西北工业大学

数字图像处理–论文翻译

原论文标题: Strategy and Skill Learning for Physics-based Table Tennis Animation

符博锦

计算机科学与技术

2024 年 11 月

学号: 2022302745

Northwestern Polytechnical University

Strategy and Skill Learning for Physics-based Table Tennis Animation

Jiashun Wang
jiashunw@cmu.edu
Carnegie Mellon University
USA

Jessica Hodgins
jkh@cmu.edu
Carnegie Mellon University
USA
The AI Institute
USA

Jungdam Won
jungdam@imo.snu.ac.kr
Seoul National University
South Korea

Figure 1: On the left, two physically-simulated agents engage in a competitive match, controlled by our strategy and skill controllers. The center panels show the five learned skills: Forehand Drive, Push and Smash, Backhand Drive and Push, showcasing the skill controller's ability to execute a diverse set of skills. On the right, a human interacts with the simulated agent through VR.

ABSTRACT

Recent advancements in physics-based character animation leverage deep learning to generate agile and natural motion, enabling characters to execute movements such as backflips, boxing, and tennis. However, reproducing the selection and use of diverse motor skills in dynamic environments to solve complex tasks, as humans do, still remains a challenge. We present a strategy and skill learning approach for physics-based table tennis animation. Our method addresses the issue of mode collapse, where the characters do not fully utilize the motor skills they need to perform to execute complex tasks. More specifically, we demonstrate a hierarchical control system for diversified skill learning and a strategy learning framework for effective decision-making. We showcase the efficacy of our method through comparative analysis with state-of-the-art methods, demonstrating its capabilities in executing various skills for table tennis. Our strategy learning framework is validated through both agent-agent interaction and human-agent interaction in Virtual Reality, handling both competitive and cooperative tasks.

CCS CONCEPTS

• **Computing methodologies** → **Physical simulation**; Motion Processing.

KEYWORDS

Character Animation, Physics-based Characters, Deep Reinforcement Learning, Multi-character Interaction, Virtual Reality, Table Tennis

ACM Reference Format:
Jiashun Wang, Jessica Hodgins, and Jungdam Won. 2024. Strategy and Skill Learning for Physics-based Table Tennis Animation. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers '24 (SIGGRAPH Conference Papers '24), July 27–August 01, 2024, Denver, CO, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3641519.3657437

SIGGRAPH Conference Papers '24, July 27–August 01, 2024, Denver, CO, USA

ACM ISBN 979-8-4007-0525-0/24/07
https://doi.org/10.1145/3641519.3657437

1 INTRODUCTION

The integration of deep learning into physics-based character animation has led to significant advancements in generating agile and natural motion, enhancing the lifelike quality of characters in complex environments. To increase the versatility of these characters, it is essential to ensure that their skills can be reused in environments or conditions that may not precisely match their training data. To achieve this goal, recent approaches have focused on learning reusable skill embeddings. These approaches are typically trained in two stages. Initially, characters learn various skill embeddings by imitating reference motions. Then, in the task training stage, they apply these skills to accomplish diverse tasks. These approaches have demonstrated remarkable success in generating natural motion in various environments.

However, when the differences between the skills are subtle, these approaches often suffer from mode collapse during the task training phase. Specifically, although agents (a.k.a. characters) can learn various skills during the imitation stage, they tend to use

a limited set of skills for the downstream tasks, neglecting the diversity of their learned skills in the imitation stage. Thus, mode collapse restricts the agents' potential in scenarios that require a diverse set of skills. Mode collapse also restricts exploration during RL training, resulting in sub-optimal task performance.

Another relatively unexplored topic relates to the decision strategy of agents, particularly their ability to dynamically formulate decision strategies that encompass skill selection and associated skill goals in response to task demands. Most previous studies either have not required a diverse skill set or have relied on a human user to manually determine skills for the agents. Agents have generally not been equipped with the capability to employ different strategies to adapt to complex and dynamic environments.

Our research introduces a learning approach to enhance both the skill and strategic decision-making capabilities of physically simulated agents. First, we develop a hierarchical skill controller that enables agents to utilize different table tennis skills and transition among them rapidly. This controller effectively addresses mode collapse during task training. Second, we develop a method for strategy learning, enabling agents to explicitly select and utilize specific skills for different types of interaction, whether competitive or cooperative. An overview of the results is in Figure 1.

We demonstrate the effectiveness of our approach through two interaction environments: a table tennis match played between two simulated agents and a match between a human and a simulated agent in virtual reality (VR). In the agent-agent environment, the agents demonstrate improved skill diversity and decision strategy in simulated table tennis matches compared to results predicted by the previous techniques. In the human-agent interaction environment, we evaluate both cooperative and competitive scenarios in real-time interactions between humans and agents. These environments not only validate our approach but also provide platforms for future research into complex agent behaviors and human-agent dynamics. Code and data for this paper are at https://jiashunwang.github.io/PhysicsPingPong/.

We summarize the contributions of this paper as follows:

- A hierarchical skill controller that empowers physically simulated agents to explicitly perform various skills, enabling rapid skill transitions. An interaction learning framework designed to create a decision strategy allows agents to continually learn and adapt, meeting the demands of competition or cooperation in dynamic environments with other agents and with humans.
- Novel results demonstrating our learning framework's capacity to generate intelligent decisions and natural motions for table tennis in two scenarios: agent-agent interactions in a simulated environment and human-agent interactions in a VR environment. The agent-agent environment is a platform for developing and testing competitive and cooperative algorithms while the VR environment allows natural human-agent interactions.

2 RELATED WORK

We review the closest related work in physics-based character animation with reusable skills and multi-character animation. We review studies on transitions among skills as we develop a method for skill selection and transition. We further discuss relevant research in human-agent interaction in VR.

2.1 Physics-based Character Animation

Incorporating physical laws into character animation allows for the development of controllers that generate more realistic behaviors [Hodgins et al. 1995; Laszlo et al. 1996]. Optimization techniques, such as trajectory optimization [de Lasa et al. 2010; Mordatch et al. 2012; Yin et al. 2008] and sampling-based methods [Liu et al. 2016, 2010] have been widely explored. Recently, deep reinforcement learning (DRL) has been shown to substantially enhance control capabilities [Liu and Hodgins 2017; Peng et al. 2017]. Due to its flexibility and ease of use, DRL methods eliminate the need for designing complex objective functions while delivering outstanding results and have attracted significant research interest as a result.

Data-driven methods have become prevalent in physics-based character animation studies since a DRL-based method was introduced by Peng et al. [2018]. The idea has been extended for handling larger datasets [Bergamin et al. 2019; Won et al. 2020] and for allowing recombination of existing state transitions [Peng et al. 2021]. Recently, much attention has been paid to reusable motor skills. The idea is to learn a latent space of reference motions and then to reuse the learnt space for downstream tasks. Various latent models have been studied such as encoder-decoders with autoregression [Merel et al. 2019; Won et al. 2021], spherical embedding [Dou et al. 2023; Peng et al. 2022; Tessler et al. 2023], conditional variational autoencoder (VAE) [Won et al. 2022; Yao et al. 2022], and vector-quantized VAE [Zhu et al. 2023]. Some researchers have also proposed part-wise models to maximize reusability of reference motions [Bae et al. 2023; Xu et al. 2023].

Our system is designed for table tennis games, involving two players (i.e., agents). Two or more agents have been created primarily with kinematic approaches [Kwon et al. 2008; Liu et al. 2006; Shum et al. 2008, 2012; Wampler et al. 2010]. There exist two recent approaches [Won et al. 2021; Zhu et al. 2023] demonstrating examples of physically simulated boxing. Zhang et al. [2023] build a system to learn tennis skills from broadcast videos and produce rallies with a mirrored opponent. In their approach, kinematics-based motion generation is utilized first, followed by physics-based tracking, relying on residual forces and extra arm control for successful strikes. Skill and target selection are not learned but rather performed manually or randomly to create a scene including two players. In contrast, our method learns not only agile and precise motor control to strike the ball but also strategies to select skills and targets based on the movement of the opponent and the ball.

2.2 Transition of skills

Option-based methods [Bagaria and Konidaris 2020; Jain et al. 2021; Klissarov et al. 2017; Konidaris and Barto 2009; Sutton et al. 1999] represent skills as *options*, which are sequentially constructed, with each option's execution in the chain enabling the agent to execute the subsequent option. Lee et al. [2019] propose learning additional transition policies to connect primitive skills and introduce proximity predictors, which yield rewards based on proximity suitable for initial states for the next skill. One challenge of transitioning between different skills to chain long-horizon tasks is addressed

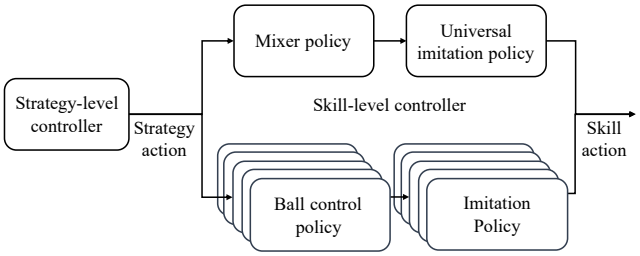

Figure 2: An overview of our method. Strategy action includes the skill command and ball's target landing location. Skill action includes the target joint angles for PD controllers, blended from the outputs of imitation policies.

by terminal state regularization [Lee et al. 2021]. Behavior Trees are also a common method for planning the transition between different states [Cheng et al. 2023; French et al. 2019; Marzinotto et al. 2014]. These methods achieve skill transitions by ensuring that the terminal state of the previous stage is close to the initial state of the next stage. While these methods work well for tasks that are not time-sensitive, table tennis, which involves high-speed movements and rapid responses, poses a challenge as players do not always hit the ball from a well-defined initial state.

2.3 Human-agent interaction

Research has focused on human sports training within VR [Liu et al. 2020; Pastel et al. 2023]. However, these studies often lack a physically simulated opponent. There are commercial games that allow people to interact with an agent in VR for sports activities, such as boxing, golf, and badminton. Eleven Table Tennis [2016] is a VR-based table tennis game similar to the one we have constructed, which enables a human to play with an agent. However, this agent is not simulated with full-body dynamics, rather it is simulated with only a floating head and a floating paddle. Advances in GPU-accelerated simulation and our control algorithm, enable us to create a physically-simulated agent with full-body dynamics that can play in real-time with humans. Another relevant area involves enhancing the agent's capabilities with *human-in-the-loop* methodologies [Brenneis et al. 2021; Li et al. 2022; Seo et al. 2023; Wang et al. 2023] using extended reality. Our work differs from previous studies by bringing humans and agents into a unified environment allowing bidirectional physical interaction, where they can cooperate and compete.

3 METHOD OVERVIEW

We propose a hierarchical approach that includes a strategy-level controller and a skill-level controller. The strategy-level controller takes the states of the agent, opponent, and ball as inputs, and outputs a strategy action, which includes the skill to use and the target landing location for the ball. Meanwhile, the skill-level controller takes the states of the agent and ball, along with the strategy action as inputs, and then generates a skill action, which includes the target joint angles for PD controllers. An overview of our method is in Figure 2 and Figure 3 shows the architecture of our method.

4 SKILL-LEVEL CONTROLLER

Three stages are required to train our skill-level controller. Initially, we train imitation policies using the motion capture data. Then the ball control policy for each skill is learned, which enables the agent to hit back balls using the corresponding imitation policy. Finally, we learn a policy that enables the agent to perform various skills sequentially while making plausible transitions among them. We call this policy the mixer policy. Once the skill-level controller is trained, the agent can proficiently and continuously execute various skills, sending balls to diverse target locations.

4.1 Imitation Policy

We first categorize the motion capture dataset into five subsets corresponding to each skill. This subdivision allows us to train the skill-specific imitation policies. We also utilize all the data to train a universal imitation policy. The imitation policy is represented as $\pi^i(a^i|s, z^i)$, where $i \in \{1, 2, 3, 4, 5, u\}$, $1 \sim 5$ are indices of different skills and u is the index of the universal imitation policy. z^i is a latent variable sampled from a hyper-sphere distribution, and s is the agent's state. The goal of the imitation policy is to output an action a^i that leads to simulated motions similar to the reference motions. Thus, each skill-specific imitation policy generates motions similar to its corresponding reference motion in each skill subset, while the universal imitation policy generates motions encompassing the entire motion capture dataset. When solving specific tasks in the later stage, using a single universal imitation policy trained with a variety of motions often leads to the mode collapse problem. The agent does not explore various available skills enough; instead, it repeats very limited skills, and the task performance remains suboptimal. Our controller design is inspired by mixture-of-experts and mitigates this problem. Each imitation policy $\pi^i(a^i|s, z^i)$ is built by the adversarial framework ASE [Peng et al. 2022], where the policy is updated so that it tricks a motion discriminator D^i. The transitions $d_{M^i}(s, s')$ existing in the motion capture dataset are used as positive samples while the transitions $d_{\pi^i}(s, s')$ generated from the policy π^i are used as negative samples. The discriminator is trained by minimizing:

$$\begin{aligned}\min_{D^i} & -\mathbb{E}_{d_{M^i}(s,s')}\log(D^i(s,s')) - \mathbb{E}_{d_{\pi^i}(s,s')}\log(1-D^i(s,s')) \\ & + \lambda_{gp}\mathbb{E}_{d_{M^i}(s,s')}\left\|\nabla_\phi D^i(\phi)\big|_{\phi=(s,s')}\right\|^2, \end{aligned} \tag{1}$$

where the last term is a gradient penalty regularization with a constant factor λ_{gp}. We train encoders q^i to encourage correspondence between the transition (s, s') and the latent variable z^i. The encoder is modeled as a von Mises-Fisher distribution and it is trained by maximizing its log-likelihood:

$$\begin{aligned}\max_{q^i} & \mathbb{E}_{p(z^i)}\mathbb{E}_{d^{\pi^i}(s,s'|z^i)}[\log q^i(z^i|s,s')], \\ & q^i(z^i|s,s') = \frac{1}{Z}exp(\mu_{q^i}(s,s')^T z^i)\end{aligned} \tag{2}$$

where $\mu_{q^i}(s, s')$ is the mean of the distribution, and Z is a normalization constant. Given a discriminator D^i, the reward to train π^i is defined as:

$$r_t = -\log(1 - D^i(s_t, s_{t+1})) + \beta \log q^i(z_t^i|s_t, s_{t+1}). \tag{3}$$

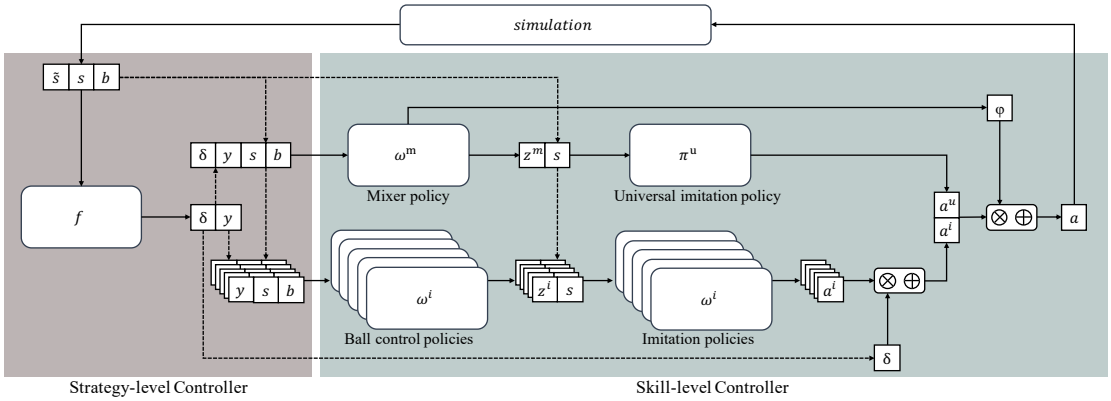

Figure 3: The architecture of our method. We train the skill-level controller through the stages of imitation policies, ball control policies, and finally, the mixer policy. We train the strategy-level controller after the skill-level controller is ready and its weight is frozen. ⊗⊕ stands for the weighted sum in Equation 8.

where β is the relative weight. Additionally, a diversity term is included to encourage different latent variables to represent distinct motions. Bringing everything together, the objective of π^i becomes:

$$\max_{\pi^i} \mathbb{E}_{p(Z)}\mathbb{E}_{p(\tau|\pi^i,Z)}[\sum_{t=0}^{T}\gamma^t(r_t)] \tag{4}$$
$$-\lambda_{D^i}\mathbb{E}_{d^{\pi^i}(s)}\mathbb{E}_{z_1^i,z_2^i\sim p(z^i)}\left[\left(\frac{D_{KL}(\pi^i(\cdot|s,z_1^i),\pi^i(\cdot|s,z_2^i))}{0.5(1-z_1^i z_2^i)}-1\right)^2\right],$$

where $D_{KL}(\cdot||\cdot)$ measures the KL-divergence between two distributions, z_1^i and z_2^i refer two different latent variables, γ is the discount factor and λ is a constant used to balance the weight.

