MAAIP: Multi-Agent Adversarial Interaction Priors for imitation from fighting demonstrations for physics-based characters

Mohamed Younes
Inria, IRISA, University of Rennes
Rennes, France
mohamed.younes@inria.fr

Ewa Kijak
University of Rennes, Inria, IRISA
Rennes, France
ewa.kijak@irisa.fr

Richard Kulpa
University of Rennes, Inria, M2S
Rennes, France
richard.kulpa@irisa.fr

Simon Malinowski
University of Rennes, Inria, IRISA
Rennes, France
simon.malinowski@irisa.fr

Franck Multon
University of Rennes, Inria, IRISA, M2S
Rennes, France
franck.multon@irisa.fr

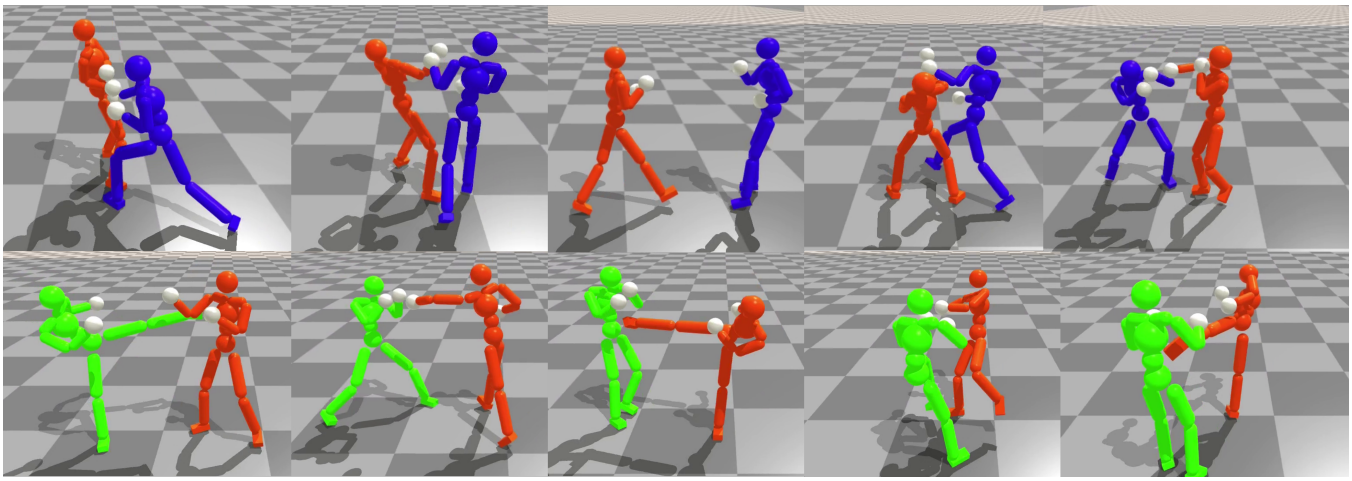

Figure 1: Two examples of imitation-based simulation with two different styles. Top: boxing scenario trained with single-actor and multiple-actors motion capture boxing datasets. Bottom: Qwankido (martial art) scenario based on the same approach, simply replacing the boxing datasets by Qwankido datasets. In these examples, the sequences show the ability of the characters to avoid attacks and then to counter-attack.

ABSTRACT

Simulating realistic interaction and motions for physics-based characters is of great interest for interactive applications, and automatic secondary character animation in the movie and video game industries. Recent works in reinforcement learning have proposed impressive results for single character simulation, especially the ones that use imitation learning based techniques. However, imitating multiple characters interactions and motions requires to also model their interactions. In this paper, we propose a novel Multi-Agent Generative Adversarial Imitation Learning based approach that generalizes the idea of motion imitation for one character to deal with both the interaction and the motions of the multiple physics-based characters. Two unstructured datasets are given as inputs: 1) a single-actor dataset containing motions of a single actor performing a set of motions linked to a specific application, and 2) an interaction dataset containing a few examples of interactions between multiple actors. Based on these datasets, our system trains control policies allowing each character to imitate the interactive skills associated with each actor, while preserving the intrinsic style. This approach has been tested on two different fighting styles, boxing and full-body martial art, to demonstrate the ability of the method to imitate different styles.

