

PingpongDuo: Agile Humanoid Table Tennis Doubles with Safe Team Coordination

Haoran Jiang^{1,2,3} Yichao Zhong³ Kai Zhang³ Cheng Liu³ Yixuan Pan³
Zhenjie Yang³ Jingbo Wang⁴ Tianyu Li^{2,1} Li Chen²

¹Shanghai Innovation Institute ²Archon Robotics ³The University of Hong Kong ⁴NVIDIA

<https://www.archon.tech/research/PingpongDuo>



Fig. 1: **PingpongDuo: the first table tennis *doubles* system on Humanoid.** After returning a ball on the right side of the table, the red robot clears space as the blue robot moves left to receive the next ball, achieving the complete movement within one second.

Abstract—Unlike singles, humanoid table tennis doubles requires teammates to strike in strict alternation while sharing a confined workspace. Rapid repositioning can compromise striking accuracy, while uncoordinated teammate movements introduce collision risks. We present PingpongDuo, a hierarchical framework for agile humanoid doubles with safe team coordination. At the team level, a planner jointly selects positioning targets and uses learned risk estimates to filter risky proposals, preferentially preserving the current striker’s target. At the individual level, separate footwork and striking skills combine agile movement with precise racket control. The skills are guided by human motion references, including footwork collected from human doubles play, and trained from states produced by one another to support rapid transitions. A racket-arm residual further refines strikes after movement using return feedback from a simulator calibrated with real ball-flight and contact measurements. In simulation, PingpongDuo improves the ball-return success rate by 13.2% over the state-of-the-art baseline, while achieving approximately 2 cm positioning error during clearance. In real-world experiments, a robot completes a 2.15 m clearance-and-recovery cycle in 2.4 s with a peak speed of 2.23 m/s, while two robots sustain more than 10 consecutive strikes in strict alternation.

I. INTRODUCTION

Recent advances in humanoid learning and control [1, 2] have enabled robots to perform dynamic interactions with fast-moving balls [3, 4, 5, 6, 7]. Racket sports combine rapid movement, precise contact timing, and whole-body balance.

Progress in these individual capabilities provides a foundation for studying humanoid team sports, where successful execution also depends on coordination with other players.

Existing humanoid table tennis systems such as HITTER [3] and SMASH [8] combine strike planning with learned control to support balance and consecutive ball returns. Their focus is on individual striking performance. We study table tennis doubles with two humanoid teammates on the same side of the table (Fig. 1). Unlike singles, doubles requires teammates to strike in strict alternation and share the space needed to approach, strike, and recover.

However, doubles requires each robot to coordinate its movement with its teammate while remaining ready for its next strike. After each strike, one robot must clear the central hitting area while its teammate moves into position for the next shot. Insufficient clearance can obstruct the teammate or cause collisions, whereas excessive displacement increases the time needed for subsequent recovery. At the individual level, robots must combine agile footwork with precise striking. Unlike singles, where movement primarily serves the incoming shot, doubles robot also requires clearance and recovery movements that may conflict with the next strike. The robot must therefore adapt its movement direction and speed while maintaining accurate racket control.

We present **PingpongDuo**, a hierarchical framework for

agile humanoid doubles with safe team coordination. A team planner jointly selects positioning targets, and a learned risk filter adjusts risky proposals while preferentially preserving the striker’s target. For agile movement and precise striking, we train separate footwork and striking skills with human motion references, including footwork collected from human doubles play. Each skill is trained from states produced by the other to support repeated transitions. A racket-arm residual further adjusts the nominal strike to the current robot and ball states using simulated return feedback. The system demonstrates more than 10 consecutive alternating strikes and a 2.15 m clearance-and-recovery cycle completed in 2.4 s in real-world deployment.

In sum, the contributions are:

- A hierarchical framework for safe team coordination and agile individual control in humanoid doubles, combining risk-aware joint positioning with footwork and striking skills trained for continuous execution.
- An arm refinement policy that preserves striking accuracy after rapid footwork by adapting nominal actions to robot and ball states using simulated return feedback.
- Simulation and real-world evaluations of individual agility and team coordination, including the first real-world demonstration of humanoid-humanoid teammates completing more than 10 consecutive alternating strikes.

II. RELATED WORK

A. Humanoid Racket Sports

Humanoid racket-sport research has advanced whole-body striking, repositioning, and perception [3, 4, 5, 7, 8]. Related studies in badminton also investigate coordinated locomotion and perception [9, 10, 11]. Moving beyond individual skills, these capabilities become increasingly important in multi-robot settings, where robots must remain agile while safely coordinating their motions and avoiding collisions with teammates. PingpongDuo further develops a low-level controller that supports agile footwork and striking skills.

Accurate racket-ball contact modeling and closed-loop reinforcement learning (RL) are essential for precise and robust strike performance [12]. Human motion references also guide tennis skill learning and task-specific adaptation [4, 5, 13]. In the context of table tennis, the small size and high speed of the ball make accurate trajectory modeling particularly important for precise interaction. Ball trajectory and racket–ball interaction modeling support accurate returns [12, 14]. PingpongDuo adopts this trajectory-based formulation. By incorporating simulated return feedback into training, PingpongDuo closes the loop between interaction and striking.

B. Safe Multi-Embodiment Coordination

Coordination in dynamic sports has been studied in robot teams [15] and human–robot table tennis doubles [16]. Simulated humanoid football [17] and cooperative quadrupedal soccer [18] combine learned skills with team-level policies. HCSP [19] jointly refines an event-driven team strategy and motor skills for multi-drone volleyball. Vision-driven humanoid soccer [20] learns reactive soccer skills while

using a high-level behavior tree to assign roles and coordinate behavior across the team.