4.2 Ball Control Policy

Once the agent can imitate each skill $i \in \{1,2,3,4,5\}$, we train ball control policies $\omega^i(z^i|s,b,y)$ to enable the agent to hit and move a ball launched from a random location to the desired location, where s denotes the state of the agent, b represents the state of the ball and y is the target landing location for the ball. The task reward r is a composite of three terms: the paddle reward r_p, the ball reward r_b, and the style reward r_s,

$$r(t) = w_p r_p(t) + w_b r_b(t) + w_s r_s(t), \tag{5}$$

where w_p, w_b, and w_s are the relative weights. The paddle reward r_p encourages the agent to position the paddle close to the ball. The reward is defined as:

$$r_p(t) = \begin{cases} \exp(-4||x_p(t)-x_b(t)||^2), & \text{if } C_{bp}(t)=0, \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

where $x_p(t)$ and $x_b(t)$ represent the positions of paddle and ball, respectively, $C_{bp}(t)$ is a binary variable representing contact states. $C_{bp}(t)=0$ means the ball has not contacted the paddle until time t, while $C_{bp}(t)=1$ indicates the ball has contacted the paddle at time t or contacted previously before time t. It will be reset to 0 whenever the next ball is launched. The ball reward r_b is given by:

$$r_b(t) = \begin{cases} 1+\exp(-4||x_c(t)-x_t(t)||^2), \\ \qquad \text{if } C_{bp}(t)=1 \text{ and } C_{bt}(t)=0, \\ 0, \text{ otherwise,} \end{cases} \tag{7}$$

where $x_t(t)$ is the target landing location of the ball, $x_c(t)$ represents the anticipated landing location on the table, calculated using Newton's equation of motion for the point mass (i.e., quadratic trajectory), with its state corresponding to the current position and velocity of the ball. $C_{bt}(t)$ is a binary variable checking the contact history between the ball and the table, which is updated similarly to $C_{bp}(t)$. The agent receives the maximum reward when it successfully hits the ball and the ball moves toward the target location. We also apply the style reward, $r_s = -\log(1-D^i(s_t,s_{t+1}))$ in the task training similarly to ASE [Peng et al. 2022], where D^i is the discriminator learned during the previous stage.

4.3 Mixer Policy

While our agent can play table tennis using the ball control policies with the corresponding imitation policies, its capability is limited to repeating a single skill. Simply transitioning from one controller to another during play often leads to failure due to a mismatch between the end state of one skill to the start state of the next. To create plausible transitions among the different skills, we learn a mixer policy $\omega^m(z^m|s,b,\delta,y)$, which takes the agent state s, the ball state b, and the strategy action (δ,y) as input, where δ is a one-hot vector determining the skill to use and y is the target ball landing location, then generates the latent variable for the universal imitation policy π^u and a set of blending weights φ, mixing the skill actions in a joint-wise manner. In other words, φ determines which policy the agent relies on among the transition and five different skills. The target joint angles for PD controllers are computed as

$$a = \varphi \odot \pi^u(\cdot|s,z^u) + (1-\varphi)\odot\sum_{i=1}^{5}\delta_i\pi^i(\cdot|s,z^i) \tag{8}$$

ALGORITHM 1: Strategy learning

Input: Number of iterations N, interaction environment Env.
Output: Updated policy f.
$f \leftarrow$ Random initialization ;
for $i \leftarrow 1$ **to** N **do**
  $\{(o_k^{\text{expert}}, c_k^{\text{expert}})\}_{k=1}^{K} \leftarrow \text{Interact}(Env, f)$;
  Apply stochastic gradient descent to update f using Equation 9
end

where $\delta = (\delta_1, \delta_2, \delta_3, \delta_4, \delta_5)$ is a one-hot vector indicating the skill selected. While training the mixer policy, the agent is asked to perform the ball control task with randomly launched balls, randomly selected skills, and random target locations. The same rewards used for learning ball control policies are employed, and the weights of all other policies remain frozen.

5 STRATEGY-LEVEL CONTROLLER

The strategy-level controller is developed by iterative behavior cloning inspired by [Oh et al. 2018]. More specifically, we first collect interaction data by randomly sampling strategy actions during agent-agent play or human-agent interactions with VR. This data is then used to update the strategy-level controller, and we repeat this process by collecting new interaction data with the latest strategy-level controller. When collecting interaction data, there are two options: competition and cooperation. To train a competitive strategy, we choose data that results in victories, whereas in a cooperative strategy, we choose sequences where the opponent successfully catches the ball.

A strategy-level controller produces a skill index and a target landing location repeatedly so that they satisfy the requirements of different applications. More specifically, the strategy-level controller f takes the strategy observation $o = (s, \tilde{s}, b)$ as input where s, $\tilde{s}$, and b are the agent state, the opponent state, and the ball state, respectively, then outputs the strategy action $c = (\delta, y)$, where δ is a one-hot vector determining the skill to use, and y is the target landing location of the ball. The strategy action is updated when the ball starts moving from the opponent to the agent. To effectively learn a strategy-level controller, we adopt a behavior cloning approach with iterative refinement, aiming to learn strategies from available expert demonstrations $\{(o_k^{\text{expert}}, c_k^{\text{expert}})\}_{k=1}^{K}$ (see Algorithm 1). As a structure of the controller, we utilize a Conditional Variational Autoencoder (CVAE) to model the stochastic nature inherent in sports gameplay. During training, the CVAE encoder takes o and c as inputs and generates the mean μ and variance σ^2 of the posterior Gaussian distribution $Q(u|\mu, \sigma^2)$. We then sample a latent variable u from this distribution and concatenate it with observation o as input for the decoder, which reconstructs the action c'. The training loss is defined as:

$$\sum_{k=1}^{K} ||c_k^{\text{expert}} - c_k'|| + \beta_{KL} D_{KL}(Q(u|\mu_k, \sigma_k^2)||\mathcal{N}(0, I)), \qquad (9)$$

where $D_{KL}(\cdot||\cdot)$ measures the KL divergence between the two distributions and β_{KL} is the relative weight. During inference the decoder is utilized solely, it takes a randomly sampled latent variable u and the observation o, and then generates the strategy action that guides the agent to perform a corresponding skill. If the opponent successfully returns the ball, this process repeats. We collect expert demonstrations from two different interaction environments (Env in Algorithm 1). The details of each environment will be explained in Section 6.

6 INTERACTION ENVIRONMENT

We introduce the agent-agent and human-agent interaction environments that we build to validate the strategy learning approach.

The agent-agent interaction environment is an environment where two virtual agents play table tennis with each other (Figure 1 left column). We name one agent as *our agent* and the other as *the opponent*. In the process of learning a strategy-level controller for our agent, the opponent uses a fixed heuristic strategy-level controller while the controller for our agent is updated iteratively. More specifically, we let our agent and the opponent play with each other using their own strategy-level controllers, collect those demonstrations, and then use them to update our agent's controller. If our goal is to learn a competitive strategy, that can beat the opponent, we selectively use demonstrations leading to wins. On the other hand, we use demonstrations where the opponent successfully returns the ball when aiming to learn a cooperative strategy. In our system, we utilize two types of heuristic strategy-level controllers: a random strategy and a video strategy. The random strategy selects skills and target landing locations randomly from a uniform distribution. The video strategy is constructed by using broadcast videos. We extract expert demonstrations $\{(o_k^{\text{video}}, c_k^{\text{video}})\}_{k=1}^{K}$ from existing broadcast videos (20 minutes in total). Subsequently, we train a CVAE using the behavior cloning method.

The human-agent interaction environment allows a human user to play with a virtual agent. In our system, the user interacts with an agent by using a VR device, including a head-mount display and a hand controller ((Figure 1 right column)). The VR interface operates through Unity while the physics-based simulation runs on Isaac Gym [Makoviychuk et al. 2021]. To enable the simulated agent to interact with a human user, we physically simulate the user's paddle, with its position and orientation controlled via signals from the VR interface. Specifically, for paddle control, we use the VR hand controller's Cartesian pose q_{user} and the simulated paddle pose q_{sim} to calculate the target velocity $\dot{q}_{\text{target}} = (q_{\text{user}} - q_{\text{sim}})/\Delta t$, where Δt is the simulation step. We use this target velocity as an input to the velocity controller provided by the simulator. For visualization, Unity takes the state of the simulated agent, user's paddle, and ball as inputs and renders them using visualization assets. This implementation significantly reduces the amount of information exchange compared to a previous study that sent stereo images [Seo et al. 2023], enabling real-time interaction and gameplay. By considering a human user as the opponent, the strategy-level controller of the agent can be built through the same pipeline used for the agent-agent interaction environment.

7 EXPERIMENTS

We evaluate the skill-level controller based on motion quality and task performance. We assess the strategy-level controller by examining its effectiveness in agent-agent and human-agent interaction

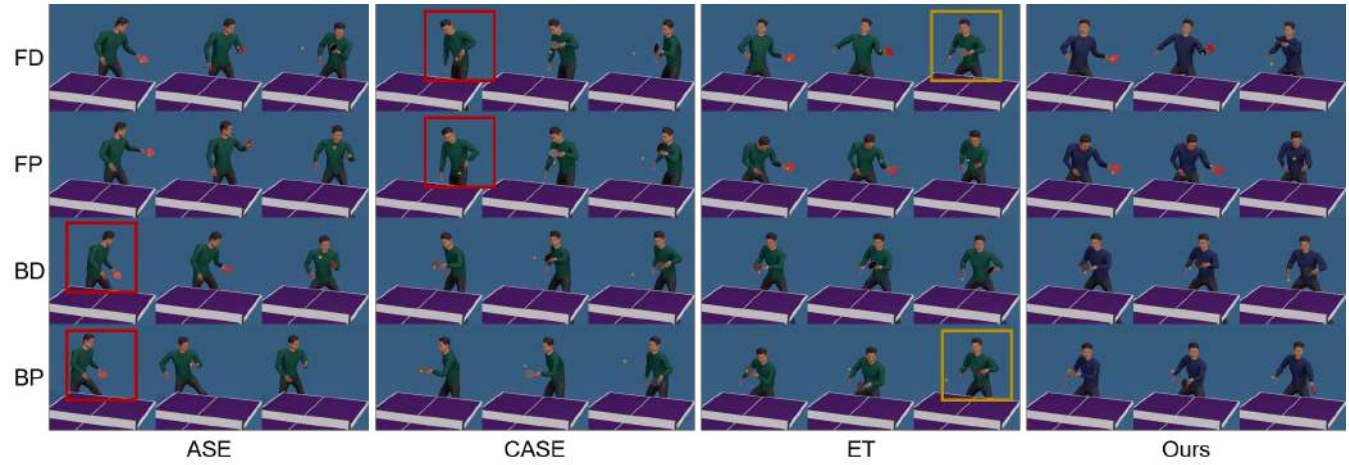

Figure 4: Comparison with other methods with four skill commands. ASE and CASE may use wrong skills as shown in the red box. ET may terminate earlier to return to a preparation pose, as shown in the yellow boxes.

Figure 5: Transition results with only using forehand and backhand drive controllers. Both controllers are trained with random initialized configurations from the motion capture data. As shown in the red boxes, the agent attempts to use another forehand drive before the next ball is launched, which prevents it from switching back to a backhand drive in time.

environments, with the demands of both competition and cooperation scenarios.

7.1 Skill evaluation

We evaluate the skill performance from motion quality and task performance. The evaluation of motion quality measures the naturalness of generated motions when given the desired skill command and whether the agent performs the correct skill. The evaluation of task performance measures the overall proficiency in playing table tennis. We compare our method with two state-of-the-art methods, ASE [Peng et al. 2022] and CASE [Dou et al. 2023], as well as an explicit transition model (ET) which is a variant of our method with the mixer policy ω^m removed from our model. We train an explicit controller to handle skill transitions by taking over the control when the ball passes the net until it is returned to the agent. The controller is also built using the ball control-imitation architecture. The key difference between our approach and ET is that ours provides continuous action blending with the selected skill's action at every time step, whereas ET does not.

7.1.1 Motion quality. We design three metrics to evaluate the motion quality, particularly to evaluate the naturalness and mode collapse. The first metric is *Discriminator Score*, which measures how similar the current strike motion is to the reference motion of i-th target skill. Because we have five skills, we train a discriminator D^i_{test} for each skill and utilize the following equation to calculate the score:

$$\text{Discriminator Score i} = \frac{1}{T}\sum_{t=0}^{T-1} -\log(1 - D^i_{test}(s_t, s_{t+1})), \quad (10)$$

where T is the length of a single strike motion. The details of training D^i_{test} will be introduced in Appendix D. The second metric is *Skill Accuracy*, to measure whether the agent performs the correct skill given the target skill command. Specifically, given a motion sequence, we first classify it by taking the index of the discriminator which provides the highest value. Then, we compare it with the target skill command to calculate accuracy. The third metric, *Diversity Score*, is designed to test whether motions for drive and push commands are distinctive enough. In table tennis, motions within each skill category (e.g., forehand drive vs. forehand push) might exhibit subtle differences, even though their roles in gameplay can be significantly distinct. *Diversity Score* measures the capability of distinguishing motions that are visually similar. It is calculated by

$$\text{Diversity Score} = \frac{1}{2N^2}\sum_{i\in\{1,3\}}\sum_{m=1}^{N}\sum_{n=1}^{N}||s^i_m - s^{i+1}_n||, \quad (11)$$

where s^i is the state that the agent hits the ball under skill command i. Specifically $i \in \{1, 3\}$ stands for forehand drive and backhand drive and $i+1 \in \{2, 4\}$ stands for forehand push and backhand push respectively. N is the total number of hits for each skill command. We only take into account the moment when the agent's paddle makes contact with the ball to calculate this score. The first two metrics evaluate a general skill mode collapse problem, for example,

using forehand motions when being asked to use backhand. The third metric is specifically designed to measure if the agent has the ability to accurately perform drive and push skills.

