SPIKEPINGPONG: SPIKE VISION-BASED FAST-SLOW PINGPONG ROBOT SYSTEM

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning to control high-speed objects in dynamic environments represents a fundamental challenge in robotics. Table tennis serves as an ideal testbed for advancing robotic capabilities in dynamic environments. This task presents two fundamental challenges: it requires a high-precision vision system capable of accurately predicting ball trajectories under complex dynamics, and it necessitates intelligent control strategies to ensure precise ball striking to target regions. High-speed object manipulation typically demands advanced visual perception hardware capable of capturing rapid motion with exceptional temporal resolution. Drawing inspiration from Kahneman's dual-system theory, where fast intuitive processing complements slower deliberate reasoning, there exists an opportunity to develop more robust perception architectures that can handle high-speed dynamics while maintaining accuracy. To this end, we present ***SpikePingpong***, a novel system that integrates spike-based vision with imitation learning for high-precision robotic table tennis. We develop a Fast-Slow system architecture where System 1 provides rapid ball detection and preliminary trajectory prediction with millisecond-level responses, while System 2 employs spike-oriented neural calibration for precise hittable position corrections. For strategic ball striking, we introduce Imitation-based Motion Planning And Control Technology, which learns optimal robotic arm striking policies through demonstration-based learning. Experimental results demonstrate that ***SpikePingpong*** achieves a remarkable 92% success rate for 30 cm accuracy zones and 70% in the more challenging 20 cm precision targeting. This work demonstrates the potential of Fast-Slow architectures for advancing robotic capabilities in time-critical manipulation tasks.

1 INTRODUCTION

Current research in robot learning primarily focuses on manipulation tasks involving static or slow-moving objects (Avigal et al., 2022; Wu et al., 2024; Wang et al., 2024; Luo et al., 2024; Vuong et al., 2024; Shao et al., 2020). While these achievements represent significant progress, they predominantly address scenarios with relatively simple dynamics and predictable object behaviors. However, real-world environments are replete with dynamic scenarios involving high-speed moving objects that demand rapid perception and precise control, from catching falling items (Zhang et al., 2024b) and intercepting projectiles (Natarajan et al., 2024) to navigating through crowded environments (Gao et al., 2023). These high-speed scenarios present fundamentally more challenging problems requiring millisecond-level decision making and robust handling of dynamic uncertainties.

Table tennis provides an ideal testbed for developing such capabilities, as it constitutes an optimal paradigm for high-speed robotic interaction while exhibiting exceptional generalizability. This task embodies Moravec's paradox (Moravec, 1988) in its purest form: what appears as a simple recreational activity to humans represents one of the most challenging domains for robotics, demanding the integration of high-speed perception, predictive modeling, and precise motor control. Beyond its apparent simplicity, this task systematically encapsulates the fundamental challenges of dynamic robotics: millisecond-scale perception and prediction, precise manipulation under temporal constraints, and real-time strategic planning. The core competencies developed, including high-speed object tracking, precision manipulation, and adaptive control, demonstrate direct transferability to industrial automation Deka et al. (2024), medical robotics Wah (2025), and aerospace trajectory interception systems Baradaran (2025). This inherent scalability positions table tennis as a system-

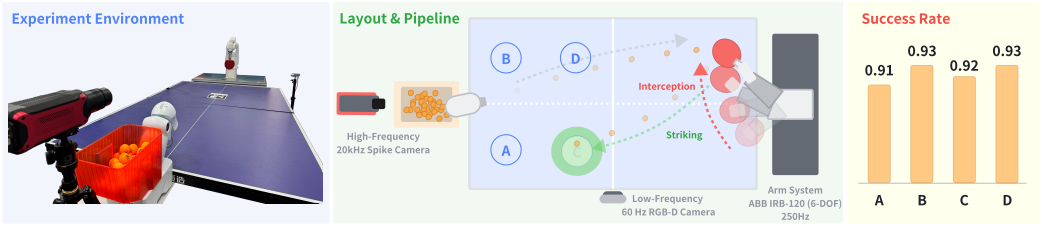

Figure 1: **Overview of *SpikePingpong*.** Our framework decomposes table tennis into two specialized phases: (1) the interception phase employing a Fast-Slow system architecture to achieve accurate ball interception, and (2) the striking phase utilizing IMPACT, which applies imitation learning to strike the ball to specified target locations. Experimental validation demonstrates 92% success rate for primary accuracy zones and 70% for high-precision targeting across four distinct regions.

atic methodology for developing foundational capabilities essential for advanced robots operating in dynamic, time-critical environments.

When tackling high-speed dynamic tasks like robotic table tennis, existing approaches can be broadly categorized into control-based methods and learning-based methods. Control-based approaches (Acosta et al., 2003; Mülling et al., 2010; Zhang et al., 2018; Silva et al., 2005; Yang et al., 2010) rely on precise physical modeling and predefined motion planning, but despite being mathematically rigorous and computationally efficient, they struggle with real-world complexities due to their requirement for precise calibration and inability to adaptively adjust to varying ball trajectories or unexpected disturbances. Learning-based approaches (Mülling et al., 2013; Abeyruwan et al., 2023; D'Ambrosio et al., 2023; Gao et al., 2020; Ding et al., 2022; Zhao et al., 2024; DAmbrosio et al., 2025; Su et al., 2025), offer greater theoretical adaptability. However, they often suffer from the persistent sim-to-real gap, where policies trained in simulation perform poorly in physical systems. This is particularly pronounced in table tennis, where subtle factors like ball spin and contact dynamics significantly impact performance. Additionally, existing methods typically rely on high-precision hardware vision systems, which are expensive and may still struggle with the rapid temporal dynamics required for accurate trajectory prediction. Drawing inspiration from Kahneman's dual-system theory (Kahneman, 2011), where fast intuitive processing (System 1) complements slower deliberate reasoning (System 2), there exists an opportunity to develop more robust perception architectures that can handle high-speed dynamics while maintaining accuracy.

To this end, we propose ***SpikePingpong***, a high-precision robotic table tennis system integrating spike vision-based Fast-Slow system and advanced control techniques as illustrated in Figure 1. Our system addresses the fundamental challenge of robotic table tennis through a principled decomposition into interception and striking phases, each employing specialized technical innovations. **For the interception phase,** we employ a Fast-Slow system architecture where System 1 provides rapid ball detection and preliminary trajectory prediction with millisecond-level responses, while System 2 leverages high-frequency spike camera data for refined trajectory analysis through neural error correction, effectively addressing physical model inaccuracies caused by environmental variables and spin dynamics. **For the striking phase,** we develop IMPACT (Imitation-based Motion Planning And Control Technology), which learns strategic ball striking through demonstration-based learning, mapping incoming trajectory characteristics to optimal robotic arm striking policies. In summary, our contributions are as follows:

- We design and implement a comprehensive robotic table tennis system that systematically addresses high-speed dynamic manipulation through task-specific decomposition and Fast-Slow architecture.
- We develop a Fast-Slow system perception framework that enables accurate trajectory prediction using conventional cameras through neural error correction, complemented by real-world imitation learning for precise ball striking control.
- We conduct extensive experimental evaluation demonstrating superior performance with 92% success rate in 30cm zones and 70% accuracy in challenging 20cm precision targeting, validating the effectiveness of our integrated approach.

2 Related Work

2.1 Agile Policy Learning

Agile policy learning addresses the challenge of generating fast, adaptive, and robust behaviors in highly dynamic environments. It has been widely studied across various domains such as autonomous driving (Pan et al., 2017; Pomerleau, 1988; Muller et al., 2005; Anzalone et al., 2022; Pan et al., 2020), legged locomotion (Nguyen et al., 2017; Tan et al., 2018; Haarnoja et al., 2018; Zhong et al., 2025), humanoid skills (He et al., 2025; Ben et al., 2025; He et al., 2024; Zhang et al., 2024a), and dynamic manipulation tasks like throwing and catching (Zhang et al., 2024b; Hu et al., 2023; Kim et al., 2014; Huang et al., 2023; Lan et al., 2023). These tasks require policies capable of maintaining high inference frequencies, handling disturbances, and generalizing across a wide range of conditions. Perception plays a critical role in enabling robots to adapt to environmental changes (Wang & Wang, 2022) and to understand the dynamic interaction between objects and agents (Zeng et al., 2020; Kober et al., 2011; Fu et al., 2025). When tracking fast-moving objects, systems often rely on high-speed motion capture setups (Mori et al., 2019). However, in table tennis scenarios, the high speed and abrupt motion of the ball cause significant motion blur with standard RGB cameras, leading to inaccurate position estimates and trajectory predictions. Unlike previous systems that rely on high-precision hardware setups, our ***SpikePingpong*** system introduces a Fast-Slow system architecture that integrates innovative spike-based vision technology with imitation learning. The Fast system provides rapid ball detection and preliminary trajectory estimation, while the Slow system leverages high-frequency spike camera data for refined trajectory analysis and error correction. This integrated approach compensates for systematic errors without complex physical modeling, significantly improving both interception accuracy and strategic ball striking.

2.2 Robotic Table Tennis

Robotic table tennis has long served as a benchmark task in robotics due to its requirement for real-time perception, prediction, planning, and control. Since Billingsley initiated the first robot table tennis competition in 1983 (Billingsley, 1983), the task has attracted continuous attention from the research community. Existing approaches can be broadly categorized into two groups: control-based methods and learning-based methods. **Control-based approaches** (Acosta et al., 2003; Mülling et al., 2010; Zhang et al., 2018) rely on mathematical modeling and predefined control strategies, typically following a perception-prediction-control pipeline. While these methods benefit from mathematical rigor, they often require precise calibration and struggle with adapting to environmental variations. **Learning-based methods**, particularly reinforcement learning and imitation learning, have gained prominence recently. RL approaches (Büchler et al., 2020; Tebbe et al., 2021; Gao et al., 2020; DAmbrosio et al., 2025; Su et al., 2025) directly map sensory inputs to motor commands, offering greater adaptability. Abeyruwan et al. (Abeyruwan et al., 2023) proposed iterative sim-to-real transfer, while GoalsEye (Ding et al., 2022) employs imitation learning through demonstrations and self-supervised practice, but its sim2real reliance limits performance. Our ***SpikePingpong*** system differs by introducing a Fast-Slow system perception architecture and real-world imitation learning for ball striking, training directly on data collected from real-world interactions without complex human assistance or simulation dependencies, achieving superior performance and practical deployment.

3 Method

In this section, we present our robotic table tennis framework ***SpikePingpong***, consisting of two integrated components as shown in Figure 2. Our approach features a Fast-Slow system for perception: System 1 employs physics-based trajectory prediction using an RGB-D camera for rapid ball detection, while System 2 leverages a high-frequency spike camera to refine predictions by compensating for real-world physical effects. For action generation, our Imitation-based Motion Planning And Control Technology module generates strategic hitting motions through imitation learning, enabling tactical control over return placement.