SCA '23, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

CCS CONCEPTS

• **Computing methodologies** → **Procedural animation**; **Adversarial learning**; **Multi-agent reinforcement learning**.

KEYWORDS

Character Animation, Multi-Agent Reinforcement Learning, Adversarial Imitation learning, Physics-based Simulation, Motion Capture

ACM Reference Format:

Mohamed Younes, Ewa Kijak, Richard Kulpa, Simon Malinowski, and Franck Multon. 2018. MAAIP: Multi-Agent Adversarial Interaction Priors for imitation from fighting demonstrations for physics-based characters. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (SCA '23).* ACM, New York, NY, USA, 13 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

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 Virtual Reality fighting training, for example, the user is not expecting an optimal behavior 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.

Multiple physics-based character interactions can be simulated using spacetime constraints and optimal control [18, 44]. 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 [37, 38]. 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 [16].

Reinforcement learning has been explored for designing physics-based controllers capable of imitating motions given a small unstructured database of examples while achieving different tasks [33]. This Adversarial Motion Priors (AMP) approach is based on the Generative Adversarial Imitation Learning (GAIL) [11] framework. It mainly uses an adversarial discriminator output as a reward instead of manually designing an imitation reward. However, it has only been applied to single character control.

In this paper, we propose Multi-Agents Adversarial Interaction Priors (MAAIP), a method 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) [39] 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. To the best of our knowledge, MAAIP is one of the first adversarial learning system for physics-based multiple-character animation that combines adversarial motion prior and interaction prior, allowing different characters to imitate interaction from a set of unstructured motion clips performed by multiple actors.

We evaluate our method by simulating competitive interactions between two physics-based characters, with different styles: boxing (hands only) and a Qwankido (a Sino-Vietnamese martial art with full body interactions). We use a few minutes of single-actor motion clip examples, and short sequences of interaction motion clips. We show the ability of our method to simulate interactions between two fighters, while imitating the style of each fighter contained in the datasets, without the need of designing specific constraints or rewards. We then explore the limits of the generalization ability of the method, when dealing with situations that have not been captured in the interaction dataset used for training. We also show how to control the interaction, by simply adding new rewards, such as interactively controlling the direction of the simulated fight, making the fighter be more aggressive, or more defensive.

2 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 2.1) and multiple (section 2.2) characters. We then introduce Imitation Learning techniques used for physics-based character simulation in section 2.3.

2.1 Single 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 [12], abstract models [3], optimal control [27], model predictive control [7, 26] and reinforcement learning [48, 50]. 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 [53]. More recent works tracked reference motions by learning policies that get feedback from the physics simulation [15, 19]. With the development of deep reinforcement

learning techniques, it became possible to robustly track agile human motions [30] and to generalize to various morphologies [47] and environments [48].

2.2 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 [10], but also more complex parameters, such as gaze, orientations or time coordination. Optimal control with spacetime optimization has been used to solve complex interaction problems involving multiple physics-based characters [18, 44]. 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 [37, 38]. 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. [9] 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. [46] proposed a two-steps approach that first learns an imitation policy from single-actor motion capture data, then transfers it into competitive policies. [21] 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.

2.3 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. [4, 14, 15, 19, 20, 30]. 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 [31].

Adversarial imitation learning [11, 52] 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 [25, 45, 49]. 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) [33] proposed a number of tweaks to address those issues, such as using gradient penalty, but did not handle interaction imitation of multiple characters.

Multi-Agent Generative Adversarial Imitation Learning (MA-GAIL) [39] is a promising framework to design controllers that imitate the interaction behavior of multiple characters, given a small set of unstructured motion capture examples. However, this framework has never been developed 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.

