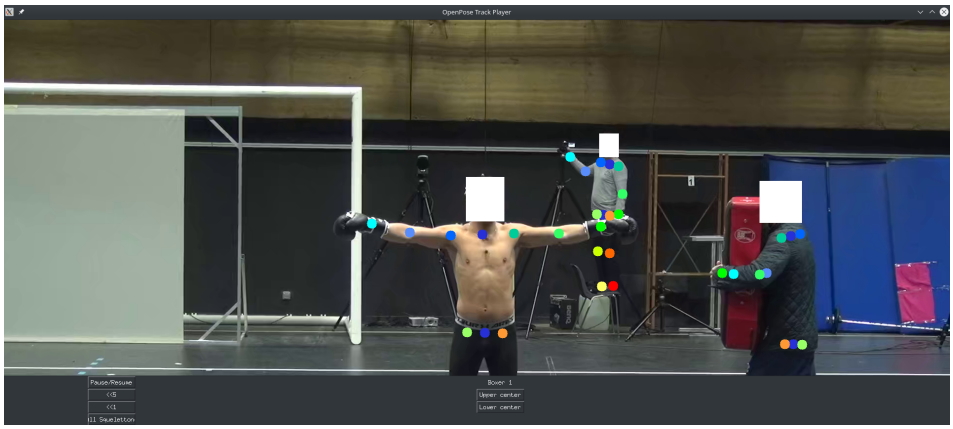


Figure 2.10 – The tool used for tracking keypoints of the subject in a video recording based on distance through successive frames and upper/lower body color regions. Initially, all skeletons detected by the HPE method are displayed. The user then choose the skeleton of interest then the tracking starts for the rest of the video

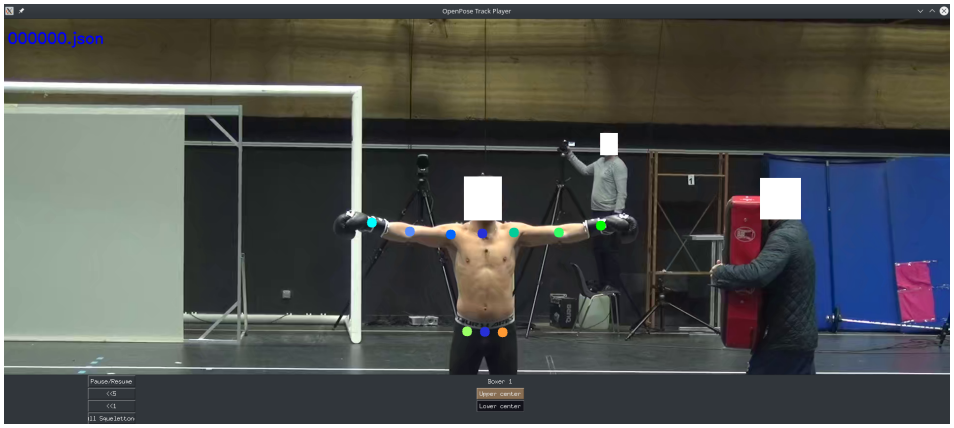


Figure 2.11 – The tool also allows for re-assigning the tracked skeleton by the user and correcting tracking in previous frames.

Chapter 3

Interaction Imitation from motion capture

In the previous chapter, we highlighted the challenges in extracting fast-paced motions of interacting fighters from RGB videos. These challenges include occlusions, cumulative errors, and hierarchical dependencies on extraction systems. Additionally, semantically segmenting motions before modeling interactions between fighters is difficult.

To address these issues, we propose an alternative approach that leverages expert demonstrations and physics-based simulations of motion. By treating the problem of simulating fighter interactions as an imitation learning problem, we aim to control physically capable agents to generate interactions that resemble expert demonstrations captured in motion capture data. This approach aims to guarantee a level of realism in the interaction simulations. This approach also comes with its own set of challenges including the modeling of interaction of two fighters and the leveraging of the motion capture demonstrations to train physically simulated agents to imitate reactive behavior and also adding control to it.

Multiple physics-based character interactions can be simulated using space-time constraints and optimal control [72, 133]. These approaches can find an optimal solution given a set of manually edited constraints, but may fail to imitate the style given in a small set of examples. Data-driven approaches select the optimal actions available in a database of examples, using game tree based methods [116, 117]. However, due to the simplicity of the rules and the high computational complexity, the intelligence of rule-based simulated characters is too limited to handle stylized interactions [65].

Physics-based character control is an active field of research, as it enables to generate physically valid animations in complex interactive environments. In applications with multiple characters (simulated, or real-time avatars of users), these physics-based characters have to realistically interact with others in a large variety of situations. In the case of Virtual Reality fighting training, the user is not necessarily expecting an optimal be-

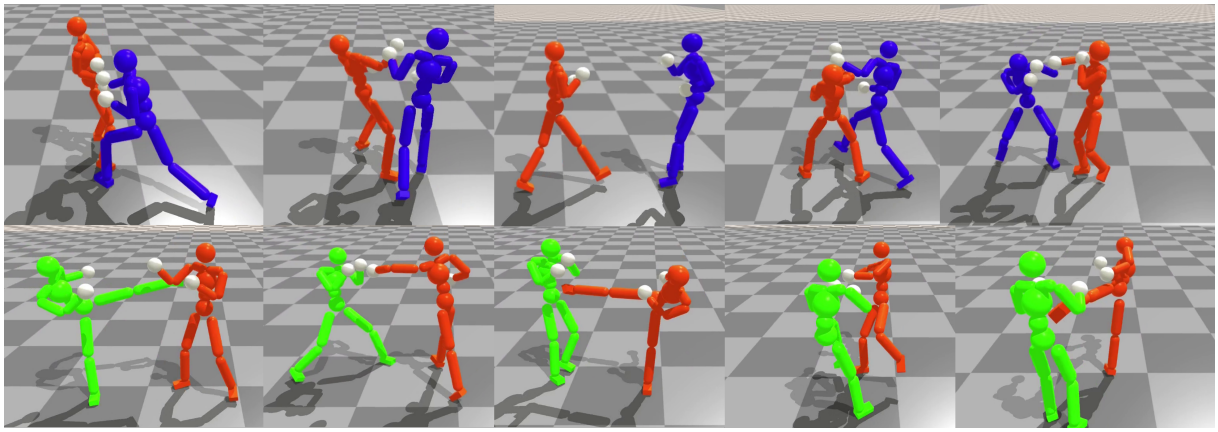


Figure 3.1

Two examples of imitation-based simulation with two different styles. Top: boxing scenario trained with single-actor and multiple-actors motion capture boxing datasets. Bottom: Qwankido (martial art) scenario based on the same approach, simply replacing the boxing datasets by Qwankido datasets. In these examples, the sequences show the ability of the characters to avoid attacks and then to counter-attack.

havior for the virtual opponent, but may prefer to prepare fighting against an opponent that imitates the behavior and the motions of a particular real boxer. Such simulation should take the current state of the interaction into account, select the most relevant action to perform and compute the physically-valid corresponding motion, as a specific human would do, by imitating a small set of examples.

Reinforcement learning based Imitation Learning has been explored for designing physics-based controllers capable of imitating motions given unstructured database of examples while achieving different tasks [102, 138, 99, 139, 140]. Recently, Adversarial Motion Priors (AMP) approach, based on the Generative Adversarial Imitation Learning (GAIL) [46] framework has been proposed. It mainly uses an adversarial discriminator output as a reward instead of manually designing an imitation reward opposite to tracking-based prior work that rely on specifying step-based handcrafted imitation rewards [102, 138]. However, it has only been applied to control a single character to generate motions similar to motion dataset while performing simple tasks such as locomotion, running and hitting a target.

In this chapter, we seek to develop a system, Multi-Agents Adversarial Interaction Priors (MAAIP), for imitating interactions and motions of multiple physics-based characters from unstructured motion clips. Our method is based on the Multi-Agent Generative Adversarial Imitation Learning (MAGAIL) [120] framework, and aims at extending AMP to deal with both the interaction and the motion of the controlled physics-based characters.

Two unstructured datasets are used by the system: 1) a single-actor dataset containing motions of single actors performing a set of motions linked to a specific application, and 2) an interaction dataset containing few examples of interactions between multiple actors. Our system trains control policies allowing each character to imitate the interactive skills associated with each actor from the demonstrations, while preserving the intrinsic style. Similarly to AMP, the single-actor dataset is used to train a single motion prior, while the interaction dataset offers a novel complementary interaction prior to train each agent on how to behave in different interactive situations with other agents. The interaction prior is therefore acting as a measure of similarity between the motions of the characters when interacting with each other, and the interaction examples in the datasets. The single motion prior offers a complementary repertoire of individual possible motions that may not appear, or not sufficiently, in the multiple-actors dataset, which is a flexible framework as interaction data is harder to obtain than single actor data in practice.

In the following sections, we will present our approach for simulating interactions between fighters using imitation learning and physics-based simulation. Our initial results focused on imitating interactions from unstructured data of two fighters performing light shadow-boxing with minimal physical contacts, presented as a poster at SIGGRAPH Asia 2022 [151].

We extended the method to handle boxing with physical contact, as well as another fighting activity with full-body attacks. The approach was also applied to various applications beyond imitation, including user commands and interaction constraints. We evaluated the method by simulating competitive interactions between two physics-based characters with different styles: boxing (fists only) and Qwankido (a Sino-Vietnamese martial art with full-body interactions).

We also showcase how to control the interaction by adding new rewards, such as interactively controlling the direction of the simulated fight, making the fighter more aggressive, or more defensive. These findings were presented at The Symposium on Computer Animation (SCA'23) and published in the journal Proceedings of the ACM in Computer Graphics and Interactive Techniques (PACMCGIT) [152].

3.1 Related Work

Physics-based simulation relies on the dynamic equation of motion to generate joint angles trajectories for a character. However, the main challenge with these methods is to

design a controller that generates realistic motions, with a desired style, and given a set of goals to achieve. In the two next sections, we review relevant physics-based simulation methods for a single (section 3.1.1) and multiple (section 3.1.3) characters. We then introduce Imitation Learning techniques used for physics-based character simulation in section 3.1.2.

3.1.1 Single Character physics-based character control

Physics-based character simulation has a long history in computer animation. Early efforts focused on developing locomotion control using motion analysis and hand-crafted controllers [47], abstract models [22], optimal control [86], model predictive control [37, 84] and reinforcement learning [144, 149]. These approaches typically require prior knowledge and hand-tuned parameters, which can make them difficult to apply to complex motions and scenarios. To address these difficulties, several physics-based controllers have been supplemented with the motion capture data, using trajectory tracking to follow motion clips and a balance controller to keep the character upright [164]. More recent works tracked reference motions by learning policies that get feedback from the physics simulation [63, 74]. With the development of deep reinforcement learning techniques, it became possible to robustly track agile human motions [102] and to generalize to various morphologies [141] and environments [144].

3.1.2 Imitation Learning for physics-based character simulation

Imitation learning in physics-based animation uses reference motion data to improve the quality of the simulated motions. This is typically done by implementing a tracking objective, where the goal is to minimize the error between the simulated poses and example poses. This can be achieved through the use of a phase variable provided as an additional input to the controller for synchronization, or by providing target poses from the reference motion as inputs to the controller [75, 25, 62, 63, 74, 102]. However, using a single phase variable may not allow scaling to datasets containing multiple disparate motions, and using a reference pose as a target for the controller requires a high level controller to select the motions to imitate from as well as the manual definition of the pose error metrics [101].

Adversarial imitation learning [46, 163] is an alternative approach to avoid manually designing and tuning specific pose error metrics. It showed promising results to imitate

motions, given a database of examples [83, 136, 145]. This approach relies on an adversarial discriminator, aiming to distinguish simulated motions from those depicted in the demonstration data. The discriminator is then used as a reward function to train a control policy to imitate the type of motion observed in the demonstration data. However, adversarial learning algorithms can be unstable during training, and the quality of the resulting motion can still be low compared to tracking-based methods. Adversarial Motion Priors (AMP) [99] proposed a number of tweaks to address those issues, such as using gradient penalty, but did not handle interaction imitation of multiple characters.