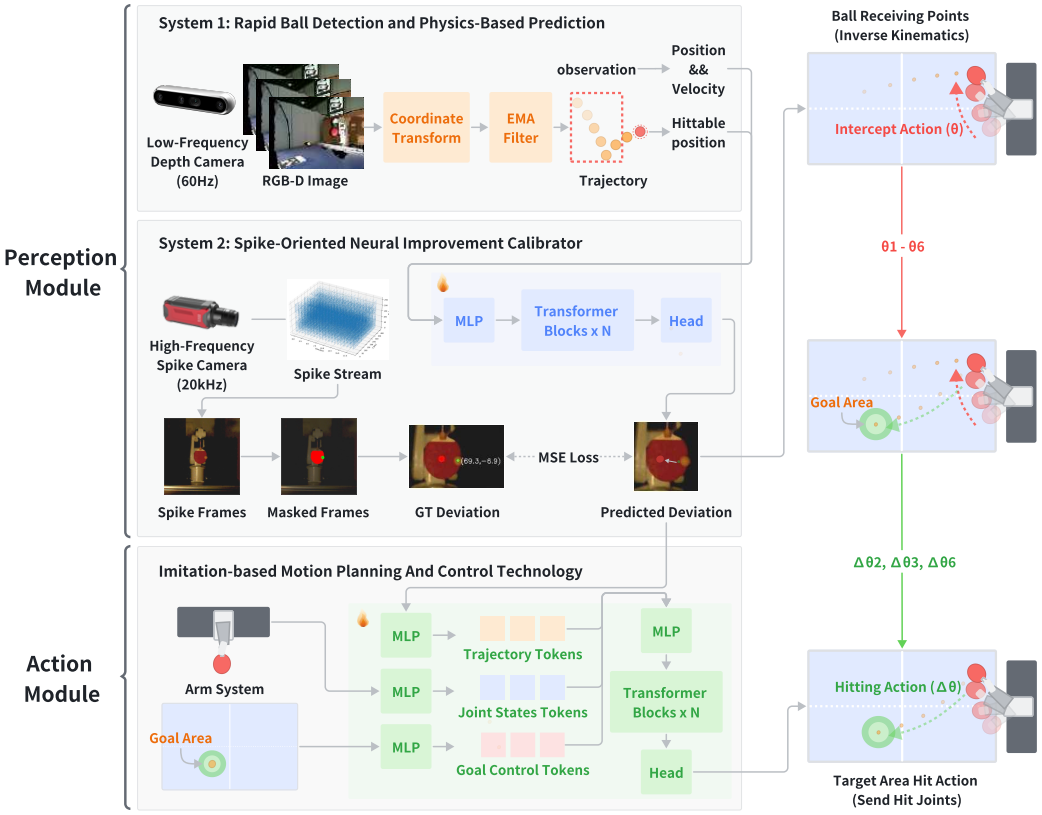

Figure 2: **Framework of *SpikePingpong*.** Our system integrates two key components: (1) A Fast-Slow system architecture for perception, where System 1 provides rapid trajectory prediction using RGB-D camera data and System 2 serves as a Spike-Oriented Neural Improvement Calibrator for refined hittable position estimation through spike camera data, and (2) IMPACT module for strategic motion planning and control.

3.1 Fast-Slow System Ball Interception Framework

3.1.1 System 1: Rapid Ball Detection and Physics-Based Prediction.

System 1 serves as the foundation of our Fast-Slow system architecture, providing rapid response capabilities for high-speed ball tracking through two core components: real-time ball detection and physics-based trajectory prediction. The detection module extracts ball positions from RGB-D camera streams, while the prediction module utilizes classical physics models to estimate future ball trajectories and determine optimal hittable positions. Additionally, System 1 provides essential contextual information to System 2, including ball state estimates and predicted hittable position that inform the learning-based decision-making process.

Ball Detection. We employ YOLOv4-tiny (bubbliiiing, 2020) for its computational efficiency, achieving detection frequencies up to 150 Hz. Our training methodology adopts a two-phase approach: initial pre-training on public datasets (Roboflow (Alexandrova et al., 2015), TT2 (desigproject, 2022), and Ping Pong Detection (pingpong, 2024)), followed by domain-specific fine-tuning. Following object detection, we perform coordinate transformation from image space to world coordinates using calibrated camera parameters, enabling accurate 3D ball position acquisition for subsequent motion planning.

Physics-Based Trajectory Prediction. Our physics-based approach represents another key component of System 1, enabling the system to anticipate the ball's path and determine optimal hittable positions. The prediction model employs Exponential Moving Average filtering to obtain reliable estimates of the ball's current position (x, y, z) and velocity (v_x, v_y, v_z). These filtered state estimates

serve as input to our physics model, which outputs the predicted hittable position $(x_{\text{hit}}, y_{\text{hit}}, z_{\text{hit}})$ and corresponding velocity $(v_x^{\text{hit}}, v_y^{\text{hit}}, v_z^{\text{hit}})$. We calculate the time t required for the ball to reach the predetermined hitting plane at y_{hit}: $t = \frac{y_{\text{hit}} - y}{v_y}$. Using this time value, we predict the x-coordinate at the hittable position: $x_{\text{hit}} = x + v_x \cdot t$. If x_{hit} falls outside the robot's operational workspace, the ball is classified as unhittable. For the z-coordinate prediction, we consider two scenarios:

- **Direct trajectory:** If the ball doesn't contact the table before reaching y_{hit}, we compute $z_{\text{hit}} = z + v_z \cdot t + \frac{1}{2} g t^2$. When $z_{\text{hit}} > h_{table}$, no rebound occurs.
- **Rebound trajectory:** If the ball impacts the table, we calculate the rebound time t_{rb} by solving: $z + v_z \cdot t_{\text{rb}} + \frac{1}{2} g t_{\text{rb}}^2 = h_{table}$.
  We then determine the impact velocity $v_{z,\text{in}}$ and post-rebound velocity $v_{z,\text{out}}$ using:

$$v_{z,\text{in}} = -\sqrt{-2g\,(z - h_{table}) + v_z^2}, \tag{1}$$

$$v_{z,\text{out}} = -e \cdot v_{z,\text{in}}. \tag{2}$$

  where e represents the coefficient of restitution. The system further evaluates potential secondary rebounds to determine the final z_{hit} and v_z^{hit}.

3.1.2 System 2: Spike-Oriented Neural Improvement Calibrator

System 2 serves as the learning-based enhancement layer of our Fast-Slow system architecture, addressing the limitations of physics-based predictions through neural calibration. While System 1 provides rapid trajectory estimation under ideal conditions, real-world scenarios introduce deviations due to air resistance, ball spin, and sensor noise that simplified physics models cannot capture. To bridge this gap, System 2 functions as a Spike-Oriented Neural Improvement Calibrator, which learns to predict the systematic discrepancy between System 1's theoretical hittable position and actual optimal interception position. By leveraging high-frequency spike camera observations and contextual information from System 1, System 2 provides precise calibration corrections that significantly enhance overall system accuracy.

Data Collection. System 2 integrates ball trajectory data, velocity measurements, and physics-based predictions to precisely quantify the discrepancy between theoretical and actual hittable positions. For training purposes, we developed an extensive dataset that meticulously documents the systematic variations between physics-model predictions and empirically observed real-world interception positions, enabling our system to learn these complex error patterns.

Specifically, during each trial, we systematically record the ball's 3D position and velocity vectors throughout its trajectory, along with the corresponding hittable position predicted by System 1's physics-based model. Based on these predictions, we compute the required joint angles through inverse kinematics and execute the corresponding robotic arm motion to position the paddle center at the theoretically optimal hittable position. Subsequently, a Spike camera (Dong et al., 2021) captures images of the actual ball-paddle interaction at the moment of contact. The pixel distance between the ball's observed position and the paddle center in these images provides a quantitative measure of the spatial deviation, which serves as the ground truth for System 2.

Network Architecture. System 2's network processes three input modalities: historical position vectors $\mathbf{p}_i \in \mathbb{R}^{K\times 3}$, velocity vectors $\mathbf{v}_i \in \mathbb{R}^{K\times 3}$ from the preceding K frames, and the physics-based predicted hittable position $\mathbf{h}_i \in \mathbb{R}^3$. Each modality is processed through dedicated MLPs with ReLU activation and dropout regularization for feature extraction. The concatenated features are then processed through a Transformer (Vaswani et al., 2017) encoder to capture temporal dependencies and contextual relationships across trajectory segments. Finally, a regression head with fully connected layers maps the refined representation to the predicted deviation vector.

Training Objective. We optimize the model using the mean squared error (MSE) loss function, which minimizes the difference between the predicted and ground-truth deviation vectors:

$$L_{MSE}(\theta) = \frac{1}{N} \sum_{i=1}^{N} ||\hat{D}_i - D_i||^2; \text{where } \hat{D}_i = f_\theta([p_i, v_i, h_i]). \tag{3}$$

f_θ represents the neural network with parameters θ, $p_i \in \mathbb{R}^{K\times 3}$ denotes the position history, $v_i \in \mathbb{R}^{K\times 3}$ denotes the velocity history, $h_i \in \mathbb{R}^3$ denotes the expected hittable position, $D_i \in \mathbb{R}^2$ is the ground truth deviation vector, and $\hat{D}_i \in \mathbb{R}^2$ is the predicted deviation vector.

Drawing inspiration from dual-system theory in cognitive science (Kahneman, 2011), this Fast-Slow system architecture combines the speed of immediate heuristic reasoning with the accuracy of experience-based learning, ensuring both real-time operation and precise task execution. Once trained, System 2 operates as a lightweight neural predictor that directly estimates deviation vectors from trajectory features without requiring spike camera feedback during deployment. This design enables the system to benefit from high-fidelity spike-based training data while maintaining computational efficiency and real-time performance in operational scenarios.

3.2 IMPACT: Imitation-based Motion Planning And Control Technology

Building upon our Fast-Slow system framework, where System 1 provides rapid physics-based predictions and System 2 delivers precise neural calibration through spike-oriented improvement, we introduce **IMPACT** (Imitation-based Motion Planning And Control Technology) for strategic striking behaviors. This module learns tactical ball-striking through imitation learning, enabling strategic returns that achieve targeted gameplay beyond mere interception.