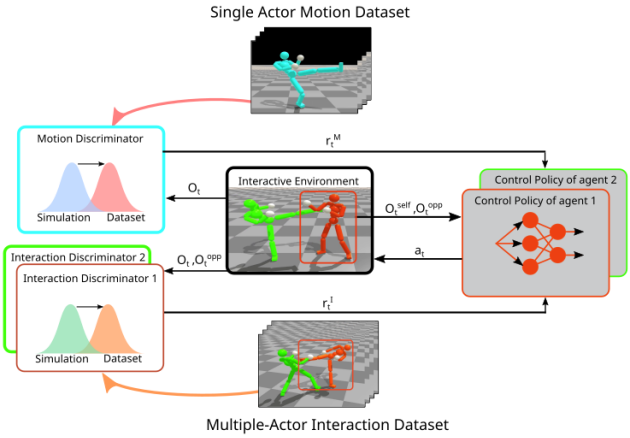

Figure 2: Overview of the system. Multiple-actors motion capture clips are used to train an interaction discriminator assigned to each agent, aiming at returning an interaction reward r^I. Single-actor motion clips are used to train a motion discriminator that returns a motion reward r^M. The rewards learned by the two discriminators are combined to train each agent's policy in order to imitate the interactive behavior depicted in the datasets.

3 SYSTEM OVERVIEW

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 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. 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 4.2, we give more details about the agent's observations.

We define the controller for each character using a policy:

$$\pi(a_t | o_t^{self}, o_t^{opp}) \tag{1}$$

where a_t is the action that specifies the set of target joint angles (target poses) used by a Proportional Derivative (PD) controller [41]. 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 his 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.

4 METHOD

We formulate the interaction imitation problem 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 [1, 17]. In the following, we detail our architecture adapted from Multi-Agent Generative Adversarial Imitation Learning [39], with two major contributions: modeling the interaction with an opponent, and new objectives for training the system.

4.1 Multi-Agent Generative Adversarial Imitation Learning

Multi-Agent Generative Adversarial Imitation Learning (MAGAIL) [39] is a variant of the Generative Adversarial Imitation Learning (GAIL) [11] 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 [5].

For each agent i, a parametrized 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 [39]. The objective to be optimized is the following:

$$\min_{\theta} \max_{\omega} \mathbb{E}_{\pi_E}[\sum_{i=1}^{N} \log D_{\omega_i}(s, a_i)] + \mathbb{E}_{\pi_\theta}[\sum_{i=1}^{N} \log(1 - D_{\omega_i}(s, a_i))] \tag{2}$$

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:

$$r_t^i = -\log(1 - D_{\omega_i}(s_t, a_t)_i) \tag{3}$$

4.2 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 o_t^{opp}. 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 [46]. 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 use 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 [32, 33], 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 y 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 PD controller to compute joint torques for the character's joints. The target pose for spherical joints is represented by 3D exponential map $q \in \mathbb{R}^3$ [6] 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 [33]. The target rotations for revolute joints are specified as 1D rotation angles $q = \theta$. The resulting action space has 28 dimensions.

4.3 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:

$$r(o_t, a_t, o_{t+1}) = w^M r^M(o_t^{self}, o_{t+1}^{self}) + w^I r^I(o_t, o_{t+1}^{self}) \quad (4)$$

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

Following [33], 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 4.1. 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 [43]. In this case, the reward function based on the motion discriminator is given by:

$$r_t^M = -\log(1 - D^M(o_t^{self}, o_{t+1}^{self})) \quad (5)$$

while the reward based on the interaction discriminators is:

$$r_t^I = -\log(1 - D^I(o_t, o_{t+1}^{self})) \quad (6)$$

We also use the gradient penalty regularization [33] 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:

$$\min_{D^M} -\mathbb{E}_{\pi_E}[\log D^M(\phi)] - \mathbb{E}_{\pi_i}[\log(1 - D^M(\phi))] + w_{gp}\mathbb{E}_{\pi_E}\left[\left\|\nabla_\phi D^M(\phi)|_\phi\right\|^2\right] \quad (7)$$

