BADMINTONLAB: A UNIFIED PHYSICS-BASED PLATFORM FOR HUMANOID BADMINTON LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in embodied intelligence are pushing humanoid robots beyond locomotion toward dynamic tasks that require active interaction with the physical world. Racket sports provide a particularly demanding setting, where control must be concentrated within a brief window of precise contact, while the effectiveness of each stroke becomes evident only from the ensuing trajectory. Badminton makes these challenges especially pronounced due to the fast and highly dynamic flight of the shuttlecock and the stringent spatiotemporal precision required for racket control. We introduce BadmintonLab, a unified physics-based platform that provides three progressively richer learning settings for task-oriented skill learning, reference-guided style learning, and competitive rally learning. To enable effective learning across these settings, the platform employs compact event-driven task objectives that provide efficient learning signals for racket–shuttle interaction and shot outcomes. Experiments demonstrate distinct badminton skills, reference-guided motion styles, and competitive rally learning, and further show reuse of the same task and learning interfaces across different humanoid embodiments.

1 INTRODUCTION

Embodied intelligence is moving humanoid robots beyond locomotion toward tasks that require coordinated whole-body interaction with a dynamic physical world (Cheng et al., 2024). Recent progress in dynamic locomotion, physics-based skill learning, and robotic sports reflects this transition (Peng et al., 2018; Haarnoja et al., 2024). Racket sports provide a particularly demanding setting for such interaction: effective control must be concentrated within a brief window of precise racket–object contact, while the quality of each stroke becomes evident only through the subsequent object trajectory (D'Ambrosio et al., 2025). Badminton makes these challenges especially pronounced. A robot must continuously track the rapidly evolving shuttle trajectory, reposition and stabilize its body, coordinate whole-body motion with racket control, and establish accurate racket–shuttle contact within a narrow spatiotemporal window (Ma et al., 2025). Whether that interaction is successful, however, can only be assessed from the ensuing shuttle flight and its eventual outcome on the court. These characteristics make badminton a compelling setting for studying dynamic, contact-driven embodied interaction, rather than merely another application of sports robotics.

Recent robotic badminton systems have demonstrated substantial progress in dynamic shuttle tracking and striking (Yuan et al., 2025; Ma et al., 2025; Chen et al., 2026; Liu et al., 2025). Early systems based on robotic arms or legged manipulators achieve effective interception and return behaviors (Yuan et al., 2025; Ma et al., 2025), but do not address the balance and tightly coupled whole-body coordination required by humanoid embodiments. Liu et al. (2025) demonstrate coordinated humanoid locomotion and striking for goal-directed returns, but focus primarily on stroke execution rather than style learning or competitive rally learning. Chen et al. (2026) further incorporate motion references for human-like stroke learning, but rely on a progressive imitation-to-interaction pipeline with explicitly time-conditioned objectives.

We introduce BadmintonLab, a unified physics-based platform for embodied badminton learning with humanoid robots. It provides a shared simulation substrate and an event-driven task-objective interface across three progressively richer learning settings: task-oriented skill learning, reference-guided style learning, and competitive rally learning. The first setting focuses on acquiring physically effective badminton skills from task objectives, the second incorporates motion references to

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

shape how those skills are performed, and the third extends learning from individual strokes to competitive rally-level behavior. Across these settings, compact event-driven objectives attach learning signals to physically meaningful opportunities and outcomes, including racket–shuttle approach, contact, outgoing shot, net crossing, and landing. The objective specifies the desired physical outcome while leaving the robot free to discover how to prepare and execute the motion. The same formulation can be instantiated across different humanoid embodiments, separating the learning interface from any particular robot morphology.

Our contributions are summarized as follows:

- **A unified platform for multi-level badminton learning.** BadmintonLab supports task-oriented skill learning, reference-guided style learning, and competitive rally learning within a common physics-based simulation framework.
- **Compact event-driven task objectives.** BadmintonLab formulates racket–shuttle interaction and shot outcomes through physically meaningful events and task parameters, without prescribing explicit racket trajectories or precise contact timing.
- **Validation across learning settings and humanoid embodiments.** Experiments demonstrate the autonomous acquisition of distinct badminton skills, reference-guided motion styles, and competitive rally behavior, and further show reuse of the same task and learning interfaces on a second humanoid embodiment.

2 Related Work

2.1 Whole-body control and motion-prior learning

Motion-based control has developed along three related directions. Early imitation methods, represented by DeepMimic, train simulated agents to track reference demonstrations, and have also been applied to agile robot locomotion (Peng et al., 2018; 2020). Style-learning methods instead use learned motion priors, including AMP, SMP, and ASE, to encourage motions consistent with reference data while task objectives specify the desired outcome (Peng et al., 2021; Mu et al., 2026; Peng et al., 2022). More recent systems such as PHC, ExBody, and BeyondMimic extend reference-motion tracking toward diverse whole-body motions and task-directed humanoid control (Luo et al., 2023; Cheng et al., 2024; Liao et al., 2026). BadmintonLab adopts these established techniques: Level 1 studies task-oriented skill learning without reference motions, while Level 2 adds an AMP-based motion prior for reference-guided style learning.

2.2 Learning-based robotic sports

Recent work has taught robots interaction tasks in which brief contact determines a later outcome, including soccer (Haarnoja et al., 2024), tennis (Zhang et al., 2026), table tennis (Mülling et al., 2013; Büchler et al., 2022; D'Ambrosio et al., 2025; Su et al., 2025), and badminton (Yuan et al., 2025; Ma et al., 2025; Liu et al., 2025; Chen et al., 2026). Badminton provides a representative setting, combining fast shuttlecock dynamics, precise racket–shuttle contact, and whole-body repositioning. At ETH Zurich, Ma et al. (2025) learn a whole-body visuomotor policy for a legged manipulator that integrates shuttle tracking, locomotion, and striking. At Phybot, Liu et al. (2025) train a humanoid through a three-stage curriculum for footwork, swing acquisition, and task refinement, demonstrating sim-to-real transfer and rallies. Chen et al. (2026) combine human motion tracking, distillation, AMP, and interaction refinement in a four-stage pipeline, transferring styled badminton skills to an EngineAI humanoid.

These three systems use temporally gated racket-shuttle objectives: Ma and Liu activate swing rewards at commanded interception times, while Chen uses a narrow time-to-hit window. Their target-conditioned controllers include interception timing in the observations, although Liu also evaluates a prediction-free actor. Interception heights are specified through target generation and can vary across targets, as explicitly demonstrated by Chen. BadmintonLab instead uses compact event-driven task objectives: the racket–shuttle approach objective is evaluated during a receive opportunity, and contact and shot outcomes trigger subsequent rewards. No commanded time-to-hit is required, and the policy can discover when and at what height to strike within the task-specified receive interval. This

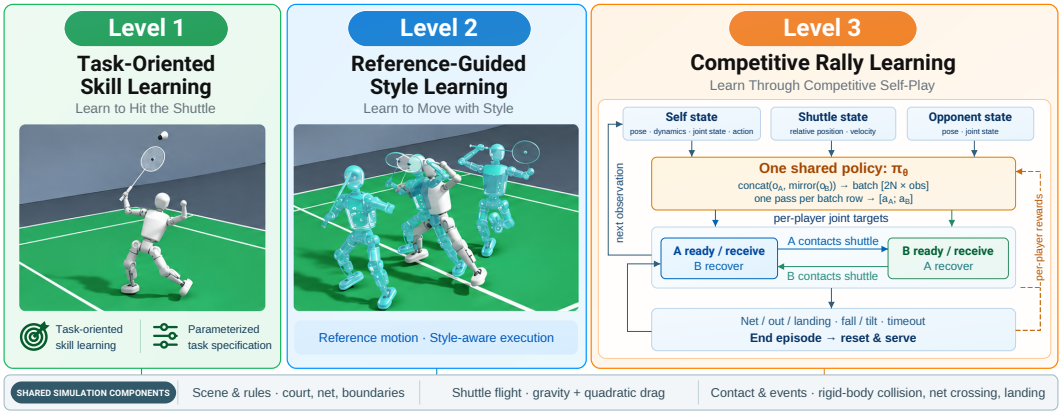

Figure 1: Overview of the BADMINTONLAB. Built on a shared physics-based simulation substrate, the platform supports three progressively richer learning settings: task-oriented skill learning, reference-guided style learning, and competitive rally learning.

formulation supports single-stage training for each skill under a parameterized task specification, as well as reference-guided style learning, without requiring a staged imitation pipeline or teacher–student distillation. The current badminton systems primarily investigate motor-skill acquisition. BadmintonLab additionally exposes competitive rally learning in Level 3.

2.3 COMPETITIVE GAME ENVIRONMENTS AND BADMINTON POLICY LEARNING

Progress in reinforcement learning depends on environments that provide reproducible dynamics, observations, actions, opponents, and evaluation protocols. Representative examples include PySC2 for StarCraft II (Vinyals et al., 2017), OpenSpiel for multi-agent games (Lanctot et al., 2019), Honor of Kings Arena for competitive policy generalization (Wei et al., 2022), and Google Research Football for physics-based football scenarios (Kurach et al., 2020). Together with self-play research (Bansal et al., 2018), these platforms show how a well-defined environment enables systematic study of competitive and multi-agent policies.

Badminton research has developed a complementary rally-level abstraction. CoachAI established a pipeline for match-data collection and tactical analysis (Hsu et al., 2019); ShuttleSet provides expert-annotated stroke-level rallies (Wang et al., 2023); and subsequent work models stroke and movement forecasting (Wang et al., 2022; Chang et al., 2023). Data-driven RL environments then simulate tactical choices and realistic opponents (Huang et al., 2023; Wang et al., 2024), while offline RL learns tactical policies from match data (Liu et al., 2026). ShuttleEnv further provides explicit probabilistic rally dynamics for interactive strategy learning (Li et al., 2026). These systems represent actions and transitions at the stroke or rally level. BadmintonLab Level 3 fills the complementary physics-based gap by coupling two humanoid policies through racket–shuttle contact, shuttle flight, whole-body control, court rules, and rally outcomes, allowing motor skills and competitive rally-level decisions to be studied in the same environment.

3 BADMINTONLAB PLATFORM

BADMINTONLAB separates physical interaction modeling, task-objective specification, and learning problem formulation through three layered components: a shared physics-based simulation substrate, a shared event-driven task-objective interface, and three progressively richer learning settings. Built on the same simulation substrate and task-objective interface, these settings progress from task-oriented skill learning to reference-guided style learning and competitive rally learning as shown in Figure 1.

3.1 SHARED PHYSICS-BASED SIMULATION SUBSTRATE

Scene and rule components. BADMINTONLAB provides a regulation-size court and net, a shuttlecock, and racket-equipped Booster T1 and T2 humanoids. The two embodiments share task interfaces for contact, net crossing, landing, and court boundaries, supporting both single returns and alternating-return rallies. These components separate badminton task definitions from robot morphology. Appendix A details the scene configuration and physical parameters.

Dynamics and contact. Shuttle flight combines gravity with quadratic aerodynamic drag, following standard shuttlecock trajectory models (Le Personnic et al., 2011; Darbois Texier et al., 2012). This simplified model captures speed decay without resolving detailed aerodynamics. Racket–shuttle collisions and contact sensing use the rigid-body simulation infrastructure of Isaac Lab (Mittal et al., 2025), while net faults are detected geometrically at shuttle crossings. A shooting solver generates incoming trajectories from specified start positions, arrival points, and flight times. Together, these mechanisms expose shared physical events for the objectives and termination conditions of all three learning levels.