3.1.3 Multiple characters animation

Simulation of multiple characters interacting with each other involves defining properly the interaction between characters: relative positions between body parts of the characters [45], but also more complex parameters, such as gaze, orientations or time coordination. Optimal control with space-time optimization has been used to solve complex interaction problems involving multiple physics-based characters [72, 133]. It requires careful design and tuning of the cost functions to obtain realistic simulations. Other works proposed an offline game tree expansion to explore all the possible interactions between characters to simulate multiple characters competing or collaborating in a given scenario [116, 117]. All these approaches have been designed to find an optimal solution, but cannot easily imitate realistic behaviors contained in example motion capture clips. When a few examples of interactions are available, reinforcement learning is a promising way to control physics-based characters. [41] proposed a hierarchical policy that incorporates navigation, footstep planning, and bipedal walking skills, for controlling navigation of pedestrians. Unlike previous approaches, this method learns control policies that can handle interactions between multiple simulated humanoids. [139] proposed a two-steps approach that first learns an imitation policy from single-actor motion capture data, then transfers it into competitive policies. The authors of [76] trained football teams of physically simulated humanoids in a sequence of training stages using a combination of imitation learning, single/multi-agent reinforcement learning and population-based methods. However, these approaches have not been designed to leverage available interaction data of a few examples.

Multi-Agent Generative Adversarial Imitation Learning (MAGAIL) [120] is a promising framework to design controllers that imitate the interaction behavior of multiple characters given a small set of unstructured motion capture examples. We explore the use of

this framework to control multiple physics-based characters. This raises the question of how to define the state representation to model the interaction of multiple characters, and how to build a discriminator based on interactions instead of just the motion of a single character.

3.2 Multi Agent Interaction Priors for Fighting Sport

Interaction imitation problem could be viewed as a Partially Observable Markov Game, where the goal is to learn optimal policies of multiple agents interacting with each other in the same environment [70, 15]. To produce realistic motions and interaction behaviors between multiple characters, we use two main databases (also denoted demonstrations):

— a Multiple-actors motion capture dataset $\mathbb{M}^I$ that includes interaction between multiple actors. For our application, we use a dataset of fighting motions between two fighters of two different styles: Boxing (only the upper-body attacks) and QwanKiDo (full-body movements)

— a Single-actor motion capture dataset $\mathbb{M}^S$ that includes basic skills of the same activity. It enables simulated physics-based characters to have access to a larger repertoire of realistic motions than those included in the Multiple-actors dataset.

Figure 3.2 illustrates the overview of our approach. Each dataset $\mathbb{M}^I$ and $\mathbb{M}^S$ contains motion clips $\{m_S^i \in \mathbb{M}^S\}$ and $\{m_I^i \in \mathbb{M}^I\}$. The goal of the method is to simulate interaction behaviors and motions that imitate the style contained in the Multiple-actors interaction dataset $\mathbb{M}^I$. The Single Actor dataset $\mathbb{M}^S$ is used to 1) offer a wide variety of possible motions to the simulated characters, and 2) make the physics controller be more robust in handling situations not present in the interaction demonstration.

Each motion clip can be seen as a sequence of character poses $m_S^i = \{q_t^i\}$ for the motion dataset $\mathbb{M}^S$, and as a sequence of two interacting characters poses $m_I^i = \{q_t^{i,0}, q_t^{i,1}\}$ for the interaction dataset $\mathbb{M}^I$, with two fighters denoted 0 and 1 respectively. Based on these poses m_I^i, we propose to build an observation at time t, o_t=[o_t^{self}, o_t^{opp}], for each character ($self$ for the agent, and opp for the opponent). In Section 3.2.2, we give more details about the agent's observations.

We define the controller for each character using a policy:

$$\pi(a_t|o_t^{self}, o_t^{opp}) \tag{3.1}$$

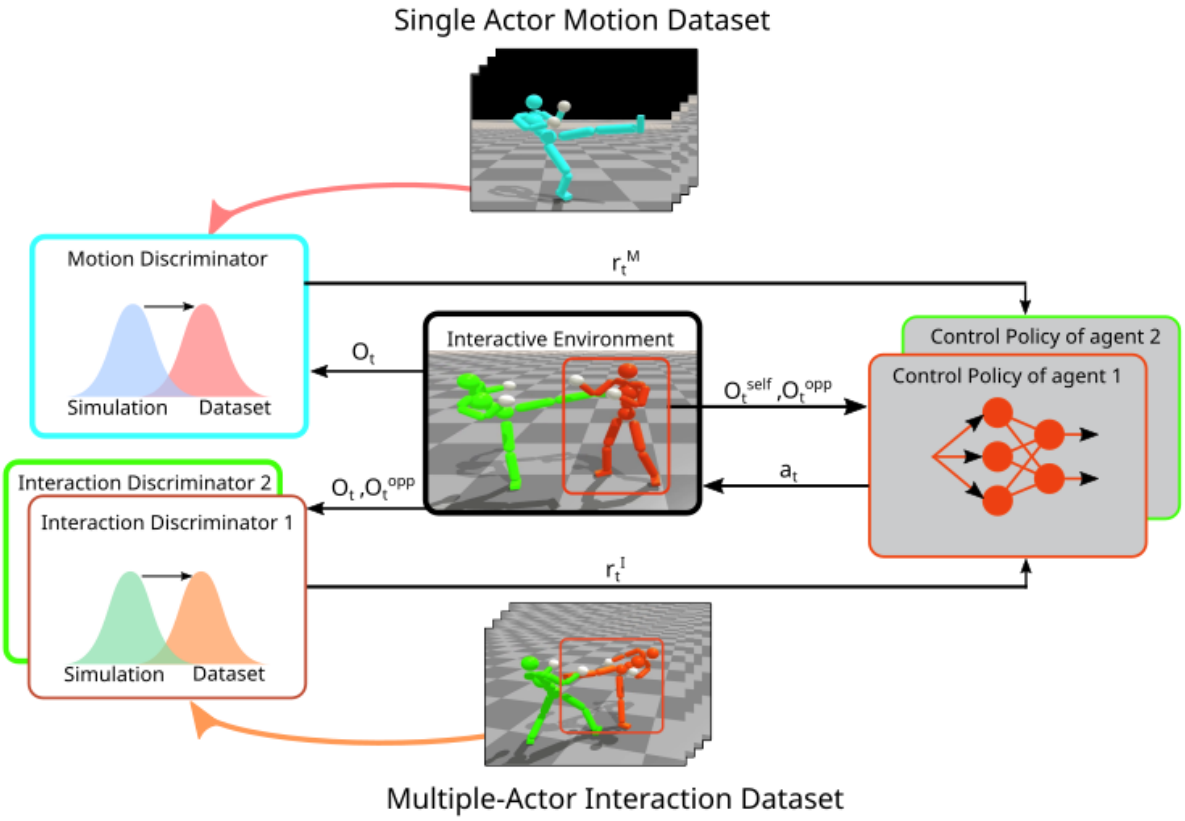


Figure 3.2 – Overview of the system. Multiple-actors motion capture clips are used to train an interaction discriminator assigned to each agent, aiming at returning an interaction reward r^I. Single-actor motion clips are used to train a motion discriminator that returns a motion reward r^M. The rewards learned by the two discriminators are combined to train each agent's policy in order to imitate the interactive behavior depicted in the datasets.

where a_t is the action that specifies the set of target joint angles (target poses) used by the PD controller [125]. Based on the physical model, the contact forces are computed during the simulation, both during the training and simulation phases. Thus, they can be used to simulate impacts, or design specific rewards minimizing self-damages or maximizing damages on the opponent. An adversarial discriminator is trained to compute a reward $r^I([o_t^{self}, o_t^{opp}], o_{t+1}^{self})$. For each character, this discriminator is trained to distinguish between interactions simulated by the simulated agent from those shown in the demonstrations (Multiple-actors motion capture dataset). Hence, it is possible to train specific discriminators for each character, with its specific style. This is a key idea here, as we expect to be able to generate individual style for each character in the final multiple-characters simulation.

The observation's transition $(o_t^{self}, o_{t+1}^{self})$ is also used to compute a motion reward $r^M(o_t^{self}, o_{t+1}^{self})$ that measures the naturalness of the simulated motion. Similarly, r^M is the output of an adversarial discriminator trained to differentiate between generated motions and demonstrations stored in the Single-actor motion capture dataset.

The two learned rewards could be combined with other rewards r_t^C, to offer control facilities, such as maximizing physical contact on a specific body part of the opponent, or driving the interaction to a given direction.

In the following sections, we introduce the framework of Multi-Agent Generative Adversarial Imitation Learning [120] in section 3.2.1. We then present our architecture adapted from it, that we dub Adversarial Interaction Priors in section 3.2.3 with two main contributions: modeling the interaction with an opponent: section 3.2.2, and new objectives for training the system: section 3.2.3. Then, we provide additional details about network architecture: section 3.2.4 and the algorithm for training the system: section 3.2.5. Finally, we showcase different experiments: section 3.3 and ablation studies: section 3.4 for validating the proposed system, then end the chapter with the limitations of the system and the future directions: section 3.5

3.2.1 Preliminary on Multi-Agent Generative Adversarial Imitation Learning

Multi-Agent Generative Adversarial Imitation Learning (MAGAIL) [120] is a variant of the Generative Adversarial Imitation Learning (GAIL) [46] that is used to deal with multi-agent interactions. In MAGAIL, multiple agents i (each with their own policy π_{θ_i}) are trained to imitate the behavior of one or many expert policies π_{E_i}, using a Generative Adversarial Network framework [35]. For each agent i, a parameterized discriminator D_{ω_i} maps state-action pairs $(s_t, a_t)_i$ to scores that are optimized to discriminate expert demonstrations generated by unknown expert policy π_{E_i} from behaviors produced by the agent's policy π_{θ_i}. D_{ω_i} plays the role of a reward function for the generator π_{θ_i}, which in turn attempts to train the agent to maximize its reward, therefore fooling the discriminator [120]. The objective to be optimized is the following:

$$\min_{\theta} \max_{\omega} \mathbb{E}_{\pi_E}[\sum_{i=1}^{N} \log D_{\omega_i}(s, a_i)] + \mathbb{E}_{\pi_\theta}[\sum_{i=1}^{N} \log(1 - D_{\omega_i}(s, a_i))] \tag{3.2}$$

where π_θ denotes the joint policy for N agents $\pi_\theta = \prod_{i=1}^{N} \pi_{\theta_i}$ and $\pi_E = \prod_{i=1}^{N} \pi_{E_i}$ denotes

the joint policy for N experts. The policies π_{θ_i} are updated through reinforcement learning by using as a reward function for each agent i:

$$r_t^i = -\log(1 - D_{\omega_i}(s_t, a_t)_i) \tag{3.3}$$

3.2.2 Modeling Self and opponent observations

The observation of each agent $o_t = [o_t^{self}, o_t^{opp}]$ consists of a set of features describing the proprioceptive configuration of its own body o_t^{self} at the current time t, as well as features describing the current observation about the opponent. The features used to model o_t^{self} are:

— Root's height from the ground $\in \mathbb{R}$

— All body parts' positions in the character's local coordinate frame $\in \mathbb{R}^{42}$

— All body parts' local rotations $\in \mathbb{R}^{90}$

— All body parts' local linear and angular velocities $\in \mathbb{R}^{45}$

We used a reduced set of features for observations about the opponent compared to the one used in [139]. Each agent's features about the opponent o_t^{opp} include:

— Opponent's root position $\in \mathbb{R}^3$, orientation $\in \mathbb{R}^6$, linear and angular velocities $\in \mathbb{R}^3$ in the current character's local coordinate frame

— Opponent's head, torso, hands and feet positions and velocities in the current character's local coordinate frame $\in \mathbb{R}^{18}$

We used the linear and angular velocities as relevant information for deciding the appropriate reaction to the opponent. In the context of physical interaction between two characters, we assume that the controller should benefit from potential anticipation skills thanks to this type of information. Indeed, in real competitive or collaborative interactions between people, this anticipation skill is important.