The evaluation results are reported in Table 1, where the values are computed with 10k balls randomly launched toward the agent equipped with the respective skill controller. For the *Discriminator Score*, our method significantly surpasses ASE and CASE, and achieves 15.6% higher score than ET. These results prove our method generates motions that are the most similar to the reference target skill. As shown in the *Skill Accuracy* results, our method uses the correct skills to hit the ball in most cases (0.76 in Table 1). While ASE and CASE only use the correct skill with an accuracy of 0.38 and 0.47. In *Diversity Score*, our method achieves 30.7%, 32.3%, and 9.4% higher scores than ASE, CASE, and ET respectively. We also show a qualitative comparison in Figure 4. We find ASE and CASE often use forehand skills when asked to use backhand skills, or vice-versa, as shown in the red boxes in Figure 4. And we can't observe any forehand smash skill. Even when the correct skills are used, the naturalness remains insufficient. ET often does not complete the skills; instead, the skills are terminated earlier to return to a preparation pose, as shown in the yellow box in Figure 4. ASE and CASE often overlook skill commands, tending to use relatively fewer skills. This error occurs because, during the task training, these methods fall into mode collapse, making it challenging to effectively explore various skills. In contrast, our approach leverages an idea of the mixture-of-experts approach to avoid this problem. We further test the use of individual skill controllers without any design for transitions. Each skill controller is trained with randomly initialized configurations sampled from the motion capture data. As shown in Figure 5, after executing a forehand drive, the agent attempts another forehand drive before the next ball is launched—a typical behavior for single-skill controllers. This unnecessary movement prevents it from switching to a backhand drive in time, ultimately causing a missed shot.

7.1.2 Task performance. To assess the task performance of the skill controller, we evaluate two aspects: sustainability and accuracy. Sustainability is determined by the average number of successful continuous returns, while accuracy is measured by the average distance in meters between the target landing location and the actual contact location on the table. Besides testing on the training distribution, we collect some ball tracking data with faster ball trajectories from a match between high-ranking players and evaluate whether each method can perform well with the testset of the ball tracking data. We report the evaluation results in Table 2. The numbers in parentheses are the results of the fine-tuning experiments. Our method can achieve the largest number of average hits and the second-best accuracy. Although ET can achieve higher accuracy, it is not sustainable, especially for more challenging balls. It only achieves an average of 3.66 hits because it often lacks time to respond to the next ball due to the explicit transition design.

7.1.3 Blending weights of the mixer policy. We test the agent with different skills to hit the ball and visualize the average blending weights φ of the shoulder, elbow, and wrist joints in Figure 8. We can observe that the weights of the mixer policy are usually lowest at the moment the paddle contacts the ball, and higher before and after transitions between different skills. It indicates a reliance on the pre-trained ball control policy during ball strikes, and on the mixer policy during transitions.

Table 1: Comparisons on *Discriminator Score*, *Skill Accuracy*, and *Diversity Score*.

	ASE	CASE	ET	Ours
Discriminator Score	1.62	2.28	4.95	**5.72**
Skill Accuracy	0.38	0.47	0.69	**0.76**
Diversity Score	6.13	6.05	7.32	**8.01**

Table 2: Task performance evaluation. Our method can achieve the longest average hits and the second best accuracy.

	ASE	CASE	ET	Ours
Avg Hits	9.54 (5.94)	8.79 (5.28	6.55 (3.66)	**10.93 (6.28)**
Avg error	0.28 (0.33)	0.35 (0.39)	**0.25 (0.28)**	0.26 (0.31)

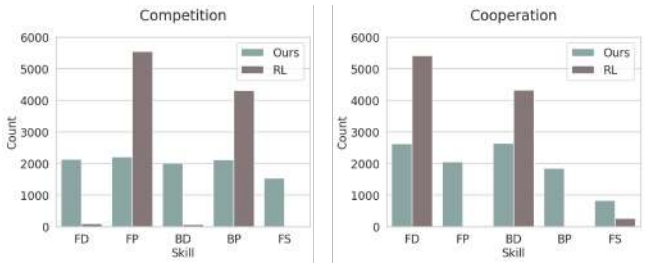

Figure 6: Skill command distribution of our method and RL.

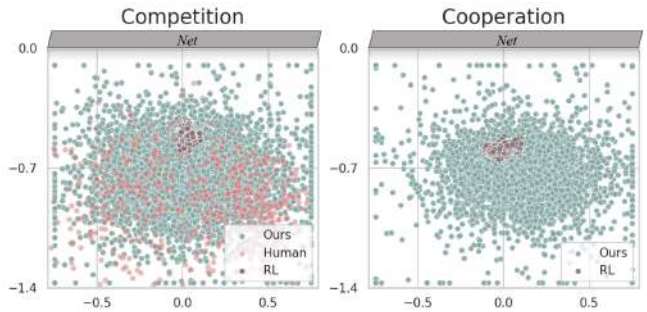

Figure 7: Target landing locations of our method, RL and Human.

7.2 Evaluation for agent-agent interaction

We evaluate the performance of learned strategies in the agent-agent interaction environment under both competition and cooperation settings. The competition strategy aims to develop an agent that achieves a higher winning rate than the opponent. The cooperation strategy develops an agent that can play gently with the opponent to increase the length of rallies. As a baseline, we learn a strategy policy via reinforcement learning (RL). Please refer to Appendix E for details on training the RL baseline. Our method and the RL baseline are compared by having them play with two types of opponents: the random strategy and the video strategy opponent introduced in Section 6. Each evaluation is computed over

Table 3: Strategy evaluation. We report the winning rates for the competition setting and average rounds for the cooperation setting.

	Competition		Cooperation	
	RL	Ours	RL	Ours
Random op	0.641	**0.687**	14.9	**16.4**
Video op	0.637	**0.681**	15.6	**18.2**

10k points. Table 3 shows the winning rate and the average rounds for the competition and cooperation settings. Our strategy learning algorithm can achieve higher winning rates for the competition setting and can maintain longer rallies for the cooperation setting for both opponents.

In Figure 6 and 7, we visualize the histogram of skill commands from the strategy policies, and the target landing locations. In Figure 7, we also provide ball landing locations captured from real players during competitive matches. We observe that our method has a more similar distribution of landing locations to humans than RL. RL converges to less diverse skill commands and it only hits to a small region of the table. In contrast, our method utilizes various skills and target locations throughout the gameplay. We also include qualitative gameplay visualizations in Figure 9. We further let RL and our method compete with each other, and report the winning rates in Table 4. Each method has two strategy policies trained with two opponents, therefore, we have four matches in total. Because RL falls into a local minimum and overfits to a specific opponent, our method achieves a much higher winning rate.

7.3 Evaluation of Human-agent interaction

Before learning strategies for the human-agent interaction environment, we finetune the skill-level controller using the play data of a human user interacting with the agent equipped with the original skill-level controller. The finetuning is required because of the domain gap between what the simulated agent has experienced and the styles of a real human user in the VR environment. After finetuning the skill controller, strategies are learned by following similar procedures as the agent-agent interaction environment. For training a competition strategy, we use demonstrations that result in the agent winning points, thus presenting more challenging returns for the human opponent. In contrast, for training a cooperative strategy, we use demonstrations where the human can maintain rallies, emphasizing easier ball returns for the human. These demonstrations serve as expert demonstrations in Algorithm 1. We report the winning rate of the agent and the average hits between the user and the agent in Table 5. When playing with the initial policies, the agent can achieve a winning rate of 64% and a rally with 4.04 hits on average. After two iterations of refinement of the competition strategy, the agent can achieve a winning rate of 78% , and the average number of hits drops to 3.75. For the cooperative strategy, the winning rate drops to 58%, and the user can achieve a rally with an average of 5.34 hits. These results demonstrate that our strategy learning algorithm is also effective for the human-agent interaction environment. We provide screenshots of real-time human-agent gameplay video in Figure 10.

Table 4: Winning rates between our method and RL. The opponent in parentheses is the opponent during training of the strategy policy.

	Ours (random op)	Ours (video op)
RL (random op)	0.45 vs 0.55	0.47 vs 0.53
RL (video op)	0.42 vs 0.58	0.42 vs 0.58

Table 5: Evaluation of human-agent interaction.

	Initial policies	Competition	Cooperation
Winning rate	0.64	0.78	0.58
Avg hits	4.04	3.75	5.34

8 DISCUSSION AND CONCLUSION

Although our method produces agents that play competitively and more naturally, it still has several limitations. First, although building individual policies for each skill and combining them via the mixer policy clearly improves the generated motion quality and task performance, our model would not scale well to a dataset including hundreds of different skills. Developing a hybrid model that combines our approach with a model learnable from unlabeled motions to achieve both high motion quality and scalability would be an interesting future research topic. Second, because our method is data-driven, the captured motion quality significantly affects the final motion quality. For example, the player tends to use large arm motions, and this motion style appears in our results as well. However, in matches, using less arm motion could be a way to conserve energy, and concealed movements can also confuse the opponent. Lastly, although we employ a rigid-body simulation for every component, including the ball, player, and table, where the ball can spin as well, air resistance is modeled using only damping based on velocity, rather than incorporating the Magnus effect, which bends the ball trajectory due to air pressure differences. This omission could impact the realism of our animations and the final strategies our system learned.

In this paper, we introduce a learning approach for physics-based table tennis animation. We develop a hierarchical controller structure, which overcomes the mode collapse problem that appears frequently in reusable latent-based models. Our approach not only improves overall motion quality but also enables us to learn effective decision strategies for two types of environments: agent-agent and human-agent interactions.

ACKNOWLEDGMENTS

This work was partially completed during Jiashun Wang's internship at The AI Institute. Jungdam Won was partially supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) [NO.2021-0-01343-004, Artificial Intelligence Graduate School Program (Seoul National University)] and ICT(Institute of Computer Technology) at Seoul National University. We would like to thank Murphy Wonsick for helping to build the VR system and Melanie Danver for rendering the results.

REFERENCES

Jinseok Bae, Jungdam Won, Donggeun Lim, Cheol-Hui Min, and Young Min Kim. 2023. PMP: Learning to Physically Interact with Environments using Part-wise Motion Priors. In *ACM SIGGRAPH 2023 Conference Proceedings, SIGGRAPH 2023*. ACM, 64:1–64:10. https://doi.org/10.1145/3588432.3591487

Akhil Bagaria and George Konidaris. 2020. Option Discovery using Deep Skill Chaining. In *8th International Conference on Learning Representations, ICLR 2020*. OpenReview.net. https://openreview.net/forum?id=B1gqipNYwH

Kevin Bergamin, Simon Clavet, Daniel Holden, and James Richard Forbes. 2019. DReCon: data-driven responsive control of physics-based characters. *ACM Trans. Graph.* 38, 6 (2019), 206:1–206:11. https://doi.org/10.1145/3355089.3356536

Dylan J. A. Brenneis, Adam S. R. Parker, Michael Bradley Johanson, Andrew Butcher, Elnaz Davoodi, Leslie Acker, Matthew M. Botvinick, Joseph Modayil, Adam White, and Patrick M. Pilarski. 2021. Assessing Human Interaction in Virtual Reality With Continually Learning Prediction Agents Based on Reinforcement Learning Algorithms: A Pilot Study. *arXiv preprint arXiv:2112.07774* (2021). https://arxiv.org/abs/2112.07774

Xuxin Cheng, Ashish Kumar, and Deepak Pathak. 2023. Legs as Manipulator: Pushing Quadrupedal Agility Beyond Locomotion. In *IEEE International Conference on Robotics and Automation, ICRA 2023*. IEEE, 5106–5112. https://doi.org/10.1109/ICRA48891.2023.10161470

Martin de Lasa, Igor Mordatch, and Aaron Hertzmann. 2010. Feature-based locomotion controllers. *ACM Trans. Graph.* 29, 4 (2010), 131:1–131:10. https://doi.org/10.1145/1778765.1781157

Zhiyang Dou, Xuelin Chen, Qingnan Fan, Taku Komura, and Wenping Wang. 2023. C·ASE: Learning Conditional Adversarial Skill Embeddings for Physics-based Characters. In *SIGGRAPH Asia 2023 Conference Papers, SA 2023*. ACM, 2:1–2:11. https://doi.org/10.1145/3610548.3618205

Kevin French, Shiyu Wu, Tianyang Pan, Zheming Zhou, and Odest Chadwicke Jenkins. 2019. Learning Behavior Trees From Demonstration. In *International Conference on Robotics and Automation, ICRA 2019*. IEEE, 7791–7797. https://doi.org/10.1109/ICRA.2019.8794104

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, SIGGRAPH 1995*. ACM, 71–78. https://doi.org/10.1145/218380.218414

Arushi Jain, Khimya Khetarpal, and Doina Precup. 2021. Safe option-critic: learning safety in the option-critic architecture. *Knowl. Eng. Rev.* 36 (2021), e4. https://doi.org/10.1017/S0269888921000035

Martin Klissarov, Pierre-Luc Bacon, Jean Harb, and Doina Precup. 2017. Learnings Options End-to-End for Continuous Action Tasks. *arXiv preprint arXiv:1712.00004* (2017). http://arxiv.org/abs/1712.00004

George Dimitri Konidaris and Andrew G. Barto. 2009. Skill Discovery in Continuous Reinforcement Learning Domains using Skill Chaining. In *Advances in Neural Information Processing Systems 22*. 1015–1023. https://proceedings.neurips.cc/paper/2009/hash/e0cf1f47118daebc5b16269099ad7347-Abstract.html

Taesoo Kwon, Young-Sang Cho, Sang Il Park, and Sung Yong Shin. 2008. Two-Character Motion Analysis and Synthesis. *IEEE Trans. Vis. Comput. Graph.* 14, 3 (2008), 707–720. https://doi.org/10.1109/TVCG.2008.22

Joseph Laszlo, Michiel van de Panne, and Eugene Fiume. 1996. Limit Cycle Control and Its Application to the Animation of Balancing and Walking. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1996*. ACM, 155–162. https://doi.org/10.1145/237170.237231

Youngwoon Lee, Joseph J. Lim, Anima Anandkumar, and Yuke Zhu. 2021. Adversarial Skill Chaining for Long-Horizon Robot Manipulation via Terminal State Regularization. In *Conference on Robot Learning, 2021 (Proceedings of Machine Learning Research, Vol. 164)*. PMLR, 406–416. https://proceedings.mlr.press/v164/lee22a.html

Youngwoon Lee, Shao-Hua Sun, Sriram Somasundaram, Edward S. Hu, and Joseph J. Lim. 2019. Composing Complex Skills by Learning Transition Policies. In *7th International Conference on Learning Representations, ICLR 2019*. OpenReview.net. https://openreview.net/forum?id=rygrBhC5tQ

Chengxi Li, Pai Zheng, Shufei Li, Yat Ming Pang, and Carman K. M. Lee. 2022. AR-assisted digital twin-enabled robot collaborative manufacturing system with human-in-the-loop. *Robotics Comput. Integr. Manuf.* 76 (2022), 102321. https://doi.org/10.1016/J.RCIM.2022.102321