3.2 SHARED EVENT-DRIVEN TASK-OBJECTIVE INTERFACE

Badminton combines a short, precise contact opportunity with whole-body motion and delayed physical consequences. BADMINTONLAB therefore attaches task objectives to receive opportunities, outgoing shuttle states, net clearance, and landing outcomes without prescribing a full racket trajectory, exact contact time, or time-indexed swing phase. This shared event-driven task-objective interface lets individual learning settings activate or augment the terms required by their task.

A continuous proximity reward over a long horizon may encourage the racket to remain close to the shuttle rather than strike actively. We instead evaluate the best approach within a bounded physical opportunity. For a rollout τ, let $\mathcal{I}_{\mathcal{H}}(\tau)$ be the receive interval during which the incoming shuttle passes through the desired interception-height interval $\mathcal{H} = [H_{\min}, H_{\max}]$. With racket sweet-spot and shuttle positions $\mathbf{p}_t^r$ and $\mathbf{p}_t^s$, the racket–shuttle approach objective minimizes

$$d^*_{\text{app}}(\tau) = \min_{t \in \mathcal{I}_{\mathcal{H}}(\tau)} \|\mathbf{p}_t^r - \mathbf{p}_t^s\|_2 \,. \tag{1}$$

Unlike objectives that specify a contact time or time-indexed motion phase (Yuan et al., 2025; Liu et al., 2025), this formulation requires neither a commanded contact time nor an explicit time-to-hit observation. The shuttle state activates the opportunity, the policy chooses when to intercept within it, and $\mathcal{H}$ remains a task parameter.

Following the interval-approach objective, three event-triggered objectives evaluate progressively later outcomes of the same interaction. At the first post-contact state, the outgoing-shot objective assesses whether the shuttle has sufficient speed and a feasible direction toward the opponent's court. At the first crossing of the net plane, the net-clearance objective evaluates whether the shuttle clears the configured net threshold. Finally, at landing, the target-landing objective measures the planar distance to the target region. Together, these objectives capture the physical progression from racket–shuttle contact to flight, net crossing, and final landing, while leaving preparation, swing generation, and contact timing largely to the policy. Detailed definitions and shaping functions are provided in Appendix B.

3.3 LEVEL 1: TASK-ORIENTED SKILL LEARNING

Level 1 directly instantiates the shared event-driven task-objective interface. In each episode, a single robot receives one incoming shuttle and learns to execute a return without access to a reference motion or a prescribed swing trajectory. Task success is determined solely by the resulting shuttle outcomes. This setting focuses on learning functional badminton motions directly from task objectives, with particular emphasis on whether the robot can autonomously discover active striking behaviors rather than merely making contact with the shuttle.

We represent each badminton skill using a compact task specification

$$\boldsymbol{\theta}_{\text{skill}} \left(\mathcal{D}_{\text{in}},\ \mathcal{H},\ \mathcal{G}\right), \tag{2}$$

where $\mathcal{D}_{\text{in}}$ denotes the incoming-shuttle distribution, $\mathcal{H}$ the desired interception-height interval, and $\mathcal{G}$ the target landing region. The incoming-shuttle distribution is parameterized by ranges over the

shuttle's initial position, arrival point, and flight time. A shooting solver then converts sampled task parameters into dynamically consistent initial shuttle velocities. In the experiments, we train one policy for each selected task specification. Changing $\boldsymbol{\theta}_{\text{skill}}$ alters the desired physical outcome while leaving the robot interface and learning formulation unchanged. For example, a backcourt incoming distribution $\mathcal{D}_{\text{in}}$ combined with a deep target region defines an overhead clear; shifting the target region toward the forecourt yields a drop; and pairing a lower interception-height interval with a deep target region specifies a lift.

3.4 Level 2: Reference-Guided Style Learning

Level 2 introduces an AMP-based motion prior while preserving the same underlying badminton task. The shared task objectives specify what physical outcome should be achieved, whereas the motion prior shapes how the robot executes the corresponding movement. Accordingly, Level 2 reuses the dynamics, observations, actions, interaction events, and task objectives from Level 1, while augmenting the training procedure with Adversarial Motion Priors (AMP) (Peng et al., 2021). Rather than prescribing a time-indexed trajectory for the policy to track, the reference motion defines a distribution over plausible whole-body movements. Task success is still determined by racket–shuttle interaction and the resulting shuttle outcomes. Source motions are obtained from Unity assets and retargeted to the T1 and T2 embodiments using OmniRetarget (Yang et al., 2025) and GMR (Araujo et al., 2025).

3.5 Level 3: Competitive Rally Learning

Level 3 extends isolated stroke execution to a competitive multi-agent rally setting while retaining the shared court, shuttle dynamics, racket interaction, physical event interface, and rally rules. A second humanoid is introduced on the opposite court, and the two agents alternate returns of the same shuttle. Each agent additionally observes the opponent's court position, orientation, and joint configuration, enabling its return behavior to adapt to both the opponent and the evolving rally state. Competitive feedback is provided through opposing win/loss rewards at rally termination, while a rally state machine tracks receiving and recovery phases, determines which agent is eligible to return, verifies legal contacts, and governs rally termination. The shared physical events and event-driven task-objective interface remain available in this setting and can be selected or augmented according to the rally objective. Together, these components support opponent-conditioned behavior and competitive decision-making, including training paradigms such as self-play.

4 Experiments

We organize the experiments around three questions corresponding to the learning settings introduced in Section 3. First, can the event-driven task objectives induce active striking and support distinct functional badminton skills without reference motions? Second, can reference-guided learning shape whole-body motion style while preserving effective racket-shuttle interaction and shot outcomes? Third, can the platform extend beyond isolated stroke execution to competitive rally-level learning? We additionally evaluate whether the same task and learning interfaces can be instantiated on a second humanoid embodiment. Detailed task configurations, training settings, and evaluation protocols are provided in Appendix C.

4.1 Active Striking and Skill Learning with Parameterized Task Specifications

In this experiment, we evaluate whether the Level-1 task formulation can induce active striking behaviors and support reliable execution of distinct badminton skills under parameterized task specifications, including clear, drop, and lift. We train a separate policy for each skill rather than a single task-conditioned multi-skill policy.

Setup and metrics. We train Level-1 policies for three representative badminton skills: clear, drop, and lift. All policies are trained with PPO (Schulman et al., 2017) implemented in RSL-RL (Schwarke et al., 2025) using 4,096 parallel environments at a 50 Hz control frequency for 20,000 training iterations. Clear and drop share the same rear-court incoming-shuttle distribution

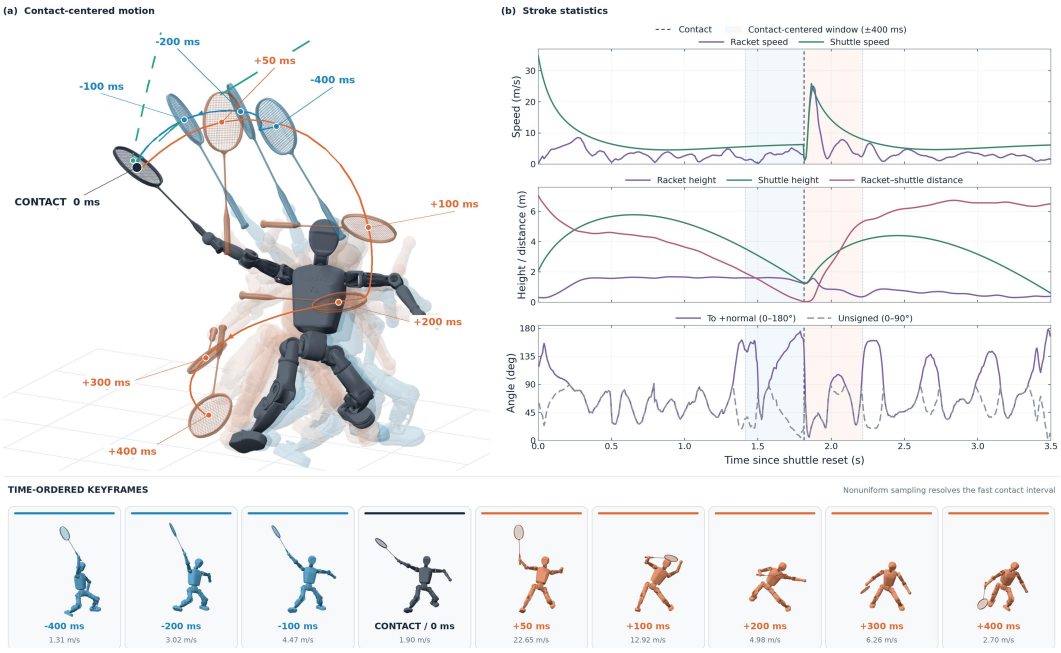

Figure 2: Active striking in a T1 Level-1 rollout. Left and bottom: motion snapshots before the detected contact (blue), at contact (dark), and during follow-through (orange), with timestamps relative to contact. Right: temporal traces from the same rollout.

but differ in their requested landing regions, whereas lift uses a frontcourt incoming distribution and targets the opponent's backcourt. Detailed task parameters and training settings are provided in Appendix C. We evaluate each trained policy deterministically over $409{,}600$ stroke attempts ($4{,}096$ parallel environments for 100 evaluation episodes each). Stroke quality is measured by the racket-shuttle hit rate R_{hit}, valid-return rate R_{valid}, target-landing rate R_{target}, and landing error e_{land}. The first three metrics are computed over all stroke attempts, including misses and faults, while e_{land} measures the mean planar distance from the landing point to the target region over valid returns.

Active striking. To show that the task objective elicits an active swing rather than merely bringing the racket into contact with the shuttle, we collect contact-centered statistics including racket and shuttle speed, contact height, and racket-face orientation, and visualize the learned striking motion as shown in Figure 2. The illustrated T1 rollout shows a preparation motion (backswing), a swing through the contact configuration, and subsequent follow-through. Near contact, the racket-speed transient accompanies a sharp increase in shuttle speed. The descending shuttle meets the racket at a similar height and subsequently rises, while the racket lowers during follow-through. The unsigned angle between racket velocity and the racket-face normal approaches zero around contact, indicating motion approximately perpendicular to the string plane. Together, these traces illustrate coordinated interception and active striking under Level-1 learning without a prescribed trajectory.

Skill learning and stroke quality. We evaluate clear, drop, and lift to examine whether the shared task-oriented learning formulation can support distinct functional skills with reliable physical outcomes when each skill is defined by its own task specification. Detailed task configurations are provided in Appendix B.2. The T1 Level-1 results in Table 1 report hit rate, valid-return rate, target-landing rate, and landing error for the three separately trained policies. Across all skills, the learned policies achieve hit rates close to 100%, while valid-return rates remain around or above 90%, indicating that the agents reliably acquire the basic ability to intercept and return the shuttle. Target-landing rates are lower than the corresponding hit and valid-return rates, reflecting the greater difficulty of controlling the final shot placement. Nevertheless, the landing errors remain relatively small, suggesting that unsuccessful target landings typically deviate only moderately from the desired region. In particular, the lower target-landing rate of the T1 clear is consistent with the configured actuator torque–speed limits of the T1 embodiment, as further analyzed in Appendix F.