Similarly to previous works [100, 99], the pelvis segment is assumed to be the root of the character. The local coordinates are then expressed in this reference frame, with the x-axis oriented along the root facing direction, and the z-axis is up. The body parts' rotations are encoded using two 3D vectors corresponding to the tangent and normal of its link local coordinate frame, expressed in the link parent's coordinate frame. The observation space obtained from these features has a dimension of 274. The actions a_t correspond to target poses used by the Proportional Derivative (PD) controller to compute joint

torques for the character's joints. The target pose for spherical joints is represented by 3D exponential map $q \in \mathbb{R}^3$ [36] such that the rotation axis v is computed by $v = \frac{q}{||q||_2}$ and the rotation angle $\theta = ||q||_2$. This representation is more compact than 4D axis-angle or quaternion representations, and also avoids the gimbal lock issue in Euler angles [99]. The target rotations for revolute joints are specified as 1D rotation angles $q = \theta$. The resulting action space has 28 dimensions.

3.2.3 Adversarial Motion and Interaction Priors

In order to imitate close interaction from motion capture demonstrations, we use a learned reward function r^M that takes into account the motions generated by each simulated character i. We also use a learned interaction reward r^I that takes into account its behavior with respect to the opponent. We use a combination of these two rewards to train each agent with RL:

$$r(o_t, a_t, o_{t+1}) = w^M r^M(o_t^{self}, o_{t+1}^{self}) + w^I r^I(o_t, o_{t+1}^{self}) \tag{3.4}$$

where w^M and w^I are weights associated with the two rewards functions r^M and r^I respectively.

Following [99], the single motion prior D^M is modeled by a learned discriminator trained to predict whether an observation transition $(o_t^{self}, o_{t+1}^{self})$ is a real sample from the dataset, or a sample simulated by the agent. We model the interaction reward by learned discriminators, each one assigned to an agent. Given the interaction dataset $\mathbb{M}^I$ of multiple actors, each discriminator D^I is trained to predict if the transition (o_t, o_{t+1}^{self}), i.e. the reaction of the agent with respect to the other one, is within the distribution of the demonstrations.

Since we use demonstrations from unlabeled and unstructured motion capture clips, we do not have access to actions needed by MAGAIL, as introduced in section 3.2.1. Therefore, we train the motion discriminator D^M with the observation transitions $(o_t^{self}, o_{t+1}^{self})$, and the interaction discriminators D^I with transitions (o_t, o_{t+1}^{self}) as inputs, as suggested in previous works [130]. In this case, the reward function based on the motion discriminator is given by:

$$r_t^M = -\log(1 - D^M(o_t^{self}, o_{t+1}^{self})) \tag{3.5}$$

while the reward based on the interaction discriminators is:

$$r_t^I = -\log(1 - D^I(o_t, o_{t+1}^{self})) \tag{3.6}$$

We also use the gradient penalty regularization [99] in order to stabilize the training of the discriminators and improve the quality of generated behaviors. Therefore, with $\phi = (o_t^{self}, o_{t+1}^{self})$, the objective for training the single motion prior D^M is formulated by:

$$\min_{D^M} -\mathbb{E}_{\pi_E}[\log D^M(\phi)] - \mathbb{E}_{\pi_i}[\log(1 - D^M(\phi))] + w_{gp}\mathbb{E}_{\pi_E}\left[\left|\left|\nabla_\phi D^M(\phi)|_\phi\right|\right|^2\right] \tag{3.7}$$

where π_E denotes an unknown expert policy that generated the demonstration transitions, w_{gp} is a manually specified coefficient. On the other hand, with $\psi = (o_t, o_{t+1}^{self})$, the objective for training each interaction prior D^I is:

$$\min_{D^I} -\mathbb{E}_{\pi_E}[\log D^I(\psi)] - \mathbb{E}_{\pi_i}[\log(1 - D^I(\psi))] + w_{gp}\mathbb{E}\pi_E\left[\left|\left|\nabla_\psi D^I(\psi)|_\psi\right|\right|^2\right] \tag{3.8}$$

3.2.4 Network Architecture

We model the different components of the system (agents policies, single motion prior and interaction priors) as neural networks whose parameters are optimized using the previously defined objectives. In this section, we present the different networks' architectures. Since the agents are homogeneous (i.e. they have the same observation and action spaces), we used parameter sharing for their policies, so that all agents share the same network. Previous works have shown that this makes the learning be more efficient [153, 127, 20]. Therefore, the policies π are modeled by a neural network for which the inputs are the full observation o_t of each agent i as well as an indicator of the identity of the agent i, and outputs the mean $\mu(o_t,\text{i})$ of a Gaussian distribution over actions $\pi(a_t|o_t,\text{i}) = N(\mu(o_t,\text{i}); \Sigma)$ where the covariance matrix Σ is fixed during training. It is a fully-connected network consisting of 3 hidden layers of 1024, 1024, 512 units with ReLU activations [89], followed by a linear output layer. We also use centralized training and decentralized execution (CTDE) for training the agents [79]. Therefore, we use a centralized value function $V(s_t = (o_t^0, o_t^1))$ shared by the two agents during training that takes as input the concatenation of all agents' local observations to build a global state s_t [79]. The value function $V(s_t)$, the interaction discriminators D^I and motion discriminator D^M, are modeled as networks with similar architecture.

3.2.5 Training Algorithm

After defining the training objectives for the different components of the system modeled using neural networks, we present in this section the training algorithm used for their optimization. We use the framework of MAGAIL [120] with the multi-agent proximal policy optimization algorithm MAPPO [153, 115]: at each time step t, each agent receives a local observation $o_t = [o_t^{self}, o_t^{opp}]$ from the environment and decides an action a_t. Then, it receives an interaction reward r_t^I and a motion reward r_t^M, computed from their respective discriminators, and possibly a control reward r_t^C specified by the user, to add a level of control to the interaction of the characters. Similar to [99], we use a combination of these rewards to get the final imitation reward r_t at time t according to Equation (3.4).

To stabilize the training in tasks where additional control rewards are used, we use reward scheduling so that at the beginning of the training, agents learn first to imitate motions from the single motion datasets then we introduce later the rewards for interaction and then the control reward. We find that by using this strategy, the resulting interaction is more convincing and does not collapse to unwanted behavior because of opposing rewards. After collecting a batch of trajectories with the policies, we record them in buffers to update the policy networks, the centralized value function V, and the discriminators D^I and D^M, similarly to [99]. We also add replay buffers B_i^I for each interaction discriminator D^I associated with each agent i.

We use Generalized Advantage Estimation GAE(λ) [114] to compute advantages for updating the policies. The centralized value function is updated using TD(λ) [122]. We follow the recommendations from [153] to choose the hyper-parameters of the multi-agent PPO algorithm The training process is described in Algorithm 2.

3.3 Experiments and results

We carried out experiments on two scenarios: boxing, where the agents only used displacements and upper-body actions, and QwanKiDo, a Sino-Vietnamese martial art involving full-body actions.

We first evaluated the standard case, using the imitation reward only (section 3.4), in an application where the two characters had to imitate interactions of the demonstrations. Then, we showed that adding a task-specific reward for minimizing (resp. maximizing) the damage received by (resp. given to) each character leaded to simulate more defensive (resp. aggressive) behaviors. We also demonstrated an example of controlling the moving

Algorithm 2 Training Algorithm for Multi-Agent Interaction policies

Require: Initialized policies π, interaction discriminators D^I, motion discriminator D^M and value function V ; Single-Actor motion dataset $\mathbb{M}^S$; Multi-Actor interaction dataset $\mathbb{M}^I$
Ensure: Learned policies π and reward functions D^I and D^M
while learning is not done **do**
    $\mathbb{B}^\pi$, $\mathbb{B}^M$, $\mathbb{B}^I \leftarrow \emptyset$ initialize data buffers for each agent.
    for trajectory $k = 1, ..., m$ of length T **do**
        $\tau^k \leftarrow (o_t, a_t)_{t=0}^{T-1}$ collect trajectory rolled out with policies π for all agents
        for timestep $t = 0, ..., T-1$ **do**
            $d_t^M \leftarrow D^M(o_t^{self}, o_{t+1}^{self})$ get score from the single motion prior for all agents
            $d_t^I \leftarrow D^I(o_t, o_{t+1}^{self})$ get scores from interaction priors for all agents
            $r_t^M \leftarrow$ calculate motion reward according to formula 3.5. for all agents
            $r_t^I \leftarrow$ calculate interaction reward according to formula 3.6. for all agents
            $r_t \leftarrow$ combine r_t^M and r_t^I according to formula 3.4.
            record r_t in the trajectory τ^k for each agent.
            store transitions $(o_t^{self}, o_{t+1}^{self})$ in $\mathbb{B}^M$ for all agents.
            store transitions (o_t, o_{t+1}^{self}) in $\mathbb{B}^I$ for each agent.
        end for
        store trajectory τ^k in $\mathbb{B}^\pi$ for each agent.
    end for
    for update steps $i = 1, ..., n$ **do**
        update D^M using K transitions sampled from $\mathbb{M}^S$ and from $\mathbb{B}^M$ according to formula 3.7.
        update each D^I using K transitions sampled from $\mathbb{M}^I$ and from $\mathbb{B}^I$ according to formula 3.8.
    end for
    update π and V using samples from $\mathbb{B}^\pi$ for all agents using MAPPO.
end while

direction while keeping the interaction. Finally, we pushed the system to the limits by simulating interaction between characters that were trained on different sets of demonstrations.

3.3.1 Experimental setup

The unstructured dataset used for training agents on fighting interactions contains motions of two different fighting styles: boxing and QwanKiDo. We used a Qualisys opto-electronic motion capture system, composed of 22 Oqus 200Hz cameras, to track 46 anatomical landmarks placed according to the Qualisys animation marker set guidelines. When contact occurred, some markers may fly away, so that the corresponding samples were eliminated. The data were down sampled to 30Hz and re-targeted to the character's skeleton used in the simulation. Some examples of motion capture sessions in boxing and QwanKiDo are given in the supplementary video [150].

Isaac Gym [80] was used for the physics-based simulation engine for GPU-based accelerated training. We simulated 2048 environments in parallel on a single NVIDIA A6000 GPU, each with 2 agents. We ran the simulations at a frequency of 60Hz with 2 substeps, while the policies were queried at 30Hz. All policies were trained for 2 billion steps, which takes approximately 15 hours of training time. The algorithm's hyper-parameters are available in Table 3.1.

Boxing Scenario. The boxing scenario involves two characters who can displace and use their upper-body to attack (with jabs, crosses, hooks and uppercuts), or defend (using guard, slipping, swaying, parrying, blocking and clinching). For the Single-actor motion dataset, 4 high-level volunteer boxers (1 professional and 3 regional-level competitors) participated in a single full-body motion capture session. The resulting single-boxer dataset contained approximately 15 minutes of boxing. For the Multiple-characters dataset, we asked pairs of the above boxers to perform 30s to 90s rounds. For each trial, the opponents started far away from each other, to capture some displacement toward a real opponent. Two pairs of boxers participated in this experiment, with different personalized "specials" (considered as styles). The total duration of multiple-actors dataset was 3 minutes.