Prior works typically deploy robots sparsely in large workspaces, while our table tennis doubles setting requires two humanoids to operate concurrently within a confined workspace. This makes multi-robot co-safety, particularly collision avoidance and motion obstruction, a critical requirement. ABS series [21, 22] integrate agile and recovery policies with reachability estimation to achieve high-speed locomotion, while safe RL approaches [23] introduce cost functions and recovery policies to enforce safety constraints during learning. In contrast to these single-robot safety formulations, temporal logic planning [24] and Control Barrier Functions [25, 26] have also been explored for multi-robot coordination and safety. However, acquiring full observations and humanoid Jacobians in the real world is often infeasible. We therefore combine risk estimation with constrained RL [27] for co-safe coordination, while a role-weighted safety filter prioritizes the striker’s target and adjusts the non-striker’s planned position to maintain ball-striking performance.

III. PRELIMINARIES AND SYSTEM OVERVIEW

We first summarize the ball prediction, strike planning, and striking policy adopted from prior single-player systems. We then introduce the doubles task and system architecture.

A. Preliminaries: Foundations for Singles Play

We use a modular formulation that separates ball prediction and strike planning from low-level motion control [3, 7, 8]. An adaptive extended Kalman filter estimates ball position and velocity from successive observations. A physics-based model predicts the trajectory under gravity, aerodynamic drag, and table bounces, from which a contact position and time are selected. Given the predicted incoming velocity and a desired landing location, a racket–ball interaction model determines the required racket velocity. For robot i , the resulting strike command is

$$c_t^{i,\text{hit}} = (p_r^{i,*}, v_r^{i,*}, \tau_t^i), \quad (1)$$

where $p_r^{i,*}$ and $v_r^{i,*}$ denote the desired racket position and velocity at contact, and τ_t^i is the remaining time to contact.

The striking policy combines this command with proprioceptive observations and reference motion information:

$$a_t^{i,\text{hit}} = \pi_{\text{strike}}(o_t^{i,\text{prop}}, m_t^{i,\text{hit}}, c_t^{i,\text{hit}}), \quad (2)$$

where $o_t^{i,\text{prop}}$ contains proprioceptive observations and recent control history, $m_t^{i,\text{hit}}$ encodes the whole-body reference motion, and $a_t^{i,\text{hit}}$ specifies whole-body joint-position targets. The reference is selected according to the strike target [8]. Training combines motion imitation, task rewards, and control regularization [1, 28]: imitation shapes the overall swing, while task rewards encourage the racket to reach the commanded position and velocity and to align its face normal with the desired strike direction within a short window immediately before and after the commanded contact time.

These single-player components do not explicitly coordinate the non-striker’s clearance, so independent positioning

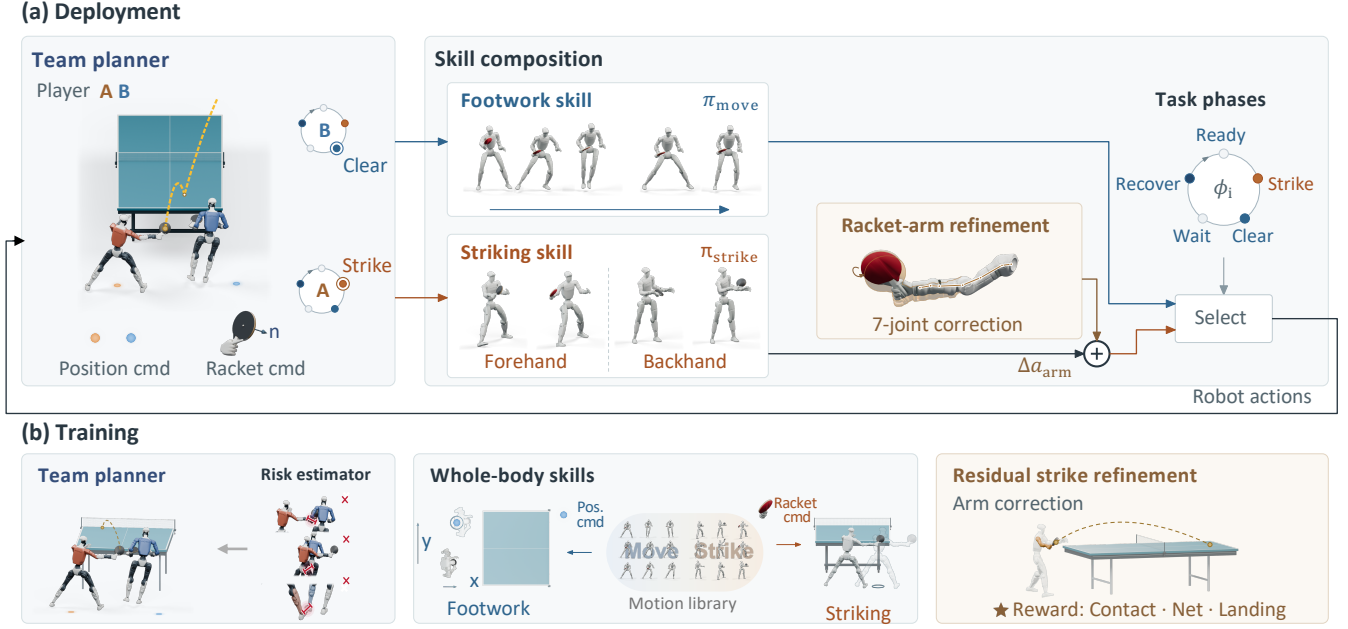


Fig. 2: **System overview.** (a) The team planner coordinates positioning under a learned risk-aware target filter, while ball prediction and strike planning provide racket commands (Sec. III-B). A state machine selects footwork or striking actions, with racket-arm refinement applied to the striking branch. (b) Training combines safety-aware team coordination (Sec. IV), skill learning from motion references, and strike refinement using simulated return feedback (Sec. V).

can lead to obstruction or collisions. Doubles also requires frequent repositioning, for which dense movement rewards can conflict with striking rewards concentrated around contact.