Table 1: Single-stroke task performance. R_{hit}, R_{valid}, and R_{target} denote hit, valid-return, and target-landing rates, respectively, and are reported as percentages. e_{land} is the mean planar landing error over valid returns.

Robot	Level	Skill	R_{hit} ↑	R_{valid} ↑	R_{target} ↑	e_{land} (m)↓
T1	1	Clear	99.92	99.88	79.88	0.037
T1	1	Drop	99.97	91.81	82.23	0.005
T1	1	Lift	99.80	89.08	64.65	0.172
T1	2	Clear	99.95	99.93	82.48	0.036
T1	2	Lift	99.82	96.86	56.24	0.182
T2	2	Clear	100.0	100.0	99.2	0.00
T2	2	Lift	99.8	99.6	99.5	0.007

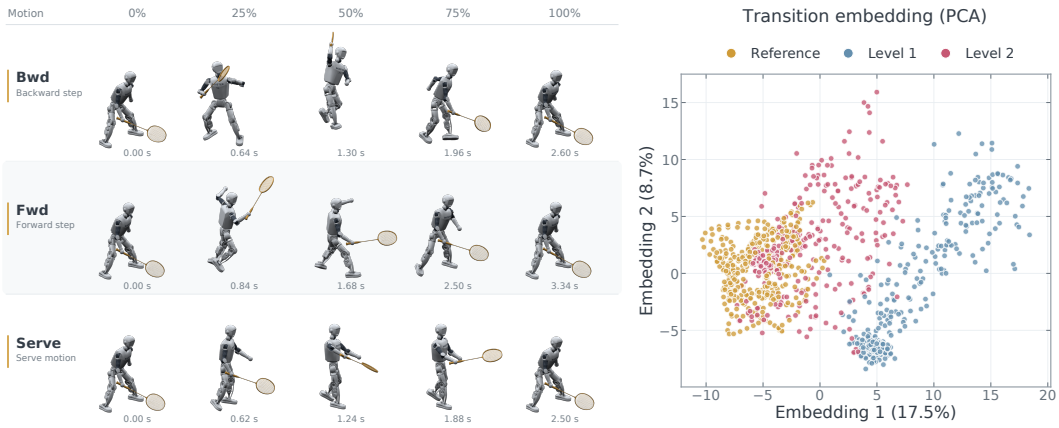

Figure 3: Reference-guided style learning on T1. Left: representative retargeted reference motions, including a backward step, forward step, and serve. Right: PCA embedding of length-four transitions from the reference motions, the Level 1 policy, and the Level 2 policy. The first two principal components explain 17.5% and 8.7% of the variance, respectively.

4.2 Reference-Guided Motion Style and Task Performance

Setup and metrics. We evaluate Level 2 on the T1 humanoid using the clear and lift tasks introduced in Level 1. The same badminton task specifications and event-driven task objectives are retained, while the body-regularization terms are relaxed to allow the reference motion prior to shape whole-body execution. An AMP-based motion prior is then introduced to guide the learned motion style. Reference motions are obtained from the Polygoncraft Unity Asset Store and retargeted to T1 using OmniRetarget (Yang et al., 2025) and GMR (Araujo et al., 2025). Representative retargeted motions are shown in Figure 3. We evaluate the resulting policies from two complementary perspectives: physical task execution using the same stroke-level metrics as in Level 1, and motion style by comparing learned transition distributions with the retargeted reference motions. Detailed differences in the Level 2 reward configuration are provided in Appendix B.3.

Physical task performance. The Level 2 results in Table 1 show that introducing the motion prior preserves reliable racket–shuttle interaction. Both skills achieve hit rates above 99.8%, while valid-return rates reach 99.93% for clear and 96.86% for lift. Target-placement performance is also largely preserved: for clear, the target-landing rate increases from 79.88% in Level 1 to 82.48% in Level 2, with a comparable landing error; for lift, the target-landing rate decreases from 64.65% to 56.24%, while the landing error remains similar.

Reference-guided motion analysis. To examine how the motion prior changes the learned whole-body behavior, we collect length-four transitions from the Level 1 policy, the Level 2 policy, and the retargeted reference motions. All transitions use the same motion features, temporal spacing, and normalization, and are jointly projected using PCA. As shown in Figure 3, Level 1 transitions

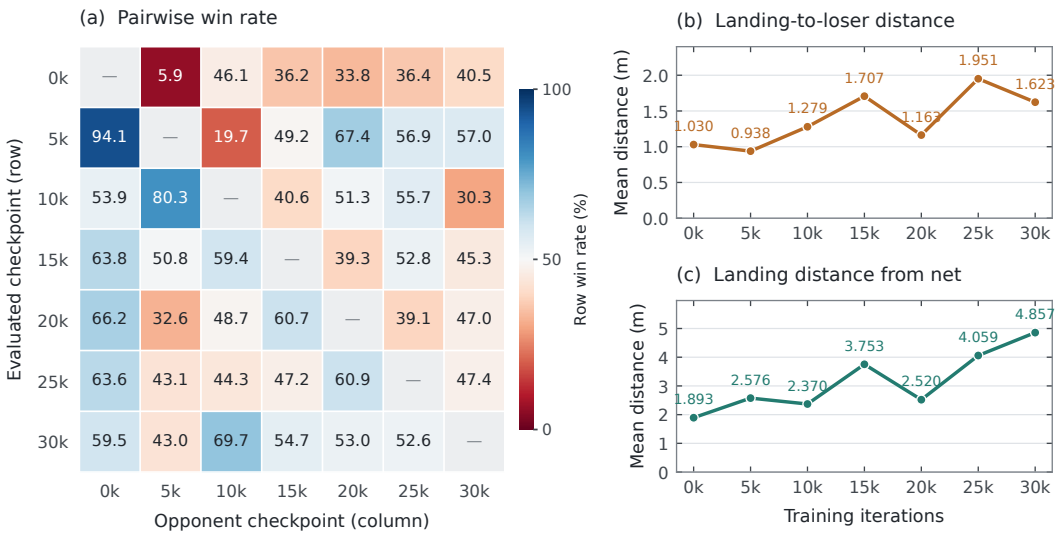

Figure 4: Competitive learning during 30k iterations of T1 self-play after pretraining. (a): pairwise win rates between checkpoints saved every 5,000 iterations, where each cell reports the win rate of the row policy against the column policy over 1,000 rallies. (b): mean distance between the shuttle landing point and the losing agent. (c): mean landing distance from the net.

occupy a largely separate region at higher values of the first principal component, whereas Level 2 transitions shift toward and partially overlap the reference distribution. This distributional shift suggests that the motion prior changes how the badminton task is executed toward the reference motion distribution. Supplementary videos provide direct qualitative demonstrations of the learned reference-guided policy on both T1 and T2.

4.3 Competitive Rally Learning

Setup and metrics. We conduct 30k training iterations starting from a pretrained checkpoint that has already acquired reliable shuttle interception and return capabilities. During Level 3 training, we retain only the racket–shuttle approach reward and the win/loss rewards. By removing explicit shaping of outgoing flight and landing, rally outcomes become the primary learning signal for improving return behavior. We save checkpoints every 5,000 iterations, with 0k denoting the pretrained policy before Level 3 training. To evaluate competitive performance, we match every pair of checkpoints over 1,000 rallies while balancing court side and serving assignment, and report the corresponding pairwise win rates. We additionally record landing-related spatial statistics to characterize how return behavior evolves throughout training.

Competitive performance across checkpoints. The pairwise evaluation in Figure 4 shows that self-play improves competitive performance beyond the pretrained initialization. Every checkpoint from 5k to 30k achieves a win rate above 50% against the 0k policy, ranging from 53.9% to 94.1%. This indicates that continued optimization under rally-level win/loss feedback produces policies that are competitively stronger than the initial stroke-capable policy.

Evolution of return placement. The landing statistics in Figure 4 reveal how return behavior changes under rally-outcome optimization. The mean landing distance from the net increases overall from 1.89 m at 0k to 4.86 m at 30k, indicating a shift toward deeper returns into the opponent's court. At the same time, the mean distance between the landing point and the losing opponent increases from 1.03 m to 1.62 m. Importantly, neither backcourt placement nor opponent separation is explicitly rewarded during this stage. The observed changes therefore arise while optimization is driven primarily by successful interception and rally outcome. Figure 5 illustrates an initial incoming flight followed by five alternating returns. Throughout the rally, both agents continuously reposition while responding to the evolving shuttle trajectory and the opponent's location. In the final

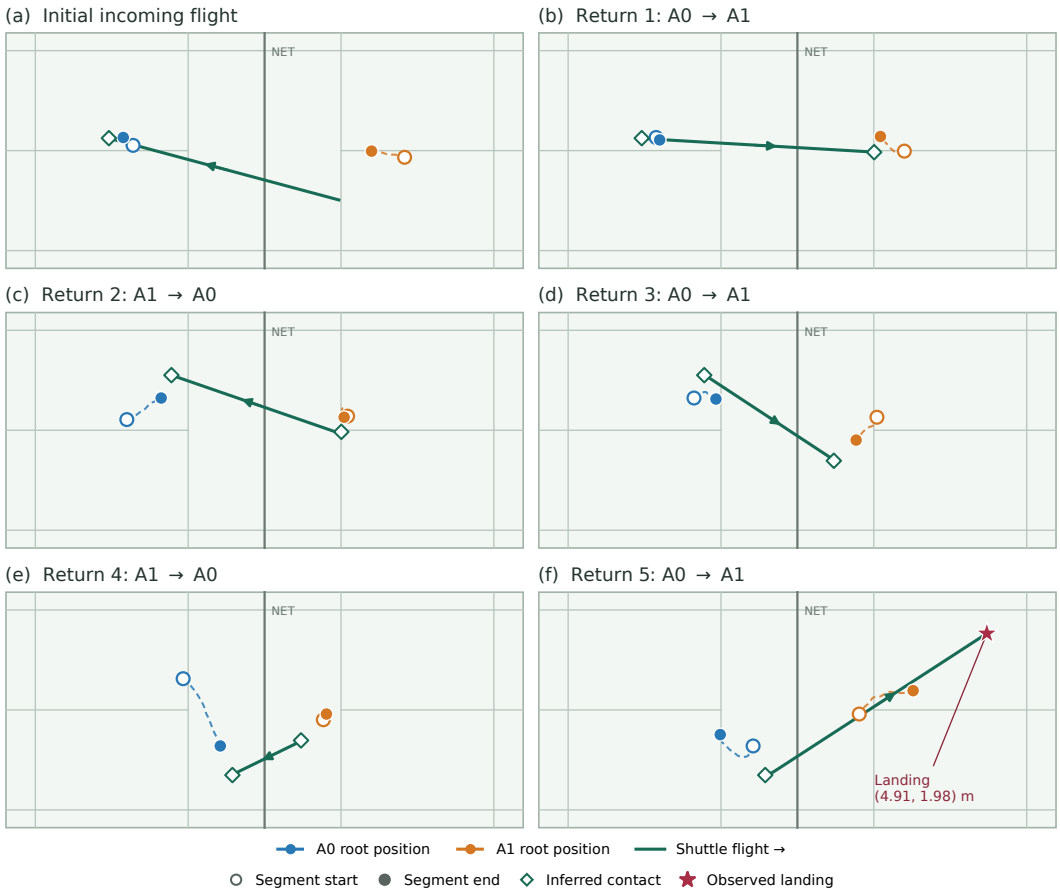

Figure 5: Top-view T1 Level-3 rally.

exchange, A0 sends the shuttle diagonally into A1's backcourt, landing at $(x, y) = (4.91, 1.98)$ m while A1 remains positioned closer to the net.