C. Karen Liu, Aaron Hertzmann, and Zoran Popovic. 2006. Composition of complex optimal multi-character motions. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2006*. Eurographics Association, 215–222. https://doi.org/10.2312/SCA/SCA06/215-222

Huimin Liu, Zhiquan Wang, Christos Mousas, and Dominic Kao. 2020. Virtual Reality Racket Sports: Virtual Drills for Exercise and Training. In *2020 IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2020*. IEEE, 566–576. https://doi.org/10.1109/ISMAR50242.2020.00084

Libin Liu and Jessica K. Hodgins. 2017. Learning to Schedule Control Fragments for Physics-Based Characters Using Deep Q-Learning. *ACM Trans. Graph.* 36, 3 (2017), 29:1–29:14. https://doi.org/10.1145/3083723

Libin Liu, Michiel van de Panne, and KangKang Yin. 2016. Guided Learning of Control Graphs for Physics-Based Characters. *ACM Trans. Graph.* 35, 3 (2016), 29:1–29:14. https://doi.org/10.1145/2893476

Libin Liu, KangKang Yin, Michiel van de Panne, Tianjia Shao, and Weiwei Xu. 2010. Sampling-based contact-rich motion control. *ACM Trans. Graph.* 29, 4 (2010), 128:1–128:10. https://doi.org/10.1145/1778765.1778865

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. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. https://openreview.net/forum?id=fgFBtYgJQX_

Alejandro Marzinotto, Michele Colledanchise, Christian Smith, and Petter Ögren. 2014. Towards a unified behavior trees framework for robot control. In *2014 IEEE International Conference on Robotics and Automation, ICRA 2014*. IEEE, 5420–5427. https://doi.org/10.1109/ICRA.2014.6907656

Josh Merel, Leonard Hasenclever, Alexandre Galashov, Arun Ahuja, Vu Pham, Greg Wayne, Yee Whye Teh, and Nicolas Heess. 2019. Neural Probabilistic Motor Primitives for Humanoid Control. In *7th International Conference on Learning Representations, ICLR 2019*. OpenReview.net. https://openreview.net/forum?id=BJl6TjRcY7

Igor Mordatch, Emanuel Todorov, and Zoran Popovic. 2012. Discovery of complex behaviors through contact-invariant optimization. *ACM Trans. Graph.* 31, 4 (2012), 43:1–43:8. https://doi.org/10.1145/2185520.2185539

Junhyuk Oh, Yijie Guo, Satinder Singh, and Honglak Lee. 2018. Self-Imitation Learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018 (Proceedings of Machine Learning Research, Vol. 80)*. PMLR, 3875–3884. http://proceedings.mlr.press/v80/oh18b.html

Stefan Pastel, Katharina Petri, C. H. Chen, Ana Milena Wiegand Cáceres, M. Stirnatis, C. Nübel, Lasse Schlotter, and Kerstin Witte. 2023. Training in virtual reality enables learning of a complex sports movement. *Virtual Real.* 27, 2 (2023), 523–540. https://doi.org/10.1007/S10055-022-00679-7

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 Trans. Graph.* 37, 4 (2018), 143. https://doi.org/10.1145/3197517.3201311

Xue Bin Peng, Glen Berseth, KangKang Yin, and Michiel van de Panne. 2017. DeepLoco: dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Trans. Graph.* 36, 4 (2017), 41:1–41:13. https://doi.org/10.1145/3072959.3073602

Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. 2022. ASE: large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Trans. Graph.* 41, 4 (2022), 94:1–94:17. https://doi.org/10.1145/3528223.3530110

Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. 2021. AMP: adversarial motion priors for stylized physics-based character control. *ACM Trans. Graph.* 40, 4 (2021), 144:1–144:20. https://doi.org/10.1145/3450626.3459670

Mingyo Seo, Steve Han, Kyutae Sim, SeungHyeon Bang, Carlos Gonzalez, Luis Sentis, and Yuke Zhu. 2023. Deep Imitation Learning for Humanoid Loco-manipulation Through Human Teleoperation. In *22nd IEEE-RAS International Conference on Humanoid Robots, Humanoids 2023*. IEEE, 1–8. https://doi.org/10.1109/HUMANOIDS57100.2023.10375203

Hubert P. H. Shum, Taku Komura, Masashi Shiraishi, and Shuntaro Yamazaki. 2008. Interaction patches for multi-character animation. *ACM Trans. Graph.* 27, 5 (2008), 114. https://doi.org/10.1145/1409060.1409067

Hubert Pak Ho Shum, Taku Komura, and Shuntaro Yamazaki. 2012. Simulating Multiple Character Interactions with Collaborative and Adversarial Goals. *IEEE Trans. Vis. Comput. Graph.* 18, 5 (2012), 741–752. https://doi.org/10.1109/TVCG.2010.257

Richard S. Sutton, Doina Precup, and Satinder Singh. 1999. Between MDPs and Semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning. *Artif. Intell.* 112, 1-2 (1999), 181–211. https://doi.org/10.1016/S0004-3702(99)00052-1

Eleven Table Tennis. 2016. https://elevenvr.com

Chen Tessler, Yoni Kasten, Yunrong Guo, Shie Mannor, Gal Chechik, and Xue Bin Peng. 2023. CALM: Conditional Adversarial Latent Models for Directable Virtual Characters. In *ACM SIGGRAPH 2023 Conference Proceedings, SIGGRAPH 2023*. ACM, 37:1–37:9. https://doi.org/10.1145/3588432.3591541

Kevin Wampler, Erik Andersen, Evan Herbst, Yongjoon Lee, and Zoran Popovic. 2010. Character animation in two-player adversarial games. *ACM Trans. Graph.* 29, 3 (2010), 26:1–26:13. https://doi.org/10.1145/1805964.1805970

Chao Wang, Anna Belardinelli, Stephan Hasler, Theodoros Stouraitis, Daniel Tanneberg, and Michael Gienger. 2023. Explainable Human-Robot Training and Cooperation with Augmented Reality. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems, CHI EA 2023*. ACM, 449:1–449:5. https://doi.org/10.1145/3544549.3583889

Jungdam Won, Deepak Gopinath, and Jessica K. Hodgins. 2020. A scalable approach to control diverse behaviors for physically simulated characters. *ACM Trans. Graph.* 39, 4 (2020), 33. https://doi.org/10.1145/3386569.3392381

Jungdam Won, Deepak Gopinath, and Jessica K. Hodgins. 2021. Control strategies for physically simulated characters performing two-player competitive sports. *ACM Trans. Graph.* 40, 4 (2021), 146:1–146:11. https://doi.org/10.1145/3450626.3459761

Jungdam Won, Deepak Gopinath, and Jessica K. Hodgins. 2022. Physics-based character controllers using conditional VAEs. *ACM Trans. Graph.* 41, 4 (2022), 96:1–96:12.

https://doi.org/10.1145/3528223.3530067

Pei Xu, Xiumin Shang, Victor B. Zordan, and Ioannis Karamouzas. 2023. Composite Motion Learning with Task Control. *ACM Trans. Graph.* 42, 4 (2023), 93:1–93:16. https://doi.org/10.1145/3592447

Heyuan Yao, Zhenhua Song, Baoquan Chen, and Libin Liu. 2022. ControlVAE: Model-Based Learning of Generative Controllers for Physics-Based Characters. *ACM Trans. Graph.* 41, 6 (2022), 183:1–183:16. https://doi.org/10.1145/3550454.3555434

KangKang Yin, Stelian Coros, Philippe Beaudoin, and Michiel van de Panne. 2008. Continuation methods for adapting simulated skills. *ACM Trans. Graph.* 27, 3 (2008), 81. https://doi.org/10.1145/1360612.1360680

Haotian Zhang, Ye Yuan, Viktor Makoviychuk, Yunrong Guo, Sanja Fidler, Xue Bin Peng, and Kayvon Fatahalian. 2023. Learning Physically Simulated Tennis Skills from Broadcast Videos. *ACM Trans. Graph.* 42, 4 (2023), 95:1–95:14. https://doi.org/10.1145/3592408

Qingxu Zhu, He Zhang, Mengting Lan, and Lei Han. 2023. Neural Categorical Priors for Physics-Based Character Control. *ACM Trans. Graph.* 42, 6 (2023), 178:1–178:16. https://doi.org/10.1145/3618397

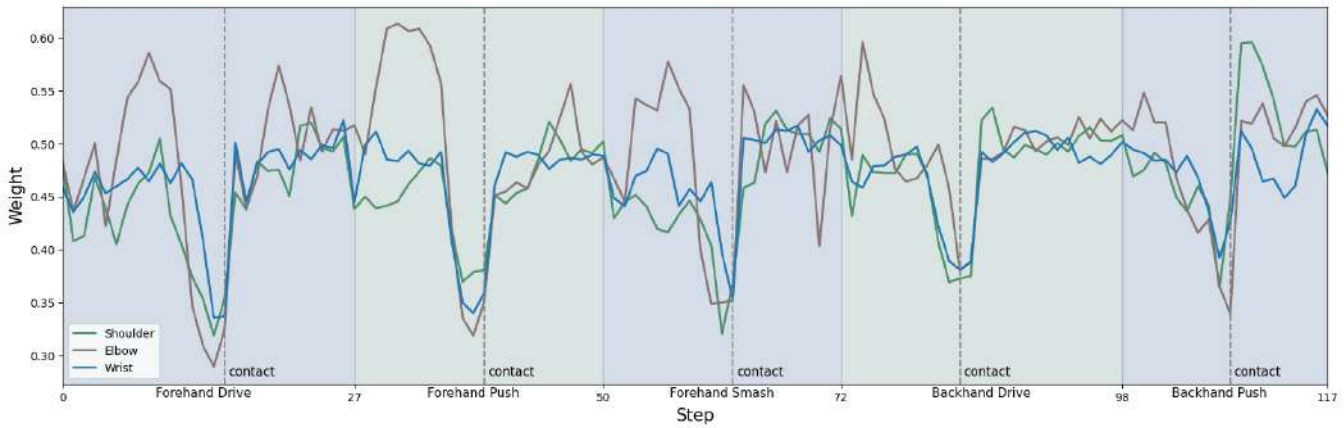

Figure 8: Visualization of the average blending weights φ of the shoulder, elbow, and wrist joints. The weights of the mixer policy are usually lowest when the paddle contacts the ball, and higher before and after transitions between different skills.

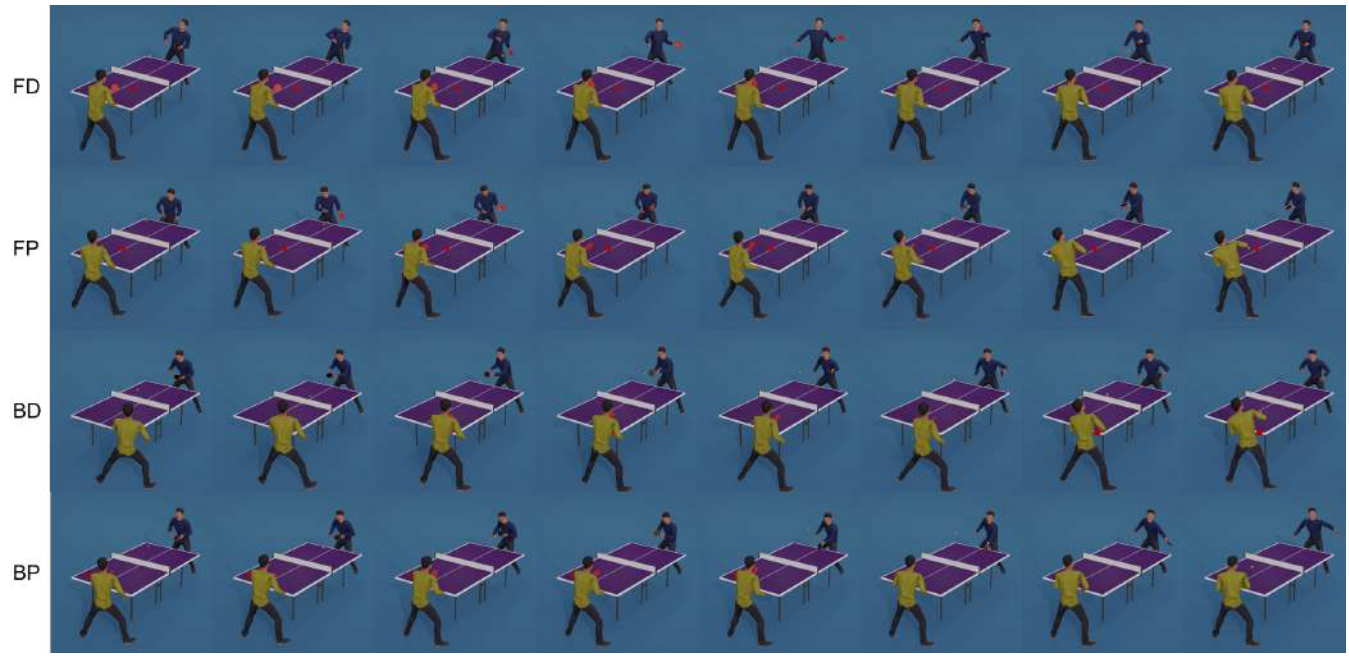

Figure 9: Agent-agent gameplay. Blue agent is applying our strategy-level controller. The red dot is the target. We demonstrate four skills; the forehand smash is less obvious because the opponent does not deliver high and slow shots.

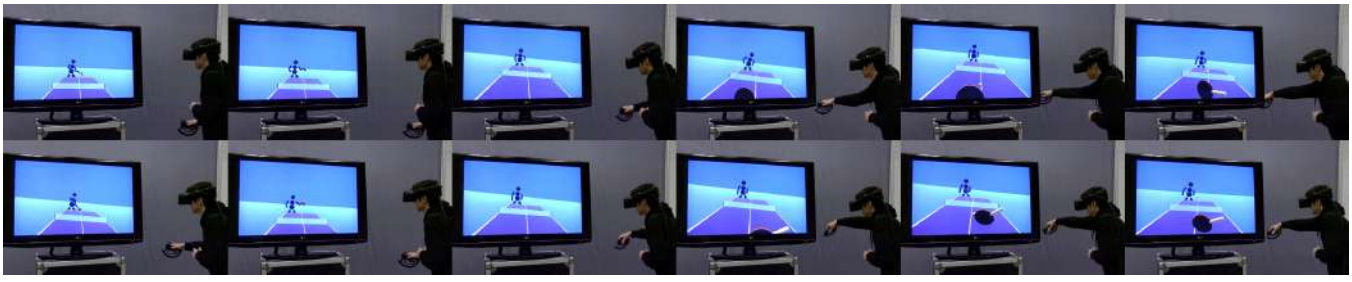

Figure 10: Human-agent interaction screenshots. A human controls a simulated paddle and the agent is simulated and controlled by our method.