Data Collection. The **IMPACT** module requires high-quality training data that captures the complete striking process from ball trajectory to final landing outcomes. Firstly, we record the incoming ball trajectory and estimate the optimal hitting position using our Fast-Slow system framework. Based on this prediction, we compute the corresponding robot joint configurations through inverse kinematics and position the robotic arm accordingly. To generate diverse striking behaviors, we apply random angular perturbations to three critical robot joints before executing the stroke. We retain only successful trials where the ball returns to the opponent's side, recording the perturbed joint angles and resulting landing positions. Each sample is labeled according to its specific landing region for fine-grained strategic control. This approach is highly efficient compared to teleoperation methods (Takada et al., 2022), as it leverages accurate hitting predictions to automatically position the robot, significantly reducing collection time while ensuring consistent data quality.

Network Architecture. The **IMPACT** module employs a transformer-based neural network that processes three input modalities: ball trajectory sequences, robot joint configurations, and desired landing region specifications (see Figure 2). Each modality is independently encoded into token representations using dedicated multi-layer perceptrons (MLPs). These tokens are concatenated to form a unified input sequence, which is processed by a Transformer encoder that leverages self-attention mechanisms to capture inter-modal dependencies. The network outputs optimal joint angle adjustments for precise ball placement control, effectively integrating trajectory information, kinematic constraints, and strategic objectives within a unified framework.

Training Objective. To train the model, we employ the mean squared error (MSE) loss function, which minimizes the discrepancy between the predicted and ground-truth joint adjustments:

$$L_{MSE}(\theta') = \frac{1}{N}\sum_{i=1}^{N} ||\hat{J}_i - J_i||^2; \text{where } \hat{J}_i = f_{\theta'}([p_i, v_i, j_i, c_i]). \tag{4}$$

Here, $f_{\theta'}$ represents the neural network with parameters θ', $p_i \in \mathbb{R}^{K\times 3}$ and $v_i \in \mathbb{R}^{K\times 3}$ denote the ball's position and velocity history, $j_i \in \mathbb{R}^6$ represents the 6-DOF robot joint configuration, $c_i \in \mathbb{R}^4$ represents the one-hot encoded control signal for the desired landing region, $J_i \in \mathbb{R}^3$ is the ground truth adjustment vector, and $\hat{J}_i \in \mathbb{R}^3$ is the predicted joint adjustment vector.

Through this imitation learning framework, **IMPACT** enables the robot to dynamically adapt its striking strategy to varying ball trajectories while precisely targeting specific regions on the opponent's court. This capability transforms the system from merely achieving technical accuracy to executing sophisticated tactical gameplay, significantly enhancing the robot's strategic competitiveness in table tennis matches.

4 EXPERIMENTS

We present comprehensive evaluations of ***SpikePingpong*** across multiple dimensions of table tennis performance. We begin with our experimental setup (Section 4.1), followed by the main results evaluating contact precision, spatial control, and computational efficiency (Section 4.2). We then conduct ablation studies to validate our architectural choices (Section 4.3).

4.1 EXPERIMENT SETTING

Dataset. We introduce the ***SpikePingpong*** Dataset containing two distinct parts for training our table tennis system. The first part includes 1k samples with trajectory information, expected hittable positions, and ground truth deviation vectors collected using RGB-D cameras (60 Hz) and high-frequency spike cameras (20 kHz). The second part consists of 2k successful return demonstrations containing ball trajectories, joint configurations, and adjustments for effective returns to specific court regions. All data were collected using an automated ball launching machine with randomized trajectories, spins, speeds, and angles under controlled conditions. This comprehensive dataset provides the foundation for training both our trajectory prediction and strategic return planning components. Detailed information can be found in Appendix D.

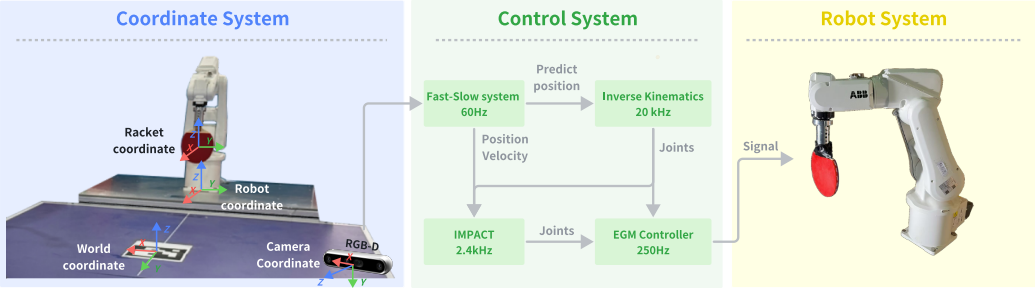

Figure 3: **System Overview.** Our system integrates three key subsystems: (1) a coordinate system for spatial tracking and transformation, (2) a multi-frequency control system with Fast-Slow system, IMPACT, and an EGM controller, and (3) a robot system based on the ABB IRB-120 arm equipped with a standard table tennis racket.

Implementation Details. As shown in Fig. 3, our system consists of three integrated subsystems. The coordinate system handles spatial transformations between world, camera, robot, and racket coordinates using ArUco marker calibration. The control system operates at multiple frequencies: Fast-Slow system processes trajectory data at 60Hz and converts predictions to joint configurations via inverse kinematics at 20kHz, while IMPACT operates at 2.4kHz for striking adjustments with commands transmitted to the EGM controller at 250Hz. The robot system employs an ABB IRB-120 robotic arm with a standard table tennis racket, implemented on a workstation with NVIDIA RTX 4090 GPU. Training details are provided in the Appendix E.

Baselines. We benchmark against ACT (Zhao et al., 2023) and Diffusion Policy (Chi et al., 2023). To ensure fair comparison, we optimized both baselines by using state-based inputs (identical to IMPACT) instead of raw images to eliminate visual latency. Additionally, we accelerated Diffusion Policy using a 10-step DDIM sampler. All methods employ a synchronous just-in-time one-shot inference strategy, executing a single forward pass at the latest decision point to leverage the most accurate state estimation.

Evaluation. To evaluate our system's precision and accuracy, we adopt quantitative metrics based on spatial accuracy. Our evaluation protocol is designed to rigorously assess generalization through standard offline validation, stochastic online testing, and challenging out-of-distribution scenarios. First, for ball-racket contact precision (MAE/RMSE), we strictly adhere to standard machine learning practices by reporting performance on a held-out test set, derived from an 80/10/10 (train/validation/test) split of our collected trajectory dataset. For landing accuracy, we divide the table tennis court into four regions (A, B, C, D) and measure successful hit rates using two thresholds: primary accuracy zone (within 30cm of target center) and high precision zone (within 20cm). Crucially, these success rates are evaluated via online physical experiments where the ball launcher gen-

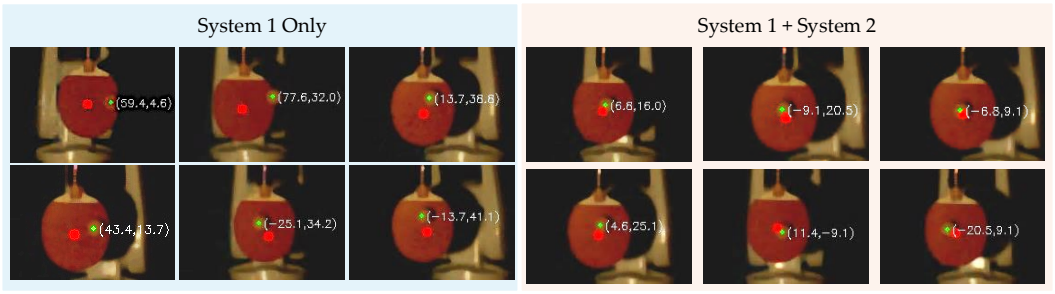

Figure 4: **Visualization of ball-racket contact precision with and without System 2.** Spike-based camera images show the ball center (green) and racket center (red) at contact moment. The reduced offset with Fast-Slow System demonstrates improved ball interception accuracy.

Table 1: **Ball Hittable Position Prediction Error.** Our Fast-Slow system approach achieves superior precision in predicting the actual ball-racket contact point across both axes.

Method	Y-axis		Z-axis		Overall	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
System 1 Only	53.65	60.39	34.62	38.45	44.13	50.62
RNN-based Method (Elman, 1990)	24.10	24.89	21.50	22.52	22.80	23.73
System 1 + System 2	**9.87**	**11.16**	**14.82**	**16.10**	**12.34**	**13.85**

erates completely randomized trajectories (varying spin, speed, and location) for each trial, ensuring the test scenarios are distinct from the training distribution. Furthermore, to assess strong generalization, we conducted Out-of-Distribution (OOD) experiments involving physically relocated launcher positions and zero-shot transfer against unseen human opponents. Due to the absence of publicly available hardware implementations, we rely on these standardized metrics and rigorous controlled validation to demonstrate system performance.

4.2 Main Results

Ball-Racket Contact Precision Evaluation Our Fast-Slow system design demonstrated significant improvements in predicting ball-racket contact deviations. Table 1 presents the quantitative results, showing our Fast-Slow system approach achieves superior precision with MAE of 12.34 and RMSE of 13.85 for overall deviation prediction. As shown in Figure 4, it substantially improves contact precision between ball and racket, with spike camera captures revealing minimal separation between ball center (green) and racket center (red), indicating highly accurate interception.

Spatial Control Capabilities We conducted rigorous comparative evaluations against established baseline methods across four strategically positioned target regions (A, B, C, and D). Table 2 presents success rates across various target regions and precision thresholds. ***SpikePingpong*** exhibits exceptional performance, achieving an average success rate of 92% within the 30cm accuracy zone and 70% within the 20cm high-precision zone, with consistent performance across all target regions. Our system substantially outperforms the human average (53% at 30cm) and all baseline methods. Notably, while baselines like ACT performed better without visual inputs (improving from 12% to 19%), our approach still demonstrates a significant performance leap.

Table 2: **Single-Target Return Accuracy.** Success rates for ball striking across four distinct target regions (A-D) at both 30cm and 20cm precision thresholds. The table compares human players, previous robotic approaches, and our SpikePingpong system. Higher percentages indicate better performance, with all values presented as mean ± standard deviation.