4.4 Cross-Embodiment Interface Reuse

We further evaluate whether the same BadmintonLab interfaces can be instantiated on a different humanoid embodiment. We consider Level-2 clear and lift tasks on Booster T2. The simulation substrate, task formulation, event interfaces, and learning procedure follow the same design used for T1. The T2 policies are trained independently for their own morphology, actuator configuration, and retargeted reference motions. The T2 rows in Table 1 are evaluated using the same physical task metrics as T1. Both tasks achieve reliable contact and valid returns, with high target-placement success, while the learned motions remain consistent with their corresponding retargeted references.

5 Conclusion

BadmintonLab is a unified physics-based platform for embodied badminton learning, spanning task-oriented skill learning, reference-guided style learning, and competitive rally learning. Its shared simulation substrate and event-driven task-objective interface decouple desired physical outcomes from prescribed motion trajectories or contact timing, enabling a progression from functional stroke acquisition to style-guided execution and rally-level interaction. Experiments demonstrate active striking, separately trained badminton skills with parameterized task specifications, reference-guided changes in whole-body motion, competitive rally behavior, and reuse of the same learning interfaces across humanoid embodiments.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on conceptual framing, research methodology, and experimental design; to assist with software development and interpretation of experimental results; and to support manuscript drafting and editing. We additionally used generative AI tools for literature search and summarization, reference verification and formatting, and figure-design assistance. All AI-assisted suggestions and outputs were independently reviewed by the authors. Any AI-assisted code incorporated into the project was inspected and tested by the authors, references were verified against their original sources, and all experimental results and scientific claims were checked against our own implementations and data. We take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We provide detailed information required to reproduce the reported experiments throughout the main paper and appendices. Appendix A specifies the simulation setup, physical parameters, shuttle dynamics, and incoming-trajectory generator; Appendix B gives the event-driven task objectives and reward configurations for all three learning levels; and Appendix C documents the observation and action interfaces, network architectures, training hyperparameters, and evaluation protocols. Appendix D describes the reference-motion collection and retargeting setup, while Appendix E details the Level-3 rally-training protocol and competitive reward formulation. Appendix F provides additional actuator-level diagnostics for the T1 clear experiment. The supplementary material further includes qualitative videos illustrating the learned reference-guided motions and rally behaviors.

REFERENCES

Joao Pedro Araujo, Yanjie Ze, Pei Xu, Jiajun Wu, and C. Karen Liu. Retargeting matters: General motion retargeting for humanoid motion tracking. *arXiv preprint arXiv:2510.02252*, 2025.

Trapit Bansal, Jakub Pachocki, Szymon Sidor, Ilya Sutskever, and Igor Mordatch. Emergent complexity via multi-agent competition. In *International Conference on Learning Representations*, 2018.

Dieter Büchler, Simon Guist, Roberto Calandra, Vincent Berenz, Bernhard Schölkopf, and Jan Peters. Learning to play table tennis from scratch using muscular robots. *IEEE Transactions on Robotics*, 38(6):3850–3860, 2022. doi: 10.1109/TRO.2022.3176207.

Kai-Shiang Chang, Wei-Yao Wang, and Wen-Chih Peng. Where will players move next? dynamic graphs and hierarchical fusion for movement forecasting in badminton. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6998–7005, 2023. doi: 10.1609/aaai.v37i6.25855.

Lung-Ming Chen, Yi-Hsiang Pan, and Yung-Jen Chen. A study of shuttlecock's trajectory in badminton. *Journal of Sports Science and Medicine*, 8(4):657–662, 2009. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3761540/`.

Yeke Chen, Shihao Dong, Xiaoyu Ji, Jingkai Sun, Zeren Luo, Liu Zhao, Jiahui Zhang, Wanyue Li, Ji Ma, Bowen Xu, Yimin Han, Xuanyi Li, Yudong Zhao, Liyun Li, and Peng Lu. Learning human-like badminton skills for humanoid robots, 2026.

Xuxin Cheng, Yandong Ji, Junming Chen, Ruihan Yang, Ge Yang, and Xiaolong Wang. Expressive whole-body control for humanoid robots. In *Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.107.

David B. D'Ambrosio, 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 *IEEE International Conference on Robotics and Automation*, 2025.

Baptiste Darbois Texier, Caroline Cohen, David Quéré, and Christophe Clanet. Shuttlecock dynamics. *Procedia Engineering*, 2012. doi: 10.1016/j.proeng.2012.04.031.

Tuomas Haarnoja et al. Learning agile soccer skills for a bipedal robot with deep reinforcement learning. *Science Robotics*, 9(89), 2024. doi: 10.1126/scirobotics.adi8022.

Tzu-Han Hsu, Ching-Hsuan Chen, Nyan Ping Jut, Tsì-Uí Ík, Wen-Chih Peng, Yu-Shuen Wang, Yu-Chee Tseng, Jiun-Long Huang, Yu-Tai Ching, Chih-Chuan Wang, et al. Coachai: A project for microscopic badminton match data collection and tactical analysis. In *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 1–4. IEEE, 2019. doi: 10.23919/APNOMS.2019.8893039.

Li-Chun Huang, Nai-Zen Hsueh, Yen-Che Chien, Wei-Yao Wang, Kuang-Da Wang, and Wen-Chih Peng. A reinforcement learning badminton environment for simulating player tactics. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 16232–16233, 2023. doi: 10.1609/aaai.v37i13.26976.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Karol Kurach, Anton Raichuk, Piotr Stanczyk, Michal Zajac, Olivier Bachem, Lasse Espeholt, Carlos Riquelme, Damien Vincent, Marcin Michalski, Olivier Bousquet, and Sylvain Gelly. Google research football: A novel reinforcement learning environment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 4501–4510, 2020. doi: 10.1609/aaai.v34i04.5878.

Marc Lanctot, Edward Lockhart, Jean-Baptiste Lespiau, et al. OpenSpiel: A framework for reinforcement learning in games, 2019.

Julien Le Personnic, Firoz Alam, Laurent Le Gendre, Harun Chowdhury, and Aleksandar Subic. Flight trajectory simulation of badminton shuttlecocks. *Procedia Engineering*, 2011. doi: 10.1016/j.proeng.2011.05.096.

Ang Li, Xinyang Gong, Bozhou Chen, Yunlong Lu, Jiaming Ji, Yongyi Wang, Yaodong Yang, and Wenxin Li. Shuttleenv: An interactive data-driven RL environment for badminton strategy modeling, 2026.

Qiayuan Liao, Takara E. Truong, Xiaoyu Huang, Yuman Gao, Guy Tevet, Koushil Sreenath, and C. Karen Liu. Beyondmimic: From motion tracking to versatile humanoid control via guided diffusion. *Science Robotics*, 11(117), 2026. doi: 10.1126/scirobotics.adx8924.

Chenhao Liu, Leyun Jiang, Yibo Wang, Kairan Yao, Jinchen Fu, and Xiaoyu Ren. Humanoid whole-body badminton via multi-stage reinforcement learning. *arXiv preprint arXiv:2511.11218*, 2025.

Mingjiang Liu, Weizhi Tao, and Hailong Huang. Offline reinforcement learning for badminton tactical decision-making. *Engineering Applications of Artificial Intelligence*, 164:113395, 2026. doi: 10.1016/j.engappai.2025.113395.

Zhengyi Luo, Jinkun Cao, Alexander Winkler, Kris Kitani, and Weipeng Xu. Perpetual humanoid control for real-time simulated avatars. In *IEEE/CVF International Conference on Computer Vision*, pp. 10895–10904, 2023. doi: 10.1109/ICCV51070.2023.01000.

Yuntao Ma, Andrei Cramariuc, Farbod Farshidian, and Marco Hutter. Learning coordinated badminton skills for legged manipulators. *Science Robotics*, 10:eadu3922, 2025. doi: 10.1126/scirobotics.adu3922.

Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Octi Zhang, Peter Du, Antonio Serrano-Muñoz, Xinjie Yao, René Zurbrügg, Nikita Rudin, et al. Isaac lab: A GPU-accelerated simulation framework for multi-modal robot learning, 2025.

Yuxuan Mu, Ziyu Zhang, Yi Shi, Dun Yang, Minami Matsumoto, Kotaro Imamura, Guy Tevet, Chuan Guo, Michael Taylor, Chang Shu, Pengcheng Xi, and Xue Bin Peng. SMP: Reusable score-matching motion priors for physics-based character control. *ACM Transactions on Graphics*, 45 (4):1–21, 2026. doi: 10.1145/3811282.

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), 2013.

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics*, 37(4), 2018. doi: 10.1145/3197517.3201311.

Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Lee, Jie Tan, and Sergey Levine. Learning agile robotic locomotion skills by imitating animals. In *Robotics: Science and Systems*, 2020. doi: 10.15607/RSS.2020.XVI.064.

Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. AMP: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics*, 40 (4), 2021. doi: 10.1145/3450626.3459670.

Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. ASE: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Transactions on Graphics*, 41(4), 2022. doi: 10.1145/3528223.3530110.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016.

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

Clemens Schwarke, Mayank Mittal, Nikita Rudin, David Hoeller, and Marco Hutter. RSL-RL: A learning library for robotics research. *arXiv preprint arXiv:2509.10771*, 2025.

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, 2025.

Oriol Vinyals, Timo Ewalds, Sergey Bartunov, et al. StarCraft II: A new challenge for reinforcement learning, 2017.

Kuang-Da Wang, Yu-Tse Chen, Yu-Heng Lin, Wei-Yao Wang, and Wen-Chih Peng. The CoachAI badminton environment: Bridging the gap between a reinforcement learning environment and real-world badminton games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 23844–23846, 2024. doi: 10.1609/aaai.v38i21.30584.

Wei-Yao Wang, Hong-Han Shuai, Kai-Shiang Chang, and Wen-Chih Peng. ShuttleNet: Position-aware fusion of rally progress and player styles for stroke forecasting in badminton. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 4219–4227, 2022. doi: 10.1609/aaai.v36i4.20341.

Wei-Yao Wang, Yung-Chang Huang, Tsi-Ui Ik, and Wen-Chih Peng. ShuttleSet: A human-annotated stroke-level singles dataset for badminton tactical analysis. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5126–5136, 2023. doi: 10.1145/3580305.3599906.

Hua Wei, Jingxiao Chen, Xiyang Ji, Hongyang Qin, Minwen Deng, Siqin Li, Liang Wang, Weinan Zhang, Yong Yu, Linc Liu, Lanxiao Huang, Deheng Ye, Qiang Fu, and Wei Yang. Honor of kings arena: An environment for generalization in competitive reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 11881–11892, 2022. doi: 10.52202/068431-0863.

Lujie Yang, Xiaoyu Huang, Zhen Wu, Angjoo Kanazawa, Pieter Abbeel, Carmelo Sferrazza, C. Karen Liu, Rocky Duan, and Guanya Shi. OmniRetarget: Interaction-preserving data generation for humanoid whole-body loco-manipulation and scene interaction. *arXiv preprint arXiv:2509.26633*, 2025.