基于物理的乒乓球动画的策略与技能学习

Jiashun Wang
jiashunw@cmu.edu
Carnegie Mellon University
USA

Jessica Hodgins
jkh@cmu.edu
Carnegie Mellon University
USA
The AI Institute
USA

Jungdam Won
jungdam@imo.snu.ac.kr
Seoul National University
South Korea

图 1: 左侧，两个人工物理模拟体在竞争比赛中进行对抗，由我们的策略和技能控制器控制。中间面板展示了五项已学会的技能：正手击球、推球与扣杀、反手击球与推球，展示了技能控制器执行多种技能的能力。右侧，一名用户通过虚拟现实与模拟代理进行互动。

摘要

最近，基于物理的角色动画在深度学习的帮助下取得了进展，能够生成灵活自然的运动，使角色能够执行诸如后空翻、拳击和网球等动作。然而，像人类一样在动态环境中选择和运用多样的运动技能以解决复杂任务仍然是一项挑战。我们提出了一种用于基于物理的乒乓球动画的策略和技能学习方法。我们的方法解决了模式崩溃的问题，即角色未能充分利用其执行复杂任务所需的运动技能。更具体而言，我们展示了一个层次控制系统，用于多样化技能学习，以及一个策略学习框架，用于有效决策。通过与最先进的方法进行比较分析，我们展示了我们方法的有效性，证明其在乒乓球中执行各种技能的能力。我们的策略学习框架通过在虚拟现实中的代理-代理交互和人类-代理交互得到了验证，能够处理竞争和合作任务。

SIGGRAPH Conference Papers'24, July 27–August 01,2024, Denver, CO, USA

ACM ISBN 979-8-4007-0525-0/24/07
https://doi.org/10.1145/3641519.3657437

CCS CONCEPTS

• **Computing methodologies** → **Physical simulation**; • **Computing methodologies** → **Motion processing**;

KEYWORDS

角色动画, 基于物理的角色, 深度强化学习, 多角色互动, 虚拟现实, 乒乓球

ACM Reference Format:
Jiashun Wang, Jessica Hodgins, and Jungdam Won . 2024. **基于物理的乒乓球动画的策略与技能学习. In *Proceedings of conference title from your rights confirmation emai (SIGGRAPH Conference Papers'24).* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3641519.3657437**

1 引言

将深度学习与基于物理的角色动画相结合，显著提高了生成灵活和自然运动的能力，增强了角色在复杂环境中的真实感。为了提高这些角色的多功能性，确保它们的技能能够在与训练数据不完全匹配的环境或条件中重复使用至关重要。为实现这一目标，近期的方法集中于学习可重用的技能嵌入。这些方法通常分为两个阶段进行训练。最初，角色通过模仿参考动作学习各种技能嵌入。然后，在任务训练阶段，它们应用这些技能来完成多样的任务。这些方法在各种环境中生成自然运动方面表现出了显著的成功。

然而，当技能之间的差异微妙时，这些方法在任务训练阶段通常会遭遇模式崩溃（mode collapse）。具体来说，尽管代理（即角色）可以在模仿阶段学习各种技能，但在下游任务中，它们往往只使用有限的技能集合，忽视了在模仿阶段学习到的技能多样性。因此，模式崩溃限制了代理在需要多样技能的场景中的潜力。模式崩溃还限制了在强化学习（RL）训练中的探索，导致任务表现次优。

另一个相对未被探索的话题与代理的决策策略有关，特别是它们在应对任务需求时动态制定技能选择和相关技能目标的决策策略的能力。大多数以前的研究要么没有要求多样的技能集合，要么依赖人类用户手动为代理确定技能。代理通常没有能力采用不同的策略来适应复杂和动态的环境。

我们的研究提出了一种学习方法，以增强物理模拟代理的技能和战略决策能力。首先，我们开发了一个分层技能控制器，使代理能够快速利用不同的乒乓球技能并在它们之间切换。该控制器有效地解决了任务训练中的模式崩溃问题。其次，我们开发了一种策略学习方法，使代理能够明确选择和利用特定技能，以适应不同类型的互动，无论是竞争还是合作。结果的概述见于图 1。

我们通过两个互动环境展示了我们方法的有效性：两个模拟代理之间进行的乒乓球比赛和虚拟现实（VR）中人类与模拟代理之间的比赛。在代理-代理环境中，与之前技术预测的结果相比，代理在模拟乒乓球比赛中展示了更好的技能多样性和决策策略。在人类-代理互动环境中，我们评估了人类与代理之间实时互动中的合作和竞争场景。这些环境不仅验证了我们的方法，还为未来研究复杂的代理行为和人类-代理动态提供了平台。本论文的代码和数据可以在https://jiashunwang.github.io/PhysicsPingPong/找到。

我们总结本论文的贡献如下：

- 一个分层技能控制器，使物理模拟代理能够明确执行各种技能，支持快速技能转换。一个交互学习框架设计用于创建决策策略，使代理能够持续学习和适应，与其他代理和人类在动态环境中的竞争或合作需求。
- 新颖的结果展示了我们的学习框架在两个场景中生成智能决策和自然运动的能力：在模拟环境中的代理-代理互动和在 VR 环境中的人类-代理互动。代理-代理环境是开发和测试竞争和合作算法的平台，而 VR 环境则允许自然的人类-代理互动。

2 相关工作

我们回顾了在基于物理的角色动画中与可重用技能和多角色动画最相关的工作。我们审视了技能之间过渡的研究，因为我们正在开发一种技能选择和过渡的方法。我们还讨论了在虚拟现实（VR）中人类与代理交互的相关研究。

2.1 基于物理的角色动画

将物理法则纳入角色动画中，可以开发出生成更真实行为的控制器 [Hodgins et al. 1995; Laszlo et al. 1996]。优化技术，如轨迹优化 [de Lasa et al. 2010; Mordatch et al. 2012; Yin et al. 2008] 和基于采样的方法 [Liu et

al. 2016, 2010] 已被广泛探讨。最近，深度强化学习（DRL）被证明能显著增强控制能力 [Liu and Hodgins 2017; Peng et al. 2017]。由于其灵活性和易用性，DRL 方法消除了设计复杂目标函数的需求，同时提供了卓越的结果，因此吸引了大量研究兴趣。

自从 Peng 等人于 2018 年提出基于深度强化学习（DRL）的方法以来，数据驱动的方法在基于物理的角色动画研究中变得普遍。该思想已扩展到处理更大数据集的情况 [Bergamin et al. 2019; Won et al. 2020]，并允许对现有状态转换进行重组 [Peng et al. 2021]。最近，越来越多的关注被放在可重用的运动技能上。其理念是学习参考动作的潜在空间，然后将学习到的空间用于下游任务。已经研究了多种潜在模型，例如自回归的编码器-解码器 [Merel et al. 2019; Won et al. 2021]、球形嵌入 [Dou et al. 2023; Peng et al. 2022; Tessler et al. 2023]、条件变分自编码器（VAE）[Won et al. 2022; Yao et al. 2022] 和向量量化 VAE [Zhu et al. 2023]。一些研究人员还提出了部分模型，以最大化参考动作的可重用性 [Bae et al. 2023; Xu et al. 2023]。

我们的系统旨在用于乒乓球比赛，涉及两个玩家（即代理）。两个或更多的代理主要采用运动学方法创建 [Kwon et al. 2008; Liu et al. 2006; Shum et al. 2008, 2012; Wampler et al. 2010]。最近有两种方法 [Won et al. 2021; Zhu et al. 2023] 展示了物理模拟拳击的示例。Zhang 等人 [2023] 构建了一个系统，从广播视频中学习网球技能，并生成与镜像对手的拉锯战。在他们的方法中，首先利用基于运动学的动作生成，然后进行基于物理的跟踪，依赖残余力量和额外的手臂控制以成功击球。技能和目标选择不是通过学习而是手动或随机执行，以创建包含两个玩家的场景。相比之下，我们的方法不仅学习灵活而精确的运动控制以击打球，还基于对手和球的运动选择技能和目标的策略。

2.2 技能过渡

基于选项的方法 [Bagaria 和 Konidaris 2020; Jain 等人 2021; Klissarov 等人 2017; Konidaris 和 Barto 2009; Sutton 等人 1999] 将技能表示为选项，这些选项按顺序构建，每个选项的执行使代理能够执行下一个选项。Lee 等人 [2019] 提出了学习额外的过渡策略来连接基本技能，并引入接近度预测器，根据适合下一技能初始状态的接近度提供奖励。

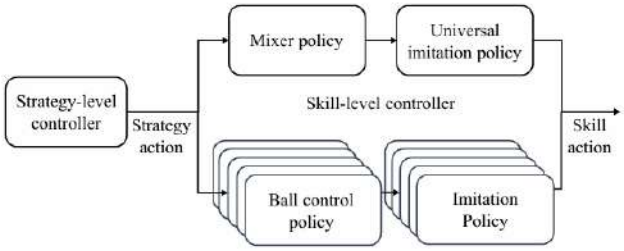

图 2: 我们方法的概述。策略动作包括技能指令和球的目标落地点。技能动作包括 PD 控制器的目标关节角度，这些角度从模仿策略的输出中混合生成。

Lee 等人 [2021] 通过终端状态正则化解决了在链式执行长时间任务时不同技能之间的过渡难题。行为树也是一种常用的方法，用于规划不同状态之间的过渡 [Cheng 等人 2023; French 等人 2019; Marzinotto 等人 2014]。这些方法通过确保前一阶段的终端状态接近下一阶段的初始状态来实现技能转换。尽管这些方法适用于时间不敏感的任务，但乒乓球涉及高速运动和快速反应，因此存在挑战，因为玩家不总是从明确的初始状态击球。

2.3 人机交互

研究集中于在虚拟现实（VR）中进行的人类运动训练 [Liu 等人 2020; Pastel 等人 2023]。然而，这些研究通常缺乏物理模拟的对手。也有一些商业游戏允许人们在 VR 中与代理进行体育互动，如拳击、高尔夫和羽毛球。Eleven Table Tennis [2016] 是一个基于 VR 的乒乓球游戏，类似于我们构建的游戏，允许人类与代理对战。然而，该游戏的代理仅模拟了漂浮的头部和球拍，而没有全身动态。借助 GPU 加速的模拟技术和我们的控制算法，我们得以创建一个具有全身动态的物理模拟代理，能够实时与人类对战。另一个相关领域涉及利用扩展现实技术，通过人类参与的方法 [Brenneis 等人 2021; Li 等人 2022; Seo 等人 2023; Wang 等人 2023]

增强代理能力。我们的工作区别于以往的研究，通过将人类和代理带入统一的环境中，实现了双向的物理互动，使他们能够进行合作和竞争。

3 方法概述

我们提出了一种分层方法，包括策略层控制器和技能层控制器。策略层控制器将代理、对手和球的状态作为输入，并输出策略动作，其中包括所要使用的技能以及球的目标落地点。同时，技能层控制器将代理和球的状态以及策略动作作为输入，然后生成技能动作，其中包括 PD 控制器的目标关节角度。图 2 展示了我们方法的概述，图 3 展示了我们方法的架构。

4 技能层控制器

训练我们的技能层控制器需要三个阶段。首先，我们使用动作捕捉数据训练模仿策略。然后，学习每项技能的球控制策略，使代理能够使用相应的模仿策略回击球。最后，我们学习一种策略，使代理能够顺序执行各种技能，同时在它们之间进行合理的过渡。我们将这种策略称为混合策略（mixer policy）。一旦技能层控制器训练完成，代理就能够熟练且连续地执行各种技能，将球发送到不同的目标位置。

4.1 模仿策略

我们首先将动作捕捉数据集分类为五个子集，对应于每种技能。这个细分允许我们训练特定于技能的模仿策略。我们还利用所有数据训练一个通用模仿策略。模仿策略表示为 $\pi^i(a^i|s,z^i)$，其中 $i \in \{1,2,3,4,5,u\}$，1 到 5 是不同技能的索引，而 u 是通用模仿策略的索引。z^i 是从超球面分布中采样的潜变量，s 是代理的状态。模仿策略的目标是输出一个动作 a^i，使得模拟的运动与参考运动相似。因此，每个特定技能的模仿策略生成的运动与其对应的参考运动在各自的技能子集中相似，而通用模仿策略生成的运动则涵盖整个动作捕捉数据集。在后期解决特定任务时，使用经过各种动作训练的单一通用模仿策略往往会导致模式崩溃问题。代理未能充分探索可用的各种技能，而是重复非常有限的技能，导致任务表现仍然次优。我们的控制器设计受到混合专家模型的启发，以缓解这个问题。每个模仿策略 $\pi^i(a^i|s,z^i)$ 是通过对抗框架 ASE [Peng et al. 2022] 构建的，其中策略被更新以欺骗运动判别器 D^i。动作捕捉数据集中存在的过渡 $d_{M^i}(s,s')$ 用作正样本，而由策略 π^i 生成的过渡 $d_{\pi^i}(s,s')$ 则用作负样本。通过最小化以下公式来训练判别器：

$$\min_{D^i} -\mathbb{E}_{d_{M^i}}(s,s')\log(D^i(s,s'))-\mathbb{E}_{d_{\pi^i}}(s,s')\log(1-D^i(s,s'))$$
$$+\lambda_{gp}\mathbb{E}_{d_{M^i}}(s,s')\|\nabla_\phi D^i(\phi)|_{\phi=(s,s')}\|^2, \quad (1)$$

最后一项是带有常数因子 λ_{gp} 的梯度惩罚正则化。我们训练编码器 q^i 以促进状态转移 (s,s') 与潜在变量 z^i 之间的对应关系。编码器被建模为 von Mises-Fisher 分布，并通过最大化其对数似然来进行训练：

$$\max_{q^i} \mathbb{E}_{p(z^i)}\mathbb{E}_{d^{\pi^i}(s,s'|z^i)}[\log q^i(z^i|s,s')],$$
$$q^i(z^i|_{S,S'}) = \frac{1}{Z}exp(\mu_{q^i}(s,s')^T z^i) \quad (2)$$

其中 $\mu_{q^i}(s,s')$ 是分布的均值，Z 是归一化常数。给定一个判别器 D^i，训练 π^i 的奖励定义为：

$$r_t = -\log(1-D^i(s_t,s_{t+1})) + \beta\log q^i(z_t^i|s_t,s_{t+1}). \quad (3)$$

其中 β 是相对权重。此外，包含一个多样性项，以鼓励不同的潜在变量表示不同的动作。综合所有内容，π^i 的目标变为:

$$\max_{\pi^i} \mathbb{E}_p(Z)\mathbb{E}_{p(\tau|\pi^i,Z)}[\sum_{t=0}^{T}\gamma^t(r_t)] \quad (4)$$
$$-\lambda_{D^i}\mathbb{E}_{d^{\pi^i}}\mathbb{E}_{z_1^i,z_2^i\sim p(z^i)}[(\frac{D_{KL}(\pi^i(\cdot|s,z_1^i),\pi^i(\cdot|s,z_2^i))}{0.5(1-z_1^i z_2^i)})^2],$$

其中 $D_{KL}(\cdot\|\cdot)$ 衡量两个分布之间的 KL 散度，z_1^i 和 z_2^i 指代两个不同的潜变量，γ 是折扣因子，λ 是用于平衡权重的常数。