B. Doubles Task and System Overview

We consider two humanoid robots $i \in \{A, B\}$ on the same side of the table. Let x_t^i denote robot i 's state, b_t the ball state, and ϕ_t^i its state-machine phase. $h_k \in \{A, B\}$ identifies the striker for the k th incoming ball. Strict alternation requires $h_{k+1} \neq h_k$. The other robot is the non-striker. We use t for low-level control time, n for team-planning updates, and k for incoming-ball events. After each strike, the roles alternate: the previous striker becomes the non-striker and clears space for the new striker. Positioning must therefore balance clearance for the upcoming shot with timely recovery between shots.

Our framework combines team planning with footwork and striking skills (Fig. 2). The single-player planning module provides racket contact targets, while a team planner jointly selects lateral base targets for both robots. A learned risk estimator supports a role-weighted target filter that adjusts the paired targets while preferentially preserving the striker's target. Each skill is trained from states produced by the other to support repeated transitions. A racket-arm residual uses robot and ball feedback to refine the strike after movement.

During execution, a state machine combines the accepted positioning targets with the racket commands from ball prediction and strike planning. It tracks each robot's phase and enforces the alternating strike order. For the designated robot, a strike command takes priority over ongoing clearance, recovery, or waiting and immediately activates the striking skill. Otherwise, after completing its strike and follow-through,

the robot proceeds through clearance, waiting, and recovery, then holds its ready position until its next strike command. The corresponding low-level skill executes each phase.

IV. SAFETY-AWARE TEAM COORDINATION

Between incoming balls, fixed rules guide the next striker toward the table center and its teammate outward for clearance. The RL planner is invoked per ball, using the predicted strike command to select both robots' positioning targets. These targets are checked by the risk filter before execution.

A. Team Positioning Policy

At the planning update triggered by each incoming ball, the team positioning policy proposes

$$u_n = (y_n^A, y_n^B)^\top \sim \pi_\theta(\cdot | o_n^{\text{plan}}), \quad (3)$$

where u_n is the paired lateral base target and θ denotes the policy parameters. Each robot is constrained to remain within a longitudinal (x) range suitable for striking, while lateral targets y_n^i guide teammate clearance and recovery for subsequent shots. The planner observation o_n^{plan} contains the strike command computed from the latest ball observations using the physics-based prediction and planning module (Eq. (1)), together with both robots' current base positions and the previously accepted target pair. The striker's component supports the upcoming strike, whereas the non-striker's component provides clearance and supports subsequent recovery.

We train π_θ over sequences of incoming balls using constrained policy optimization based on PPO [27, 29], while holding the low-level controller and risk estimator fixed. The objective maximizes the accumulated team task return subject

to a limit on the expected cost of inter-robot separation violations. At deployment, each proposal u_n is evaluated by the risk-aware target filter before execution.

B. Learned Risk Estimation

Let $\mathcal{Q} = \{\text{hand, foot, base}\}$ index the monitored inter-robot separations. For a candidate target pair executed over rollout interval $\mathcal{I}_n$, we assign the binary labels

$$z_{n,q} = \mathbb{I} \left[\min_{t \in \mathcal{I}_n} \delta_q(t) < \delta_q^{\min} \right], \quad q \in \mathcal{Q}, \quad (4)$$

where $\delta_q(t)$ is the corresponding separation and $\delta_q^{\min}$ is its violation threshold. The dataset $\mathcal{D}_{\text{risk}} = \{(o_n^{\text{risk}}, u_n, z_n)\}$ is collected from closed-loop simulated rollouts with the low-level controller fixed and targets generated by random and perturbed heuristic planners. An MLP $r_\psi(o_n^{\text{risk}}, u_n)$ is trained in a supervised manner to predict the per-separation risk vector. The scalar score used by the target filter is the mean $\bar{r}_\psi(o, u) = |\mathcal{Q}|^{-1} \sum_{q \in \mathcal{Q}} r_{\psi,q}(o, u)$.

C. Role-Weighted Target Filtering

If $\bar{r}_\psi(o_n^{\text{risk}}, u_n) \leq \eta$, the proposal is accepted without modification. Otherwise, the filter searches for the closest admissible target pair whose predicted mean risk satisfies the threshold:

$$\begin{aligned} \tilde{u}_n = \arg \min_{u \in \mathcal{U}} \quad & w_{\text{str}}(u_{\text{str}} - u_{n,\text{str}})^2 + w_{\text{nstr}}(u_{\text{nstr}} - u_{n,\text{nstr}})^2 \\ \text{subject to} \quad & \bar{r}_\psi(o_n^{\text{risk}}, u) \leq \eta. \end{aligned} \quad (5)$$

Here $\mathcal{U}$ specifies the admissible lateral-target bounds, while str and nstr denote the striker and non-striker components. Choosing $w_{\text{str}} > w_{\text{nstr}} > 0$ assigns a larger modification cost to the striker's target because changing its position near contact can disrupt the immediate return. The role assignment follows h_k and changes after each incoming ball. We evaluate candidate target pairs by grid search over $\mathcal{U}$. If no admissible pair is found, both robots return to their predefined default targets and the next strike handoff is withheld.

V. LEARNING AGILE FOOTWORK AND PRECISE STRIKING

Doubles requires rapid repositioning between successive strikes while maintaining accurate racket control. We train footwork and striking skills separately, incorporate states from the preceding skill into their training, and refine the nominal strike using simulated ball-interaction outcomes.

A. Footwork and Striking

We collect whole-body motion data from human table tennis doubles and segment the recordings into around 200 footwork clips covering different movement distances. The clips capture movements for strike preparation, non-striker clearance, and recovery. We use GMR [30] to retarget them to the robot and annotate each clip with its lateral displacement. Retaining the temporal structure of these motions provides references for completing the required movements within the time available between strikes. These references guide coordinated stepping and body motion across

different repositioning distances, providing motion priors for both short adjustments and larger lateral movements.