Yanyan Yuan, Yucheng Tao, Shaowen Cheng, Yanhong Liang, Yongbin Jin, and Hongtao Wang. Imitation-relaxation reinforcement learning for sparse badminton strikes via dynamic trajectory generation. *Frontiers in Neurorobotics*, 19:1649870, 2025.

Zhikai Zhang, Haofei Lu, Yunrui Lian, Ziqing Chen, Yun Liu, Chenghuai Lin, Han Xue, Zicheng Zeng, Zekun Qi, Shaolin Zheng, Qing Luan, Jingbo Wang, Junliang Xing, He Wang, and Li Yi. Learning athletic humanoid tennis skills from imperfect human motion data, 2026.

A Simulation and Trajectory Generation Details

This appendix documents the physical scene and the incoming-shuttle trajectory generator shared by the learning settings. Unless stated otherwise, positions and velocities use SI units and are expressed in the court-local frame.

A.1 Court, Shuttlecock, and Physical Parameters

Court frame and geometry. The court-local frame is right-handed. The origin is at the center of the court on the ground plane, x runs between the two baselines, y runs across the court, and z points upward. The net plane is $x = 0$. The court has length $13.40\,\mathrm{m}$, doubles width $6.10\,\mathrm{m}$, and singles width $5.18\,\mathrm{m}$. The rule checks use the singles sideline by default, while the net-crossing check uses the doubles width. The net top height is $1.524\,\mathrm{m}$ at the center and $1.55\,\mathrm{m}$ at the posts; between these points it is linearly interpolated as a function of $|y|$. The static net mesh is a visual asset without collision geometry. A net fault is instead detected geometrically when the shuttle crosses $x = 0$ within the net width and its interpolated crossing height is below the local net height. The court dimensions and net specifications follow the Badminton World Federation's official Laws of Badminton, available through the BWF Statutes page: `https://corporate.bwfbadminton.com/statutes/`.

Shuttle proxy and translational dynamics. The simulated shuttle uses a spherical collision proxy with radius $r_\mathrm{s} = 0.025\,\mathrm{m}$ and mass $m = 0.00519\,\mathrm{kg}$. Its rendered cork and skirt are a separate visual child and do not change the collision proxy. The dynamic model applies gravity and isotropic quadratic drag,

$$\dot{\mathbf{p}} = \mathbf{v}, \qquad m\dot{\mathbf{v}} = m\mathbf{g} - k_d\|\mathbf{v}\|_2\mathbf{v}, \qquad \mathbf{g} = (0, 0, -9.81)\,\mathrm{m\,s^{-2}}, \quad k_d = 0.00110. \tag{3}$$

The shuttle mass and drag coefficient are selected following the physical parameterization reported by Chen et al. (2009). The orientation controller applies a bounded torque that tends to align the shuttle's cork axis with its translational velocity.

Time stepping and contact handling. The physics timestep is $\Delta t_\mathrm{phys} = 0.005\,\mathrm{s}$ (200 Hz), and four physics steps are grouped into one environment step, giving a nominal control interval of $0.02\,\mathrm{s}$. The racket–shuttle contact sensor stores a four-step history and uses a force threshold of $1.0\,\mathrm{N}$. A stroke is marked as having a racket–shuttle contact when the maximum norm over the recorded contact history exceeds this threshold; only the first contact is latched for the episode.

A.2 Task-Space Trajectory Parameterization

Incoming shots are specified in task space rather than by manually chosen initial velocities. A task designer samples a start position, an arrival point, and a flight time,

$$\mathbf{p}_0 \sim \mathcal{D}_\mathrm{start}, \qquad \mathbf{p}_1 \sim \mathcal{D}_\mathrm{arrival}, \qquad T \sim \mathcal{D}_T, \tag{4}$$

and the trajectory generator computes

$$\mathbf{v}_0 = f_\mathrm{solve}(\mathbf{p}_0, \mathbf{p}_1, T; m, k_d, \mathbf{g}). \tag{5}$$

Here, $\mathcal{D}_\mathrm{start}$ and $\mathcal{D}_\mathrm{arrival}$ are distributions over physically meaningful court locations and heights, and $\mathcal{D}_T$ is a distribution over flight durations. In the Level-1 configuration these distributions are implemented as independent uniform ranges over the start coordinates, target coordinates, target height, and flight time.

This parameterization separates task design from dynamics inversion. The designer chooses where and when the shuttle should arrive, while the solver determines the initial velocity required by the configured physical model. Consequently, changing an incoming-shuttle distribution does not require hand-tuning a collection of velocity vectors. It also makes the same task description readable across robot embodiments and learning settings.

Let $D_h = \|\mathbf{p}_1^{xy} - \mathbf{p}_0^{xy}\|_2$ and $\mathbf{e}_h = (\mathbf{p}_1^{xy} - \mathbf{p}_0^{xy})/D_h$ when $D_h > 0$. The initial horizontal velocity is constrained to point along $\mathbf{e}_h$, so the solver searches for a horizontal speed $v_{h,0}$ and a vertical speed $v_{z,0}$. When $D_h = 0$, the horizontal velocity is set to zero. The resulting vector is

$$\mathbf{v}_0 = \begin{bmatrix} v_{h,0}\mathbf{e}_h \\ v_{z,0} \end{bmatrix}. \tag{6}$$

The solver seeks the two-dimensional boundary condition $\big(h(T), z(T)\big) = (D_h, z_1)$. This restriction preserves the sampled horizontal direction while allowing the drag model to determine the required speed and vertical launch component.

A.3 Batched Shooting Solver

The reset event solves all sampled trajectories in a single Torch batch. For each batch element, a rollout integrates the scalar horizontal speed and vertical velocity for $K = 32$ RK4 substeps with $\delta t = T/K$. The acceleration uses the instantaneous three-dimensional speed, so the horizontal and vertical components remain coupled through the quadratic-drag magnitude. Starting from $(v_{h,0}, v_{z,0}) = (10, 8)$ m/s, the solver performs five Newton iterations. At each iteration it evaluates the residual

$$\mathbf{r} = \begin{bmatrix} h(T) - D_h \\ z(T) - z_1 \end{bmatrix} \tag{7}$$

and estimates the 2×2 Jacobian by forward finite differences with a 0.01 m/s perturbation in each initial-velocity component. The Newton increment is obtained from the two-by-two linear system. Each component of the increment is clipped to $[-5, 5]$ m/s, and determinants with magnitude below 10^{-8} are replaced by a signed 10^{-8} value before division. These guards keep nearly singular batches numerically bounded while leaving the task-space boundary conditions unchanged.

The computed horizontal speed is converted to a Cartesian velocity using $\mathbf{e}_h$, concatenated with $v_{z,0}$, and written into the rigid-body root state. Reset solutions are cached in a pool of up to 4096 samples keyed by the sampled ranges and solver parameters. Subsequent resets can draw from this pool instead of repeating the shooting solve, which reduces reset cost without changing the distribution of task-space samples.

B Task Objectives and Reward Configurations

This appendix gives the implemented event-driven objectives and the reward configurations used by the three learning settings.

B.1 Shared Event-Driven Objective Definitions

For a rollout τ, let t_h be the first racket–shuttle contact time, $\mathbf{v}_h^+$ the first post-contact shuttle velocity, t_n the first post-contact crossing of the net plane, t_l the landing time, and $z_{t_n}^s$ the shuttle height at the net crossing. The receive interval $\mathcal{I}_{\mathcal{H}}(\tau)$ is the portion of the incoming flight in which the shuttle descends through the task-specified interception-height interval $\mathcal{H} = [H_{\min}, H_{\max}]$. With racket sweet-spot and shuttle positions $\mathbf{p}_t^r$ and $\mathbf{p}_t^s$, define

$$d_{\text{app}}^*(\tau) = \min_{t \in \mathcal{I}_{\mathcal{H}}(\tau)} \|\mathbf{p}_t^r - \mathbf{p}_t^s\|_2 \,. \tag{8}$$

The receive opportunity is evaluated once when the interval ends through a contact or a descending interval exit. A contact event is represented by

$$J_{\text{contact}}(\pi) = \mathbb{E}_{\tau \sim \pi}\left[\mathbf{1}[t_h < \infty]\right]. \tag{9}$$

In the rally setting, the contact also has to be made by the expected hitter.

Let $\rho_k(d)$ be a non-increasing shaping function for a non-negative deficit d, with $\rho_k(0) = 1$. The implementation uses event-local exponential or Gaussian decay, together with a full-reward radius where appropriate. For an outgoing speed target v^* and feasible direction set $\mathcal{C}(\mathbf{p}_h^s)$, where $\mathbf{p}_h^s$ is the contact position, define

$$\begin{aligned} d_{\text{speed}}(\tau) &= \left[v^* - \|\mathbf{v}_h^+\|_2\right]_+ , \\ d_{\text{dir}}(\tau) &= \operatorname{dist}\left(\frac{\mathbf{v}_h^+}{\|\mathbf{v}_h^+\|_2}, \mathcal{C}(\mathbf{p}_h^s)\right), \\ J_{\text{out}}(\pi; v^*) &= \mathbb{E}_{\tau \sim \pi}[\mathbf{1}[t_h < \infty] \rho_{\text{speed}}(d_{\text{speed}}) \rho_{\text{dir}}(d_{\text{dir}})] \,. \end{aligned} \tag{10}$$

The feasible directions point toward the opponent's court and satisfy the net and lateral-boundary constraints. The outgoing-shot event is evaluated on the first post-contact velocity sample.

Table 2: Level-1 reward configuration. Positive weights reward task progress; negative weights penalize violations or unnecessary motion.

Term	Weight	Event or parameterization
Anti-fall	$+0.1$	root height above $0.55\,\mathrm{m}$
Root orientation	$+0.1$	roll, pitch, and yaw limit shaping
Racket–ground support	-2.0	sustained force $> 3\,\mathrm{N}$; scale 10; cap 1
Joint limits	-2.0	all robot joints
Action rate	-0.1	squared action change
Foot slip	-0.05	horizontal foot speed near the ground
Racket–shuttle approach	$+1.0$	$H = [H_{\min}, H_{\max}]\,\mathrm{m}$; radius $0.10\,\mathrm{m}$
Contact bonus	$+1.0$	first valid racket–shuttle contact
Outgoing shot	$+1.0$	target speed $10\,\mathrm{m/s}$; clearance $1.55\,\mathrm{m}$
Net clearance	$+1.0$	crossing height normalized by $1.55\,\mathrm{m}$
Target landing	$+1.0$	landing height $0.03\,\mathrm{m}$
Post-contact stability	$+0.5$	follow time $0.35\,\mathrm{s}$; heading limit 90°

At the first post-contact net crossing, the net-clearance deficit and objective are

$$\begin{aligned} d_{\mathrm{net}}(\tau) &= \left[H_{\mathrm{net}} - z^s_{t_n}\right]_+, \\ J_{\mathrm{net}}(\pi) &= \mathbb{E}_{\tau\sim\pi}[\mathbf{1}[t_h < t_n < \infty]\rho_{\mathrm{net}}(d_{\mathrm{net}})]\,. \end{aligned} \tag{11}$$

The local net height is used for rule checking, while the reward configuration uses $H_{\mathrm{net}} = 1.55\,\mathrm{m}$ as the full-reward threshold.