Method	A		B		C		D		Avg.	
	30cm (%)	20cm (%)	30cm (%)	20cm (%)	30cm (%)	20cm (%)	30cm (%)	20cm (%)	30cm (%)	20cm (%)
Human Avg.	48 ± 6	28 ± 4	52 ± 5	32 ± 3	56 ± 7	38 ± 5	54 ± 4	34 ± 6	53 ± 6	33 ± 5
Diffusion Policy (Chi et al., 2023) (w/ vision)	3 ± 2	1 ± 1	4 ± 3	0 ± 0	2 ± 2	1 ± 1	3 ± 2	0 ± 0	3 ± 2	1 ± 1
Diffusion Policy (Chi et al., 2023) (w/o vision)	6 ± 3	2 ± 1	7 ± 4	2 ± 2	5 ± 2	1 ± 1	6 ± 3	1 ± 1	6 ± 3	2 ± 1
ACT (Zhao et al., 2023) (w/ vision)	11 ± 4	4 ± 2	12 ± 5	4 ± 1	10 ± 3	2 ± 1	14 ± 4	5 ± 2	12 ± 4	4 ± 2
ACT (Zhao et al., 2023) (w/o vision)	18 ± 5	7 ± 3	20 ± 6	8 ± 3	17 ± 4	6 ± 2	19 ± 5	7 ± 2	19 ± 5	7 ± 3
SpikePingpong (Ours)	**91 ± 3**	**69 ± 4**	**93 ± 2**	**72 ± 3**	**92 ± 4**	**70 ± 5**	**93 ± 3**	**71 ± 4**	**92 ± 3**	**70 ± 4**

Table 3: **Sequential Target Execution Performance.** Success rates for 100-shot random target sequences. The Success Rate column shows the percentage of individual shots successfully reaching their designated targets (30cm), while subsequent columns present region-specific success rates.

Method	Success Rate(%)	A		B		C		D	
		30cm (%)	20cm (%)	30cm (%)	20cm (%)	30cm (%)	20cm (%)	30cm (%)	20cm (%)
Human Avg.	45 ± 5	44	26	47	28	43	25	46	29
Diffusion Policy (Chi et al., 2023) (w/ vision)	1 ± 1	1	0	2	0	1	0	1	0
Diffusion Policy (Chi et al., 2023) (w/o vision)	2 ± 2	2	1	3	1	2	0	2	0
ACT (Zhao et al., 2023) (w/ vision)	8 ± 3	7	2	9	3	6	2	10	3
ACT (Zhao et al., 2023) (w/o vision)	15 ± 4	14	5	16	6	13	4	17	5
SpikePingpong (Ours)	**78 ± 3**	**76**	**52**	**79**	**54**	**77**	**51**	**80**	**55**

Sequential Target Execution Capabilities To evaluate our system's ability to execute tactical sequences, we designed experiments involving random target sequences spanning 100 consecutive returns across four target regions (A, B, C, D), with 25 targets per region presented in random order. We define success using a 30cm precision threshold, with additional 20cm precision metrics provided for detailed analysis. Table 3 presents the results of our sequential target execution experiment. SpikePingpong achieves an overall sequence success rate of 78%, significantly outperforming the human baseline of 45%. Our system demonstrates balanced performance across all court regions with consistent precision under stricter 20cm thresholds. These results demonstrate that our system maintains high precision while executing complex, extended tactical sequences that surpass human-level performance, representing a significant advancement toward sustained strategic gameplay rather than merely returning balls.

Out-of-Distribution Generalization on Robotic Launchers To evaluate the generalization capability of our policy beyond the training distribution, we conducted a challenging out-of-distribution (OOD) experiment. While all training data was collected with the ball launcher at a fixed central position on the opponent's table, for the OOD test, we physically moved the launcher to two new, unseen off-center positions. This fundamentally altered the entire distribution of incoming ball trajectories, including their angles, speeds, and bounce locations.

As shown in Table 4, while there is a predictable drop in performance compared to the in-distribution setting, our system maintained a remarkable average success rate of 74% for 30cm targets. This result is significant, as the off-center launcher positions create entirely novel trajectories that the policy has never encountered. The sustained high performance strongly suggests that our policy has learned a robust, underlying model of ball dynamics and striking control, rather than simply memorizing patterns from the training data. This validates the generalization capabilities of our framework.

Table 4: **Out-of-Distribution Generalization.** Success rates on seen vs. unseen trajectory distributions.

Condition	Avg. Success Rate	
	30cm (%)	20cm (%)
In-Distribution	**92 ± 3**	**70 ± 4**
Out-of-Dist.	74 ± 5	52 ± 6

Adaptation and Generalization to Human Demonstrations To push the boundaries of generalization, we tested our system's ability to adapt to and generalize from complex, human-generated trajectories. Unlike the structured patterns from a robotic launcher, human shots introduce significant, unstructured variability, representing a more challenging data distribution. We conducted a two-stage experiment to assess this capability, with results summarized in Table 5.

First, to test adaptability, we fine-tuned our model on a small dataset of 100 demonstrations from a single human player (Person A). As shown in the "Seen Player" rows of Table 5, when tested on this same player, the system achieved a meaningful success rate of 47% for 30cm targets, demonstrating strong sample efficiency.

Table 5: **Adaptation and Generalization to Human Demonstrations.** Success rates on precision targeting tasks. "Seen Player" refers to testing on the same human demonstrator (Person A) whose data was used for fine-tuning. "Unseen Player" refers to zero-shot testing on a new human player (Person B).

Precision	**Condition**	**A (%)**	**B (%)**	**C (%)**	**D (%)**	**Avg. (%)**
30cm	Seen Player (Person A)	51±5	47±3	42±3	50±4	**47±3**
	Unseen Player (Person B)	35±6	31±5	27±5	32±4	**31±5**
20cm	Seen Player (Person A)	29±2	23±4	27±2	28±3	**27±3**
	Unseen Player (Person B)	18±3	13±4	16±3	14±4	**15±4**

Next, to perform a stricter test of generalization, we evaluated this fine-tuned model in a zero-shot setting against a new, unseen player (Person B). The results in the "Unseen Player" rows show a drop in performance to 31%, as expected in a challenging OOD scenario. Nevertheless, achieving over 31% accuracy on a completely new person without any specific fine-tuning is a non-trivial result. It provides promising evidence that our model captures generalizable features of human-like dynamics rather than simply overfitting to an individual's style, highlighting its potential for collaborative human-robot scenarios.

Computational Efficiency Real-time responsiveness is crucial in table tennis robotics. As shown in Table 6, ***SpikePingpong*** achieves an inference latency of only 0.407ms, dramatically outpacing Diffusion Policy and substantially exceeding ACT. This rapid inference capability provides adequate time for physical actuation, enabling responsive gameplay in real-world conditions.

Table 6: **Computational Performance Comparison.** Average inference times in milliseconds for generating return actions across different methods.

Method	Inference time (ms)
Diffusion Policy (Chi et al., 2023)	25.18
ACT (Zhao et al., 2023)	7.15
SpikePingpong	**0.407**

4.3 Ablation Study

We conducted ablation experiments across four target regions (A, B, C, D) within a 30cm radius threshold to evaluate each component's contribution. Our Fast-Slow system architecture achieves 92% accuracy in single-target returns, representing a 25-percentage-point improvement over the RNN (Elman, 1990) baseline. For sequential target execution, our Fast-Slow system achieves 78% success rate compared to 52% for the RNN-based method (Elman, 1990). This substantial enhancement stems from the system's ability to account for ball-racket interaction sensitivity, where identical striking motions can produce vastly different trajectories depending on precise contact positioning. The consistent improvements across both single-target and sequential tasks validate the robustness of our architectural design.

Table 7: **Ablation Study.** Performance comparison of different trajectory prediction components in our SpikePingpong system.

Single-Target Return Accuracy					
Method	A (%)	B (%)	C (%)	D (%)	Avg. (%)
System 1 + IMPACT	22 ± 6	25 ± 5	21 ± 7	24 ± 5	23 ± 6
RNN-based Method (Elman, 1990) + IMPACT	65 ± 5	68 ± 4	66 ± 6	69 ± 3	67 ± 5
Fast-Slow System + IMPACT	**91 ± 3**	**93 ± 2**	**92 ± 4**	**93 ± 3**	**92 ± 3**
Sequential Target Execution Performance					
Method	A (%)	B (%)	C (%)	D (%)	Success Rate (%)
System 1 + IMPACT	18 ± 5	20 ± 4	17 ± 6	19 ± 5	15 ± 4
RNN-based Method (Elman, 1990) + IMPACT	55 ± 4	58 ± 5	54 ± 6	57 ± 4	52 ± 6
Fast-Slow System + IMPACT	**76 ± 4**	**79 ± 3**	**77 ± 4**	**80 ± 3**	**78 ± 3**

5 Conclusions

In this paper, we presented ***SpikePingpong***, a high-precision robotic table tennis system integrating spike vision and advanced control techniques from a cognitive perspective. Our Fast-Slow system architecture emulates human visual perception processes for enhanced ball detection and trajectory prediction, while IMPACT handles strategic motion planning for accurate ball striking to specified target regions. Experiments demonstrate superior performance with 92% success in the primary accuracy zone and 70% in the high-precision zone, significantly outperforming previous systems. ***SpikePingpong*** enables real-time decision-making while executing extended tactical sequences with a 78% success rate, showcasing sustained strategic gameplay capabilities. The success of our approach stems from the Fast-Slow architecture that combines rapid physics-based prediction with neural calibration, and the IMPACT module that enables strategic ball placement through imitation learning. Beyond table tennis, the core competencies developed in this work, including high-speed object tracking, precision manipulation, and adaptive control, demonstrate broad applicability to industrial automation, medical robotics, and aerospace systems. This work advances robotic capabilities in time-critical manipulation tasks requiring precise spatiotemporal coordination.

ETHICS STATEMENT

We acknowledge and adhere to the ICLR Code of Ethics. All data collection focused solely on ball trajectories and paddle movements without storing personally identifiable information. This research contributes to recreational robotics applications and presents minimal ethical concerns when developed responsibly.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our work, we provide comprehensive implementation details and experimental specifications throughout the paper and supplementary materials. The experimental setup, including hardware specifications, data collection protocols, and evaluation metrics, is thoroughly documented in Appendix B. The dataset collection methodology and preprocessing steps are detailed in Appendix D, enabling replication of our data generation process. Appendix E contains detailed network architectures, hyperparameters, and training procedures for all components of our system. We've provided the code for our method in the supplementary material. Our codebase, including model implementations, training scripts, and evaluation protocols, will be made publicly available upon publication.

REFERENCES

Saminda Wishwajith Abeyruwan, Laura Graesser, David B D'Ambrosio, Avi Singh, Anish Shankar, Alex Bewley, Deepali Jain, Krzysztof Marcin Choromanski, and Pannag R Sanketi. i-sim2real: Reinforcement learning of robotic policies in tight human-robot interaction loops. In *Conference on Robot Learning*, pp. 212–224. PMLR, 2023.

Leopoldo Acosta, JJ Rodrigo, Juan A Mendez, G Nicolás Marichal, and Marta Sigut. Ping-pong player prototype. *IEEE robotics & automation magazine*, 10(4):44–52, 2003.

Sonya Alexandrova, Zachary Tatlock, and Maya Cakmak. Roboflow: A flow-based visual programming language for mobile manipulation tasks. In *2015 IEEE international conference on robotics and automation (ICRA)*, pp. 5537–5544. IEEE, 2015.