QwanKiDo scenario. The QwanKiDo scenario also involves two characters, but the repertoire of possible motions is larger, including kicks, elbow or knee strikes, and sweeping. The protocol was similar to the one used for boxing, with 2 participants, single actor and two-actors sessions. The total usable motion capture duration for the single-actor

Parameter	Value
T Episode length	1200
w_{gp} Gradient Penalty Weight	5
Samples Per Update Iteration	131072
Horizon	32
PPO Mini Batch Number	2
PPO Learning Epoch Number	6
Adam Step size	2×10^{-5}
B Discriminators Replay Buffer Size	100000
K Discriminators Batch Size	4096
γ Discount	0.99
TD(λ)	0.95
GAE(λ)	0.95
Σ_π Action Distribution Variance	0.0025
PPO Clip Threshold	0.02

Table 3.1 – Hyper-parameters used and implementation details for reproducibility. We followed the recommendations from [153] for choosing the hyper-parameters of the multi-agent PPO algorithm.

dataset was 10 minutes. This scenario raises more challenges for the imitation approach, as the quantity of available demonstrations is smaller, whereas the number of possible actions is larger. Moreover, the "specials" for each fighter are visually more different than those observed for the boxing scenario. The total duration for the multiple-actors dataset was 3 minutes.

3.3.2 Fighting simulation using the priors only

In this first application, we only used the rewards computed from the discriminators' outputs. We used the weighting values of $w^M = 0.2$ for the motion reward and $w^I = 0.8$ for the interaction reward in Equation 3.4. For the interaction, each agent was associated with the same opponent in all the demonstrations, assuming that it should enable to provide this specific opponent style of interaction to this agent.

Visual results are depicted in figures 3.3 and 3.4. In the resulting sequences, one can see that the fighters learned basic fighting skills, such as getting closer to the opponent, staying in guard stance when approaching, anticipating openings for attacks and evading incoming attacks. They also learned footwork skills for fighting as they move around the opponent and remain at a safe distance before switching to attack. The experts who participated in the motion capture sessions were able to recognize the participant who

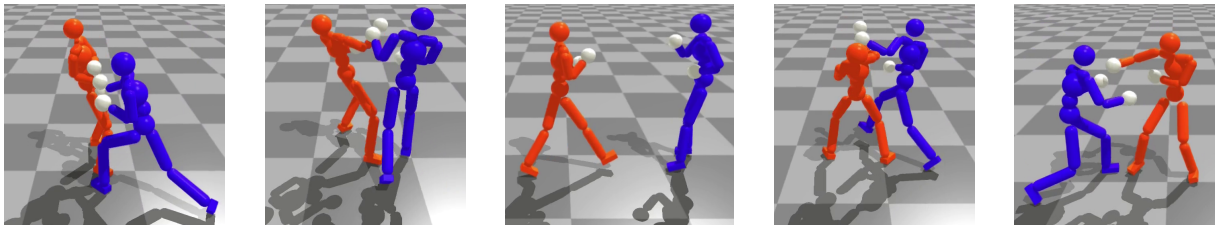


Figure 3.3 – Simulation of boxing interaction between two agents. The boxers show agility in the movements, interactive skills such as getting closer to the opponent, dodging and blocking attacks as well as finding attack openings.

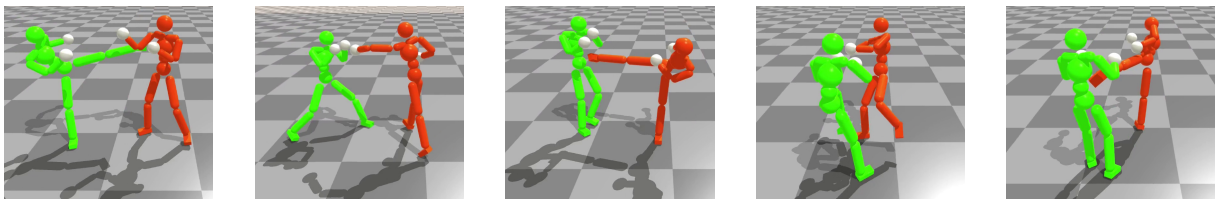


Figure 3.4 – Simulation of QwanKiDo interaction between two agents. The agents show highly-dynamic motions, such as using full body for attacks, unique fighting styles similar to the actor motions used for training them.

served as a demonstration for each avatar, in all the simulations. This is a promising result, as they enabled to recognize the participants only looking at neutral synthetic characters which policy was trained with this participant dataset. This result should of course be confirmed by a scientific perceptual study.

We ran numerous simulations, with random initialization states (global position and orientation) and obtained very convincing results, as shown in the supplementary video [150] . In very few cases, we could obtain clearly unrealistic results, which demonstrates one of the fundamental limits of imitation-based approaches: too few examples in the demonstrations may lead to simulate unrealistic behaviors. These unrealistic behaviors could be strange following behaviors, or repeating the same motion many times (due to mode col-

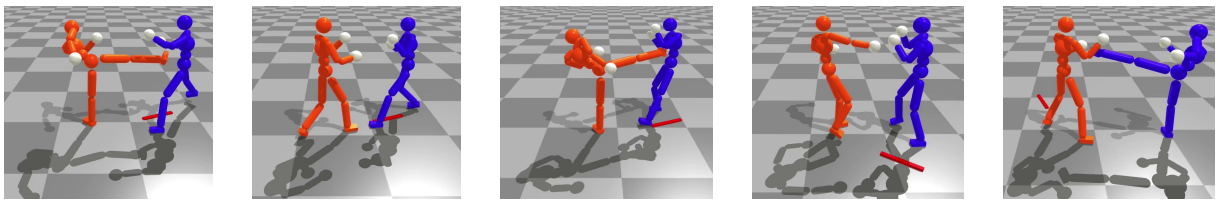


Figure 3.5 – An example of interaction simulation with heading controls. The fighters are constrained to move towards a given target direction, represented by the red line.

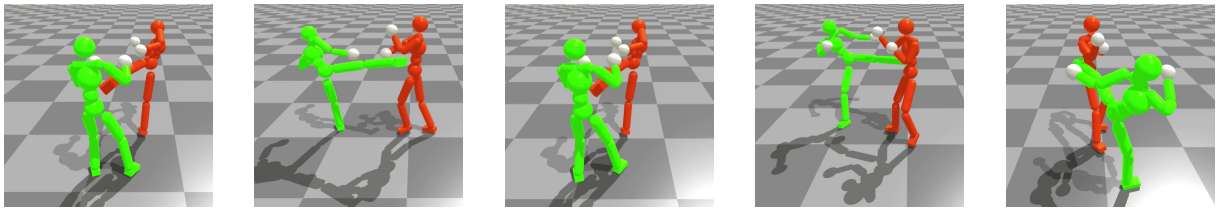


Figure 3.6 – An example of fighting simulation using additional control rewards, that encourages the agents to minimize the damage dealt by the opponent to specific body parts: head, torso and pelvis. Top: without control reward ; Bottom: with control reward. The reward drives the agents into simulating interactive behavior where they act more defensively, and they block attacks more often.

lapse of the discriminators). To partly mitigate this risk, the system could be trained with more examples, and could also use additional rewards, such as minimizing or maximizing impacts, which should provide a wider set of potential solutions.

3.3.3 Fighting simulation using additional task-dependent control rewards

To control the generated interaction and guide the selection of the motions the agents should imitate from the dataset, we tested additional task rewards r^C. Firstly, we introduced a reward that encourages the agents to minimize the damage dealt by the opponent to specific body parts. Secondly, we designed another reward that encourages maximizing damages on the opponent. These task-specific rewards are reasonable choices for both boxing and QwanKiDo.

The additional rewards could enable the system to find acceptable solutions when facing a new situation that was not captured in the single motion and interaction priors. Compared to previous works, the new behaviors are generated automatically, without the need of designing a specific motion planner for motion selection.

Let $|f_{opp \rightarrow self}|$ be the magnitude of external contact normal forces applied by an opponent (considered as "damages") to the head, torso and pelvis of a character. The damage minimization reward is then given by:

$$r^C = exp(-w \cdot |f_{opp \rightarrow self}|). \tag{3.9}$$

Similarly, the damage maximization reward is expressed by:

$$r^C = 1 - exp(-w \cdot |f_{self \to opp}|) \tag{3.10}$$

The weighting for the different rewards becomes: $w^M = 0.1$, $w^I = 0.4$ and $w^C = 0.5$. We computed the "damages" applied to each character by averaging the total damage received over 32 trials, with an episode length of 1200 frames, with and without using these task rewards. The quantitative results (see Table 3.2) showed a significant decrease of the "damages" with the damage minimization reward compared to using the imitation reward only. Reversely, we noticed an increase in the received damage when using the damage maximization reward. The top part of Figure 3.6 depicts a QwanKiDo sequence simulated

Scenario	Imitation only		Damage min.		Damage max.	
Boxing Duo 1	2210	3261	820	862	6759	6143
Boxing Duo 2	1135	2010	957	1393	9861	8146
Qwankido	4038	2216	123	215	8623	9435

Table 3.2 – Mean damage values (in Newton) for 32 randomly initialized episodes of length 1200 each, with imitation reward only, minimizing or maximizing damage additional reward. The damages are cumulative contact forces applied to the head, the torso and the pelvis, either of the controlled character (to minimize damages) or of the opponent (to maximize damages).

without the damage minimization reward, leading to a series of attacks. The bottom part of Figure 3.6 depicts the resulting sequence when adding the damage minimization reward, which shows more defensive and less engaging behavior.

3.3.4 Target Heading Task

In this task, the objective for the characters is to move along an imposed target heading direction d^*, while still fighting one against each other. We conditioned the policies of the agents on the given target direction in the local coordinate frame for each character d_t^* at time t, and we used a reward function similar to the one used in [99]:

$$r^C = exp(-w \cdot (d^* \cdot v^{root})) \tag{3.11}$$

where v^{root} is the root velocity for each character.

The weighting used for this task is $w^M = 0.1$, $w^I = 0.4$ and $w^C = 0.5$. Figure 3.5 shows the interaction of two QwanKiDo fighters moving towards a given direction. The

resulting task return of the heading control task for QwanKiDo and Boxing is reported in Table 3.3. The resulting animation is shown in the supplementary video.

This task in particular illustrates the interest of the single motion prior. The results show that characters trained with the single motion prior slightly better follow the heading direction, with slightly better task return r^C. Although the agents trained without the single motion prior might obtain a good task return, they only can imitate the motions included in the interaction dataset, which can lead to unnatural behavior. Indeed, some selected displacements may exhibit hits or avoidance to satisfy the heading constraints, while these actions are not appropriate in the current situation: avoidance without opponent attack, or punches while the opponent is too far. This type of artifacts was not observed when also using the single motion prior.

Let us notice that the training for this task is very sensitive to the weights associated with each component. Indeed, when giving more importance to the single motion prior with a high w^M weight, the simulated agents follow the given direction without interacting with each other, as some displacement without interaction are available in the single motion prior. Reversely, when giving more importance to the interaction prior w^I, the agents mainly use displacements based on hits and avoidance, as interaction-free displacements are rare in the interaction prior (see the supplementary video for some examples [150]).

Scenario	With Single MP	Without Single MP
Boxing Duo1	0.86	0.82
Boxing Duo2	0.80	0.76
QwanKiDo	0.90	0.89

Table 3.3 – Performance of the trained agents in the heading control task when using or not the single motion prior (MP). The performance is quantified by the average normalized task return r^C for 32 episodes of 500 length each.

3.3.5 Transfer to unseen fighting situations

In our approach, the main idea is to imitate an interaction given in a multiple-characters motion capture dataset. To evaluate if our method could generalize to handle novel and unseen fighting situations, we trained two agents with different interaction datasets. Let us consider for example that the agent 0 is trained with the boxing dataset, and another agent 1 with the QwanKiDo dataset. The agent 1 had seen some examples of attacks performed with the arms, such as jabs or uppercuts, although they may have been performed with a different style. However, agent 0 had never seen any kick or sweeping

attacks. Again, it is hard to quantify the ability of the system to generalize, as there are no real metrics to quantify the realism of the resulting simulation.

We found that the agents were able to keep the basic interactive skills, such as getting closer to the opponent, facing him and staying on guard, even though they were not trained against those specific opponents. However, we also noticed that they performed fewer attacks and are less engaging, as attacks are conditioned by a given observation of the opponent, and there is no such an attack signal for observations that have never been seen during training. We believe that enhancing the datasets used for training and using policy architectures that account for past observations should help to handle a larger variety of fighting situations, but it may still suffer from distributional shift [110].