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:

$$\min_{D^I} -\mathbb{E}_{\pi_E}[\log D^I(\psi)] - \mathbb{E}_{\pi_i}[\log(1 - D^I(\psi))] + w_{gp}\mathbb{E}\pi_E\left[\left\|\nabla_\psi D^I(\psi)|_\psi\right\|^2\right] \quad (8)$$

4.4 Network Architecture

Since the agents are homogeneous (i.e. they have the same observation and action spaces), we use 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 [2, 42, 51]. 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 [28], followed by a linear output layer. We also use centralized training and decentralized execution (CTDE) for training the agents [22]. 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 [22]. The value function $V(s_t)$, the interaction discriminators D^I and motion discriminator D^M, are modeled as networks with similar architecture.

4.5 Training

We use the framework of MAGAIL [39] with the multi-agent proximal policy optimization algorithm MAPPO [36, 51]: 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 [33], we use a combination of these rewards to get the final imitation reward r_t at time t according to Equation (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 [33]. 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) [35] to compute advantages for updating the policies. The centralized value

function is updated using TD(λ) [40]. We follow the recommendations from [51] to choose the hyperparameters of the multi-agent PPO algorithm. The training process is described in algorithm 1.

ALGORITHM 1: 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 4. for all agents
      $r_t^I \leftarrow$ calculate interaction reward according to formula 5. for all agents
      $r_t \leftarrow$ combine r_t^M and r_t^I according to formula 2.
      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 6.
    update each D^I using K transitions sampled from $\mathbb{M}^I$ and from $\mathbb{B}^I$ according to formula 7.
  end for
  update π and V using samples from $\mathbb{B}^\pi$ for all agents using MAPPO.
end while

5 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 (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 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.

5.1 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 downsampled to 30Hz and retargeted 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. We plan to share our datasets upon acceptance of this work to support future research in filling the gap between individual motor skills for single characters and interactive skills between multiple characters.

Isaac Gym [23] 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 hyperparameters are available in supplementary material.

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 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.

5.2 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 (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 and 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 served as a demonstration for each avatar, in all the simulations. 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. 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 collapse 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.

5.3 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 $|f_{opp \to self}|$ 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:

$$r^C = exp(-w \cdot |f_{opp \to self}|). \tag{9}$$

Similarly, the damage maximization reward is expressed by:

$$r^C = 1 - exp(-w \cdot |f_{self \to opp}|) \tag{10}$$

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 1) 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 6 depicts a QwanKiDo sequence simulated without the damage minimization reward, leading to a series of attacks. The bottom part of Figure 6 depicts the resulting sequence when adding the damage minimization reward, which shows more defensive and less engaging behavior.

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 1: 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).

5.4 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 [33]:

$$r^C = exp(-w \cdot (d^* \cdot v^{root})) \tag{11}$$

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 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 2. 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).

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 2: 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.

5.5 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 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 [34].

6 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.

6.1 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 show 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 2), but the resulting motions were less natural. Indeed, the agents seem to exploit the motion included in the interaction 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 seamless transitions between interaction and displacement motions (see supplementary video). However, the weighting between the interaction prior, the single motion prior and the task reward needs to be tuned so that 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 on maximizing the heading task relying only on the single motion prior.

For the transfer to unseen fight situations introduced in 5.3, 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.

6.2 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 [11] 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) [24] that showed more training stability and better overall quality. The objective for optimizing the discriminator is defined as:

$$\min_{D^M} \mathbb{E}_{\pi_E}\left[\left(D^M(\phi)-1\right)^2\right] + \mathbb{E}_{\pi}\left[\left(D^M(\phi)+1\right)^2\right] \tag{12}$$

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

$$r(\phi) = \max\left[0, u - v \cdot \left(D^M(\phi)-1\right)^2\right] \tag{13}$$

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

We experimented with this objective 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.