At the start of each movement segment, we select a reference according to the commanded displacement:

$$\begin{aligned} m^* &= \arg \min_{m \in \mathcal{M}(\Delta y_t^i)} |\Delta y_t^i - \Delta y_m|, \\ \Delta y_t^i &= y_{t,\text{cmd}}^i - y_t^i, \end{aligned} \quad (6)$$

where $y_{t,\text{cmd}}^i$ and y_t^i are the commanded and current lateral base positions, Δy_m is the displacement of motion clip m , and $\mathcal{M}(\Delta y_t^i)$ contains clips for the requested direction. The team planner determines where the robot should move, while motion matching selects a reference for executing.

The footwork policy π_{move} receives proprioceptive observations, a movement command, and the matched reference. Its training combines position and velocity tracking, motion imitation, stability objectives, and control regularization. Movement phases emphasize reaching the target, while waiting phases emphasize braking and maintaining a ready posture. Target distances and waiting durations are varied to cover the movement and holding conditions in doubles.

We combine footwork with the striking skill introduced in Sec. III-A:

$$a_t^{i,\text{nom}} = \begin{cases} \pi_{\text{strike}}(o_t^{i,\text{prop}}, m_t^{i,\text{hit}}, c_t^{i,\text{hit}}), & \phi_t^i \in \Phi_{\text{hit}}, \\ \pi_{\text{move}}(o_t^{i,\text{prop}}, m_t^{i,\text{move}}, c_t^{i,\text{move}}), & \text{otherwise.} \end{cases} \quad (7)$$

Here the roman superscripts hit and move identify the skill, while i identifies the robot. The variables $m_t^{i,\text{hit}}$ and $m_t^{i,\text{move}}$ denote the respective reference-motion features, and $c_t^{i,\text{move}}$ specifies the movement command. The phase set Φ_{hit} includes striking and a brief follow-through. Footwork handles preparation, clearance, recovery, and waiting. Both skills produce whole-body joint actions.

We train the skills in a canonical configuration with right-handed striking and leftward clearance. A mirrored configuration uses left-handed striking and rightward clearance for the robot on the opposite lateral side of the same table half. Mirroring is applied consistently to observations, commands, reference motions, and joint actions, allowing both robots to share the trained policies.

B. Training for Skill Transitions

Directly composing independently trained skills can expose each policy to body states outside its training distribution. Footwork must begin from the posture and velocity left by a swing, while striking may begin before the robot has fully settled after movement. We therefore use states produced by each skill to initialize training of the other:

$$x_0^{\text{move}} \sim \mathcal{D}_{\text{hit} \rightarrow \text{move}}, \quad x_0^{\text{hit}} \sim \mathcal{D}_{\text{move} \rightarrow \text{hit}}, \quad (8)$$

where $\mathcal{D}_{\text{hit} \rightarrow \text{move}}$ contains states collected after striking and follow-through, and $\mathcal{D}_{\text{move} \rightarrow \text{hit}}$ contains states reached during footwork.

The footwork policy learns to initiate clearance and recovery from post-strike states. Conversely, the striking policy learns to execute its racket command from movement-generated states. These initial states include body position,

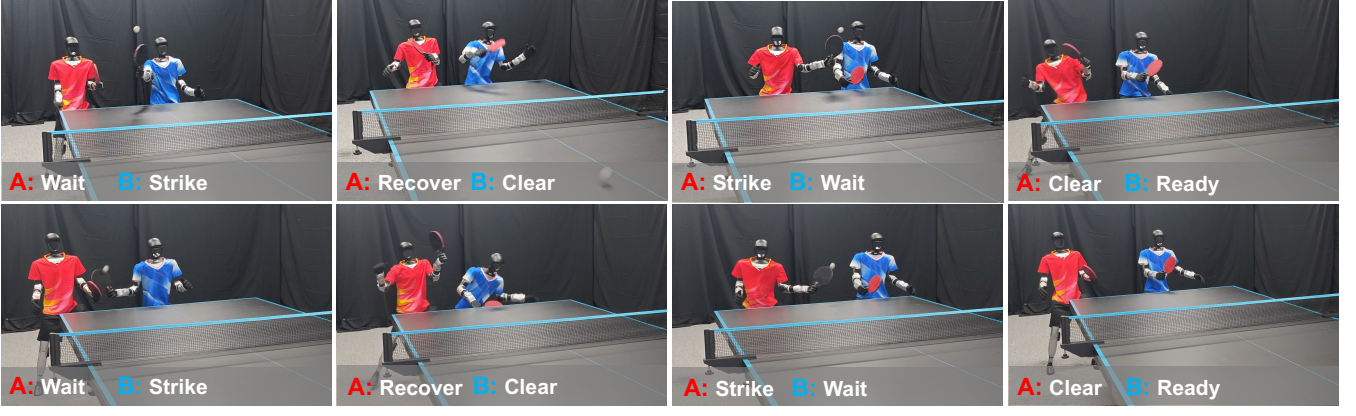


Fig. 3: **Real-world humanoid doubles.** Four consecutive alternating returns in a rally with human. Each striker clears space as its teammate prepares for the next shot.

orientation, linear and angular velocity, and joint positions and velocities, preserving the dynamics inherited from the preceding motion.

During execution, control histories are maintained across skill switches, and reference motions are transitioned over a short interval. A strike can begin during repositioning, and footwork resumes after follow-through. This supports repeated movement, striking, and clearance without resetting the robot to a standard pose between phases.

C. Strike Refinement with a Racket-Arm Residual

Reference motions are matched to the strike target, but their execution also depends on the robot’s current body state. Residual body velocity or an unsettled posture after footwork can make the selected motion less suited to the current conditions. Furthermore, position, velocity, and orientation errors jointly affect the realized contact and outgoing ball trajectory, so smaller racket-tracking errors do not necessarily yield a better return. We therefore use ball-interaction feedback to refine the nominal strike. With the nominal skills fixed, we learn a bounded correction over the seven racket-arm joints, following [4, 12]:

$$\begin{aligned} \Delta a_{i,\text{arm}}^i &= \pi_{\text{res}}(o_t^{i,\text{hit}}, a_t^{i,\text{nom}}, \hat{b}_t^i), \\ a_t^{i,\text{exec}} &= a_t^{i,\text{nom}} + \alpha g_t^i S_{\text{arm}}^i \Delta a_{i,\text{arm}}^i. \end{aligned} \quad (9)$$

The context $o_t^{i,\text{hit}}$ contains proprioceptive observations, reference motion information, and strike commands; $\hat{b}_t^i$ contains robot i ’s estimate of ball position and velocity. The residual $\Delta a_{i,\text{arm}}^i$ is a bounded seven-dimensional correction for the racket arm. The selector S_{arm}^i embeds it into the whole-body action, and α controls the overall correction magnitude. A phase-dependent temporal gate g_t^i gradually activates and removes the correction around contact.