3.4 Ablation Study

In this section, we study the importance of the components of our method by ablating the sensitivity to the weighting between the interaction prior and the single motion prior, as well as the impact of the losses used for training the discriminators.

3.4.1 Single Motion Prior Impact

The single motion prior in our framework aims at providing single actor motion examples to generate natural behavior and account for unseen situations in the interaction prior. We showed the importance of using it in the heading control task, introduced previously. In this task, we found that using only the interaction prior may lead to similar task returns (see Table 3.3), but the resulting motions were less natural. Indeed, the agents seem to exploit the motion included in the interaction dataset to achieve high reward at the cost of motion naturalness, especially when the given direction changes. As the single motion prior is trained with a larger variety of displacement motions compared to the interaction prior, it enables to generate more natural foot work and displacements. Therefore, it enables to create more natural transitions between interaction and displacement motions (see supplementary video [150]). However, the weighting between the interaction prior, the single motion prior and the task reward needs to be tuned. Hence, the agents achieve the desirable behavior as a high weighting for the single motion prior might lead to agents that completely ignore the interaction, and focus more on maximizing the heading task relying only on the single motion prior.

For the transfer to unseen fighting situations introduced in section 3.3.5, we found that adding the single motion prior helps to generate behaviors in fighting situations which are not present in the interaction dataset, and yields more plausible results in general, compared to when using only the interaction prior. Indeed, the agents trained with only the interaction prior struggle more to keep natural behavior in out-of-distribution states. However, we found that the generated behavior is sensitive to the weighting assigned to the single motion prior. By assigning more importance to the motion prior, the characters are less interactive and focus more on maximizing the motion reward. Consequently, they start punching/kicking far from each other (see the example of such case in the supplementary video). We believe that better strategies for varying the weights assigned to each term depending on the task could be beneficial to improve the quality of the resulting interaction rather than having constant weights.

3.4.2 Discriminators Training Loss Impact

The objective used for training the single motion prior and each interaction prior in our framework is the same one defined in the original GAIL [46], which uses a sigmoid cross-entropy loss function. This loss function is known for training instability because of saturation of the sigmoid function, leading to vanishing gradients. To counter this, the authors of AMP proposed to use the loss function for least-squares GAN (LSGAN) [81] that showed more training stability and better overall quality. The objective for optimizing the discriminator is defined as:

$$\min_{D^M} \mathbb{E}_{\pi_E}\left[\left(D^M(\phi)-1\right)^2\right] + \mathbb{E}_{\pi}\left[\left(D^M(\phi)+1\right)^2\right] \tag{3.12}$$

with $\phi = (o_t, o_{t+1})$. The policy π is then optimized using the following reward function:

$$r(\phi) = \max\left[0, u - v \cdot \left(D^M(\phi)-1\right)^2\right] \tag{3.13}$$

u and v are offset and scale to bound the reward between $[0, 1]$.

We experimented with this objective function for training both the single motion prior and the interaction priors in the imitation task. We found that the quality of the generated interactive behavior is worse compared to what we get with the standard GAIL objective, and that it is more prone to mode collapse by repeatedly generating the same subset of motion sequences. Although the agents were able to perform the basic fighting

motions included in the single motion dataset, their interactive capabilities were limited even when assigning more importance to the interaction priors in the total reward. We think that this degradation in interaction quality is due to the difficulty of solving the least-squares regression by the interaction priors when the environment is non-stationary in the setting of multiple characters' interaction. We show examples of these behaviors in the supplementary video.

3.5 Discussion and Limitations

We have introduced an innovative adversarial system designed to imitate the intricate fighting interactions between multiple physics-based characters, utilizing unstructured motion clips. Building upon the foundations of the MAGAIL framework, our approach incorporates crucial adaptations to effectively simulate multiple physics-based characters' behaviors. The first significant enhancement involves the modeling of reactive behavior, wherein we establish a transition from the current full observation, that includes the self-observation of the agent itself and the current observation about the opponent, to the subsequent self-observation. This transition captures the dynamic nature of the characters' responses to their opponent, resulting in a more plausible simulation. Additionally, we devised a training strategy that encompasses both single motion and interaction priors.

The resulting sequences do not simply imitate the reference motions with the same frame order, but exhibits similar interactive behaviors to the interaction dataset by maximizing the rewards assigned by each prior. Hence, our approach enabled us to imitate the personalized reaction of fighters with specific styles. We can also provide the users with some control of the simulation, by adding task-specific rewards: following a given direction, minimizing the received impacts or maximizing damages to the opponents when searching for the next action, while still imitating the style of the interaction dataset. We could imagine other rewards, such as aiming specific parts on the opponent's body. The results show that although the interaction dataset could be enough to learn motion and interaction imitation policies, associating a complementary single motion prior helps to generalize to a wider range of situations with realistic motions.

However, like other Generative Adversarial Networks (GAN)-based methods, our approach can suffer from mode collapse: repeating the same interaction behavior and generating only a small subset of the interactions contained in the demonstrations, especially because of the multi-modality of the interaction dataset. Recent work [53] tried to mit-

igate this problem by conditioning the motion prior on latents that encode each motion clip. Some other works [91, 39] propose to use multiple discriminators to handle the multi-modality of the training distribution. Although these methods introduce new challenges such as predefining the number of discriminators to be used, increasing the number of trained parameters or the assumption of having a labeled reference motion dataset, we believe that they can serve in reducing the effect of mode collapse and improving the quality of the generated behavior. For example, if the motion clips are segmented and labelled, we could imagine using a discriminator for attacks, another for defense, etc.

On the other hand, our method can also be used to simulate new individual styles, or new multi-characters activities (fencing, dancing, collaborative work, etc.), by retraining the same system but with new single-character and multiple-characters datasets. However, this can also be a limitation, as it requires providing enough examples to make the physics-based character correctly imitate the activity. Instead of fully retraining the policies, it should be possible to use transfer learning: pretraining the system with basic skills, such as moving around while maintaining balance, and then fine-tuning the resulting policies with a few new specific examples. This is specifically true for simulating different individual styles for the same activity, where the basic actions should be very similar.

For some activities, the effort required to capture interaction datasets of multiple actors would be an important obstacle. For applications in the movie industry, we could also imagine using animation sequences designed by animators to convey a specific style for imaginary characters.

While the motion generated by our framework is qualitatively similar to the ones depicted in the clips examples, the resulting motion of some sequences may still appear unnatural. As the method's goal is to imitate the style of the interactions given as examples, for safety reasons, it was difficult to ask the subjects to exert high impacts on the opponent, given that they were equipped with hard markers that could injure them. Hence, we asked them to perform shadow style combat with low impacts, which is actually imitated by the system. We have shown that the same framework works for (shadow) boxing and Qwankido by simply changing the input databases of examples, and in some cases the additional attack reward can lead to combat engagement that was not present in the original motions. We could expect that fighting motions with higher impacts would help to imitate real fights.

In this work, we only tested activities involving two fighters. Future investigations and tests are needed to check the capability of the system to scale to more characters

and to adapt to different types of interaction such as dancing, where the choreography, synchronization and long duration contacts of multiple dancers are important for generating plausible results. We note that with the current policies' architecture, our system can only imitate short term reactions, such as parrying a strike, or counter-attacking with one strike. It cannot handle continuous physical interactions (such as continuous contacts in dancing), middle or long-term strategies involving a sequence of actions. We would like to explore techniques that incorporate high-level long term planning in the imitation learning process so that fighters are equipped with strategic play that they can learn from demonstrations while being able to use the same strategic reasoning in new fighting situations. Learning basic fighting skills with a low level controller, then learning strategic play from demonstrations by a high level controller equipped with a long term memory component would be an interesting future direction for this work.

In addition to the aforementioned applications in sports training, the proposed approach can potentially be applied in various other domains that involve multi-agent interactions, such as virtual reality games, robotics, and animation. By simulating realistic interactions between virtual characters, our system can contribute to creating more engaging and immersive experiences for users.

To evaluate the effectiveness of the system in real-world scenarios, we could conduct experiments involving human participants who engage in virtual sports training sessions. By comparing their performance and reactions to both real opponents and simulated opponents using our approach, we can assess whether people respond naturally to virtual opponents and whether the system can provide a realistic and effective training experience.

However, one major challenge in scaling the system to more characters and different types of interactions is the collection and processing of large amounts of data. As the number of agents increases, the complexity of the interactions also grows, making it more difficult to capture and model the dynamics of the system. In this context, computer vision methods can play a crucial role in automating the data collection process and extracting relevant features from videos of real-life interactions.

Furthermore, a key issue in the current approach is the assumption that opponents' behaviors can be accurately modeled and predicted based on a finite set of demonstrations. However, real-world opponents can exhibit unseen behaviors and strategies that the system has not encountered before. To address this challenge, we could explore techniques that enable the system to learn and adapt to new opponents by continuously updating its models and policies based on the observed interactions.

Lastly, we acknowledge the importance of the initial work introduced in the previous chapter on fighter motion extraction from RGB videos. This work provides a foundation for future research on imitation learning in multi-agent interactions from videos, as it showcases a validated protocol for the motion extraction and also highlights the suitable category of human pose estimators for this task in order to alleviate the reliance on motion capture data. By combining high-quality interaction data with advanced imitation learning techniques, we can develop more sophisticated models and policies that can capture the nuances and complexities of human interactions in various domains.

Conclusion

The goal of this research was to explore techniques for simulating realistic interactive behaviors in the context of competitive sports and martial arts, such as boxing, with the aim of enhancing training experiences in virtual reality training environments. By modeling the complex interactions between fighters using data-driven approaches and physics-based character simulation, we aimed to create a virtual opponent capable of replicating the reactive behavior of real opponents. This opponent should provide realistic motion and behavior, but also responsiveness.

Simulation of interactive behaviors through physics-based simulation, and imitation learning techniques could help to design such simulated opponents for virtual reality sport training. In this thesis, we have taken steps along this direction by analysing methods for extracting such interactive behaviours from sport videos. We have also proposed a learning-based approach for imitating the intricate martial arts interactions between multiple physics-based characters, utilizing unstructured motion clips as demonstrations. Our first contribution (see Chapter 2) proposed a benchmark protocol for quantifying and evaluating the performance of several state of the art human pose estimation (HPE) methods for sport videos, involving fast and various boxing motions. This work targeted the evaluation of HPE methods as they usually constitute the first step in the pipeline of human motion extraction from video sources. We also highlighted the suitable category of human pose estimators for competitive sport activities such as boxing. Our second contribution (see Chapter 3) proposed an adversarial imitation learning system designed to imitate the intricate fighting interactions between multiple physics-based characters, by utilizing unstructured interaction clips. This approach led to learning interactive fighting policies controlling physically simulated agents that capture the dynamic nature of agent's responses to their opponent. As a result the physically-based simulation is more plausible. We applied this approach to two different kinds of fighting sports: boxing and Qwan Ki Do. We showed how this approach can go beyond interaction imitation, by adding constraints to the interaction and even the potential of transferring the learned interactive policies to handle unseen interactions.

The same techniques can be used for adding more realism and immersion in other

domains such as video games and movie making. However, as pointed out in previous chapters, there is still a number of challenges that need to be addressed in order to reach a truly immersive and beneficial experience in the interactive behavior of a virtual opponent. When interacting with an opponent in a real fighting situation, a fighter decides their next move by analyzing and taking into account the history of their past moves as well as those of their opponent. This history of past interactions can last from seconds to minutes. We only handled immediate or short term reactive behavior in this work. Fighters can also make decisions based on their own predictions and their opponent's potential next move, which is crucial to generate smart behavior. In addition, our approach builds on imitation learning techniques for motor control imitation from motion capture data, which is difficult to obtain in the context of intricate fighting interactions. All these challenges highlight a number of exiting directions that could be explored as perspectives for the presented work.