7 DISCUSSION AND LIMITATIONS

We presented a novel adversarial system for imitating interactions between multiple physics-based characters, using unstructured motion clips. This approach is adapted from the MAGAIL framework, and we introduced important adaptations to handle multiple physics-based characters' simulation: 1) modeling of reactive behavior as a transition from the current full observation to the next self observation, and 2) training both single motion and interactions priors by leveraging reference motions of a single actor and motions of interaction between multiple actors.

It enabled us to obtain convincing simulation of two fighting characters, using only small datasets of motions and interactions examples. The resulting sequence does 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. This is a key contribution of this work.

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 [13] tried to mitigate this problem by conditioning the motion prior on latents that encode each motion clip. Some other work [8, 29] 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.

Using this approach to simulate new individual styles, or new multi-characters activities (fencing, dancing, collaborative work, etc.), only requires the user to retrain 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 motion of the motion clips examples, the resulting motion of some sequences may still appear unnatural; Since 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. Moreover, 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 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, and so that they are 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.

ACKNOWLEDGMENTS

This work was supported by French government funding managed by the National Research Agency under the Investments for the Future program in France 2030 with the grant ANR-18-EURE-0022 (DIGISPORT). It was granted access to the HPC resources of IDRIS under the allocation 20XX-AD011013491R made by GENCI.

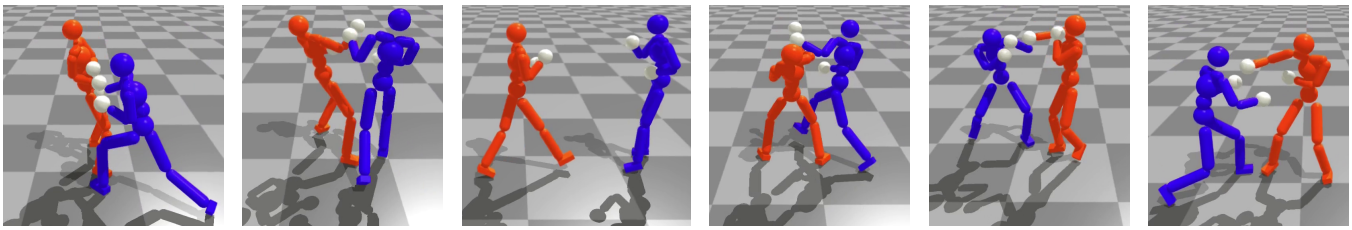

Figure 3: Simulation of boxing interaction between two agents. The boxers show agility in the movements, interactive skills such as getting closer to the opponent, dodging and blocking attacks as well as finding attack openings.

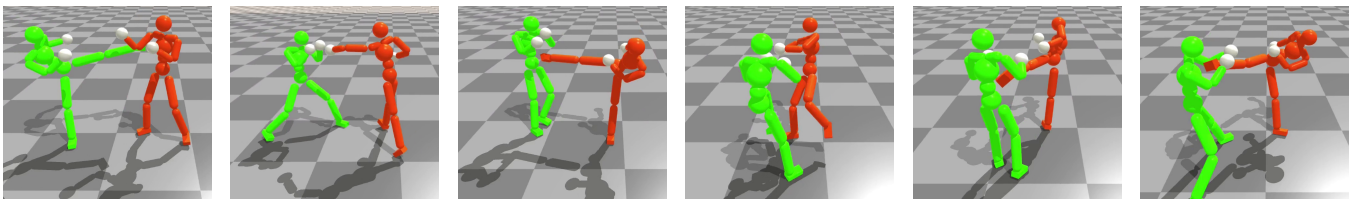

Figure 4: Simulation of QwanKiDo interaction between two agents. The agents show highly-dynamic motions, such as using full body for attacks, unique fighting styles similar to the actor motions used for training them.