Real-to-sim-to-real. As shown in Fig. 4, we calibrate the ball-interaction simulator using staged motion-capture experiments. We first identify the effective aerodynamic drag coefficient, table restitution, and friction parameters from vertical drops and oblique ball–table impacts. Grid search

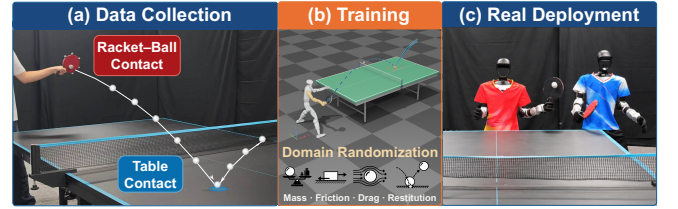


Fig. 4: **Real-to-sim-to-real training pipeline for strike refinement.** (a) We collect motion-capture trajectories covering racket–ball and table contacts to identify free-flight and contact dynamics. (b) The identified dynamics define the nominal simulator, where the refinement policy is trained with domain randomization over ball mass, friction, air drag, and restitution. (c) The trained refinement policy is transferred to the real robots without further adaptation.

selects parameters that minimize a weighted discrepancy between simulated and measured motion. The fitting objectives include time-aligned trajectory errors, rebound-height errors, and contact-time errors. For oblique impacts, post-contact velocity errors further constrain the tangential response, which vertical drops alone cannot identify.

With flight and table dynamics fixed, we fit the racket-contact model using moving-racket experiments. The fitted parameters describe speed-dependent normal restitution and effective tangential attenuation. Incoming and outgoing ball velocities are expressed relative to the moving racket contact point. We use grid search followed by local grid refinement to minimize a weighted sum of outgoing normal-velocity, tangential-speed, and direction errors. This staged procedure reduces compensation between flight and contact-model errors. The fitted parameters define the nominal simulator; separate dynamics randomization covers modeling error and physical variation during policy training.

We train the refinement policy with PPO [29] on strikes following footwork, using varied incoming balls and target landing locations. This exposes the policy to residual body motion after footwork, allowing it to adapt the strike to the

TABLE I: Team-planner comparison in simulation. Mean $\pm$ SD over five evaluation seeds.

Method	Mean returns $\uparrow$	S_2 (%) $\uparrow$	S_3 (%) $\uparrow$	S_4 (%) $\uparrow$	E_p (cm) $\downarrow$	E_v (m/s) $\downarrow$	E_n ($^\circ$) $\downarrow$
Rule-based	3.60 ± 0.40	57.40 ± 4.93	48.00 ± 5.48	38.40 ± 4.51	7.92 ± 0.21	0.56 ± 0.04	14.34 ± 2.22
Rule-based + Heuristic CBF	3.79 ± 0.38	57.40 ± 4.93	48.00 ± 5.48	39.20 ± 5.36	7.96 ± 0.23	0.56 ± 0.04	14.21 ± 2.17
Rule-based + Learned Risk	3.79 ± 0.36	57.40 ± 4.93	48.00 ± 5.48	38.80 ± 3.70	7.94 ± 0.23	0.56 ± 0.04	14.19 ± 2.48
RL + Learned Risk (ours)	4.87 ± 0.74	64.00 ± 4.30	54.80 ± 4.82	46.60 ± 5.68	7.42 ± 0.09	0.54 ± 0.05	14.72 ± 2.33