4.2 球控制策略

一旦代理能够模仿每项技能 $i \in \{1,2,3,4,5\}$，我们便训练球控策略 $\omega^i(z^i|s,b,y)$，使得代理能够击打并移动

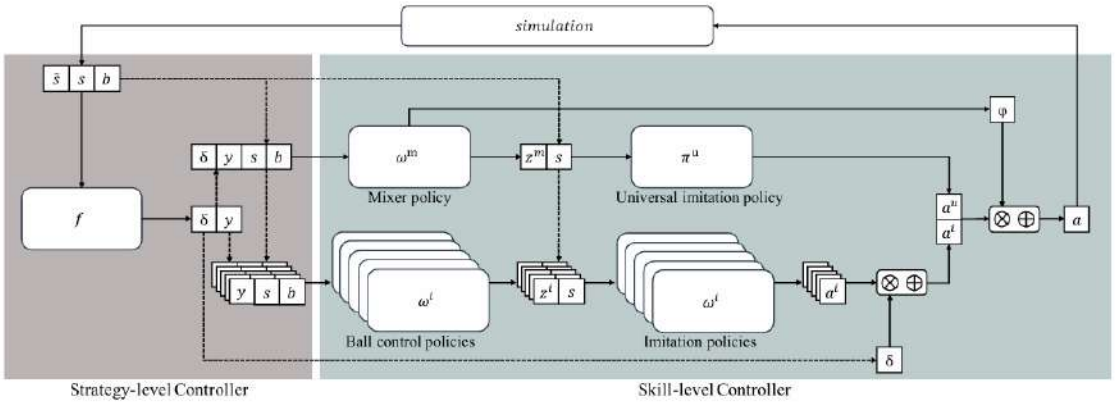

图 3: 我们的方法架构。我们通过模仿策略、球控制策略，最终是混合策略的阶段来训练技能级控制器。在技能级控制器准备好并且其权重被冻结后，我们再训练策略级控制器。符号 ⊗⊕ 表示公式 8 中的加权和。

从随机位置发出的球，使其达到目标位置。其中，s 表示代理的状态，b 表示球的状态，y 是球的目标落地点。任务奖励 r 是三个部分的组合：拍击奖励 r_p、球的奖励 r_b 和风格奖励 r_s，

$$\mathrm{r}(t) = w_p r_p(t) + w_b r_b(t) + w_s r_s(t), \tag{5}$$

其中 w_p、w_b 和 w_s 是相对权重。拍击奖励 r_p 鼓励代理将拍子定位在靠近球的位置。该奖励定义为：

$$r_p(t) = \begin{cases} \exp(-4||x_p(t) - x_b(t)||^2), \text{ if } C_{bp}(t) = 0, \\ 0, \text{ otherwise}, \end{cases} \tag{6}$$

其中 $x_p(t)$ 和 $x_b(t)$ 分别表示拍子和球的位置，$C_{bp}(t)$ 是一个表示接触状态的二进制变量。当 $C_{bp}(t)$=0 时，表示球在时间 t 前尚未接触拍子；当 $C_{bp}(t)$=1 时，表示球在时间 t 接触了拍子，或在 t 之前已经接触过拍子。每次新的球发射时，该值将重置为 0。球的奖励 r_b 义如下：

$$r_b(t) = \begin{cases} 1 + \exp(-4||x_c(t) - x_t(t)||^2), \\ \text{if } C_{bp}(t) = 1 \text{ and } C_{bt}(t) = 0, \\ 0, \text{ otherwise}, \end{cases} \tag{7}$$

其中 $x_t(t)$ 是球的目标落地点，$x_c(t)$ 表示预期的球在桌面的落地点，该落地点通过牛顿运动方程（即二次轨迹）计算，状态对应于球的当前位置和速度。$C_{bt}(t)$ 是一个用于检查球与桌子接触历史的二进制变量，其更新方式类似于 $C_{bp}(t)$。当代理成功击中球且球朝向目标位置移动时，将获得最高奖励。此外，我们还应用了风格奖励 $r_s = -\log(1 - D^i(s_t, s_{t+1}))$ 来进行任务训练，与 ASE 方法（Peng et al. 2022）类似，其中 D^i 是在前一阶段训练时学习的判别器。

4.3 混合策略

虽然我们的代理可以使用球控策略和相应的模仿策略来打乒乓球，但其能力局限于重复单一技能。在游戏过程中简单地从一个控制器切换到另一个控制器，往往会因为一个技能的结束状态与下一个技能的开始状态不匹配而导致失败。为了在不同技能之间创建合理的过渡，我们学习了一个混合策略 $\omega^m(z^m|s, b, \delta, y)$，它以代理的状态 s、球的状态 b 和策略动作（δ 和 y），其中 δ 是确定所用技能的独热向量，y 是球的目标落地点。然后，它生成用于通用模仿策略 π^u 的潜变量以及一组混合权重 φ，以逐关节方式混合技能动作。换句话说，φ 决定了在过渡和五种不同技能之间，代理依赖于哪种策略。用于 PD 控制器的目标关节角度计算如下：

$$a = \varphi \odot \pi^u(\cdot|s, z^u) + (1 - \varphi) \odot \sum_{i=1}^{5} \delta_i \pi^i(\cdot|s, z^i) \tag{8}$$

其中，$\delta = (\delta_1, \delta_2, \delta_3, \delta_4, \delta_5)$ 是一个指示所选技能的独热向量。在训练混合策略时，要求智能体在随机发球、随

机选择技能和随机目标位置的条件下执行控球任务。用于学习控球策略的奖励机制同样适用于此，且所有其他策略的权重保持不变。

Algorithm 1 Strategy learning

Input: Number of iterations N, interaction environment Env

Output: Updated policy f

$f \leftarrow$ Random initialization

for $i \leftarrow 1$ to N **do**

$\{(o_k^{\text{expert}}, c_k^{\text{expert}})\}_{k=1}^{K} \leftarrow \text{Interact}(Env, f)$

Apply stochastic gradient descent to update f using Equation 9

end for

5 策略级控制器

策略级控制器通过迭代行为克隆进行开发，灵感来自[Oh et al. 2018]。更具体地说，我们首先在智能体-智能体对战或人类-智能体与 VR 的交互中，通过随机采样策略动作来收集交互数据。然后使用这些数据来更新策略级控制器，并通过使用最新的策略级控制器收集新的交互数据重复该过程。在收集交互数据时，有两种选择：竞争和合作。为了训练竞争策略，我们选择导致胜利的数据，而在合作策略中，我们选择对手成功接住球的序列。

策略级控制器反复生成技能索引和目标落地点，以满足不同应用的需求。更具体地说，策略级控制器 f 将策略观测 $o = (s, \tilde{s}, b)$ 作为输入，其中 s、$tildes$ 和 b 分别是智能体状态、对手状态和球的状态，输出策略动作 $c = (\delta, y)$，其中 δ 是确定所用技能的独热向量，y 是球的目标落地点。当球开始从对手移动到智能体时，策略动作会更新。为了有效学习策略级控制器，我们采用带有迭代优化的行为克隆方法，旨在从现有的专家演示 $\left\{(\mathrm{o}_k^{\text{expert}}, \mathrm{c}_k^{\text{expert}})\right\}_{k=1}^{K}$ 中学习策略（见算法 1）。作为控制器的结构，我们利用条件变分自编码器（CVAE）来建模体育比赛中固有的随机性。在训练过程中，CVAE 编码器以 o 和 c 为输入，生成后验高斯分布 $Q(u|\mu, \sigma^2)$ 的均值 μ 和方差 σ^2。然后，我们从该分布中采样潜在变量 u，并将其与观测 o 连接作为解码器的输入，解码器重构动作 c'。训练损失定义为：

$$\sum_{k=1}^{K} ||c_k^{\text{expert}} - c_k'|| + \beta_{KL} D_{KL}(Q(u|\mu_k, \sigma_k^2)||\mathcal{N}(0, I)), \quad (9)$$

其中 KL 散度 $D_{KL}(\cdot||\cdot)$ 测量两个分布之间的 KL 散度，而 β_{KL} 是相对权重。在推理过程中，仅使用解码器，它接受一个随机采样的潜在变量 u 和观测值 o，然后生成策略动作，引导智能体执行相应的技能。如果对手成功地回球，则该过程会重复。我们从两个不同的交互环境（算法 1 中的 Env）收集专家演示。每个环境的具体细节将在第 6 节中解释。

6 交互环境

我们介绍了为了验证策略学习方法而构建的智能体-智能体和人类-智能体交互环境。

智能体-智能体交互环境是一个两个虚拟智能体互相打乒乓球的环境（图 1 左列）。我们将其中一个智能体称为我们的智能体，另一个称为对手。在为我们的智能体学习策略级控制器的过程中，对手使用一个固定的启发式策略级控制器，而我们的智能体的控制器则是迭代更新的。更具体地说，我们让我们的智能体和对手使用各自的策略级控制器互相对战，收集这些演示数据，然后用它们来更新我们智能体的控制器。如果我们的目标是学习一个能够击败对手的竞争性策略，我们会有选择地使用那些导致获胜的演示数据。另一方面，如果我们想学习一个合作性策略，则会使用那些对手成功回球的演示数据。在我们的系统中，我们使用两种类型的启发式策略级控制器：随机策略和视频策略。随机策略从均匀分布中随机选择技能和目标落点位置。视频策略是通过使用广播视频构建的。我们从现有的广播视频中提取专家演示数据 $\left\{(\mathrm{o}_k^{\text{video}}, \mathrm{c}_k^{\text{video}})\right\}_{k=1}^{K}$ 共 20 分钟），然后使用行为克隆方法训练一个条件变分自编码器（CVAE）。

人类-智能体交互环境允许人类用户与虚拟智能体进行互动。在我们的系统中，用户通过使用虚拟现实设备（包括头戴显示器和手持控制器）与智能体互动（见图 1 右列）。虚拟现实界面通过 Unity 操作，而

基于物理的仿真则在 Isaac Gym 中运行【Makoviychuk et al.2021】。为了使模拟智能体能够与人类用户进行互动，我们物理模拟了用户的球拍，并通过虚拟现实界面发出的信号控制其位置和方向。具体来说，在球拍控制方面，我们使用虚拟现实手持控制器的笛卡尔坐标姿态 q_{user} 和模拟球拍的姿态 q_{sim} 来计算目标速度 $\dot{q}_{\text{target}} = (q_{\text{user}} - q_{\text{sim}})/\Delta t$，其中 Δt 是仿真步骤。我们将这个目标速度作为输入，传递给模拟器提供的速度控制器。在可视化方面，Unity 将模拟智能体、用户的球拍和球的状态作为输入，并使用可视化资源进行渲染。与之前的研究（发送立体图像）相比，这种实现显著减少了信息交换量【Seo et al,2023】。通过将人类用户视为对手，智能体的策略级控制器可以通过与智能体-智能体交互环境相同的流程来构建。

7 实验

我们通过运动质量和任务表现来评估技能级控制器。我们通过检查策略级控制器在智能体-智能体和人类-智能体交互环境中的有效性，评估其在竞争和合作场景中的表现。

7.1 技能评估

我们从运动质量和任务表现两个方面评估技能表现。运动质量的评估衡量在给定所需技能指令时，生成的动作的自然性以及智能体是否执行了正确的技能。任务表现的评估则衡量打乒乓球的整体熟练度。我们将我们的方法与两种最先进的方法 ASE【Peng et al.2022】和 CASE【Dou et al.2023】进行比较，并与一个显式过渡模型（ET）进行对比，后者是我们方法的一种变体，移除了模型中的混合策略 ω^m。我们训练了一个显式控制器来处理技能过渡，当球越过网并直到被智能体回球时，该控制器接管控制。该控制器也使用球控制模仿架构构建。我们方法与 ET 之间的主要区别在于，我们的方法在每个时间步提供与所选技能动作的连续动作混合，而 ET 则没有。

7.1.1 **运动质量**. 我们设计了三个指标来评估运动质量，特别是评估自然性和模式崩溃。第一个指标是鉴别器得分，它衡量当前击打动作与第 i 个目标技能的参考动作的相似度。因为我们有五种技能，所以我们为每种技能训练了一个鉴别器 D^i_{test}，并使用以下公式来计算得分：

$$\text{Discriminator Score i} = \frac{1}{T}\sum_{t=0}^{T-1} -\log(1 - D^i_{test}(s_t, s_{t+1})), \tag{10}$$

其中 T 是单个击打动作的长度。训练 D^i_{test} 测试的细节将在附录 D 中介绍。第二个指标是技能准确度，用于衡量代理在给定目标技能命令时是否执行了正确的技能。具体来说，给定一个动作序列，我们首先通过选择提供最高值的鉴别器的索引来对其进行分类。然后，我们将其与目标技能命令进行比较以计算准确度。第三个指标是多样性得分，旨在测试驱动和推动命令的动作是否足够区分。在乒乓球中，每个技能类别内的动作（例如，正手快攻与正手推挡）可能表现出微妙的差异，尽管它们在游戏中的角色可能截然不同。多样性得分衡量区分视觉上相似动作的能力。它是通过以下方式计算的：

$$\text{Diversity Score} = \frac{1}{2N^2}\sum_{i\in\{1,3\}}\sum_{m=1}^{N} ||s^i_m - s^{i+1}_n||, \tag{11}$$

其中 s^i 是代理在技能命令 i 下击球的状态。具体来说，$i \in \{1,3\}$ 代表正手攻球和反手攻球，而 $i+1 \in \{1,3\}$ 分别代表正手推挡和反手推挡。N 是每个技能命令的总击球次数。我们仅在代理的球拍与球接触的时刻计算此得分。前两个指标评估一般性的技能模式崩溃问题，例如，在被要求使用反手时使用正手动作。第三个指标专门设计用来衡量代理是否具有准确执行攻球和推挡技能的能力。

评估结果在表 1 中报告，其中值是使用 10,000 个随机向配备相应技能控制器的代理发射的球计算得出的。对于鉴别器得分，我们的方法显著超过了 ASE 和 CASE，并且比 ET 高出 15.6% 的得分。这些结果证明了我们的方法生成的动作与参考目标技能最为相似。正如技能准确度结果所示，我们的方法在大多数情况下使用正确的技能击球（表 1 中的 0.76）。而 ASE 和 CASE 仅以 0.38 和 0.47 的准确度使用正确的技能。在

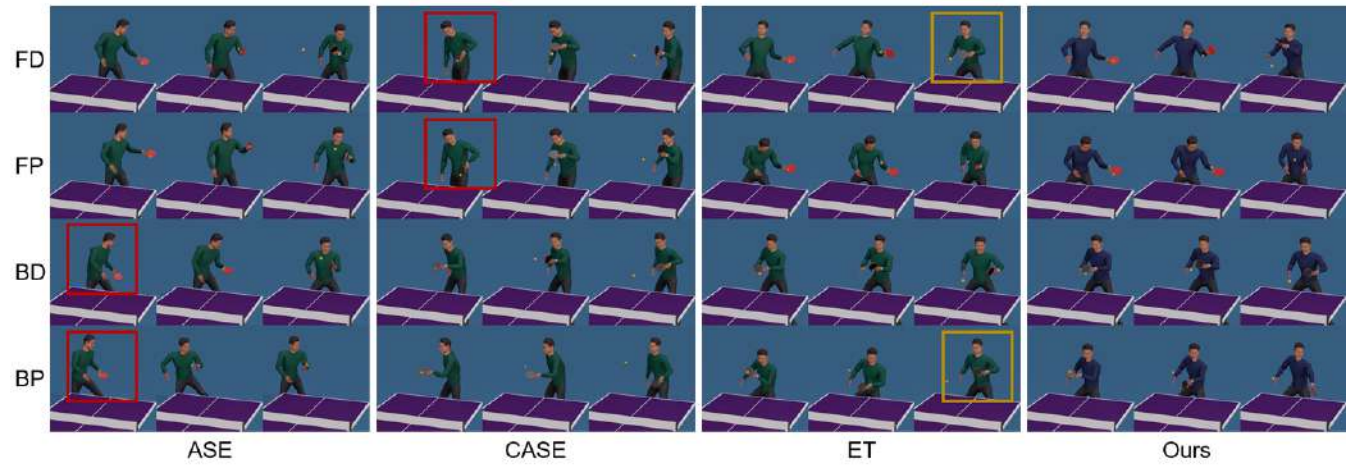

图 4: 与其他方法的比较，包含四个技能命令。ASE 和 CASE 可能会使用错误的技能，如红框所示。ET 可能会提前终止以返回准备姿势，如黄框所示。

图 5: 仅使用正手和反手击球控制器的过渡结果。这两个控制器都是通过从动作捕捉数据中随机初始化配置进行训练的。如红框所示，代理在下一球发射之前尝试使用另一个正手击球，这导致它未能及时切换回反手击球。