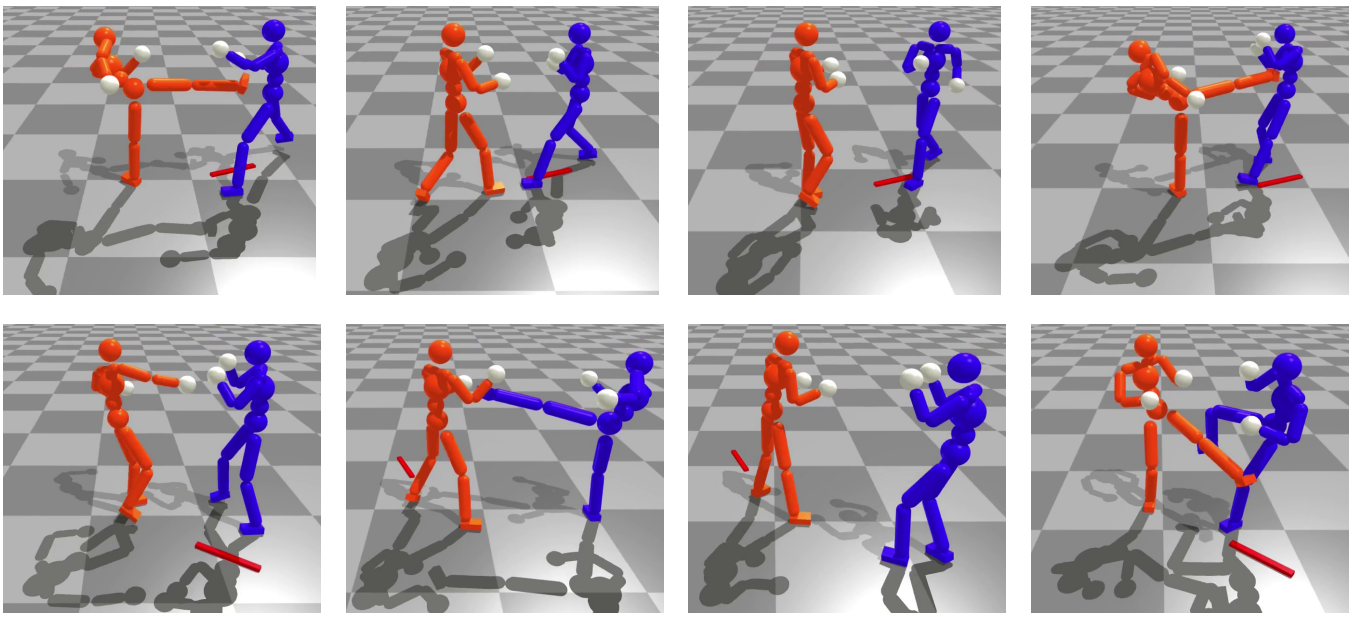

Figure 5: An example of interaction simulation with heading controls. The fighters are constrained to move towards a given target direction, represented by the red line.