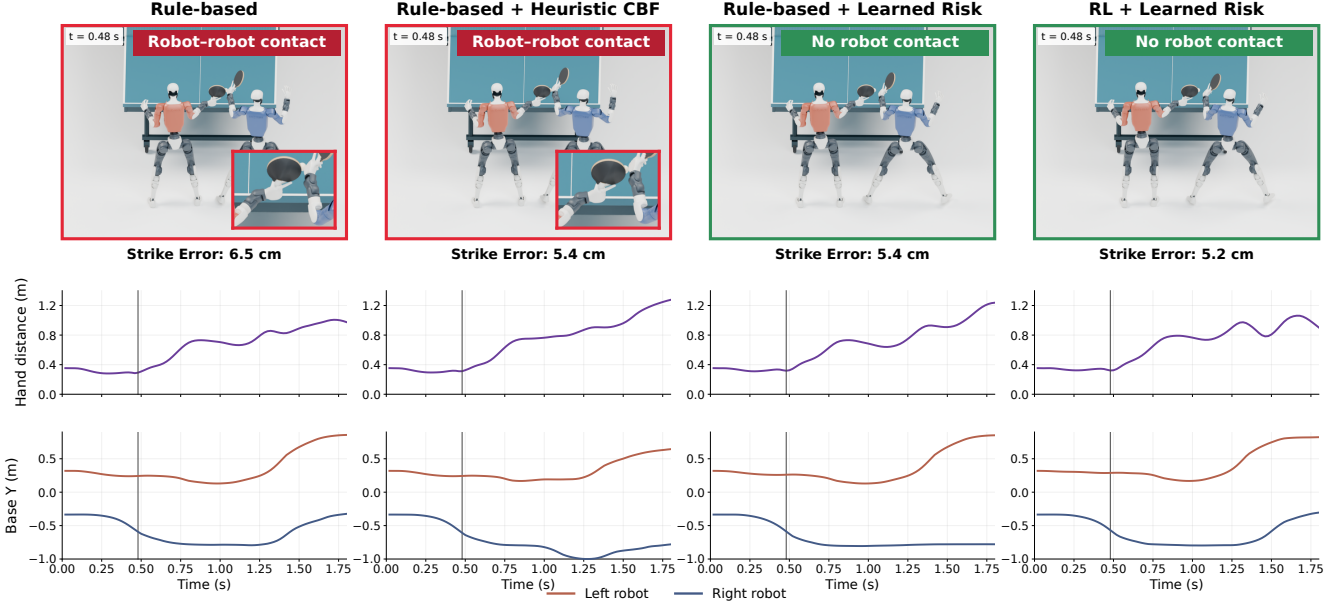


Fig. 5: Planner comparison with baselines under the same coming balls and robot initialization. Top: simulated execution snapshot at the collision moment of rule-based and rule-based+heuristic CBF; Mid: inner hand distance timeline curve; Down: two robots’ lateral base y-position timeline curve. Red boxes mark robot collision.

robot’s state at contact. For each ball trial, we randomize ball mass, aerodynamic drag, table restitution, and racket-contact responses around the nominal simulation. The corrected arm motion determines the racket state at impact, while simulated racket contact, net crossing, and landing provide task-level feedback. Rewards favor valid returns and, for successful returns, landing accuracy and net-clearance margin. Misses, low net crossings, out-of-bounds returns, and unintended ball-body contacts are penalized. Auxiliary racket-tracking and stability terms are retained together with regularization on the correction magnitude. Ball-observation perturbations and actuator delays provide additional variation during training. The resulting refinement policy is then deployed on the real robot without further adaptation.

VI. EXPERIMENTS

We evaluate PingpongDuo through three main questions: (1) Can joint positioning and risk filtering support safe, continuous alternating play? (2) Does skill composition combine agile footwork with precise striking more effectively than a joint policy? (3) Does arm refinement improve ball-return success when striking from moving states?

TABLE II: Evaluation metrics.

Metric	Unit	Definition
Mean returns	–	Mean consecutive legal returns
S_k	%	Percentage of attempts with at least k alternating legal returns
SR_C	%	Percentage of balls contacted by the racket
SR_H	%	Percentage of balls legally returned
E_p	cm	$100\ p_r(t^*) - p_r^*\ _2$
E_v	m/s	$\ v_r(t^*) - v_r^*\ _2$
E_n	deg	$\frac{180}{\pi} \arccos(n_r(t^*)^\top \hat{v}_r^*)$
E_M	cm	Lateral target error at the end of clearance
MPJPE	mm	Mean body-point error to the motion reference after pelvis-translation alignment

A. Experimental Setup

We deploy two 29-DoF Unitree G1 robots with external motion capture (MoCap) for robot and ball state estimation. Simulation uses Isaac Lab [31], with physics at 200 Hz and control at 50 Hz. The planner benchmark uses five evaluation seeds, with 100 attempts per seed. Policy tests use five seeds, with 100 balls and clearance trial per seed. The refinement study uses 100 movement-and-striking trials per condition.

TABLE III: **Policy comparison in simulation.** Mean $\pm$ SD over five seeds (100 balls and one clearance trial each). Joint: one policy; Composed: separate footwork and striking skills; w/o arm refine.: arm refinement disabled.

Method	Skill design	Striking					Clearance	Motion
		SR_C (%) $\uparrow$	SR_H (%) $\uparrow$	E_p (cm) $\downarrow$	E_v (m/s) $\downarrow$	E_n ($^\circ$) $\downarrow$	E_M (cm) $\downarrow$	MPJPE (mm) $\downarrow$
HITTER [3]	Composed	55.20 $\pm$ 6.14	42.40 $\pm$ 8.26	12.47 $\pm$ 2.63	1.09 $\pm$ 0.14	4.83 $\pm$ 1.46	3.76 $\pm$ 1.18	—
	Joint	47.20 $\pm$ 6.76	37.80 $\pm$ 8.17	14.66 $\pm$ 3.06	1.05 $\pm$ 0.08	3.91 $\pm$ 1.27	4.95 $\pm$ 1.05	—
SMASH [8]	Composed	53.80 $\pm$ 1.92	35.80 $\pm$ 0.84	9.62 $\pm$ 0.53	1.08 $\pm$ 0.08	7.20 $\pm$ 1.28	12.54 $\pm$ 3.45	92.35 $\pm$ 1.75
	Joint	9.20 $\pm$ 1.79	3.80 $\pm$ 1.92	16.98 $\pm$ 0.61	1.11 $\pm$ 0.06	8.38 $\pm$ 0.45	17.75 $\pm$ 2.94	95.08 $\pm$ 2.76
PingpongDuo (ours)	Composed	67.40 $\pm$ 4.88	55.60 $\pm$ 9.76	7.43 $\pm$ 0.44	1.13 $\pm$ 0.34	8.29 $\pm$ 2.06	2.11 $\pm$ 0.57	89.08 $\pm$ 0.80
	w/o arm refine.	55.20 $\pm$ 5.36	44.20 $\pm$ 12.97	7.77 $\pm$ 0.46	1.16 $\pm$ 0.32	8.72 $\pm$ 1.66	2.11 $\pm$ 0.57	89.04 $\pm$ 1.72
	Joint	59.00 $\pm$ 6.12	46.60 $\pm$ 7.50	6.00 $\pm$ 1.02	0.62 $\pm$ 0.16	11.74 $\pm$ 2.27	4.06 $\pm$ 3.08	93.33 $\pm$ 2.71