Luca Anzalone, Paola Barra, Silvio Barra, Aniello Castiglione, and Michele Nappi. An end-to-end curriculum learning approach for autonomous driving scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):19817–19826, 2022.

Yahav Avigal, Lars Berscheid, Tamim Asfour, Torsten Kröger, and Ken Goldberg. Speedfolding: Learning efficient bimanual folding of garments. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–8. IEEE, 2022.

Amir Hossein Baradaran. Advanced prediction of hypersonic missile trajectories with cnn-lstm-gru architectures. *arXiv preprint arXiv:2506.15043*, 2025.

Qingwei Ben, Feiyu Jia, Jia Zeng, Junting Dong, Dahua Lin, and Jiangmiao Pang. Homie: Humanoid loco-manipulation with isomorphic exoskeleton cockpit. *arXiv preprint arXiv:2502.13013*, 2025.

J. Billingsley. Robot ping pong. *Practical Computing*, 1983.

bubbliiiing. yolov4-tiny-pytorch: PyTorch implementation of YOLOv4-tiny. `https://github.com/bubbliiiing/yolov4-tiny-pytorch`, 2020.

D. Buchler, S. Guist, R. Calandra, V. Berenz, B. Scholkopf, and J. Peters. Learning to play table tennis from scratch using muscular robots. *IEEE Transactions on Robotics*, 38:3850–3860, Dec. 2022. doi: 10.1109/tro.2022.3176207. URL `https://doi.org/10.1109/tro.2022.3176207`.

Dieter Büchler, Simon Guist, Roberto Calandra, Vincent Berenz, Bernhard Schölkopf, and Jan Peters. Learning to play table tennis from scratch using muscular robots. *CoRR*, abs/2006.05935, 2020.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, pp. 02783649241273668, 2023.

David B D'Ambrosio, Jonathan Abelian, Saminda Abeyruwan, Michael Ahn, Alex Bewley, Justin Boyd, Krzysztof Choromanski, Omar Cortes, Erwin Coumans, Tianli Ding, et al. Robotic table tennis: A case study into a high speed learning system. *arXiv preprint arXiv:2309.03315*, 2023.

David B DAmbrosio, Saminda Abeyruwan, Laura Graesser, Atil Iscen, Heni Ben Amor, Alex Bewley, Barney J Reed, Krista Reymann, Leila Takayama, Yuval Tassa, et al. Achieving human level competitive robot table tennis. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 74–82. IEEE, 2025.

Rupam Deka, Partha Protim Borthakur, Elora Baruah, Pranjal Sarmah, and Madhurjya Saikia. A comprehensive review on mechanical conveyor systems: Evolution, types, and applications. *Int. J. Eng. Nat. Sci*, 18:164–183, 2024.

desigproject. tt2 dataset. `https://universe.roboflow.com/desigproject/tt2-cyt9i`, dec 2022. URL `https://universe.roboflow.com/desigproject/tt2-cyt9i`. visited on 2025-05-12.

Tianli Ding, Laura Graesser, Saminda Abeyruwan, David B D'Ambrosio, Anish Shankar, Pierre Sermanet, Pannag R Sanketi, and Corey Lynch. Goalseye: Learning high speed precision table tennis on a physical robot. *arXiv preprint arXiv:2210.03662*, 2022.

Siwei Dong, Tiejun Huang, and Yonghong Tian. Spike camera and its coding methods. *arXiv preprint arXiv:2104.04669*, 2021.

Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

Yankai Fu, Qiuxuan Feng, Ning Chen, Zichen Zhou, Mengzhen Liu, Mingdong Wu, Tianxing Chen, Shanyu Rong, Jiaming Liu, Hao Dong, et al. Cordvip: Correspondence-based visuomotor policy for dexterous manipulation in real-world. *arXiv preprint arXiv:2502.08449*, 2025.

Wenbo Gao, Laura Graesser, Krzysztof Choromanski, Xingyou Song, Nevena Lazic, Pannag Sanketi, Vikas Sindhwani, and Navdeep Jaitly. Robotic table tennis with model-free reinforcement learning. In *IROS*, 2020.

Xiaoshan Gao, Liang Yan, Zhijun Li, Gang Wang, and I-Ming Chen. Improved deep deterministic policy gradient for dynamic obstacle avoidance of mobile robot. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53(6):3675–3682, 2023.

Tuomas Haarnoja, Sehoon Ha, Aurick Zhou, Jie Tan, George Tucker, and Sergey Levine. Learning to walk via deep reinforcement learning. *arXiv preprint arXiv:1812.11103*, 2018.

Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. *arXiv preprint arXiv:2406.08858*, 2024.

Tairan He, Jiawei Gao, Wenli Xiao, Yuanhang Zhang, Zi Wang, Jiashun Wang, Zhengyi Luo, Guanqi He, Nikhil Sobanbabu, Chaoyi Pan, Zeji Yi, Guannan Qu, Kris Kitani, Jessica Hodgins, Linxi "Jim" Fan, Yuke Zhu, Changliu Liu, and Guanya Shi. Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills. *arXiv preprint arXiv:2502.01143*, 2025.

Zhe Hu, Yu Zheng, and Jia Pan. Grasping living objects with adversarial behaviors using inverse reinforcement learning. *IEEE Transactions on Robotics*, 39(2):1151–1163, 2023.

Binghao Huang, Yuanpei Chen, Tianyu Wang, Yuzhe Qin, Yaodong Yang, Nikolay Atanasov, and Xiaolong Wang. Dynamic handover: Throw and catch with bimanual hands. *arXiv preprint arXiv:2309.05655*, 2023.

Daniel Kahneman. *Thinking, fast and slow*. macmillan, 2011.

Seungsu Kim, Ashwini Shukla, and Aude Billard. Catching objects in flight. *IEEE Transactions on Robotics*, 30(5):1049–1065, 2014.

Jens Kober, Erhan Öztop, and Jan Peters. Reinforcement learning to adjust robot movements to new situations. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, pp. 2650. Citeseer, 2011.

Fengbo Lan, Shengjie Wang, Yunzhe Zhang, Haotian Xu, Oluwatosin Oseni, Ziye Zhang, Yang Gao, and Tao Zhang. Dexcatch: Learning to catch arbitrary objects with dexterous hands. *arXiv preprint arXiv:2310.08809*, 2023.

Xinyuan Luo, Shengmiao Jin, Hung-Jui Huang, and Wenzhen Yuan. An intelligent robotic system for perceptive pancake batter stirring and precise pouring. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5970–5977. IEEE, 2024.

Hans Moravec. *Mind children: The future of robot and human intelligence*. Harvard University Press, 1988.

Shotaro Mori, Kazutoshi Tanaka, Satoshi Nishikawa, Ryuma Niiyama, and Yasuo Kuniyoshi. High-speed humanoid robot arm for badminton using pneumatic-electric hybrid actuators. *IEEE Robotics and Automation Letters*, 4(4):3601–3608, 2019.

Urs Muller, Jan Ben, Eric Cosatto, Beat Flepp, and Yann Cun. Off-road obstacle avoidance through end-to-end learning. *Advances in neural information processing systems*, 18, 2005.

Katharina Mülling, Jens Kober, and Jan Peters. A biomimetic approach to robot table tennis. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1921–1926. IEEE, 2010.

Katharina Mülling, Jens Kober, Oliver Kroemer, and Jan Peters. Learning to select and generalize striking movements in robot table tennis. *The International Journal of Robotics Research*, 32(3): 263–279, 2013.

Ramkumar Natarajan, Hanlan Yang, Qintong Xie, Yash Oza, Manash Pratim Das, Fahad Islam, Muhammad Suhail Saleem, Howie Choset, and Maxim Likhachev. Preprocessing-based kinodynamic motion planning framework for intercepting projectiles using a robot manipulator. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10910–10916. IEEE, 2024.

Quan Nguyen, Ayush Agrawal, Xingye Da, William C Martin, Hartmut Geyer, Jessy W Grizzle, and Koushil Sreenath. Dynamic walking on randomly-varying discrete terrain with one-step preview. In *Robotics: Science and Systems*, volume 2, pp. 384–99, 2017.

Yunpeng Pan, Ching-An Cheng, Kamil Saigol, Keuntaek Lee, Xinyan Yan, Evangelos Theodorou, and Byron Boots. Agile autonomous driving using end-to-end deep imitation learning. *arXiv preprint arXiv:1709.07174*, 2017.

Yunpeng Pan, Ching-An Cheng, Kamil Saigol, Keuntaek Lee, Xinyan Yan, Evangelos A Theodorou, and Byron Boots. Imitation learning for agile autonomous driving. *The International Journal of Robotics Research*, 39(2-3):286–302, 2020.

pingpong. Ping pong detection dataset. `https://universe.roboflow.com/pingpong-ojuhj/ping-pong-detection-0guzq`, may 2024. URL `https://universe.roboflow.com/pingpong-ojuhj/ping-pong-detection-0guzq`. visited on 2025-05-12.

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

Lin Shao, Fabio Ferreira, Mikael Jorda, Varun Nambiar, Jianlan Luo, Eugen Solowjow, Juan Aparicio Ojea, Oussama Khatib, and Jeannette Bohg. Unigrasp: Learning a unified model to grasp with multifingered robotic hands. *IEEE Robotics and Automation Letters*, 5(2):2286–2293, 2020.

Luis Angel Silva, José Maria Sebastián, R Saltaren, Rafael Aracil, and Jose Sanpedro. Robotenis: optimal design of a parallel robot with high performance. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2134–2139. IEEE, 2005.

Zhi Su, Bike Zhang, Nima Rahmanian, Yuman Gao, Qiayuan Liao, Caitlin Regan, Koushil Sreenath, and S Shankar Sastry. Hitter: A humanoid table tennis robot via hierarchical planning and learning. *arXiv preprint arXiv:2508.21043*, 2025.

Kazuma Takada, Midori Kawaguchi, Akira Uehara, Yukiya Nakanishi, Mark Armstrong, Adrien Verhulst, Kouta Minamizawa, and Shunichi Kasahara. Parallel ping-pong: Exploring parallel embodiment through multiple bodies by a single user. In *Proceedings of the Augmented Humans International Conference 2022*, pp. 121–130, 2022.

Jie Tan, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. *arXiv preprint arXiv:1804.10332*, 2018.

Jonas Tebbe, Lukas Krauch, Yapeng Gao, and Andreas Zell. Sample-efficient reinforcement learning in robotic table tennis. In *ICRA*, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

An Dinh Vuong, Minh Nhat Vu, Hieu Le, Baoru Huang, Huynh Thi Thanh Binh, Thieu Vo, Andreas Kugi, and Anh Nguyen. Grasp-anything: Large-scale grasp dataset from foundation models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14030–14037. IEEE, 2024.