For a target landing region $\mathcal{G}$ and planar landing position $\mathbf{p}^{s,xy}_{t_l}$, the target-landing objective is

$$\begin{aligned} d_{\mathrm{land}}(\tau) &= \mathrm{dist}(\mathbf{p}^{s,xy}_{t_l}, \mathcal{G}), \\ J_{\mathrm{land}}(\pi;\mathcal{G}) &= \mathbb{E}_{\tau\sim\pi}[\mathbf{1}[t_h < t_l < \infty]\rho_{\mathrm{land}}(d_{\mathrm{land}})]\,. \end{aligned} \tag{12}$$

The distance is zero for every point inside $\mathcal{G}$. These definitions are shared task abstractions; each learning level activates the subset needed by its physical setting and adds its own regularization or competitive terms.

B.2 Level-1 Reward Configuration

The Level-1 reward is the weighted sum of the shared event-driven terms and basic feasibility regularizers. One-shot event rewards are emitted when the corresponding physical event is resolved. The weights and parameters are listed in Table 2.

Clear, drop, and lift differ through the task specification rather than through separate reward definitions. We use

$$\boldsymbol{\theta}_{\mathrm{skill}} = (\mathcal{D}_{\mathrm{in}}, \mathcal{H}, \mathcal{G})\,, \tag{13}$$

where $\mathcal{D}_{\mathrm{in}}$ is the incoming-shuttle distribution, $\mathcal{H}$ the interception-height interval, and $\mathcal{G}$ the target landing region. Table 3 records the task specifications for the three separately trained skills in the shared court frame.

B.3 Level-2 Reward Configuration

Level 2 preserves the Level-1 task terms and adds an AMP-based motion prior:

$$R_{\mathrm{L2}} = R_{\mathrm{task}} + R_{\mathrm{AMP}} + R_{\mathrm{regularization}}. \tag{14}$$

The AMP term is produced by the discriminator over short whole-body state transitions. In the configured AMP-PPO run its style-reward weight is 1.0, the reward interval is $0.02\,\mathrm{s}$, and the style reward is clipped to 1.0. The task terms retain the Level-1 approach, contact, outgoing-shot, net-clearance, and target-landing events.

Table 4 summarizes the changes to body regularization. Removing these terms from the explicit task reward leaves the motion prior to influence whole-body execution while retaining physical feasibility constraints.

Table 3: Level-1 task specifications. Each incoming distribution is parameterized by start position $\mathbf{p}_0$, arrival point $\mathbf{p}_1$, and flight time T.

Skill	$\mathbf{p}_0$ (m)	$\mathbf{p}_1$ (m)	T (s)	$\mathcal{H}$ (m)	$\mathcal{G}$ (m)
Clear	$x \in [1.80, 2.20]$ $y \in [-1.30, 1.30]$ $z \in [1.30, 1.60]$	$x \in [-6.7, -6.0]$	$[1.40, 1.80]$	$[2.00, 2.50]$	$x \in [6.0, 6.7]$ $y \in [-2.44, 2.44]$
Drop	$x \in [1.80, 2.20]$ $y \in [-1.30, 1.30]$ $z \in [1.30, 1.60]$	$x \in [-6.7, -6.0]$	$[1.40, 1.80]$	$[2.00, 2.50]$	$x \in [0.35, 1.80]$ $y \in [-2.44, 2.44]$
Lift	$x \in [6.00, 6.70]$ $y \in [-1.30, 1.30]$ $z \in [2.00, 2.50]$	$x \in [-2.00, -1.00]$	$[1.50, 1.90]$	$[1.50, 1.60]$	$x \in [6.0, 6.7]$ $y \in [-2.44, 2.44]$

Table 4: Level-2 changes relative to Level 1.

Term	Level 1	Level 2
Root orientation	active, $+0.1$	removed
Default-posture recovery matching	active in post-contact stability	removed
Post-contact position and heading	active	retained, AMP-style recovery
Anti-fall	active, $+0.1$	retained
Joint limits	active, -2.0	retained
Foot slip	active, -0.05	retained
Action rate	active, -0.1	retained
Racket–ground support	active, -2.0	retained

B.4 Level-3 Reward Configuration

Level 3 uses two symmetric player reward managers. The environment exposes receive, contact, outgoing-shot, net-clearance, post-contact stability, and competitive rally-result terms. The available Level-3 reward interface is summarized in Table 5; the transition to the reported self-play configuration is documented in Appendix E. In the reported 30k self-play stage, task shaping is reduced to the receive/approach term and the rally-result term; the explicit outgoing-shot, net-clearance, and post-contact landing terms are not used for that stage, while feasibility regularizers remain active.

The rally-result term is symmetric: if player i wins, it receives a positive event reward and the opponent receives the corresponding negative reward. The result is gated by the geometric mean of the two players' recent valid-return rates, which prevents the terminal signal from dominating before both return behaviors are established.

The rally state machine enforces alternating valid contacts. A contact by the non-expected player is ignored for hit sequencing rather than treated as a separate reward term. A rally resolves on a net fault, an out-of-bounds shuttle, or shuttle landing. The losing side is the last hitter for a fault; for a landing without a fault it is determined by the court half containing the landing shuttle. A fall or excessive tilt terminates the episode, and a time limit provides the timeout termination.

At reset, the configured serve side is player 1. The reset trajectory is treated as an implicit serve: player 1 starts in recovery, player 0 becomes the next expected hitter, and a receive opportunity is opened for player 0.

C Observation, Action, and Training Configuration

This appendix specifies the policy interfaces, optimization settings, and evaluation protocol used in the experiments. The joint counts are $N_q = 23$ for Booster T1 and $N_q = 31$ for Booster T2.

Table 5: Available Level-3 reward and event interface before the rally-stage task-shaping reduction, applied symmetrically to both players.

Component	Weight	Configuration and event
Receive / approach	$+1.0$	descending shuttle in $[1.0, 1.5]$ m
Contact	$+1.0$	only the expected hitter can register the contact
Outgoing shot	$+1.0$	target speed 10 m/s; clearance 1.55 m
Net clearance	$+1.0$	post-contact crossing height normalized by 1.55 m
Post-contact stability	$+0.5$	0.3 s follow interval; heading 90°
Rally result	$+10.0$	$+1$ for a win and -1 for a loss
Feasibility regularizers	align level 1	anti-fall, racket–ground, joint limits, action rate, foot slip

Table 6: Policy observation components for Levels 1 and 2. N_q is the number of actuated joints.

Component	Dimension	Frame / definition
Robot heading	2	court/world xy unit vector
Robot position	3	court-local root position
Shuttle relative position	3	robot-base frame
Shuttle relative velocity	3	robot-base frame; shuttle velocity minus root
Racket position	3	robot-base frame, sweet-spot body relative to root
Racket velocity	3	robot-base frame, relative linear velocity
Racket angular velocity	3	robot-base frame, relative angular velocity
Shuttle in racket frame	3	sweet-spot frame
Shuttle velocity in racket frame	3	sweet-spot frame, relative linear velocity
Root linear velocity	3	robot-base frame
Root angular velocity	3	robot-base frame
Projected gravity	3	robot-base frame
Joint position	N_q	position relative to the configured default pose
Joint velocity	N_q	relative joint velocity
Previous action	N_q	previous normalized policy action
Total	$35 + 3N_q$	T1: 104; T2: 128

C.1 Coordinate Frames and Observation Design

The single-agent observations (Table 6) combine court-frame quantities with egocentric body-frame quantities. A heading is represented by the planar unit vector of the robot's body $+x$ axis. The court frame uses the convention from Appendix A: x runs between baselines, y runs across the court, and z is vertical. Relative positions and velocities are computed after subtracting the corresponding root or racket velocity and are then rotated into the requested body frame.

The observation group is concatenated in the order defined by the task configuration. Level 2 uses the same task observation as Level 1; the AMP state history is supplied to the motion-prior discriminator and is not appended to the actor observation. In particular, no explicit contact time, time-to-hit command, reference-motion phase, or sampled incoming-shuttle parameters are exposed as an additional command vector.

C.2 Level-Specific Observations

For Level 3, each player observes its own state, the shared shuttle, and the opponent state. For player $i \in \{0, 1\}$, the policy input can be written as

$$o_i = \left[o_i^{\text{self}},\ o_i^{\text{shuttle}},\ o_i^{\text{opponent}}\right]. \tag{15}$$

The components and dimensions are listed in Table 7.

Mirrored shared-policy representation. The two court sides are mapped into the same canonical player frame before the shared policy is evaluated. Let $s_i = 1$ for player 0 and $s_i = -1$ for player

Table 7: Level-3 per-player observation components. The opponent state is deliberately smaller than the self state: it contains court pose, projected gravity, and joint positions, but not opponent actions or joint velocities.

Block	Component	Dimension	Canonical frame
o_i^{self}	heading, position	$2+3$	shared court frame
	root linear/angular velocity	$3+3$	player body frame
	projected gravity	3	player body frame
	joint position, joint velocity, previous action	$3N_q$	joint-local coordinates
	block total	$14+3N_q$	egocentric plus canonical court terms
o_i^{shuttle}	relative position, relative velocity	$3+3$	player body frame
o_i^{opponent}	heading, position	$2+3$	shared canonical court frame
	projected gravity	3	opponent body frame
	joint position	N_q	opponent default-relative coordinates
	block total	$8+N_q$	canonicalized opponent state
Total	o_i	$28+4N_q$	T1: 120; T2: 152

Table 8: PPO training configuration. The runner configuration allows up to 50,000 iterations; the reported single-agent runs use 20,000 iterations and the reported Level-3 self-play stage uses 30,000 iterations after pretraining.

Parameter	Levels 1–2	Level 3
Parallel environments	4096 courts	2048 courts, 4096 player samples
Rollout horizon	24 steps per environment	24 steps per environment
Learning rate	10^{-3}	10^{-3}
γ	0.995	0.995
GAE λ	0.95	0.95
PPO clip	0.2	0.2
Entropy coefficient	0.003	0.001
Value loss coefficient	1.0	1.0
Epochs per update	5	5
Mini-batches	16	16
Learning-rate schedule	adaptive, desired KL 0.01	adaptive, desired KL 0.01
Max gradient norm	1.0	1.0
Actor / critic	$[512, 256, 128]$ / $[512, 256, 128]$	same
Activation	ELU	ELU
Observation normalization	actor and critic	actor and critic
Additional reward normalization	none configured	none configured

1. For a court position or planar heading, the canonical transform is

$$M_i(x, y, z) = (s_i x, s_i y, z), \qquad M_i(h_x, h_y) = (s_i h_x, s_i h_y). \tag{16}$$

Thus the opponent's court pose is expressed from the receiving player's side, while the shuttle and body velocities remain in the corresponding player's local body frame. Joint positions, joint velocities, projected gravity, and previous actions are not sign-flipped: their joint ordering and local coordinates are retained for each embodiment. With this convention, the diagrammatic operation in the main text,

$$\text{concat}(o_A, \text{mirror}(o_B)) \longrightarrow \text{shared policy} \longrightarrow (a_A, a_B), \tag{17}$$

is implemented by canonicalizing the court-frame terms in each player's observation and then stacking the two player batches. The shared wrapper uses one policy and one joint-action head; it does not create separate policy parameters for the two sides.