多样性得分方面，我们的方法比 ASE、CASE 和 ET 分别高出 30.7%、32.3% 和 9.4%。我们还在图 4 中展示了定性比较。我们发现 ASE 和 CASE 在被要求使用反手技能时经常使用正手技能，反之亦然，如图 4 中的红框所示。我们没有观察到任何正手扣杀技能。即使使用了正确的技能，自然性仍然不足。ET 经常不完成技能；相反，技能提前终止以返回准备姿势，如图 4 中的黄框所示。ASE 和 CASE 经常忽视技能命令，倾向于使用相对较少的技能。这种错误发生是因为，在任务训练期间，这些方法陷入了模式崩溃，使得有效探索各种技能变得困难。相比之下，我们的方法利用了专家混合的思想来避免这个问题。我们进一步测试了没有任何过渡设计的单个技能控制器的使用。每个技能控制器都是用从运动捕捉数据中随机初始化的配置进行训练的。如图 5 所示，在执行正手攻球后，代理在下一个球发射前尝试另一个正手攻球——这是单技能控制器的典型行为。这种不必要的动作阻止了它及时切换到反手攻球，最终导致漏球。

7.1.2 **任务表现**. 为了评估技能控制器的任务表现，我们评估两个方面：持续性和准确性。持续性由成功连续回球的平均次数决定，而准确性则通过目标落点与实际接触点在桌面上的平均距离（以米为单位）来衡量。除了在训练分布上进行测试外，我们还收集了一些来自高水平选手比赛的球轨迹更快的球跟踪数据，并评估每种方法是否能在球跟踪数据的测试集上表现良好。我们在表 2 中报告了评估结果。括号中的数字是微调实验的结果。我们的方法能够实现最多的平均击球次数和第二好的准确性。尽管 ET 能够实现更高的准确性，但它的持续性不足，特别是在应对更具挑战性的球时。由于显式的过渡设计，它经常没有足够的时间来响应下一个球，仅实现了平均 3.66 次击球。

7.1.3 **混合策略中的混合权重**. 混合策略的混合权重。我们测试了具有不同技能的代理击球，并在图 8 中可视化了肩部、肘部和腕部关节的平均混合权重 φ。我们可以观察到，混合策略的权重通常在球拍接触球的时刻最低，在不同技能之间的转换前后较高。这表明在击球时依赖于预训练的球控制策略，在技能转换期间依赖于混合策略。

表 1: 鉴别器得分、技能准确度和多样性得分的比较。

	ASE	CASE	ET	Ours
Discriminator Score	1.62	2.28	4.95	**5.72**
Skill Accuracy	0.38	0.47	0.69	**0.76**
Diversity Score	6.13	6.05	7.32	**8.01**

表 2: 任务表现评估。我们的方法能够实现最长的平均命中次数和第二好的准确率。

	ASE	CASE	ET	Ours
Avg Hits	9.54 (5.94)	8.79 (5.28)	6.55 (3.66)	**10.93 (6.28)**
Avg error	0.28 (0.33)	0.35 (0.39)	**0.25 (0.28)**	0.26 (0.31)

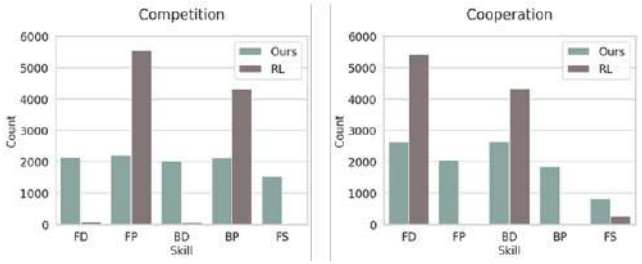

图 6: 我们方法和强化学习（RL）的技能命令分布。

7.2 代理-代理互动评估

我们评估了在代理-代理互动环境中，学习到的策略在竞争和合作设置下的表现。竞争策略旨在开发一个能够比对手取得更高胜率的代理。合作策略则开发一个能够与对手温和配合，以增加对打持续时间的代理。作为基线，我们通过强化学习（RL）学习了一种策略。有关训练 RL 基线的详细信息，请参阅附录 E。我们

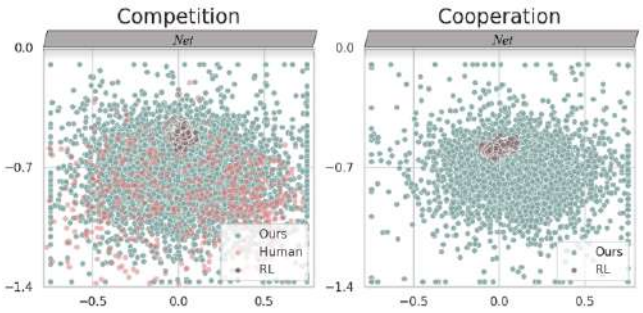

图 7: 我们方法、强化学习（RL）和人类的目标着陆位置。

的方法和 RL 基线通过与两种类型的对手进行对战进行比较：随机策略对手和第 6 节中介绍的视频策略对手。每次评估计算基于 10,000 个回合。表 3 显示了竞争和合作设置下的胜率和平均回合数。我们的策略学习算法在竞争设置中可以实现更高的胜率，并且在合作设置中可以与两种对手保持更长的对打时间。

表 3: 策略评估。我们报告了竞争设置下的胜率和合作设置下的平均回合数。

	Competition		Cooperation	
	RL	Ours	RL	Ours
Random op	0.641	**0.687**	14.9	**16.4**
Video op	0.637	**0.681**	15.6	**18.2**

在图 6 和图 7 中，我们可视化了来自策略政策的技能命令直方图和目标着陆位置。在图 7 中，我们还提供了来自真实玩家在竞争性比赛中的球着陆位置。我们观察到，我们的方法在着陆位置的分布上与人类更为相似，而与强化学习（RL）相比，RL 趋向于使用较少多样的技能命令，并且只击打到桌子上的小区域。相比之下，我们的方法在整个游戏过程中使用了多种技能和目标位置。我们还在图 9 中包含了定性游戏玩法的可视化。此外，我们让 RL 和我们的方法相互竞争，并在表 4 中报告了胜率。每种方法都有两个策略政策，分别针对两个对手进行训练，因此我们共有四场比赛。

由于 RL 陷入局部最小值并对特定对手过拟合，我们的方法取得了远高于 RL 的胜率。

7.3 人类-代理互动评估

在学习人类-代理交互环境的策略之前，我们首先使用人类用户与配备原始技能控制器的代理互动的游戏数据进行技能级别控制器的微调。之所以需要微调，是因为模拟代理所经历的情境与真实人类用户在虚拟现实（VR）环境中的互动风格之间存在领域差距。在微调技能控制器后，我们通过与代理-代理交互环境相似的程序来学习策略。对于训练竞争策略，我们使用导致代理赢得分数的示范，从而为人类对手提供更具挑战性的回报。相反，训练合作策略时，我们使用示范让人类能够维持对打，强调对人类更容易的球回击。这些示范作为算法 1 中的专家示范。我们在表 5 中报告了代理的胜率和用户与代理之间的平均击球数。当使用初始策略时，代理的胜率为 64%，且平均对打为 4.04 次击球。经过两次竞争策略的精炼后，代理的胜率提高到 78%，而平均击球数降至 3.75。对于合作策略，胜率降至 58%，用户的平均对打为 5.34 次击球。这些结果表明，我们的策略学习算法在人类-代理交互环境中也同样有效。我们在图 10 中提供了人类-代理实时游戏视频的截图。

表 4: 我们方法与 RL 之间的胜率对比。括号中的对手为策略政策训练期间的对手。

	Ours (random op)	Ours (video op)
RL (random op)	0.45 vs 0.55	0.47 vs 0.53
RL (video op)	0.42 vs 0.58	0.42 vs 0.58

表 5: 人类-代理交互评估。

	Initial policies	Competition	Cooperation
Winning rate	0.64	0.78	0.58
Avg hits	4.04	3.75	5.34

8 讨论与结论

尽管我们的方法生成的代理在竞争性和自然性方面表现良好，但它仍然存在一些局限性。首先，尽管为每个技能构建单独的策略并通过混合策略将它们结合起来，显著提高了生成的运动质量和任务表现，但我们的模型在面对包含数百种不同技能的数据集时，扩展性较差。开发一个将我们的方法与从无标签运动中学习的模型结合的混合模型，以实现高运动质量和良好的扩展性，将是一个有趣的未来研究方向。其次，由于我们的方法是数据驱动的，捕获的运动质量显著影响最终的运动质量。例如，玩家倾向于使用大幅度的手臂动作，这种运动风格也出现在我们的结果中。然而，在比赛中，使用较少的手臂动作可能是节省体力的一种方式，而且隐蔽的动作也能迷惑对手。最后，尽管我们为包括球、玩家和桌子在内的每个组件都采用了刚体仿真，并且球可以旋转，但空气阻力仅通过基于速度的阻尼模型进行建模，而没有考虑到马格努斯效应（Magnus effect），即由于气压差异而导致的球的轨迹弯曲。这一遗漏可能会影响我们动画的真实感以及系统所学习到的最终策略。

在本文中，我们提出了一种基于物理的乒乓球动画学习方法。我们开发了一个层次化的控制器结构，克服了在可重用的基于潜在空间的模型中经常出现的模式崩溃问题。我们的方法不仅提高了整体运动质量，还使我们能够为两种类型的环境——代理-代理和人类-代理交互——学习有效的决策策略。

ACKNOWLEDGMENTS

该工作部分是在王佳顺于 AI 研究所实习期间完成的。Jungdam Won 的研究部分得到了韩国政府（MSIT）资助的信息与通信技术规划评估（IITP）基金 [编号 2021-0-01343-004，人工智能研究生项目（首尔国立大学）] 和首尔国立大学计算机技术研究所（ICT）的支持。我们特别感谢 Murphy Wonsick 在构建 VR 系统方面的帮助，以及 Melanie Danver 对结果呈现的协助。

REFERENCES

[1] 2016. Eleven Table Tennis. https://elevenvr.com

[2] Jinseok Bae, Jungdam Won, Donggeun Lim, Cheol-Hui Min, and Young Min Kim. 2023. PMP: Learning to Physically Interact with Environments using Part-wise Motion Priors. In *ACM SIGGRAPH 2023 Conference Proceedings, SIGGRAPH 2023*. ACM, 64:1–64:10. https://doi.org/10.1145/3588432.3591487

[3] Akhil Bagaria and George Konidaris. 2020. Option Discovery using Deep Skill Chaining. In *8th International Conference on Learning Representations, ICLR 2020*. OpenReview.net. https://openreview.net/forum?id=B1gqipNYwH

[4] Kevin Bergamin, Simon Clavet, Daniel Holden, and James Richard Forbes. 2019. DReCon: data-driven responsive control of physics-based characters. *ACM Transactions on Graphics* 38, 6 (2019), 206:1–206:11. https://doi.org/10.1145/3355089.3356536

[5] Dylan J. A. Brenneis, Adam S. R. Parker, Michael Bradley Johanson, Andrew Butcher, Elnaz Davoodi, Leslie Acker, Matthew M. Botvinick, Joseph Modayil, Adam White, and Patrick M. Pilarski. 2021. Assessing Human Interaction in Virtual Reality With Continually Learning Prediction Agents Based on Reinforcement Learning Algorithms: A Pilot Study. *arXiv preprint arXiv:2112.07774* (2021). https://arxiv.org/abs/2112.07774

[6] Xuxin Cheng, Ashish Kumar, and Deepak Pathak. 2023. Legs as Manipulator: Pushing Quadrupedal Agility Beyond Locomotion. In *IEEE International Conference on Robotics and Automation, ICRA 2023*. IEEE, 5106–5112. https://doi.org/10.1109/ICRA48891.2023.10161470

[7] Martin de Lasa, Igor Mordatch, and Aaron Hertzmann. 2010. Feature-based locomotion controllers. *ACM Transactions on Graphics* 29, 4 (2010), 131:1–131:10. https://doi.org/10.1145/1778765.1781157

[8] Zhiyang Dou, Xuelin Chen, Qingnan Fan, Taku Komura, and Wenping Wang. 2023. C · ASE: Learning Conditional Adversarial Skill Embeddings for Physics-based Characters. In *SIGGRAPH Asia 2023 Conference Papers, SA 2023*. ACM, 2:1–2:11. https://doi.org/10.1145/3610548.3618205

[9] Kevin French, Shiyu Wu, Tianyang Pan, Zheming Zhou, and Odest Chadwicke Jenkins. 2019. Learning Behavior Trees From Demonstration. In *International Conference on Robotics and Automation, ICRA 2019*. IEEE, 7791–7797. https://doi.org/10.1109/ICRA.2019.8794104

[10] 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, SIGGRAPH 1995*. ACM, 71–78. https://doi.org/10.1145/218380.218414

[11] Arushi Jain, Khimya Khetarpal, and Doina Precup. 2021. Safe option-critic: learning safety in the option-critic architecture. *Knowledge Engineering Review* 36 (2021), e4. https://doi.org/10.1017/S0269888921000035

[12] Martin Klissarov, Pierre-Luc Bacon, Jean Harb, and Doina Precup. 2017. Learning Options End-to-End for Continuous Action Tasks. *arXiv preprint arXiv:1712.00004* (2017). http://arxiv.org/abs/1712.00004

[13] George Dimitri Konidaris and Andrew G. Barto. 2009. Skill Discovery in Continuous Reinforcement Learning Domains using Skill Chaining. In *Advances in Neural Information Processing Systems 22*. 1015–1023. https://proceedings.neurips.cc/paper/2009/hash/e0cf1f47118daebc5b16269099ad7347-Abstract.html

[14] Taesoo Kwon, Young-Sang Cho, Sang Il Park, and Sung Yong Shin. 2008. Two-Character Motion Analysis and Synthesis. *IEEE Transactions on Visualization and Computer Graphics* 14, 3 (2008), 707–720. https://doi.org/10.1109/TVCG.2008.22

[15] Joseph Laszlo, Michiel van de Panne, and Eugene Fiume. 1996. Limit Cycle Control and Its Application to the Animation of Balancing and Walking. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1996*. ACM, 155–162. https://doi.org/10.1145/237170.237231