Jack Ng Kok Wah. The rise of robotics and ai-assisted surgery in modern healthcare. *Journal of Robotic Surgery*, 19(1):311, 2025.

Yilei Wang and Ling Wang. [retracted] machine vision-based ping pong ball rotation trajectory tracking algorithm. *Computational Intelligence and Neuroscience*, 2022(1):3835649, 2022.

Zhongli Wang, Guohui Tian, and Shijie Guo. Gmm enabled by multimodal information fusion network for detection and motion planning of robotic liquid pouring. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

Ruihai Wu, Haoran Lu, Yiyan Wang, Yubo Wang, and Hao Dong. Unigarmentmanip: A unified framework for category-level garment manipulation via dense visual correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16340–16350, 2024.

Ping Yang, De Xu, Huawei Wang, and Zhengtao Zhang. Control system design for a 5-dof table tennis robot. In *2010 11th International Conference on Control Automation Robotics & Vision*, pp. 1731–1735. IEEE, 2010.

Andy Zeng, Shuran Song, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Tossingbot: Learning to throw arbitrary objects with residual physics. *IEEE Transactions on Robotics*, 36(4): 1307–1319, 2020.

Chong Zhang, Wenli Xiao, Tairan He, and Guanya Shi. Wococo: Learning whole-body humanoid control with sequential contacts. *arXiv e-prints*, pp. arXiv–2406, 2024a.

Kun Zhang, Zhiqiang Cao, Jianran Liu, Zaojun Fang, and Min Tan. Real-time visual measurement with opponent hitting behavior for table tennis robot. *IEEE Transactions on Instrumentation and Measurement*, 67(4):811–820, 2018.

Yuanhang Zhang, Tianhai Liang, Zhenyang Chen, Yanjie Ze, and Huazhe Xu. Catch it! learning to catch in flight with mobile dexterous hands. *arXiv preprint arXiv:2409.10319*, 2024b.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

Zida Zhao, Haodong Huang, Shilong Sun, Jing Jin, and Wenjie Lu. Reinforcement learning for dynamic task execution: A robotic arm for playing table tennis. In *2024 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 608–613. IEEE, 2024.

Yichao Zhong, Chong Zhang, Tairan He, and Guanya Shi. Bridging adaptivity and safety: Learning agile collision-free locomotion across varied physics. *arXiv preprint arXiv:2501.04276*, 2025.

A Overview

Due to space limitations, we provide comprehensive implementation details and additional experimental validation in the appendix. Section B presents our complete system architecture, including the ball launcher, multi-camera vision setup, and robotic execution components. Section C details our real-time perception pipeline with YOLOv4-tiny ball detection, physics-based filtering, and trajectory prediction. Section D describes our two specialized datasets for the Fast-Slow System and IMPACT models. Section E provides training specifications. Section F presents extended validation, including ultra-high precision evaluation and human demonstration integration. Section G provides a comprehensive failure case analysis to identify system limitations and improvement opportunities. Section H discusses current system limitations and outlines future research directions for enhanced spin modeling, human player adaptability, and strategic gameplay planning. Section I provides a transparent disclosure of the limited use of Large Language Models in manuscript preparation. Additionally, we provide supplementary videos demonstrating the robotic system's real-time ball-hitting performance and trajectory interception capabilities.

B System Overview

Our system consists of a ball launcher, high-speed camera, depth camera, RGB camera, and robotic arm, designed to achieve an automated table tennis playing system.

B.1 Ball Launching System

We employ the intelligent table tennis robot PONGBOT NOVA as our ball launching system. This table-mounted launcher can generate topspin and backspin, with precise landing point control ranging from -2 to +2, and adjustable ball speed between levels 1-3, providing stable and controllable ball trajectories for our experiments.

B.2 Vision System

The vision system comprises three cameras, each dedicated to different tasks:

- **High-Frequency Camera**: Spike M1K40-H2-Gen3 with a resolution of 1000×1000 and a maximum frame rate of 20,000 fps. In our experiments, we set a step size of 200, saving 100 images per second, specifically designed to capture the instantaneous contact between the ball and the paddle, providing high temporal resolution image data for stroke analysis.
- **Depth Camera**: Intel RealSense D455 with a resolution of 640×480 and a frame rate of 60Hz. This camera is calibrated using ArUco markers and employs YOLO (bubbliiiing, 2020) model for ball detection. The system transforms detected pixel coordinates into world coordinates through intrinsic and extrinsic parameters, enabling real-time tracking of the ball's position.
- **RGB Camera**: Intel RealSense LiDAR Camera L515 with a resolution of 960×540 and a frame rate of 60Hz. It primarily detects the landing position of the ball after being hit by the robot, providing feedback for evaluating stroke effectiveness.

Figure 5 illustrates the visual comparison between spike camera and conventional RGB camera captures. Due to its ultra-high frame rate, the spike camera eliminates motion blur entirely, providing crisp imagery of fast-moving objects that would appear blurred in standard cameras, which is crucial for precise ball trajectory analysis during high-speed table tennis gameplay.

B.3 Mechanical Execution System

The execution system utilizes an ABB IRB 120 6 DoF (six-degree-of-freedom) robotic arm with maximum joint rotation speeds of 250°/s, 250°/s, 250°/s, 320°/s, 320°/s, and 420°/s, respectively. The arm is equipped with a standard table tennis paddle at its end-effector to execute the optimal hitting motion calculated by the system.

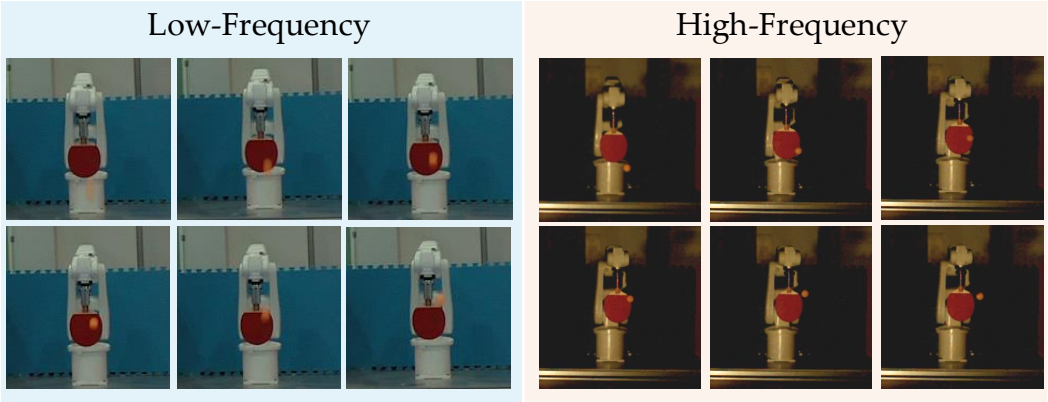

Figure 5: **Comparison of ball capture qualit.** Conventional RGB camera with motion blur at 60 fps, Spike camera with crisp imagery at 20,000 fps, demonstrating the advantage of ultra-high frame rate for fast-moving object detection.

This system achieves real-time tracking and precise hitting of table tennis balls through the tight integration of visual perception, trajectory prediction, and motion planning, providing a comprehensive experimental platform for table tennis robotics research.

C Details of Ball Detection and Trajectory Prediction

C.1 Ball Detection

In the context of high-speed robotic table tennis, where accurate timing and spatial awareness are critical, a high-frequency vision system is required to continuously track the ball's position for real-time trajectory estimation and manipulation control. Thus, we chose YOLOv4-tiny (bubbliiiing, 2020) due to its lightweight design and computational efficiency, enabling a detection frequency of up to **150 Hz**, which is crucial for the precise and timely interaction with the fast-moving ball. The initial phase of our research involves the supervised training of the detection model using publicly available datasets: Roboflow (Alexandrova et al., 2015), TT2 (desigproject, 2022), and Ping Pong Detection (pingpong, 2024). A two-phase training strategy is implemented to enhance model robustness. In the first phase, the model is trained on the complete dataset. In the second phase, samples that were misdetected during the initial training are selectively sampled and assigned higher weights for fine-tuning, with the goal of improving the detector's accuracy on difficult instances. Our empirical validation is conducted using an Intel RealSense D455 RGB-D camera, where we employ the optimized lightweight detection model. This setup achieves a detection accuracy that exceeds 99.8%, facilitating rapid and precise 2D positional detection. The results are shown in Figure 6.

C.2 Motion State Estimation Filter

To accurately track the ball's motion state, we designed a filtering algorithm that combines physical models with measurement data. This algorithm not only smooths noise in the ball position data but also provides accurate velocity estimates, establishing a foundation for subsequent trajectory prediction and interception planning.

Exponential Moving Average Filter We implemented a physics-based Exponential Moving Average (EMA) filter that simultaneously estimates both position and velocity by combining system dynamics models with real-time observation data. The core formula of the filter is:

$$\hat{x}_t = (1 - \alpha) \cdot f(\hat{x}_{t-1}) + \alpha \cdot z_t \tag{5}$$

where $\hat{x}_t$ is the current state estimate, $f(\hat{x}_{t-1})$ is the dynamics prediction based on the previous state estimate, z_t is the current observation, α is the mixing constant that determines the weight ratio between observation and prediction.

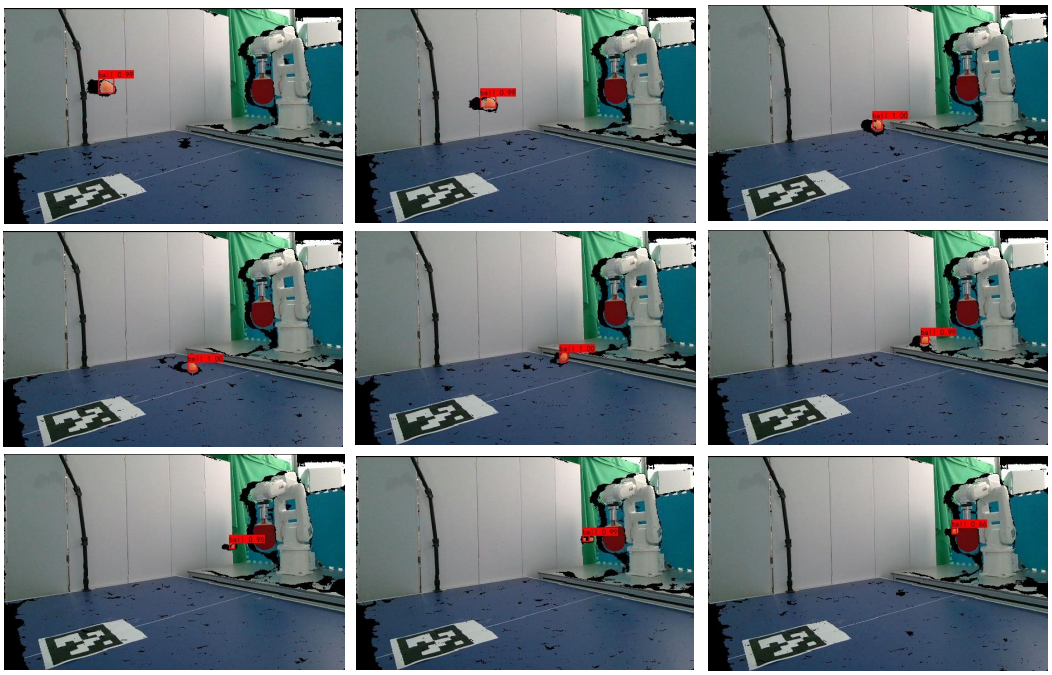

Figure 6: **Visualization of ball detection.**The sequence shows the ball's trajectory at different time points with accurately placed bounding boxes around the detected ball. Our YOLOv4-tiny model consistently identifies the ball's position even during high-speed motion, demonstrating the robustness of our detection approach under various lighting conditions and ball velocities.