C.3 PPO, AMP, AND NETWORK HYPERPARAMETERS

All policies use PPO (Schulman et al., 2017) in RSL-RL (Schwarke et al., 2025) with an actor and critic MLP of hidden widths $[512, 256, 128]$ and ELU activations. Actor and critic observations are normalized online, the initial policy standard deviation is 1.0 with log-parameterization, and the action output is clipped to the configured policy range. Advantages are computed using generalized advantage estimation (GAE) (Schulman et al., 2016). The common PPO settings are listed in Table 8.

Level 2 adds an AMP discriminator to PPO. Its transition contains four consecutive AMP states, where each state contains root linear/angular velocity and selected joint position/velocity features.

The discriminator uses hidden widths $[256, 128]$, Adam (Kingma & Ba, 2014) learning rate 3×10^{-5}, batch size 512, one discriminator update per scheduled update, and an update interval of 16 policy iterations. The AMP replay buffer stores eight rollouts, policy transitions contribute 256 samples per step, expert transitions receive Gaussian noise with standard deviation 0.02, the gradient-penalty coefficient is 10, and logit regularization is 0.001. The style reward is added with weight 1.0, accumulated at 0.02 s, and clipped to 1.0.

C.4 Evaluation Protocol

Single-stroke policies are evaluated with deterministic actor inference and the observation normalizer stored with the checkpoint. Reset randomization, physics, termination, and event handling remain enabled. For each reported Level-1 or Level-2 skill, we use 4096 parallel environments and complete 100 episodes per environment, giving $N = 409{,}600$ stroke attempts.

Let I_h, I_v, and I_t denote the indicator variables for a racket–shuttle hit, a valid return, and a target landing. The reported rates are

$$R_{\text{hit}} = \frac{1}{N}\sum_{n=1}^{N} I_h^{(n)}, \qquad R_{\text{valid}} = \frac{1}{N}\sum_{n=1}^{N} I_v^{(n)}, \qquad R_{\text{target}} = \frac{1}{N}\sum_{n=1}^{N} I_t^{(n)}. \tag{18}$$

Here $I_h = 1$ when the filtered racket contact detector records a contact before landing. $I_v = 1$ when the contacted shuttle crosses $x = 0$ toward the opponent, within the net width and above the interpolated local net height. $I_t = 1$ when the episode has a hit, a valid net crossing, and a subsequent landing inside the configured target landing region.

For every episode with a hit before landing and an observed landing, let $\mathbf{p}_l^{xy}$ be the interpolated planar landing position and let $\mathcal{G}$ be the target region. The landing error is

$$e_{\text{land}} = \frac{1}{|\mathcal{S}|}\sum_{n\in\mathcal{S}} \operatorname{dist}\left(\mathbf{p}_l^{xy,(n)}, \mathcal{G}\right), \quad \mathcal{S} = \{n : I_v^{(n)} = 1 \text{ and landing observed}\}. \tag{19}$$

The distance is zero inside $\mathcal{G}$. Episodes with no observed landing after a hit are excluded from e_{land} rather than assigned a zero error; they remain counted as unsuccessful outcomes in the rate metrics.

For Level 3, deterministic checkpoint evaluation matches pairs of policies over 1,000 rallies while balancing court side and serve assignment. Rally outcomes use the same physical event logic as training: alternating valid contacts, net faults, out-of-bounds events, landing, fall, and timeout. The pairwise win rate is computed over the completed rallies for each ordered checkpoint pair.

For the reported Level-3 self-play stage, the task reward is intentionally reduced to receive/approach feedback and the terminal rally-result reward. This isolates competitive outcome learning after pretraining has supplied basic interception and return behavior; the action, feasibility, and termination interfaces remain unchanged.

D Reference Motion Set and Retargeting

This appendix documents the reference-motion collection used by Reference-Guided Style Learning. The collection contains locomotion, preparation, receive, and serve clips. All source clips are represented at 30 Hz, and the retargeted T1 and T2 clips are represented at 50 Hz for the 50 Hz policy interface. The source motions are obtained from the Unity Asset Store package *Badminton Extra Motions* by Polygoncraft.

D.1 Source Motion Set

Table 9 lists the locomotion, preparation, neutral, receive, and serve reference clips. The retargeting pipeline preserves the clip time scale up to the expected rounding of a resampled sequence: source clips use 30 Hz and retargeted clips use 50 Hz.

Table 9: Reference-motion collection. Source clips are sampled at 30 Hz; retargeted clips use 50 Hz. Durations are computed from the source frame count and sampling rate. The same retargeted frame count and duration apply to T1 and T2.

Clip	Frames	Duration (s)	T1/T2 frames
BwdL_step_001	85	2.83	141
BwdR_step_001	85	2.83	141
BwdRun1_005	31	1.03	51
Bwd_step_002	79	2.63	131
Entry_004	209	6.97	348
FwdL_step_001	96	3.20	160
FwdR_step_001	79	2.63	131
Fwd_step_002	101	3.37	168
Idle_1_002	33	1.10	55
Idle_2_003	105	3.50	175
Left_step_001	80	2.67	133
Right_step_002	86	2.87	143
Run_005	19	0.63	31
Run_005_IP	19	0.63	31
Backhand_receive1	42	1.40	70
Backhand_receive2	51	1.70	85
Backhand_receive3	51	1.70	85
Forehand_receive1	46	1.53	76
Forehand_receive2	56	1.87	93
Forehand_receive3	51	1.70	85
Mid_receive_success	41	1.37	68
Serve_004	76	2.53	126
Slide_receive1	107	3.57	178

D.2 Full Motion Visualization

Figures 6–8 show the complete retargeted motion collection as motion strips. Each row is one clip and each column is the nearest recorded frame at 0%, 25%, 50%, 75%, or 100% of the clip duration. The strips use the same visual convention as the representative motion panel in Figure 3: the five columns encode clip time, not five independently selected poses. Each pose is reconstructed from the retargeted joint angles, root orientation, and root height with a common orthographic view; horizontal root translation is removed for visual comparison.

E Level-3 Rally Training Protocol

Level 3 is trained after a policy has acquired reliable interception and single-stroke return behavior. The training protocol then changes the source of task feedback: stroke learning specifies the physical properties of an individual return, whereas rally learning evaluates the consequence of a return within a competitive multi-agent rally. This appendix gives the reward transition and the event logic used during the reported self-play stage.

E.1 Reward Transition from Stroke Learning to Rally Learning

During stroke learning, the reward is organized around the physical sequence of one return,

$$r_{\text{stroke}} = r_{\text{approach}} + r_{\text{out}} + r_{\text{net}} + r_{\text{land}} + r_{\text{reg}}, \tag{20}$$

where r_{approach} measures racket–shuttle proximity over the receive interval, r_{out} evaluates post-contact speed and outgoing direction, r_{net} evaluates net clearance, and r_{land} evaluates landing in a task-specified target region. The regularization term contains safety and feasibility terms such as anti-fall, root orientation, racket–ground contact, joint limits, action rate, and foot slip.

After pretraining, the rally stage removes the explicit placement targets from the task objective. Its task-level reward is

$$r_{\text{rally}} = r_{\text{approach}} + r_{\text{result}} + r_{\text{reg}}, \tag{21}$$

with the receive/approach term retained and r_{result} providing the competitive signal. The reported rally-stage configuration therefore does not prescribe a preferred landing depth, a target landing region, or a desired distance from the opponent.

Table 10: Reward transition from isolated stroke learning to competitive rally learning. A check mark denotes an objective available to the stage; placement objectives are intentionally removed from the rally-stage task reward.

Objective	Pretraining / stroke learning	Level-3 rally training
Racket–shuttle approach	✓	✓
Outgoing speed / direction	✓	–
Net clearance	✓	–
Target landing	✓	–
Win reward	–	✓
Loss penalty	–	✓
Basic safety / feasibility	✓	✓

Table 11 gives the coefficients used by the reported rally-stage snapshot. The two player reward managers use the same terms and weights, with the side index changed from player to player. The approach term is shown explicitly because it is part of the protocol interface.

Table 11: Per-player reward coefficients in the reported Level-3 rally configuration. Event terms are emitted once at their corresponding physical event; regularizers remain active throughout the episode.

Term	Weight	Role
Anti-fall / root height	$+0.1$	maintain a feasible upright state
Root orientation	$+0.1$	limit excessive roll, pitch, and yaw
Racket–ground contact	-2.0	penalize support on the racket
Joint limits	-2.0	discourage limit violations
Action rate	-0.01	regularize command variation
Foot slip	-0.05	discourage sliding contacts
Receive / approach	$+1.0$	receive scaffold
Rally result	$+10.0$	terminal win/loss event

E.2 Competitive Result Reward

Let $i \in \{0, 1\}$ denote a player and let ℓ_t be the player assigned as the loser when the rally resolves. The result reward is emitted on the resolution step,

$$r_{\text{result},i}(t) = w_{\text{result}}\, g_t \left(\mathbb{1}[\ell_t = 1 - i] - \mathbb{1}[\ell_t = i] \right), \tag{22}$$

where $w_{\text{result}} = 10$ in the reported configuration. The gate g_t is the geometric mean of the two players' recent valid-return rates,

$$g_t = \sqrt{\hat{q}_{0,t}\hat{q}_{1,t}}, \qquad \hat{q}_{i,t} = \frac{1}{W}\sum_{j=1}^{W} \mathbb{1}[\text{player } i \text{ produced a valid return at } t-j], \tag{23}$$

with $W = 8$. The gate delays a strong terminal competitive signal until both players can sustain the physical exchange. It also preserves the zero-sum symmetry: the winner receives the positive event reward and the loser receives its negative counterpart.

The rally is resolved by a net fault, an out-of-bounds shuttle, or a shuttle landing. For a fault, the last hitter is assigned as the loser; for a landing without a fault, the losing side is determined by the court half containing the landing shuttle. A fall, excessive tilt, or time limit terminates the episode through the corresponding feasibility or timeout condition. These termination rules determine when the result reward is emitted; they do not introduce a placement reward.

F Actuation-Limit Analysis of the T1 Clear

The T1 clear policy frequently operates close to the actuator envelope during racket–shuttle interaction. In a 60-s deterministic rollout containing 12 episodes and 12 detected racket–shuttle contacts,

18 of the 23 joints reached their configured torque–speed limit and 18 joints experienced clipping of the requested torque. Across all physical steps, 10.53% contained at least one clipped joint, and 90.53% of saturated samples were in the speed-derating region rather than in the constant-torque region. Figure 9 gives the corresponding breakdown across all 23 joints, distinguishing saturation in the constant-torque and speed-derating regions.