Long-Term Interaction Imitation The work in this thesis mainly focused on the reactive behavior of fighters in the short term. This implies that the learned policy of the simulated fighter essentially reacts to the current action of the opponent, without taking into account the history of their interaction. To this end, architectural changes need to be applied to the fighter's decision modules. These changes should help the virtual fighter to summarize previous states by leveraging sequential modeling, such as ones used with Transformers. This direction introduces new challenges encountered in other machine learning domains, such as training and inference efficiency, as well as computational complexity. More importantly, the training paradigm of the adversarial imitation learning will have to consider the imitation of long horizon trajectories. This could be achieved by considering a hierarchical approach that combines short term reactive behavior imitation on a low level ones. Then, imitation of long trajectories of abstractions, or compressed representations of these reactive behaviors on a semantically high level, will be necessary to achieve this goal. This would enable virtual opponents to exhibit more intelligence, adaptation skills, and more fighting sports expert-like behaviors. Let us note that a new thesis project has been initiated to explore this direction in the context of generic interaction between virtual agents.

Multi-Modal Interaction Imitation In the second part of this work, we utilized motion capture data of fighters interactions to benefit from their high accuracy, when building the interaction demonstration dataset. Evaluation performed in the first part concluded that the quality of interaction data that would have been obtained from video

demonstration through current methods of human motion estimation may have insufficient quality. However, even motion capture data comes with their own challenges, as the morphology of the performer is usually different from the simulated agents. Therefore, manual re-targeting techniques must be used to overcome this limitation. Another challenge stems from the nature of fighting interactions that makes it challenging to have fighters perform intense and rapid fight engagements while equipped with motion sensors/markers for safety reasons. While several techniques have been developed recently for physically based motion imitation from video sources and marker-less setups, they mainly focus on single actor imitation. It would be of interest to extend this work for multiple people interaction imitation, and also to develop systems that could benefit from multiple sources of demonstrations, while accounting for the mismatch between demonstrators and learners.

Latent Interaction Priors The proposed interaction prior presented in this work, operates on the same low level as the motion prior, i.e. short time joint angle prediction. Therefore, while being able to train fighting policies to generate plausible reactions to the opponents moves on the short term, it requires training from scratch for each new interaction demonstration, and can only handle immediate reactions. A more efficient approach would be to train agents to simulate a wide range of motions, while encoding them in a latent representation for re-usability by high level policies in downstream tasks. This direction has been explored recently for single character control tasks. It would be interesting to imitate interactions between agents on such latent representations, instead of the low level representation of states and actions. For example, instead of having a fighter that replicates the exact trajectory of a reactive motion of a blocking move, the agent would simply attempt to generate any motion that has similar latent representation of a blocking move. This would lead to more efficient training (compared to imitating interaction from scratch), and higher Variability in the reaction generation.

We hope that the research and techniques explored in this thesis can serve as valuable foundations for creating physically-based simulated virtual opponents in martial arts training applications. These applications should exhibit human-like skill and reactivity, on a motion level, but also decision making level. As with the evolution of biological organisms, the development of artificial agents capable of mastering intricate martial arts techniques requires advanced physical capabilities and intelligent decision making. By studying and reverse engineering these movements through imitation from expert demonstrations, we aim to not only enhance the realism and challenge of virtual com-

bat simulations, but also contribute to our overall understanding of intelligence and its relationship to complex motor skills, and their use in interactive situations. The knowledge gained from constructing such systems could potentially unlock new insights into the neural mechanisms underlying expert performance.

BIBLIOGRAPHY

[1] Shailen Agrawal, Shuo Shen, and Michiel Van de Panne, « Diverse motion variations for physics-based character animation », *in*: *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2013, pp. 37–44.

[2] Thiemo Alldieck et al., « Optical flow-based 3d human motion estimation from monocular video », *in*: *Pattern Recognition: 39th German Conference, GCPR 2017, Basel, Switzerland, September 12–15, 2017, Proceedings 39*, Springer, 2017, pp. 347–360.

[3] *Alphapose Github Repository*, `https://github.com/MVIG-SJTU/AlphaPose`.

[4] Mykhaylo Andriluka et al., « 2D Human Pose Estimation: New Benchmark and State of the Art Analysis », *in*: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[5] Mykhaylo Andriluka et al., « 2d human pose estimation: New benchmark and state of the art analysis », *in*: *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, 2014, pp. 3686–3693.

[6] Mykhaylo Andriluka et al., « Posetrack: A benchmark for human pose estimation and tracking », *in*: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5167–5176.

[7] Brenna D Argall et al., « A survey of robot learning from demonstration », *in*: *Robotics and autonomous systems* 57.*5* (2009), pp. 469–483.

[8] Aritz Badiola-Bengoa and Amaia Mendez-Zorrilla, « A Systematic Review of the Application of Camera-Based Human Pose Estimation in the Field of Sport and Physical Exercise », *in*: *Sensors* 21.*18* (2021), ISSN: 1424-8220, DOI: `10.3390/s21185996`, URL: `https://www.mdpi.com/1424-8220/21/18/5996`.

[9] Michael Bain and Claude Sammut, « A Framework for Behavioural Cloning. », *in*: *Machine Intelligence 15*, 1995, pp. 103–129.

[10] Darrin C Bentivegna et al., « Humanoid robot learning and game playing using PC-based vision », *in*: *IEEE/RSJ international conference on intelligent robots and systems*, vol. 3, IEEE, 2002, pp. 2449–2454.

[11] Kevin Bergamin et al., « DReCon: data-driven responsive control of physics-based characters », *in*: *ACM Transactions On Graphics (TOG)* 38.*6* (2019), pp. 1–11.

[12] Justin Boyan and Andrew Moore, « Generalization in reinforcement learning: Safely approximating the value function », *in*: *Advances in neural information processing systems* 7 (1994).

[13] Lewis Bridgeman et al., « Multi-person 3d pose estimation and tracking in sports », *in*: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2019, pp. 0–0.

[14] Greg Brockman et al., « Openai gym », *in*: *arXiv preprint arXiv:1606.01540* (2016).

[15] Lucian Busoniu, Robert Babuska, and Bart De Schutter, « A comprehensive survey of multiagent reinforcement learning », *in*: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38.*2* (2008), pp. 156–172.

[16] Z. Cao et al., « OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields », *in*: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

[17] Yilun Chen et al., « Cascaded pyramid network for multi-person pose estimation », *in*: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7103–7112.

[18] Yucheng Chen, Yingli Tian, and Mingyi He, « Monocular human pose estimation: A survey of deep learning-based methods », *in*: *Computer Vision and Image Understanding* 192 (2020), p. 102897, ISSN: 1077-3142, DOI: https://doi.org/10.1016/j.cviu.2019.102897, URL: https://www.sciencedirect.com/science/article/pii/S1077314219301778.

[19] Evgeniy A Cherepov et al., « Maintaining postural balance in martial arts athletes depending on coordination abilities », *in*: *Journal of Physical Education and Sport* 21.*6* (2021), pp. 3427–3432.

[20] Filippos Christianos et al., « Scaling multi-agent reinforcement learning with selective parameter sharing », *in*: *International Conference on Machine Learning*, PMLR, 2021, pp. 1989–1998.

[21] Leonardo Citraro et al., « Real-time camera pose estimation for sports fields », *in*: *Machine Vision and Applications* 31 (2020), pp. 1–13.

[22] Stelian Coros, Philippe Beaudoin, and Michiel Van de Panne, « Generalized biped walking control », *in*: *ACM Transactions On Graphics (TOG)* 29.*4* (2010), pp. 1–9.

[23] Stelian Coros, Philippe Beaudoin, and Michiel Van de Panne, « Robust task-based control policies for physics-based characters », *in*: *ACM SIGGRAPH Asia 2009 papers*, 2009, pp. 1–9.

[24] *CPN Github Repository*, `https://github.com/GengDavid/pytorch-cpn`.

[25] Marco Da Silva, Yeuhi Abe, and Jovan Popović, « Simulation of human motion data using short-horizon model-predictive control », *in*: *Computer Graphics Forum*, vol. 27, 2, Wiley Online Library, 2008, pp. 371–380.

[26] Shreyansh Daftry, J Andrew Bagnell, and Martial Hebert, « Learning transferable policies for monocular reactive mav control », *in*: *2016 International Symposium on Experimental Robotics*, Springer, 2017, pp. 3–11.

[27] Qi Dang et al., « Deep learning based 2D human pose estimation: A survey », *in*: *Tsinghua Science and Technology* 24.*6* (2019), pp. 663–676, DOI: `10.26599/TST.2018.9010100`.

[28] *DCPose Github Repository*, `https://github.com/Pose-Group/DCPose`.

[29] *Detectron2-Pipeline*, `https://github.com/jagin/detectron2-pipeline`.

[30] Hao-Shu Fang et al., « Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time », *in*: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[31] Hao-Shu Fang et al., « RMPE: Regional Multi-person Pose Estimation », *in*: *ICCV*, 2017.

[32] Charles Faure et al., « Virtual reality to assess and train team ball sports performance: A scoping review », *in*: *Journal of sports Sciences* 38.*2* (2020), pp. 192–205.

[33] Roy Featherstone, *Rigid body dynamics algorithms*, Springer, 2014.

[34] Wenjuan Gong et al., « Human Pose Estimation from Monocular Images: A Comprehensive Survey », *in*: *Sensors* 16.*12* (2016), ISSN: 1424-8220, DOI: 10.3390/s16121966, URL: https://www.mdpi.com/1424-8220/16/12/1966.

[35] Ian Goodfellow et al., « Generative adversarial networks », *in*: *Communications of the ACM* 63.*11* (2020), pp. 139–144.

[36] F Sebastian Grassia, « Practical parameterization of rotations using the exponential map », *in*: *Journal of graphics tools* 3.*3* (1998), pp. 29–48.

[37] Perttu Hämäläinen, Joose Rajamäki, and C Karen Liu, « Online control of simulated humanoids using particle belief propagation », *in*: *ACM Transactions on Graphics (TOG)* 34.*4* (2015), pp. 1–13.

[38] Perttu Hämäläinen et al., « Martial arts in artificial reality », *in*: *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2005, pp. 781–790.

[39] Corentin Hardy, Erwan Le Merrer, and Bruno Sericola, « Md-gan: Multi-discriminator generative adversarial networks for distributed datasets », *in*: *2019 IEEE international parallel and distributed processing symposium (IPDPS)*, IEEE, 2019, pp. 866–877.

[40] Mohamed Hassan et al., « Synthesizing Physical Character-Scene Interactions », *in*: *arXiv preprint arXiv:2302.00883* (2023).

[41] Brandon Haworth et al., « Deep integration of physical humanoid control and crowd navigation », *in*: *Motion, Interaction and Games*, 2020, pp. 1–10.

[42] Masaki Hayashi et al., « Head and upper body pose estimation in team sport videos », *in*: *2013 2nd IAPR Asian Conference on Pattern Recognition*, IEEE, 2013, pp. 754–759.

[43] Kaiming He et al., « Mask r-cnn », *in*: *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[44] Nicolas Heess et al., « Learning and transfer of modulated locomotor controllers », *in*: *arXiv preprint arXiv:1610.05182* (2016).

[45] Edmond S. L. Ho, Taku Komura, and Chiew-Lan Tai, « Spatial Relationship Preserving Character Motion Adaptation », *in*: *ACM SIGGRAPH 2010 Papers*, SIGGRAPH '10, Los Angeles, California: Association for Computing Machinery, 2010, ISBN: 9781450302104, DOI: 10.1145/1833349.1778770, URL: https://doi.org/10.1145/1833349.1778770.

[46] Jonathan Ho and Stefano Ermon, « Generative adversarial imitation learning », *in*: *Advances in neural information processing systems* 29 (2016).

[47] Jessica K Hodgins et al., « Animating human athletics », *in*: *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 1995, pp. 71–78.