For the ball's free-fall motion, we adopted standard ballistic equations as the dynamics model. Specifically, the prediction equations for position and velocity are:

$$\hat{p}_t = \hat{p}_{t-1} + \hat{v}_{t-1} \cdot \Delta t + \frac{1}{2} a \cdot \Delta t^2, \tag{6}$$

$$\hat{v}_t = \hat{v}_{t-1} + a \cdot \Delta t. \tag{7}$$

where, $\hat{p}_t$ is the position estimate, $\hat{v}_t$ is the velocity estimate, Δt is the time step, a is the acceleration (gravity acceleration -9.81 m/s² in the z-direction, 0 in x and y directions).

Considering the different motion characteristics of the ball in different directions, we applied different mixing constants for the x, y, and z directions: $\alpha_x = 0.15, \alpha_y = 0.15, \alpha_z = 0.25$. The larger mixing constant for the z-direction was chosen to better adapt to the faster velocity changes in the vertical direction due to gravity. Additionally, we implemented special handling for the ascending phase in the z-direction to more accurately capture the motion characteristics of the initial segment of the parabolic trajectory.

Implementation Details The system maintains a fixed-length (10 frames) history data queue for calculating initial velocity and handling data interruptions. When a data stream interruption exceeding 1 second is detected, the filter automatically resets the historical data to avoid the influence of outdated information on current estimates. This adaptive mechanism enables the system to quickly resume normal operation when the ball reappears or a new throw begins.

Performance Visualization Experiments show that this filter effectively smooths noise in the raw position data while providing accurate velocity estimates. Since the ball detection algorithm already provides relatively accurate position information, the filter primarily serves to refine and stabilize estimates, particularly excelling in velocity calculation. The filtered trajectory data exhibits smooth parabolic characteristics consistent with physical laws, providing a reliable foundation for subsequent trajectory prediction as shown in Figure 7.

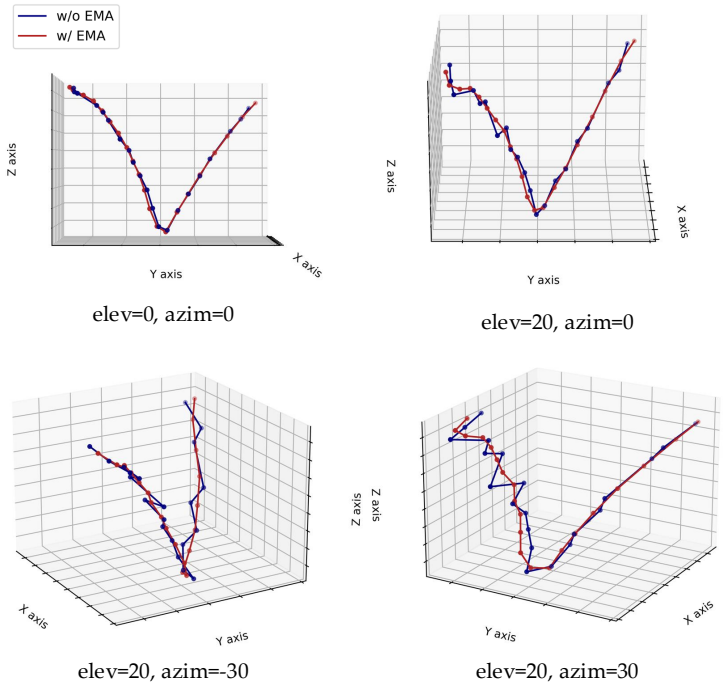

Figure 7: **Comparison of ball trajectory with and without EMA filtering.** The figure displays trajectories from multiple viewing angles, with red curves showing raw detection data and blue curves showing EMA-filtered results. The filtering effectively removes noise while maintaining the ball's natural parabolic motion, providing more reliable data for trajectory prediction.

C.3 Hittable Position Prediction

To enable the robot to intercept the ball effectively, we developed a trajectory prediction algorithm that estimates where and when the ball will reach a hittable position. This algorithm leverages the filtered position and velocity data from our EMA filter to project the ball's future path.

Ballistic Model Implementation Our prediction algorithm employs a simplified ballistic model that accounts for gravitational acceleration (9.81 m/s²), initial position and velocity vectors, and coefficient of restitution for potential bounces. The model deliberately excludes spin effects and complex aerodynamics to maintain computational efficiency and implementation simplicity across both simulation and real-world environments.

Trajectory Prediction The prediction process is implemented using the following Algorithm 1. This algorithm applies the ballistic equations to project the ball's trajectory forward in time and determine exactly when and where it will intersect the hitting plane. The computed hittable position, impact velocity, and arrival time are then passed to the robot control system for interception planning.

The prediction algorithm includes special handling for various scenarios: trajectories that never intersect the hitting plane, balls moving away from the hitting plane, multiple potential intersections (selecting the earliest valid one), and bounces off other surfaces before reaching the hitting plane.

Integration with Robot Control The predicted hittable position, impact velocity, and arrival time are continuously updated and provided to the robot control system, enabling it to plan and execute appropriate interception movements. This real-time prediction allows the robot to adjust its position and orientation to successfully hit the ball even when the ball follows an unexpected trajectory.

Algorithm 1 Predict Hittable Position

Require: p_0 (initial position), v_0 (initial velocity), $t_{current}$ (current time), y_{plane} (hitting plane y-coordinate)
Ensure: Returns hit position, velocity and arrival time, or null if not hittable

$g \leftarrow (0, 0, -9.81)$ Gravity vector
if $|v_0.y| < \epsilon$ **then**
    Ball moving parallel to hitting plane
    return null
end if
$t_{hit} \leftarrow (y_{plane} - p_0.y)/v_0.y$
if $t_{hit} < 0$ **then**
    Ball already passed the plane
    return null
end if
$p_{hit} \leftarrow p_0 + v_0 \cdot t_{hit} + \frac{1}{2} g \cdot t_{hit}^2$
$p_{hit}.y \leftarrow y_{plane}$ Ensure exact y-coordinate
$v_{hit} \leftarrow v_0 + g \cdot t_{hit}$
$t_{arrival} \leftarrow t_{current} + t_{hit}$
return $(p_{hit}, v_{hit}, t_{arrival})$

D Dataset Description

This section details the two critical datasets developed for our table tennis robot system. These datasets support the system's two core components: the Fast-Slow System, focused on precise paddle-contact control, and the IMPACT model (Imitation-based Motion Planning And Control Technology), focused on effective hitting strategies. Below, we describe the collection processes, annotation methods, and how these datasets provide the foundation for the system's performance.

D.1 Dataset for Fast-Slow System

D.1.1 Data Collection Process

Our data collection process involves the following steps: The ball launcher randomly serves balls within a predetermined range. Based on the Ball Detection and Trajectory Prediction framework described above, we use the Intel RealSense D455 camera to record trajectory information, including position and velocity vectors. Once a hittable position is predicted, this coordinate is transformed into the robotic arm's base coordinate system. Using PyBullet, we compute the inverse kinematics to determine the joint values required to move to this position, which are then sent to the robotic arm via an EGM Controller. Leveraging the high frame rate capability of the Spike camera, we capture images of the exact moment of contact between the ball and the paddle. This process is repeated multiple times to build a comprehensive dataset.

D.1.2 Data Annotation Process

Pixel-to-Real-World Conversion A critical step in our data annotation process was establishing an accurate conversion ratio between pixel measurements in images and real-world dimensions. We developed a systematic approach using the known dimensions of the table tennis paddle as a reference.

The process involves the following steps:

1. **Image Acquisition**: We captured multiple high-resolution images of the table tennis paddle using the Spike camera.
2. **Image Preprocessing**: Each image undergoes preprocessing to reduce noise and enhance contrast, facilitating more accurate edge detection. We apply a center-crop operation to focus on the region containing the paddle.

3. **Paddle Detection**: Using color-based segmentation, we isolate the paddle from the background by defining a target color range (RGB: 65, 31, 31) with an appropriate tolerance value. This creates a binary mask representing the paddle area.
4. **Morphological Operations**: To refine the mask, we apply morphological operations including opening (to remove small noise artifacts) and closing (to fill small holes), using a 5×5 kernel.
5. **Connected Component Analysis**: We identify the largest connected component in the mask, which corresponds to the paddle, and filter out smaller noise components.
6. **Minimum Area Rectangle Fitting**: For the detected paddle region, we compute the minimum area rectangle that encloses the paddle contour, providing us with the paddle's pixel dimensions (width and height).
7. **Conversion Ratio Calculation**: Knowing the actual paddle dimensions (150mm × 150mm), we calculate the conversion ratio for both width and height:

$$\text{mm\_per\_pixel\_width} = \frac{\text{real\_width\_mm}}{\text{width\_pixels}} \tag{8}$$

$$\text{mm\_per\_pixel\_height} = \frac{\text{real\_height\_mm}}{\text{height\_pixels}} \tag{9}$$

8. **Average Conversion Ratio**: To improve accuracy, we average the width and height conversion ratios:

$$\text{mm\_per\_pixel\_avg} = \frac{\text{mm\_per\_pixel\_width} + \text{mm\_per\_pixel\_height}}{2} \tag{10}$$

9. **Multiple Image Processing**: We repeat this process across multiple images and compute the overall average conversion ratio to minimize measurement errors.

This methodology yielded a reliable pixel-to-millimeter conversion factor that was subsequently used throughout our data annotation pipeline to transform pixel coordinates in images to real-world spatial coordinates.

D.1.3 BALL-PADDLE CONTACT DETECTION

After establishing the pixel-to-millimeter conversion ratio, we developed a comprehensive algorithm to detect and analyze the contact between the table tennis ball and paddle in high-speed scenarios. This process is particularly challenging due to the rapid nature of the contact event, which typically occurs within milliseconds.