[16] Youngwoon Lee, Joseph J. Lim, Anima Anandkumar, and Yuke Zhu. 2021. Adversarial Skill Chaining for Long-Horizon Robot Manipulation via Terminal State Regularization. In *Conference on Robot Learning, 2021 (Proceedings of Machine Learning Research, Vol. 164)*. PMLR, 406–416. https://proceedings.mlr.press/v164/lee22a.html

[17] Youngwoon Lee, Shao-Hua Sun, Sriram Somasundaram, Edward S. Hu, and Joseph J. Lim. 2019. Composing Complex Skills by Learning Transition Policies. In *7th International Conference on Learning Representations, ICLR 2019*. OpenReview.net. https://openreview.net/forum?id=rygrBhC5tQ

[18] Chengxi Li, Pai Zheng, Shufei Li, Yat Ming Pang, and Carman K. M. Lee. 2022. AR assisted digital twin-enabled robot collaborative manufacturing system with human-in-the-loop. *Robotics and Computer-Integrated Manufacturing* 76 (2022), 102321. https://doi.org/10.1016/J.RCIM.2022.102321

[19] C. Karen Liu, Aaron Hertzmann, and Zoran Popovic. 2006. Composition of complex optimal multi-character motions. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association, 215–222. https://doi.org/10.2312/SCA/SCA06/215-222

[20] Huimin Liu, Zhiquan Wang, Christos Mousas, and Dominic Kao. 2020. Virtual Reality Racket Sports: Virtual Drills for Exercise and Training. In *2020 IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2020*. IEEE, 566–576. https://doi.org/10.1109/ISMAR50242.2020.00084

[21] Libin Liu and Jessica K. Hodgins. 2017. Learning to Schedule Control Fragments for Physics-Based Characters Using Deep Q-Learning. *ACM Transactions on Graphics* 36, 3 (2017), 29:1–29:14. https://doi.org/10.1145/3083723

[22] Libin Liu, Michiel van de Panne, and KangKang Yin. 2016. Guided Learning of Control Graphs for Physics-Based Characters. *ACM Transactions on Graphics* 35, 3 (2016), 29:1–29:14. https://doi.org/10.1145/2893476

[23] Libin Liu, KangKang Yin, Michiel van de Panne, Tianjia Shao, and Weiwei Xu. 2010. Sampling-based contact-rich motion control. *ACM Transactions on Graphics* 29, 4 (2010), 128:1–128:10. https://doi.org/10.1145/1778765.1778865

[24] 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. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. https://openreview.net/forum?id=fgFBtYgJQX_

[25] Alejandro Marzinotto, Michele Colledanchise, Christian Smith, and Petter Ögren. 2014. Towards a unified behavior trees framework for robot control. In *2014 IEEE International Conference on Robotics and Automation, ICRA 2014*. IEEE, 5420–5427. https://doi.org/10.1109/ICRA.2014.6907656

[26] Josh Merel, Leonard Hasenclever, Alexandre Galashov, Arun Ahuja, Vu Pham, Greg Wayne, Yee Whye Teh, and Nicolas Heess. 2019. Neural Probabilistic Motor Primitives for Humanoid Control. In *7th International Conference on Learning Representations, ICLR 2019*. OpenReview.net. https://openreview.net/forum?id=BJl6TjRcY7

[27] Igor Mordatch, Emanuel Todorov, and Zoran Popovic. 2012. Discovery of complex behaviors through contact-invariant optimization. *ACM Transactions on Graphics* 31, 4 (2012), 43:1–43:8. https://doi.org/10.1145/2185520.2185539

[28] Junhyuk Oh, Yijie Guo, Satinder Singh, and Honglak Lee. 2018. Self-Imitation Learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018 (Proceedings of Machine Learning Research, Vol. 80)*. PMLR, 3875–3884. http://proceedings.mlr.press/v80/oh18b.html

[29] Stefan Pastel, Katharina Petri, C. H. Chen, Ana Milena Wiegand Cáceres, M. Stirnatis, C. Nübel, Lasse Schlotter, and Kerstin Witte. 2023. Training in virtual reality enables learning of a complex sports movement. *Virtual Reality* 27, 2 (2023), 523–540. https://doi.org/10.1007/S10055-022-00679-7

[30] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. 2018. Deep Mimic: example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics* 37, 4 (2018), 143. https://doi.org/10.1145/3197517.3201311

[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* 36, 4 (2017), 41:1–41:13. https://doi.org/10.1145/3072959.3073602

[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. *ACM Transactions on Graphics* 41, 4 (2022), 94:1–94:17. https://doi.org/10.1145/3528223.3530110

[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* 40, 4 (2021), 144:1–144:20. https://doi.org/10.1145/3450626.3459670

[34] Mingyo Seo, Steve Han, Kyutae Sim, SeungHyeon Bang, Carlos Gonzalez, Luis Sentis, and Yuke Zhu. 2023. Deep Imitation Learning for Humanoid Loco-manipulation Through Human Teleoperation. In *22nd IEEE-RAS International Conference on Humanoid Robots, Humanoids 2023*. IEEE, 1–8. https://doi.org/10.1109/HUMANOIDS57100.2023.10375203

[35] Hubert P. H. Shum, Taku Komura, Masashi Shiraishi, and Shuntaro Yamazaki. 2008. Interaction patches for multi-character animation. *ACM Transactions on Graphics* 27, 5 (2008), 114. https://doi.org/10.1145/1409060.1409067

[36] Hubert Pak Ho 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

[37] Richard S. Sutton, Doina Precup, and Satinder Singh. 1999. Between MDPs and Semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning. *Artificial Intelligence* 112, 1-2 (1999), 181–211. https://doi.org/10.1016/S0004-3702(99)00052-1

[38] Chen Tessler, Yoni Kasten, Yunrong Guo, Shie Mannor, Gal Chechik, and Xue Bin Peng. 2023. CALM: Conditional Adversarial Latent Models for Directable Virtual Characters. In *ACM SIGGRAPH 2023 Conference Proceedings, SIGGRAPH 2023*. ACM, 37:1–37:9. https://doi.org/10.1145/3588432.3591541

[39] Kevin Wampler, Erik Andersen, Evan Herbst, Yongjoon Lee, and Zoran Popovic. 2010. Character animation in two-player adversarial games. *ACM Transactions on Graphics* 29, 3 (2010), 26:1–26:13. https://doi.org/10.1145/1805964.1805970

[40] Chao Wang, Anna Belardinelli, Stephan Hasler, Theodoros Stouraitis, Daniel Tanneberg, and Michael Gienger. 2023. Explainable Human-Robot Training and Cooperation with Augmented Reality. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems, CHI EA 2023*. ACM, 449:1–449:5. https://doi.org/10.1145/3544549.3583889

[41] Jungdam Won, Deepak Gopinath, and Jessica K. Hodgins. 2020. A scalable approach to control diverse behaviors for physically simulated characters. *ACM Transactions on Graphics* 39, 4 (2020), 33. https://doi.org/10.1145/3386569.3392381

[42] Jungdam Won, Deepak Gopinath, and Jessica K. Hodgins. 2021. Control strategies for physically simulated characters performing two-player competitive sports. *ACM Transactions on Graphics* 40, 4 (2021), 146:1–146:11. https://doi.org/10.1145/3450626.3459761

[43] Jungdam Won, Deepak Gopinath, and Jessica K. Hodgins. 2022. Physics-based character controllers using conditional VAEs. *ACM Transactions on Graphics* 41, 4 (2022), 96:1–96:12. https://doi.org/10.1145/3528223.3530067

[44] Pei Xu, Xiumin Shang, Victor B. Zordan, and Ioannis Karamouzas. 2023. Composite Motion Learning with Task Control. *ACM Transactions on Graphics* 42, 4 (2023), 93:1–93:16. https://doi.org/10.1145/3592447

[45] Heyuan Yao, Zhenhua Song, Baoquan Chen, and Libin Liu. 2022. ControlVAE: Model-Based Learning of Generative Controllers for Physics-Based Characters. *ACM Transactions on Graphics* 41, 6 (2022), 183:1–183:16. https://doi.org/10.1145/3550454.3555434

[46] KangKang Yin, Stelian Coros, Philippe Beaudoin, and Michiel van de Panne. 2008. Continuation methods for adapting simulated skills. *ACM Transactions on Graphics* 27, 3 (2008), 81. https://doi.org/10.1145/1360612.1360680

[47] Haotian Zhang, Ye Yuan, Viktor Makoviychuk, Yunrong Guo, Sanja Fidler, Xue Bin Peng, and Kayvon Fatahalian. 2023. Learning Physically Simulated Tennis Skills from Broadcast Videos. *ACM Transactions on Graphics* 42, 4 (2023), 95:1–95:14. https://doi.org/10.1145/3592408

[48] Qingxu Zhu, He Zhang, Mengting Lan, and Lei Han. 2023. Neural Categorical Priors for Physics-Based Character Control. *ACM Transactions on Graphics* 42, 6 (2023), 178:1–178:16. https://doi.org/10.1145/3618397

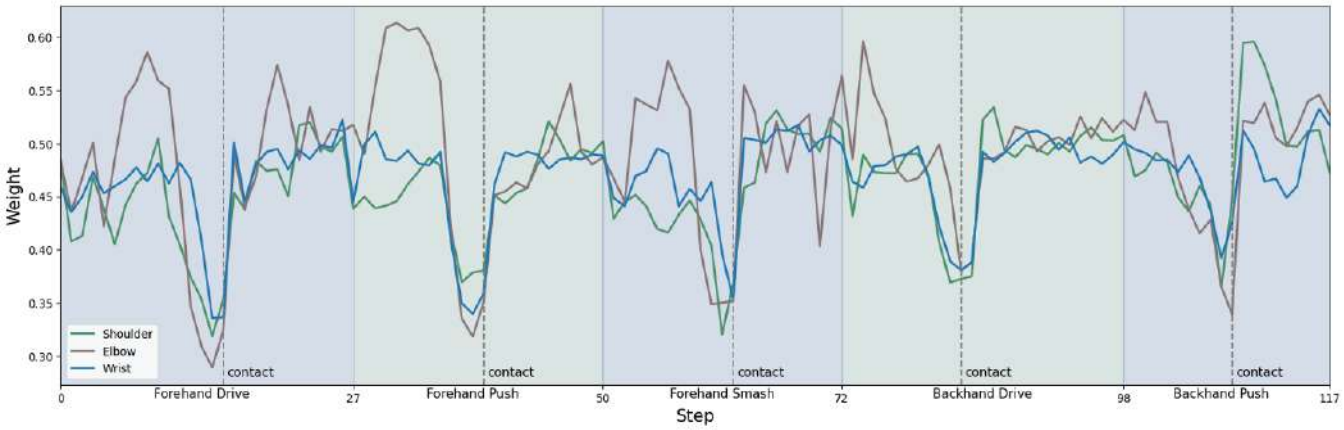

图 8: 肩膀、肘部和手腕关节的平均混合权重 φ 的可视化。混合器策略的权重通常在球拍接触球时最低，而在不同技能之间的过渡前后则较高。

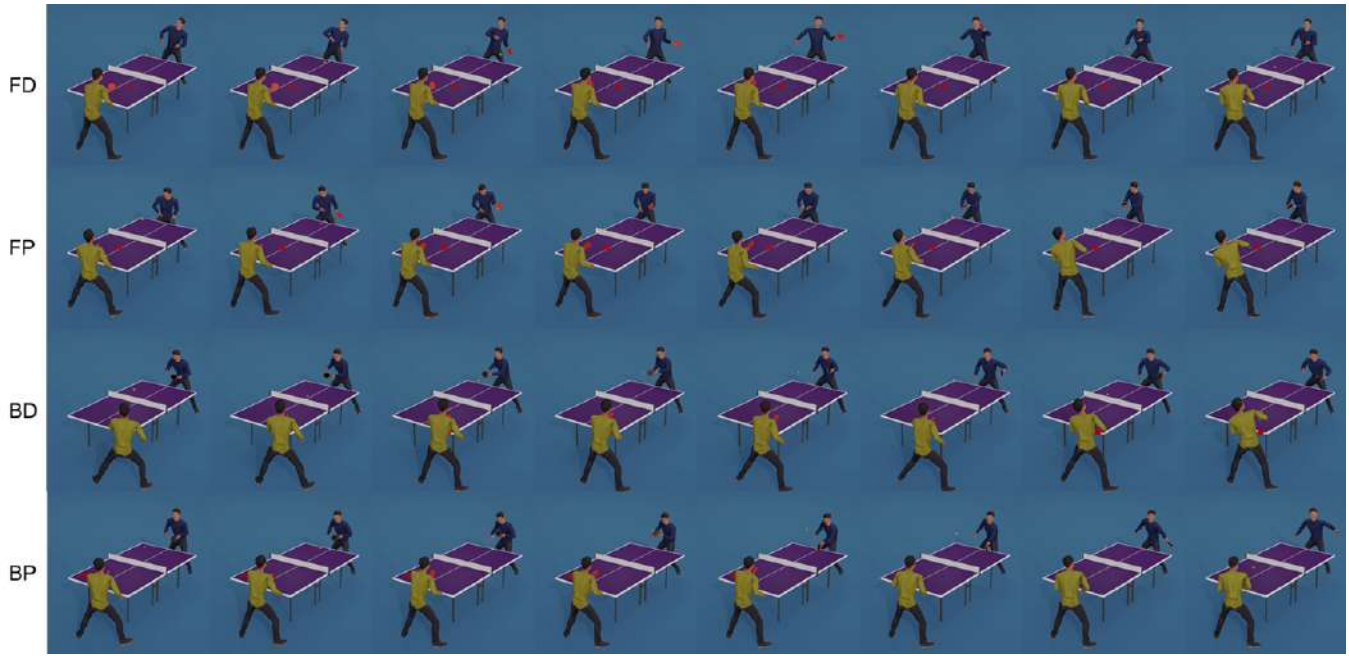

图 9: 代理之间的游戏玩法。蓝色代理应用了我们的策略级控制器。红点是目标。我们展示了四种技能；正手扣杀不太明显，因为对手没有发出高且慢的球。

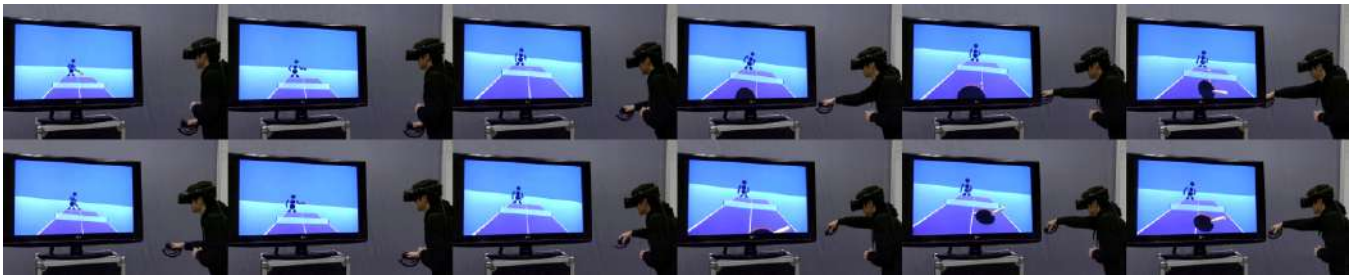

图 10: 人类-代理交互截图。人类控制一个模拟的球拍，而代理则由我们的方法进行模拟和控制。