[48] Markus Höll et al., « Efficient physics-based implementation for realistic hand-object interaction in virtual reality », *in*: *2018 IEEE conference on virtual reality and 3D user interfaces (VR)*, IEEE, 2018, pp. 175–182.

[49] Michael B Holte et al., « Human pose estimation and activity recognition from multi-view videos: Comparative explorations of recent developments », *in*: *IEEE Journal of selected topics in signal processing* 6.5 (2012), pp. 538–552.

[50] Auke Jan Ijspeert, Jun Nakanishi, and Stefan Schaal, « Movement imitation with nonlinear dynamical systems in humanoid robots », *in*: *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 2, IEEE, 2002, pp. 1398–1403.

[51] Sam Johnson and Mark Everingham, « Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation. », *in*: *bmvc*, vol. 2, 4, Citeseer, 2010, p. 5.

[52] Sam Johnson and Mark Everingham, « Learning effective human pose estimation from inaccurate annotation », *in*: *CVPR 2011*, IEEE, 2011, pp. 1465–1472.

[53] Jordan Juravsky et al., « PADL: Language-Directed Physics-Based Character Control », *in*: *SIGGRAPH Asia 2022 Conference Papers (SA '22 Conference Papers),* 2022.

[54] Vahid Kazemi et al., « Multi-view body part recognition with random forests », *in*: *2013 24th British Machine Vision Conference, BMVC 2013; Bristol; United Kingdom; 9 September 2013 through 13 September 2013*, British Machine Vision Association, 2013.

[55] Jun-Sik Kim and Jung-Min Park, « Physics-based hand interaction with virtual objects », *in*: *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, pp. 3814–3819.

[56] Takumi Kitamura et al., « Refining OpenPose With a New Sports Dataset for Robust 2D Pose Estimation », *in*: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, Jan. 2022, pp. 672–681.

[57] Taku Komura et al., « e-Learning martial arts », *in*: *Advances in Web Based Learning–ICWL 2006: 5th International Conference, Penang, Malaysia, July 19-21, 2006. Revised Papers 5*, Springer, 2006, pp. 239–248.

[58] Nenad Koropanovski and Srecko Jovanovic, « Model characteristics of combat at elite male karate competitors », *in*: *Serbian Journal of Sports Sciences* 1.*3* (2007), pp. 97–115.

[59] David M Kreps, « Nash equilibrium », *in*: *Game Theory*, Springer, 1989, pp. 167–177.

[60] Kaustubh Milind Kulkarni and Sucheth Shenoy, « Table tennis stroke recognition using two-dimensional human pose estimation », *in*: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 4576–4584.

[61] Richard Kulpa, « VR for training perceptual-motor skills of boxers and relay runners for Paris 2024 Olympic games », *in*: *ECSS 2023-28th Annual Congress of the European College of Sport Science*, 2023.

[62] Taesoo Kwon and Jessica K Hodgins, « Momentum-mapped inverted pendulum models for controlling dynamic human motions », *in*: *ACM Transactions on Graphics (TOG)* 36.*1* (2017), pp. 1–14.

[63] Yoonsang Lee, Sungeun Kim, and Jehee Lee, « Data-driven biped control », *in*: *ACM SIGGRAPH 2010 papers*, 2010, pp. 1–8.

[64] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua, « Epnp: An accurate o (n) solution to the pnp problem », *in*: *International journal of computer vision* 81.*2* (2009), p. 155.

[65] Cheng Li, Levi Fussel, and Taku Komura, « Multi-agent reinforcement learning for character control », *in*: *The Visual Computer* 37 (2021), pp. 3115–3123.

[66] Sijin Li, Weichen Zhang, and Antoni B Chan, « Maximum-margin structured learning with deep networks for 3d human pose estimation », *in*: *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2848–2856.

[67] Yanjie Li et al., « Is 2D Heatmap Representation Even Necessary for Human Pose Estimation? », *in*: *arXiv preprint arXiv:2107.03332* (2021).

[68] Yanjie Li et al., « Simcc: A simple coordinate classification perspective for human pose estimation », *in*: *European Conference on Computer Vision*, Springer, 2022, pp. 89–106.

[69] Tsung-Yi Lin et al., « Microsoft coco: Common objects in context », *in*: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, Springer, 2014, pp. 740–755.

[70] Michael L Littman, « Markov games as a framework for multi-agent reinforcement learning », *in*: *Machine learning proceedings 1994*, Elsevier, 1994, pp. 157–163.

[71] Ce Liu and William T Freeman, « A high-quality video denoising algorithm based on reliable motion estimation », *in*: *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part III 11*, Springer, 2010, pp. 706–719.

[72] Karen Liu, Aaron Hertzmann, and Zoran Popovic, « Composition of complex optimal multi-character motions », *in*: Jan. 2006, pp. 215–222, DOI: 10.1145/1218064.1218093.

[73] Libin Liu and Jessica Hodgins, « Learning to schedule control fragments for physics-based characters using deep q-learning », *in*: *ACM Transactions on Graphics (TOG)* 36.*3* (2017), pp. 1–14.

[74] Libin Liu, Michiel Van De Panne, and KangKang Yin, « Guided learning of control graphs for physics-based characters », *in*: *ACM Transactions on Graphics (TOG)* 35.*3* (2016), pp. 1–14.

[75] Libin Liu et al., « Sampling-based contact-rich motion control », *in*: *ACM SIGGRAPH 2010 papers*, 2010, pp. 1–10.

[76] Siqi Liu et al., « From motor control to team play in simulated humanoid football », *in*: *Science Robotics* 7.*69* (2022), eabo0235.

[77] Zhao Liu et al., « A survey of human pose estimation: The body parts parsing based methods », *in*: *Journal of Visual Communication and Image Representation* 32 (2015), pp. 10–19, ISSN: 1047-3203, DOI: https://doi.org/10.1016/j.jvcir.2015.06.013, URL: https://www.sciencedirect.com/science/article/pii/S1047320315001121.

[78] Zhenguang Liu et al., « Deep Dual Consecutive Network for Human Pose Estimation », *in*: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 525–534.

[79] Ryan Lowe et al., « Multi-agent actor-critic for mixed cooperative-competitive environments », *in*: *Advances in neural information processing systems* 30 (2017).

[80] Viktor Makoviychuk et al., *Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning*, 2021.

[81] Xudong Mao et al., « Least squares generative adversarial networks », *in*: *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2794–2802.

[82] Nicolas Mascret et al., « Acceptance by athletes of a virtual reality head-mounted display intended to enhance sport performance », *in*: *Psychology of Sport and Exercise* 61 (2022), p. 102201.

[83] Josh Merel et al., « Learning human behaviors from motion capture by adversarial imitation », *in*: *arXiv preprint arXiv:1707.02201* (2017).

[84] Igor Mordatch, Martin De Lasa, and Aaron Hertzmann, « Robust physics-based locomotion using low-dimensional planning », *in*: *ACM SIGGRAPH 2010 papers*, 2010, pp. 1–8.

[85] Marion Morel et al., « Advantages and limitations of virtual reality for balance assessment and rehabilitation », *in*: *Neurophysiologie Clinique/Clinical Neurophysiology* 45.*4-5* (2015), pp. 315–326.

[86] Uldarico Muico, Jovan Popović, and Zoran Popović, « Composite control of physically simulated characters », *in*: *ACM Transactions on Graphics (TOG)* 30.*3* (2011), pp. 1–11.

[87] Uldarico Muico et al., « Contact-aware nonlinear control of dynamic characters », *in*: *ACM SIGGRAPH 2009 papers*, 2009, pp. 1–9.

[88] Amir Nadeem, Ahmad Jalal, and Kibum Kim, « Automatic human posture estimation for sport activity recognition with robust body parts detection and entropy markov model », *in*: *Multimedia Tools and Applications* 80 (2021), pp. 21465–21498.

[89] Vinod Nair and Geoffrey E Hinton, « Rectified linear units improve restricted boltzmann machines », *in*: *Icml*, 2010.

[90] Andrew Y Ng, Stuart Russell, et al., « Algorithms for inverse reinforcement learning. », *in*: *Icml*, vol. 1, 2000, p. 2.

[91] Tu Nguyen et al., « Dual discriminator generative adversarial nets », *in*: *Advances in neural information processing systems* 30 (2017).

[92] Xuecheng Nie et al., « Single-stage multi-person pose machines », *in*: *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6951–6960.

[93] Štěpán Obdržálek et al., « Accuracy and robustness of Kinect pose estimation in the context of coaching of elderly population », *in*: *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE, 2012, pp. 1188–1193.

[94] *OpenPose Github Repository*, `https://github.com/CMU-Perceptual-Computing-Lab/openpose`.

[95] Takayuki Osa et al., « An algorithmic perspective on imitation learning », *in*: *Foundations and Trends® in Robotics* 7.*1-2* (2018), pp. 1–179.

[96] Xue Bin Peng, « Acquiring Motor Skills Through Motion Imitation and Reinforcement Learning », PhD thesis, EECS Department, University of California, Berkeley, Dec. 2021, URL: `http://www2.eecs.berkeley.edu/Pubs/TechRpts/2021/EECS-2021-267.html`.

[97] Xue Bin Peng, Glen Berseth, and Michiel Van de Panne, « Terrain-adaptive locomotion skills using deep reinforcement learning », *in*: *ACM Transactions on Graphics (TOG)* 35.*4* (2016), pp. 1–12.

[98] Xue Bin Peng and Michiel Van De Panne, « Learning locomotion skills using deeprl: Does the choice of action space matter? », *in*: *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2017, pp. 1–13.

[99] Xue Bin Peng et al., « Amp: Adversarial motion priors for stylized physics-based character control », *in*: *ACM Transactions on Graphics (TOG)* 40.*4* (2021), pp. 1–20.

[100] Xue Bin Peng et al., « ASE: Large-Scale Reusable Adversarial Skill Embeddings for Physically Simulated Characters », *in*: *arXiv preprint arXiv:2205.01906* (2022).

[101] Xue Bin Peng et al., « Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning », *in*: *ACM Transactions on Graphics (TOG)* 36.*4* (2017), pp. 1–13.

[102] Xue Bin Peng et al., « Deepmimic: Example-guided deep reinforcement learning of physics-based character skills », *in*: *ACM Transactions On Graphics (TOG)* 37.*4* (2018), pp. 1–14.

[103] Xue Bin Peng et al., « Sfv: Reinforcement learning of physical skills from videos », *in*: *ACM Transactions On Graphics (TOG)* 37.*6* (2018), pp. 1–14.

[104] Katharina Petri et al., « Possibilities to use a virtual opponent for enhancements of reactions and perception of young karate athletes », *in*: *Journal homepage: http://iacss. org/index. php? id* 18.*2* (2019).

[105] *Pinhole Camera Projection Forumla*, `https://qiita.com/akira108/items/a743138fca532ee193fe/`.

[106] Pierre Plantard et al., « Pose estimation with a kinect for ergonomic studies: Evaluation of the accuracy using a virtual mannequin », *in*: *Sensors* 15.*1* (2015), pp. 1785–1803.

[107] Nicolas Pronost et al., « Interactive animation of virtual characters: Application to virtual kung-fu fighting », *in*: *2008 International Conference on Cyberworlds*, IEEE, 2008, pp. 276–283.

[108] *Qualisys Track Manager*, `https://www.qualisys.com/software/qualisys-track-manager/`.

[109] D Gordon E Robertson et al., *Research methods in biomechanics*, Human kinetics, 2013.

[110] Stéphane Ross and Drew Bagnell, « Efficient reductions for imitation learning », *in*: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, 2010, pp. 661–668.

[111] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell, « A reduction of imitation learning and structured prediction to no-regret online learning », *in*: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[112] Stuart Russell, « Learning agents for uncertain environments », *in*: *Proceedings of the eleventh annual conference on Computational learning theory*, 1998, pp. 101–103.

[113] Stefan Schaal, « Learning from demonstration », *in*: *Advances in neural information processing systems* 9 (1996).