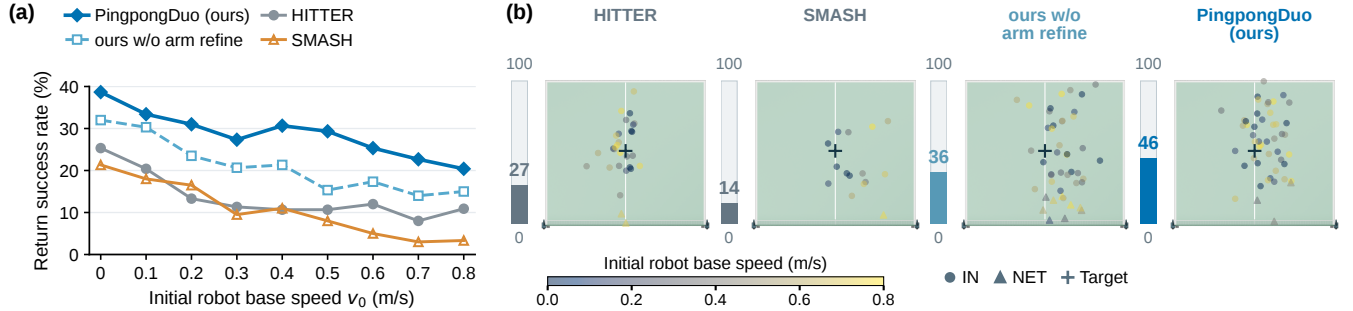


Fig. 6: **Arm refinement under initial motion.** (a) Return success versus initial base speed. (b) Landings and legal-return counts from 100 trials per method. Colors denote initial speed.

Results are reported as means and standard deviations across evaluation seeds where multiple seeds are used.

Table II summarizes the metrics. A legal return crosses the net and lands on the opponent’s table. In doubles, it must also follow the alternating order. Racket errors are evaluated at the commanded strike time t^* . Here n_r is the unit racket-face normal and $\hat{v}_r^*$ is the unit target velocity. Clearance is evaluated separately from striking. MPJPE measures tracking of each policy’s native reference.

B. Team Coordination

The full system sustains rallies with a human on hardware. Fig. 3 shows four consecutive alternating returns. After each strike, one robot clears space while its teammate approaches the next striking position. Clearance and preparation overlap in time. This requires both robots to act within the same workspace without waiting for the other to finish its entire movement. The hardware demonstrations include sequences of more than ten consecutive alternating team strikes.

In simulation, we compare rule-based positioning, heuristic CBF filtering, learned-risk filtering, and our learned positioning policy in Table I. All methods share the low-level controller. The two learned-risk variants also share the risk model and filtering settings. Their comparison separates the effect of target selection from the effect of filtering.

With the same learned-risk filter, learned positioning increases mean consecutive returns from 3.79 to 4.87 compared with fixed rule-based targets. This suggests that target

selection matters beyond avoiding interference during the current shot. The team must also retain room and time to prepare for subsequent turns. Joint positioning accounts for both robots when choosing these targets.

Fig. 5 shows different coordination behavior among different planners through one single nominal example case. The rule-based and heuristic-CBF planners fail to avoid robot contact in the shown case. Both learned-risk variants avoid that contact while ours(RL+Learned Risk) performs the lowest strike position error, retaining best striking quality. The example highlights the trade-off between clearance and striking: the farther to move a robot away conservatively, the worse quality the striking executes, to some extent. Joint positioning RL and role-weighted filtering address these requirements together.

C. Footwork and Striking Performance

We reproduce and adapt HITTER [3] and SMASH [8] to doubles movement commands. HITTER combines upper-body swing references with base and racket task rewards. SMASH uses target-matched whole-body references. SMASH and our method use the same reference motion library of striking and movement for fair comparison. For each method, we compare one joint policy with separate footwork and striking skills. Our joint or composed skill control variants use the same arm refinement policy for striking.

Across all three methods, composition improves return success in Table III. A joint policy must sustain recovery

motion and then respond to a new strike command. This may require braking or moving in the opposite direction. Separate skills allow movement and striking objectives to be trained individually, while transition training prepares each skill for the body states inherited from the preceding motion.

Our composed controller combines the highest mean return rate with the lowest clearance error. Human doubles references, including lateral shuffle steps, guide the larger movements needed for clearance and recovery. Compared with HITTER, PingpongDuo combines more accurate clearance with stronger return performance. This is consistent with the use of doubles motion references and dedicated footwork training. Our joint variant has smaller position and velocity errors but returns fewer balls. Thus, isolated tracking metrics do not fully describe the ability to move and return the ball in sequence. The results highlight the importance of coordinating footwork and striking when evaluating doubles controllers.

D. Effect of Arm Refinement

With the nominal skills fixed, arm refinement raises contact success from 55.2% to 67.4% and return success from 44.2% to 55.6% (Table III). The improvement concerns both reaching the ball and producing a valid return. Clearance performance is unchanged because the refinement acts only during striking. Fig. 6 (a) examines return success under initial motion. We set the robot’s initial motion in simulation and record its base speed v_0 when the ball appears. The refined policy retains an advantage over the nominal skills across the tested speeds. This supports the use of state-dependent arm adjustments when footwork leaves residual body motion at strike entry. Fig. 6 (b) complements the curves with landing distributions. Refinement produces more legal returns in this ablation study. The landing patterns reflect ball outcomes rather than nominal racket tracking. This comparison evaluates whether the arm correction improves the resulting ball return beyond tracking the planned racket command.

VII. CONCLUSION

We present PingpongDuo, a framework that combines agile footwork and precise striking with safe team coordination in humanoid table tennis doubles. Joint positioning and learned risk filtering coordinate the teammates’ movements, while skill composition, transition training, and arm refinement support accurate strikes after repositioning. Simulation results show improved ball-return performance and longer alternating return sequences. In real-world deployment, two robots perform more than 10 consecutive alternating strikes, and complete a 2.15 m clearance-and-recovery cycle in 2.4 s.

Limitations and future work. Our real-world system relies on external motion capture to estimate robot and ball states. The current sensing setup does not provide ball-spin measurements, and spin effects are not included in the trajectory prediction or racket-ball interaction models. Future work will investigate onboard perception and spin estimation. We also aim to extend the system with agents that autonomously select both their attacking and defensive strategies based on the evolving states of their opponents.