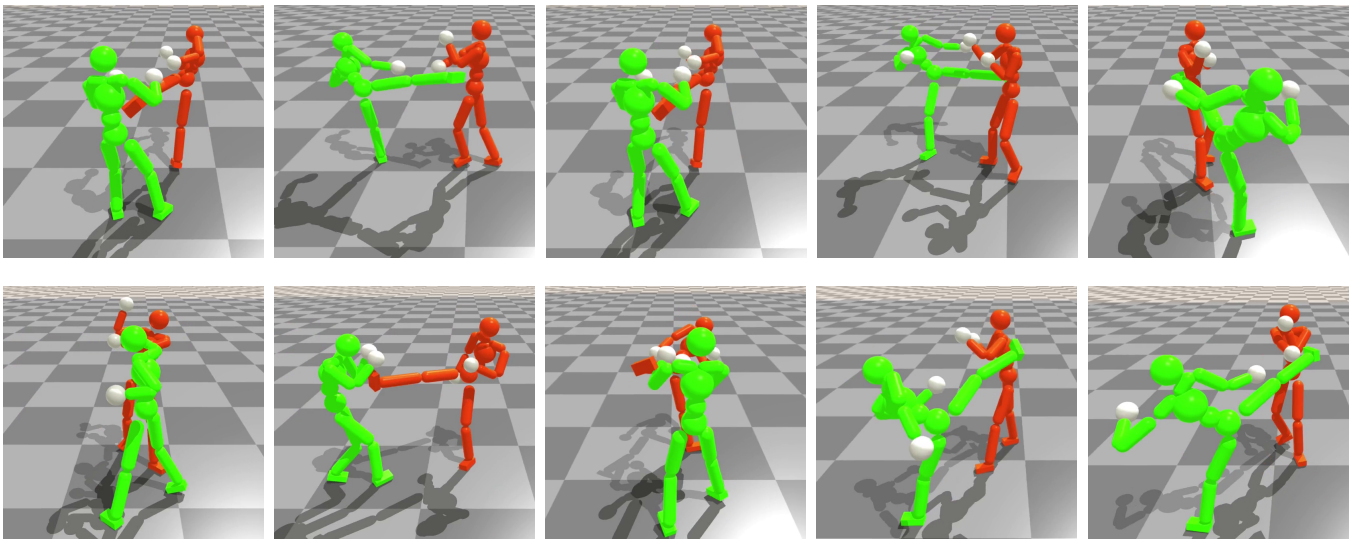

Figure 6: An example of fighting simulation using additional control rewards, that encourages the agents to minimize the damage dealt by the opponent to specific body parts: head, torso and pelvis. Top: without control reward ; Bottom: with control reward. The reward drives the agents into simulating interactive behavior where they act more defensively, and they block attacks more often.

REFERENCES

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

[2] Filippos Christianos, Georgios Papoudakis, Muhammad A Rahman, and Stefano V Albrecht. 2021. Scaling multi-agent reinforcement learning with selective parameter sharing. In *International Conference on Machine Learning*. PMLR, 1989–1998.

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

[4] Marco Da Silva, Yeuhi Abe, and Jovan Popović. 2008. Simulation of human motion data using short-horizon model-predictive control. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 371–380.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

[6] F Sebastian Grassia. 1998. Practical parameterization of rotations using the exponential map. *Journal of graphics tools* 3, 3 (1998), 29–48.

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

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

[9] Brandon Haworth, Glen Berseth, Seonghyeon Moon, Petros Faloutsos, and Mubbasir Kapadia. 2020. Deep integration of physical humanoid control and crowd navigation. In *Motion, Interaction and Games*. 1–10.

[10] Edmond S. L. Ho, Taku Komura, and Chiew-Lan Tai. 2010. Spatial Relationship Preserving Character Motion Adaptation. In *ACM SIGGRAPH 2010 Papers* (Los Angeles, California) *(SIGGRAPH '10)*. Association for Computing Machinery, New York, NY, USA, Article 33, 8 pages. https://doi.org/10.1145/1833349.1778770

[11] Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. *Advances in neural information processing systems* 29 (2016).

[12] Jessica K Hodgins, Wayne L Wooten, David C Brogan, and James F O'Brien. 1995. Animating human athletics. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. 71–78.

[13] Jordan Juravsky, Yunrong Guo, Sanja Fidler, and Xue Bin Peng. 2022. PADL: Language-Directed Physics-Based Character Control. In *SIGGRAPH Asia 2022 Conference Papers (SA '22 Conference Papers),*.

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

[15] Yoonsang Lee, Sungeun Kim, and Jehee Lee. 2010. Data-driven biped control. In *ACM SIGGRAPH 2010 papers*. 1–8.

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

[17] Michael L Littman. 1994. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*. Elsevier, 157–163.

[18] Karen Liu, Aaron Hertzmann, and Zoran Popovic. 2006. Composition of complex optimal multi-character motions. *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 215–222. https://doi.org/10.1145/1218064.1218093

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

[20] Libin Liu, KangKang Yin, Michiel Van de Panne, Tianjia Shao, and Weiwei Xu. 2010. Sampling-based contact-rich motion control. In *ACM SIGGRAPH 2010 papers*. 1–10.

[21] Siqi Liu, Guy Lever, Zhe Wang, Josh Merel, SM Ali Eslami, Daniel Hennes, Wojciech M Czarnecki, Yuval Tassa, Shayegan Omidshafiei, Abbas Abdolmaleki, et al. 2022. From motor control to team play in simulated humanoid football. *Science Robotics* 7, 69 (2022), eabo0235.