Our detection pipeline consists of the following key components:

1. **High Temporal Resolution Acquisition**: The Spike camera captures the ball-paddle interaction with exceptional temporal precision (microsecond-level), providing detailed information about the contact dynamics that conventional cameras would miss. The neuromorphic vision sensor outputs asynchronous spike signals rather than traditional frame-based images, enabling ultra-high temporal resolution.
2. **Ball Detection Strategy**: We implemented a dual-approach ball detection method:
   - **Hole-based Detection**: The primary method identifies the ball as a hole or void within the paddle region during contact. This approach is particularly effective when the ball partially occludes the paddle.
   - **Color-based Detection**: As a complementary approach, we detect the ball using a predefined color range (RGB: 79, 58, 34) with appropriate tolerance values, creating a binary mask for potential ball regions.
3. **Circularity Filtering**: To distinguish the ball from other objects or noise, we apply a circularity measure to each detected region:

$$\text{Circularity} = \frac{4\pi \times \text{Area}}{\text{Perimeter}^2} \tag{11}$$

   Regions with circularity above 0.6 are considered potential ball candidates, as table tennis balls maintain their circular appearance even during high-speed motion.

4. **Proximity Analysis**: We prioritize detected ball regions that are within a 20-pixel radius of the paddle's edge, as these are most likely to represent actual contact points.
5. **Temporal Sequence Analysis**: By analyzing the sequential spike signals from the neuromorphic camera, we track the ball's trajectory before, during, and after contact with the paddle. This allows us to determine the exact moment of impact with microsecond precision.
6. **Coordinate System Transformation**: After detecting both the paddle and ball, we establish a paddle-centered coordinate system:
   - Origin: Center of the paddle
   - X-axis: Horizontal direction (positive rightward)
   - Y-axis: Vertical direction (positive upward)
7. **Contact Point Calculation**: The ball's position is transformed from pixel coordinates to this paddle-centered coordinate system and then converted to physical units (millimeters) using our established conversion ratio:

$$\mathrm{x}_{\mathrm{mm}} = (x_{\mathrm{ball}} - x_{\mathrm{paddle}}) \times \mathrm{mm\_per\_pixel} \tag{12}$$

$$\mathrm{y}_{\mathrm{mm}} = (y_{\mathrm{paddle}} - y_{\mathrm{ball}}) \times \mathrm{mm\_per\_pixel} \tag{13}$$

This ball-paddle contact detection methodology provides unprecedented insights into the dynamics of table tennis interactions. By leveraging the unique capabilities of neuromorphic vision sensors, we can capture and analyze high-speed interactions that would be impossible to observe with conventional imaging systems. The resulting data enables quantitative analysis of contact timing, location, and dynamics, which can be valuable for both sports science research and athlete training applications.

D.2 Dataset for IMPACT

D.2.1 Data Collection Process

After successfully training the Fast-Slow System to ensure precise ball-paddle contact near the center of the paddle, we extended our data collection to include the hitting phase. Building upon our established interception capabilities, we implemented a structured process to collect data on effective hitting techniques. For the hitting phase data collection, we augmented our previous methodology with the following approach:

1. **Randomized Joint Angle Sampling**: To generate diverse hitting patterns, we implemented controlled random sampling on three critical joint axes:
   - Axis 3 (shoulder joint): Base angle of 15.0 degrees with random variation of ±10.0 degrees
   - Axis 5 (elbow joint): Base angle of 60.0 degrees with random variation of ±10.0 degrees
   - Axis 6 (wrist joint): Base angle of 0 degrees with random variation of ±20.0 degrees
2. **Hitting Execution**: For each trial, the robotic arm would:
   - First, intercept the ball using the Fast-Slow System's prediction
   - Apply the randomly generated joint angles at the moment of contact
   - Execute the hitting motion to return the ball to the opponent's side
3. **Outcome Recording**: We recorded whether the ball successfully landed on the opponent's side of the table, along with the precise landing location.

This approach allowed us to collect data on effective hitting strategies while leveraging our previously established ball interception capabilities. By systematically varying the joint angles during the hitting phase, we were able to explore a wide range of possible returns, creating a comprehensive dataset that captures the relationship between joint movements and resulting ball trajectories.

D.2.2 Data Annotation Process

For data annotation, we used an Intel RealSense L515 camera to record the landing position of the ball. We divided the opponent's side of the table into four quadrants (labeled A, B, C, and D) and encoded these landing zones using one-hot encoding. This encoded landing position information was incorporated as part of the input data, while the randomly sampled joint angles were used as labels. Using this structured dataset, we employed imitation learning techniques to train our IMPACT model, enabling it to learn the relationship between desired landing positions and the required joint movements to achieve them.

E Training Details.

Both System 2 and IMPACT modules were trained for 2000 epochs using the Adam optimizer with an initial learning rate of 1e-3 and cosine annealing schedule. System 2 used a batch size of 32 with K=10 consecutive frames as input for trajectory prediction. The IMPACT module employed a batch size of 4 to handle the complexity of return planning tasks. All positional inputs were normalized to the range [0,1], while standard scaling was applied to velocity measurements to ensure numerical stability across varying ball conditions. Training was conducted on a workstation equipped with an NVIDIA RTX 4090 GPU.

F Additional real-world experiments

F.1 Ultra-High Precision Evaluation

To further validate our system's precision capabilities and address questions regarding our target zone selection, we conducted additional experiments using 10cm radius targets following the methodology of Büchler et al. (Buchler et al., 2022).

Experimental Setup. We evaluated our system using 10cm radius target zones across the same four regions (A, B, C, D) used in our main experiments. This ultra-high precision threshold represents only 1.5% of the reachable table area, providing an extremely challenging benchmark for precision assessment.

Table 8: Ultra-high precision evaluation results with 10cm radius targets

Method	**A(%)**	**B(%)**	**C(%)**	**D(%)**	**AVG.(%)**
HYSR (Buchler et al., 2022)	-	-	-	-	8
SpikePingpong	31±4	32±3	29±3	35±2	31±3

Results and Analysis. Table 8 presents the results of our ultra-high precision evaluation. Our system achieves an average success rate of 31±3% within the 10cm target zones, substantially outperforming the 8% success rate reported by Büchler et al. (Buchler et al., 2022) using similar target specifications.

Discussion. These results validate our choice of 20cm and 30cm target zones as meaningful precision benchmarks while demonstrating that our system maintains reasonable performance even at ultra-high precision levels. The progressive degradation from 93% (30cm) to 70% (20cm) to 31% (10cm) reflects the inherent challenges of precise ball placement in table tennis, where even human players average only 53% success in 30cm zones. Our system's ability to achieve 31% success in 10cm targets represents a significant advancement in robotic table tennis precision control.

G Failure Case Analysis

To gain deeper insights into our system's limitations and identify areas for improvement, we conducted a comprehensive failure analysis by categorizing all unsuccessful attempts during our precision evaluation experiments.

Table 9: Distribution of failure types in precision targeting experiments

Failure Type	Percentage
Ball fails to cross net	4.6%
Correct quadrant, outside 30cm circle	79.1%
Wrong target quadrant	12.5%
Ball fails to land on table	3.8%

Results. Table 9 presents the distribution of failure types observed in our experiments. The analysis reveals that the majority of failures (79.1%) are near-misses where the system correctly identifies the target region but falls short of the required precision threshold.

Analysis and Implications. The failure distribution indicates robust basic control capabilities with specific areas for improvement:

- **Near-miss failures (79.1%):** These cases demonstrate that our system successfully executes the fundamental task of directing the ball toward the intended table region but requires enhanced precision in fine-grained control. This suggests that improvements in trajectory optimization and control parameter tuning could yield significant performance gains.
- **Strategic errors (12.5%):** Wrong quadrant targeting indicates occasional failures in high-level decision making, potentially due to perception errors or planning inconsistencies under challenging conditions.
- **Fundamental control failures (8.4%):** The combined percentage of balls failing to cross the net or land on the table represents basic execution errors, suggesting room for improvement in fundamental trajectory planning and power control.

Root Cause Analysis of Near-Misses. A deeper causal analysis reveals that the prevalence of "near-misses" stems from two primary root causes:

- **Unmodeled Spin Dynamics:** The most significant challenge is unmodeled ball spin. Extreme spin introduces complex aerodynamic effects (the Magnus effect) and alters bounce characteristics in ways our physics-based model does not capture. This can create small but critical, centimeter-level residual errors in our final trajectory prediction, often causing the ball to land just outside the target zone.
- **High Spatio-Temporal Sensitivity:** The task exhibits extreme sensitivity to the precise timing and location of the ball-paddle contact. Even minor variations in the robot's arrival time at the contact point can subtly alter the impact dynamics and the resulting ball trajectory. This high sensitivity means that even when the overall strategy is correct, slight execution imprecision can lead to a near-miss.

This analysis confirms that our system demonstrates robust basic control with the primary limitation being precision refinement rather than fundamental control failures, providing clear direction for future system enhancements.

H Limitations and Future Work

H.1 Current Limitations

Despite the achievements demonstrated in this work, our ***SpikePingpong*** system has several limitations that present opportunities for future enhancement:

Ball Spin Modeling: Our current system does not account for ball spin, which significantly affects optimal interception strategies for different spin types. The Magnus effect and varying bounce characteristics of topspin, backspin, and sidespin balls can lead to trajectory deviations that our physics-based models do not capture.

Human Player Adaptability: Performance against human players remains challenging due to the complex and unpredictable trajectories they generate compared to our controlled training conditions. Human players exhibit diverse playing styles, strategic variations, and adaptive behaviors that exceed the scope of our current training data.

H.2 Future Research Directions

Building upon the foundation established by ***SpikePingpong***, we identify several promising directions for future research:

Spin Dynamics Integration: Incorporating comprehensive spin modeling into both the Fast-Slow system architecture and IMPACT module to handle the full spectrum of ball spin effects on trajectory prediction and strategic planning.

Adaptive Learning Framework: Enhancing the system's adaptability to diverse playing styles through online learning mechanisms that can adjust to opponent strategies and playing patterns in real-time.

Strategic Gameplay Planning: Developing advanced strategic planning capabilities for human-robot interaction, including opponent modeling, tactical sequence planning, and adaptive game strategy formulation.

I Use of Large Language Models

Large Language Models (LLMs) were used in a limited capacity during the preparation of this manuscript. Specifically, LLMs were employed solely for:

- Grammar correction and proofreading
- Sentence structure improvement and clarity enhancement
- Minor stylistic refinements to improve readability

LLMs were not involved in research ideation, experimental design, data analysis, or the generation of scientific content. All technical contributions, methodological innovations, experimental results, and scientific insights presented in this work are entirely the product of the authors' original research.