ACKNOWLEDGMENTS

We thank Yankai Liao, Junli Ren, and Penglin Fu for their insightful feedback and fruitful discussions. We also thank Yankai Liao, Tianqi Zhang, Xinran Zhao, and the Archon Robotics team for their help with filming the demon videos.

REFERENCES

- [1] Q. Liao, T. E. Truong, X. Huang *et al.*, “BeyondMimic: From motion tracking to versatile humanoid control via guided diffusion,” *SR*, 2026.
- [2] Z. Luo, Y. Yuan, T. Wang, *et al.*, “SONIC: Supersizing motion tracking for natural humanoid whole-body control,” *SR*, 2026.
- [3] Z. Su, B. Zhang, N. A. Rahmanian *et al.*, “HITTER: A humanoid table tennis robot via hierarchical planning and learning,” in *ICRA*, 2026.
- [4] Z. Zhang, H. Lu, Y. Lian *et al.*, “Learning athletic humanoid tennis skills from imperfect human motion data,” *arXiv preprint arXiv:2603.12686*, 2026.
- [5] T. Huang, R. Liu, X. Tang *et al.*, “Towards professional tennis styles for humanoid robots with adaptive motion planning and tracking,” in *CoRL*, 2026.
- [6] M. Hu, W. Chen, W. Li *et al.*, “PACE: Physics augmentation for coordinated end-to-end reinforcement learning toward versatile humanoid table tennis,” in *ICRA*, 2026.
- [7] P. Ren, C. Qi, H. Ge *et al.*, “CyboRacket: A perception-to-action framework for humanoid racket sports,” *arXiv preprint arXiv:2603.14605*, 2026.
- [8] J. Ren, Y. Li, K. Zhang *et al.*, “SMASH: Mastering scalable whole-body skills for humanoid ping-pong with egocentric vision,” *TRO*, 2026.
- [9] Y. Ma, A. Cramariuc, F. Farshidian, and M. Hutter, “Learning coordinated badminton skills for legged manipulators,” *Science Robotics*, 2025.
- [10] C. Liu, L. Jiang, N. Tian *et al.*, “Humanoid whole-body badminton via an annealed reinforcement learning curriculum,” *arXiv preprint arXiv:2511.11218*, 2025.
- [11] Y. Chen, S. Dong, X. Ji *et al.*, “Learning human-like badminton skills for humanoid robots,” *arXiv preprint arXiv:2602.08370*, 2026.
- [12] P. Dürr, M. El Gheche, G. J. Maeda *et al.*, “Outplaying elite table tennis players with an autonomous robot,” *Nature*, 2026.
- [13] B. Werner, I. Demler, P. Perona, and A. D. Ames, “TaskNPoint: How to teach your humanoid to hit a backhand in minutes,” *arXiv preprint arXiv:2606.26215*, 2026.
- [14] D. Nguyen, Z. Zaidi, K. Karol *et al.*, “Whole-body model predictive control for spin-aware quadrupedal table tennis,” in *IROS*, 2026.
- [15] Z. Li, F. Bjelonic, V. Klemm, and M. Hutter, “MARLadona - towards cooperative team play using multi-agent reinforcement learning,” in *ICRA*, 2025.
- [16] K. M. Lee, A. Krishna, Z. Zaidi *et al.*, “The effect of robot skill level and communication in rapid, proximate human-robot collaboration,” in *HRI*, 2023.
- [17] M. Abreu, L. P. Reis, and N. Lau, “Designing a skilled soccer team for robocup: exploring skill-set-primitives through reinforcement learning,” *Neural Computing and Applications*, 2023.
- [18] Z. Su, Y. Gao, E. Lukas *et al.*, “Toward real-world cooperative and competitive soccer with quadrupedal robot teams,” in *CoRL*, 2025.
- [19] R. Zhang, S. Xiang, Z. Xu *et al.*, “Mastering multi-drone volleyball through hierarchical co-self-play reinforcement learning,” in *CoRL*, 2025.
- [20] Y. Wang, C. Luo, P. Chen *et al.*, “Learning vision-driven reactive soccer skills for humanoid robots,” *Science Robotics*, 2026.
- [21] T. He, C. Zhang, W. Xiao *et al.*, “Agile but safe: Learning collision-free high-speed legged locomotion,” in *RSS*, 2024.
- [22] Y. Zhong, C. Zhang, T. He, and G. Shi, “Bridging adaptivity and safety: Learning agile collision-free locomotion across varied physics,” in *L4DC*, 2025.
- [23] T.-Y. Yang, T. Zhang, L. Luu *et al.*, “Safe reinforcement learning for legged locomotion,” in *IROS*, 2022.
- [24] M. Kloetzer and C. Mahulea, “Multi-robot path planning for syntactically co-safe ltl specifications,” in *WODES*, 2016.
- [25] L. Lindemann and D. V. Dimarogonas, “Control barrier functions for signal temporal logic tasks,” *IEEE Control Systems Letters*, 2019.
- [26] A. D. Ames, S. Coogan, M. Egerstedt *et al.*, “Control barrier functions: Theory and applications,” in *ECC*, 2019.

- [27] J. Achiam, D. Held, A. Tamar, and P. Abbeel, “Constrained policy optimization,” in *ICML*, 2017.
- [28] H. Jiang, J. Chen, Q. Bu *et al.*, “WholeBodyVLA: Towards unified latent vla for whole-body loco-manipulation control,” in *ICLR*, 2026.
- [29] J. Schulman, F. Wolski, P. Dhariwal *et al.*, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.
- [30] J. P. Araujo, Y. Ze, P. Xu *et al.*, “Retargeting matters: General motion retargeting for humanoid motion tracking,” *ArXiv*, 2025.
- [31] M. Mittal, P. Roth, J. Tigue *et al.*, “Isaac Lab: A gpu-accelerated simulation framework for multi-modal robot learning,” *arXiv preprint arXiv:2511.04831*, 2025.