[114] John Schulman et al., « High-dimensional continuous control using generalized advantage estimation », *in*: *arXiv preprint arXiv:1506.02438* (2015).

[115] John Schulman et al., « Proximal policy optimization algorithms », *in*: *arXiv preprint arXiv:1707.06347* (2017).

[116] Hubert P. H. Shum et al., « Interaction Patches for Multi-Character Animation », *in*: *ACM Trans. Graph.* 27.*5* (Dec. 2008), ISSN: 0730-0301, DOI: 10.1145/1409060.1409067, URL: https://doi.org/10.1145/1409060.1409067.

[117] Hubert P.H. Shum, Taku Komura, and Shuntaro Yamazaki, « Simulating Multiple Character Interactions with Collaborative and Adversarial Goals », *in*: *IEEE Transactions on Visualization and Computer Graphics* 18.*5* (2012), pp. 741–752, DOI: 10.1109/TVCG.2010.257.

[118] *SimDR Github Repository*, https://github.com/leeyegy/SimDR.

[119] Kwang Won Sok, Manmyung Kim, and Jehee Lee, « Simulating biped behaviors from human motion data », *in*: *ACM SIGGRAPH 2007 papers*, 2007, 107–es.

[120] Jiaming Song et al., « Multi-Agent Generative Adversarial Imitation Learning », *in*: *Advances in Neural Information Processing Systems*, vol. 31, 2018, URL: https://proceedings.neurips.cc/paper/2018/file/240c945bb72980130446fc2b40fbb8e0-Paper.pdf.

[121] Ke Sun et al., « Deep high-resolution representation learning for human pose estimation », *in*: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5693–5703.

[122] Richard S Sutton, « Learning to predict by the methods of temporal differences », *in*: *Machine learning* 3.*1* (1988), pp. 9–44.

[123] Richard S Sutton and Andrew G Barto, *Reinforcement learning: An introduction*, MIT press, 2018.

[124] Richard S Sutton et al., « Policy gradient methods for reinforcement learning with function approximation », *in*: *Advances in neural information processing systems* 12 (1999).

[125] Jie Tan, Karen Liu, and Greg Turk, « Stable proportional-derivative controllers », *in*: *IEEE Computer Graphics and Applications* 31.*4* (2011), pp. 34–44.

[126] Yuval Tassa et al., « Deepmind control suite », *in*: *arXiv preprint arXiv:1801.00690* (2018).

[127] Justin K Terry et al., « Revisiting parameter sharing in multi-agent deep reinforcement learning », *in*: *arXiv preprint arXiv:2005.13625* (2020).

[128] Yating Tian et al., « Recovering 3d human mesh from monocular images: A survey », *in*: *IEEE transactions on pattern analysis and machine intelligence* (2023).

[129] Jonathan Tompson et al., « Efficient object localization using convolutional networks », *in*: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 648–656.

[130] Faraz Torabi, Garrett Warnell, and Peter Stone, « Generative adversarial imitation from observation », *in*: *arXiv preprint arXiv:1807.06158* (2018).

[131] Faraz Torabi, Garrett Warnell, and Peter Stone, « Recent advances in imitation learning from observation », *in*: *arXiv preprint arXiv:1905.13566* (2019).

[132] Alexander Toshev and Christian Szegedy, « Deeppose: Human pose estimation via deep neural networks », *in*: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1653–1660.

[133] Joris Vaillant, Karim Bouyarmane, and Abderrahmane Kheddar, « Multi-Character Physical and Behavioral Interactions Controller », *in*: *IEEE Transactions on Visualization and Computer Graphics* 23.*6* (2017), pp. 1650–1662, DOI: 10.1109/TVCG.2016.2542067.

[134] Jianbo Wang et al., « Ai coach: Deep human pose estimation and analysis for personalized athletic training assistance », *in*: *Proceedings of the 27th ACM international conference on multimedia*, 2019, pp. 374–382.

[135] Tingwu Wang et al., « Unicon: Universal neural controller for physics-based character motion », *in*: *arXiv preprint arXiv:2011.15119* (2020).

[136] Ziyu Wang et al., « Robust imitation of diverse behaviors », *in*: *Advances in Neural Information Processing Systems* 30 (2017).

[137] Alexander Winkler, Jungdam Won, and Yuting Ye, « QuestSim: Human motion tracking from sparse sensors with simulated avatars », *in*: *SIGGRAPH Asia 2022 Conference Papers*, 2022, pp. 1–8.

[138] Jungdam Won, Deepak Gopinath, and Jessica Hodgins, « A scalable approach to control diverse behaviors for physically simulated characters », *in*: *ACM Transactions on Graphics (TOG)* 39.*4* (2020), pp. 33–1.

[139] Jungdam Won, Deepak Gopinath, and Jessica Hodgins, « Control strategies for physically simulated characters performing two-player competitive sports », *in*: *ACM Transactions on Graphics (TOG)* 40.*4* (2021), pp. 1–11.

[140] Jungdam Won, Deepak Gopinath, and Jessica Hodgins, « Physics-based character controllers using conditional VAEs », *in*: *ACM Transactions on Graphics (TOG)* 41.*4* (2022), pp. 1–12.

[141] Jungdam Won and Jehee Lee, « Learning body shape variation in physics-based characters », *in*: *ACM Transactions on Graphics (TOG)* 38.*6* (2019), pp. 1–12.

[142] Yuxin Wu et al., *Detectron2*, `https://github.com/facebookresearch/detectron2`, 2019.

[143] Kevin Xie et al., « Physics-based human motion estimation and synthesis from videos », *in*: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11532–11541.

[144] Zhaoming Xie et al., « Allsteps: curriculum-driven learning of stepping stone skills », *in*: *Computer Graphics Forum*, vol. 39, 8, Wiley Online Library, 2020, pp. 213–224.

[145] Pei Xu and Ioannis Karamouzas, « A GAN-Like Approach for Physics-Based Imitation Learning and Interactive Character Control », *in*: *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 4.*3* (2021), pp. 1–22.

[146] Bangpeng Yao and Li Fei-Fei, « Modeling mutual context of object and human pose in human-object interaction activities », *in*: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 2010, pp. 17–24.

[147] Yuting Ye and C Karen Liu, « Synthesis of responsive motion using a dynamic model », *in*: *Computer Graphics Forum*, vol. 29, 2, Wiley Online Library, 2010, pp. 555–562.

[148] KangKang Yin, Kevin Loken, and Michiel Van de Panne, « Simbicon: Simple biped locomotion control », *in*: *ACM Transactions on Graphics (TOG)* 26.*3* (2007), 105–es.

[149] Zhiqi Yin et al., « Discovering diverse athletic jumping strategies », *in*: *ACM Transactions on Graphics (TOG)* 40.*4* (2021), pp. 1–17.

[150] Mohamed YOUNES, *[SCA'23] MAAIP:Multi-Agent Adversarial Interaction Priors for imitation from fighting demonstrations*, 2023, URL: `https://www.youtube.com/watch?v=wQfIiw_rQ3w` (visited on 11/04/2023).

[151] Mohamed Younes et al., « AIP: Adversarial Interaction Priors for Multi-Agent Physics-based Character Control », *in*: *SIGGRAPH Asia 2022 Posters*, 2022, pp. 1–2.

[152] Mohamed Younes et al., « MAAIP: Multi-Agent Adversarial Interaction Priors for imitation from fighting demonstrations for physics-based characters », *in*: *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 6.*3* (2023), pp. 1–20.

[153] Chao Yu et al., « The surprising effectiveness of ppo in cooperative, multi-agent games », *in*: *arXiv preprint arXiv:2103.01955* (2021).

[154] Ye Yuan et al., « SimPoE: Simulated Character Control for 3D Human Pose Estimation », *in*: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 7159–7169.

[155] Feng Zhang et al., « Distribution-aware coordinate representation for human pose estimation », *in*: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 7093–7102.

[156] Jason Y Zhang et al., « Predicting 3d human dynamics from video », *in*: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7114–7123.

[157] Junyu Zhang et al., « Variational policy gradient method for reinforcement learning with general utilities », *in*: *Advances in Neural Information Processing Systems* 33 (2020), pp. 4572–4583.

[158] Liang Zhang et al., « KaraKter: An autonomously interacting Karate Kumite character for VR-based training and research », *in*: *Computers & Graphics* 72 (2018), pp. 59–69.

[159] Song Zhang and Peisen S Huang, « Novel method for structured light system calibration », *in*: *Optical Engineering* 45.*8* (2006), p. 083601.

[160] Weiyu Zhang, Menglong Zhu, and Konstantinos G Derpanis, « From actemes to action: A strongly-supervised representation for detailed action understanding », *in*: *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2248–2255.

[161] Yan Zhang, Michael J Black, and Siyu Tang, « We are more than our joints: Predicting how 3d bodies move », *in*: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3372–3382.

[162] Ce Zheng et al., « Deep learning-based human pose estimation: A survey », *in*: *arXiv preprint arXiv:2012.13392* (2020).

[163] Brian D Ziebart et al., « Maximum entropy inverse reinforcement learning. », *in*: *Aaai*, vol. 8, Chicago, IL, USA, 2008, pp. 1433–1438.

[164] Victor Brian Zordan and Jessica K Hodgins, « Motion capture-driven simulations that hit and react », *in*: *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2002, pp. 89–96.

COLLEGE DOCTORAL BRETAGNE
MATHS, TELECOMS
INFORMATIQUE, SIGNAL
SYSTEMES, ELECTRONIQUE

Université de Rennes

Titre : Apprentissage et Simulation des Stratégies de Sport (la Boxe) pour l'Entrainement en Réalité Virtuelle

Mot clés : Simulation, Deep Reinforcement Learning, Sport

Résumé : Cette thèse étudie l'extraction et la simulation des interactions entre combattants, principalement pour la boxe, en utilisant des techniques d'apprentissage profond : l'estimation du mouvement humain à partir de vidéos, l'apprentissage par imitation basé sur l'apprentissage par renforcement, et la simulation de personnages basée sur la physique. Dans le contexte de l'analyse sportive à partir de vidéos, un protocole de référence est proposé dans lequel diverses méthodes contemporaines d'extraction de poses humaines en 2D sont évaluées pour leur précision à dériver des informations positionnelles à partir d'enregistrements vidéo RVB de boxeurs lors de mouvements complexes et dans des circonstances de tournage défavorables.

Dans une deuxième partie, la thèse se concentre sur la reproduction d'interactions réalistes entre boxeurs à partir de données de mouvement et d'interaction grâce à une méthodologie innovante permettant d'imiter les interactions et les mouvements de plusieurs personnages simulés physiquement à partir de données de capture de mouvement non organisées. Initialement, cette technique a été démontrée pour simuler une boxe légère entre deux combattants sans contact physique significatif. Par la suite, elle a été étendue pour prendre en compte des données d'interaction supplémentaires concernant la boxe avec du contact physique réel et d'autres activités de combat, ainsi que pour gérer les instructions de l'utilisateur et les restrictions d'interaction.

Title: Learning and Simulation of Sport Strategies (Boxing) for Virtual Reality Training

Keywords: Simulation, Deep Reinforcement Learning, Sport

Abstract: This thesis investigates the extraction and simulation of fighter interactions, mainly for boxing, by utilizing deep learning techniques: human motion estimation from videos, reinforcement learning-based imitation learning, and physics-based character simulation. In the context of sport analysis from videos, a benchmark protocol is proposed where various contemporary 2D human pose extraction methods are evaluated for their precision in deriving positional information from RGB video recordings of boxers during complex movements and unfavorable filming circumstances.

In a second part, the thesis focuses on replicating realistic fighter interactions given motion and interaction data through an innovative methodology for imitating interactions and motions among multiple physically simulated characters derived from unorganized motion capture data. Initially, this technique was demonstrated for simulating light shadow boxing between two fighters without significant physical contact. Subsequently, it was expanded to accommodate additional interaction data featuring boxing with actual physical contact and other combat activities, along with handling user instructions and interaction restrictions.