[22] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems* 30 (2017).

[23] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. 2021. Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning.

[24] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. 2017. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*. 2794–2802.

[25] Josh Merel, Yuval Tassa, Dhruva TB, Sriram Srinivasan, Jay Lemmon, Ziyu Wang, Greg Wayne, and Nicolas Heess. 2017. Learning human behaviors from motion capture by adversarial imitation. *arXiv preprint arXiv:1707.02201* (2017).

[26] Igor Mordatch, Martin De Lasa, and Aaron Hertzmann. 2010. Robust physics-based locomotion using low-dimensional planning. In *ACM SIGGRAPH 2010 papers*. 1–8.

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

[28] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Icml*.

[29] Tu Nguyen, Trung Le, Hung Vu, and Dinh Phung. 2017. Dual discriminator generative adversarial nets. *Advances in neural information processing systems* 30 (2017).

[30] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel Van de Panne. 2018. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)* 37, 4 (2018), 1–14.

[31] Xue Bin Peng, Glen Berseth, KangKang Yin, and Michiel Van De Panne. 2017. Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–13.

[32] Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. 2022. ASE: Large-Scale Reusable Adversarial Skill Embeddings for Physically Simulated Characters. *arXiv preprint arXiv:2205.01906* (2022).

[33] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. 2021. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–20.

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

[35] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2015. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438* (2015).

[36] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

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

[38] Hubert P. H. Shum, Taku Komura, Masashi Shiraishi, and Shuntaro Yamazaki. 2008. Interaction Patches for Multi-Character Animation. *ACM Trans. Graph.* 27, 5, Article 114 (dec 2008), 8 pages. https://doi.org/10.1145/1409060.1409067

[39] Jiaming Song, Hongyu Ren, Dorsa Sadigh, and Stefano Ermon. 2018. Multi-Agent Generative Adversarial Imitation Learning. In *Advances in Neural Information Processing Systems*, Vol. 31. https://proceedings.neurips.cc/paper/2018/file/240c945bb72980130446fc2b40fbb8e0-Paper.pdf

[40] Richard S Sutton. 1988. Learning to predict by the methods of temporal differences. *Machine learning* 3, 1 (1988), 9–44.

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

[42] Justin K Terry, Nathaniel Grammel, Ananth Hari, Luis Santos, and Benjamin Black. 2020. Revisiting parameter sharing in multi-agent deep reinforcement learning. *arXiv preprint arXiv:2005.13625* (2020).

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

[44] Joris Vaillant, Karim Bouyarmane, and Abderrahmane Kheddar. 2017. Multi-Character Physical and Behavioral Interactions Controller. *IEEE Transactions on Visualization and Computer Graphics* 23, 6 (2017), 1650–1662. https://doi.org/10.1109/TVCG.2016.2542067

[45] Ziyu Wang, Josh S Merel, Scott E Reed, Nando de Freitas, Gregory Wayne, and Nicolas Heess. 2017. Robust imitation of diverse behaviors. *Advances in Neural Information Processing Systems* 30 (2017).

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

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

[48] Zhaoming Xie, Hung Yu Ling, Nam Hee Kim, and Michiel van de Panne. 2020. Allsteps: curriculum-driven learning of stepping stone skills. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 213–224.

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

[50] Zhiqi Yin, Zeshi Yang, Michiel Van De Panne, and KangKang Yin. 2021. Discovering diverse athletic jumping strategies. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–17.

[51] Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre Bayen, and Yi Wu. 2021. The surprising effectiveness of ppo in cooperative, multi-agent games. *arXiv preprint arXiv:2103.01955* (2021).

[52] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. 2008. Maximum entropy inverse reinforcement learning.. In *Aaai*, Vol. 8. Chicago, IL, USA, 1433–1438.

[53] Victor Brian Zordan and Jessica K Hodgins. 2002. Motion capture-driven simulations that hit and react. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics*

symposium on Computer animation. 89–96.