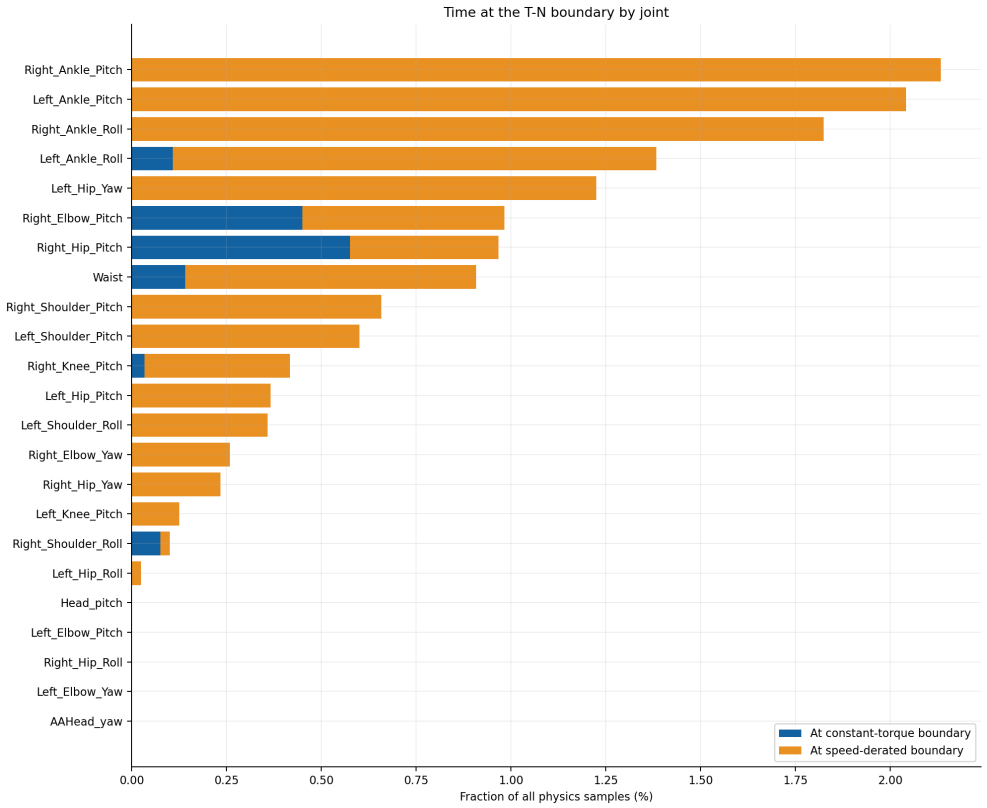

Figure 9: Full-rollout saturation for all 23 T1 joints. Stacked bars separate the constant-torque boundary (blue) and speed-derated boundary (orange). Values are percentages of all physical samples for each joint.

F.1 Torque–speed model

For each joint, the configured actuator model uses a symmetric torque–speed envelope

$$T_{\text{limit}}(\dot{q}) = T_{\text{peak}} \,\text{clip}\left(\frac{v_{\max} - |\dot{q}|}{\max(v_{\max} - v_{\text{knee}}, 10^{-6})}, 0, 1\right), \tag{24}$$

where T_{peak} is the low-speed torque limit, v_{knee} is the onset of speed derating, and $v_{\max}$ is the zero-torque speed.

F.2 Contact-centered utilization

To separate the striking motion from preparation, recovery, and locomotion, we center the analysis at the detected contact sample and use a window from -300 to $+200$ ms. Figure 10 shows the median requested and applied torques over the 12 contacts, with the 10–90% range shaded. The dashed curves are the positive and negative values of $T_{\text{limit}}(\dot{q})$. In several striking-chain joints, the requested torque continues toward the boundary while the applied torque remains clipped at the envelope around contact.

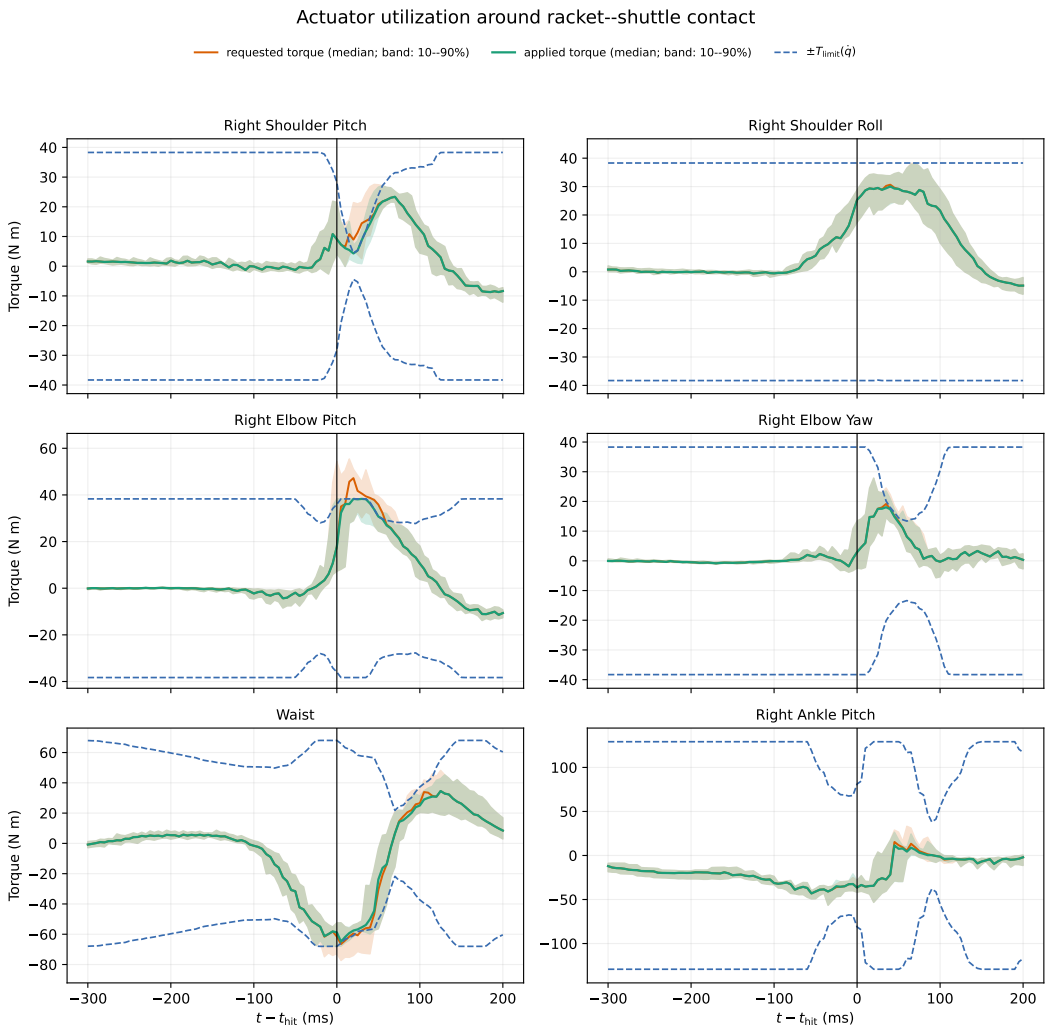

Figure 10: Contact-centered actuator traces for six joints. Curves show the median over 12 racket–shuttle contacts; shaded regions show the 10–90% range. The vertical line marks the detected contact sample.

For a compact multi-contact summary, we define the utilization of joint j at sample t as

$$u_j(t) = \frac{|\tau_j^{\text{applied}}(t)|}{T_{\text{limit},j}(\dot{q}_j(t))}, \tag{25}$$

for samples with nonzero available torque. Table 12 reports the median per-contact peak utilization in the $[-100, +50]$ ms window, the number of contacts whose peak exceeds 95%, and the number with at least one clipped request in that window. The shoulder, elbow, and waist results show that envelope interaction is repeated across the striking chain; it is not attributable to a single lower-body example.

Table 12: Contact-centered actuator utilization over 12 clear executions.

Joint	Median peak u_j	Contacts with $u_j > 0.95$	Contacts clipped
Right shoulder pitch	100.0%	12/12	12/12
Right shoulder roll	90.7%	4/12	2/12
Right elbow pitch	100.0%	12/12	12/12
Right elbow yaw	95.1%	6/12	5/12
Waist	100.0%	11/12	9/12
Right ankle pitch	61.6%	4/12	3/12

F.3 Full-episode traces and supplementary diagnostics

Figure 11 presents the first complete 5-s episode for Right Shoulder Pitch, Right Elbow Pitch, and Waist, using the original actuator archive. Unlike the contact-aligned aggregate, these traces retain the temporal correspondence between torque, velocity, and utilization within one execution. The shoulder trace shows the available torque contracting as speed increases; the elbow trace also shows requested torque exceeding the low-speed torque limit. The waist trace provides the corresponding trunk-motion context. Red markers identify samples where the requested torque differs from the applied torque by more than the tolerance defined above.

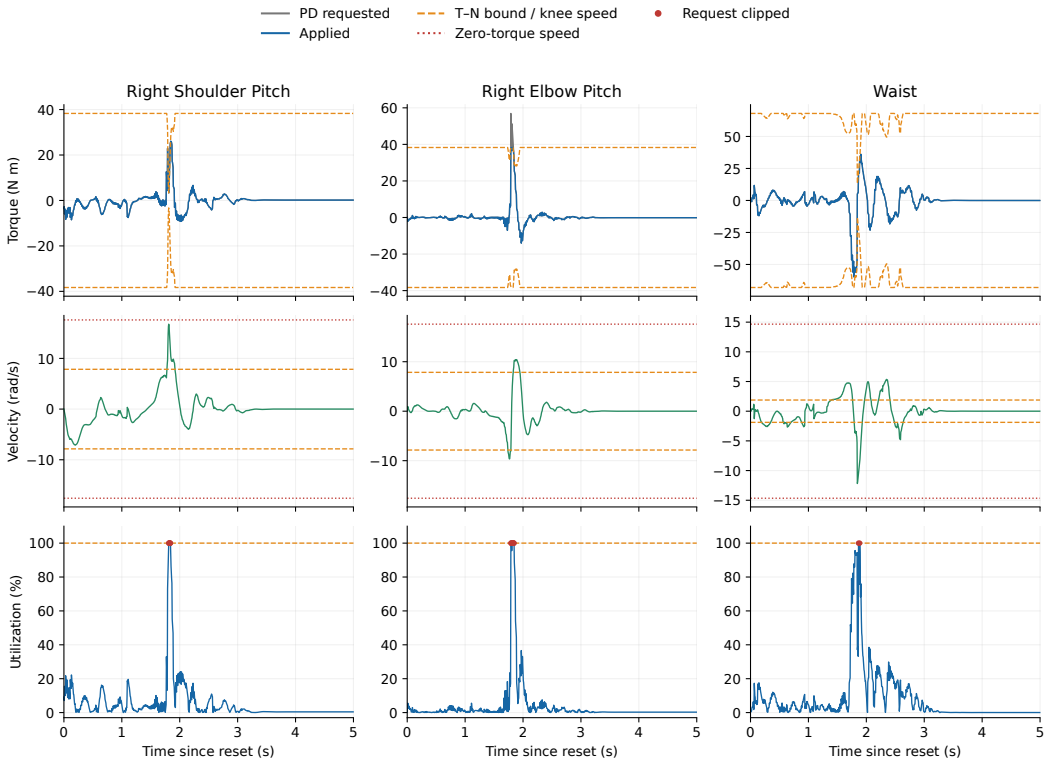

Figure 11: Full-episode actuator traces for three joints in the striking chain, redrawn from the same data as the per-joint diagnostics. Columns correspond to Right Shoulder Pitch, Right Elbow Pitch, and Waist; rows show requested/applied torque with the speed-dependent bounds, joint velocity with the knee and zero-torque speeds, and torque–speed utilization with clipped requests marked in red. All columns show the same first episode; the horizontal axis is time since reset.

The accompanying supplementary document, *T1 Clear: Per-Joint Actuator Diagnostics*, provides the full set of 23 per-joint plots, including joints that never saturate. Each page shows the first episode's torque, velocity, and utilization traces; the RMS, peak torque, and clipping percentage in its header summarize the complete 60-s rollout.

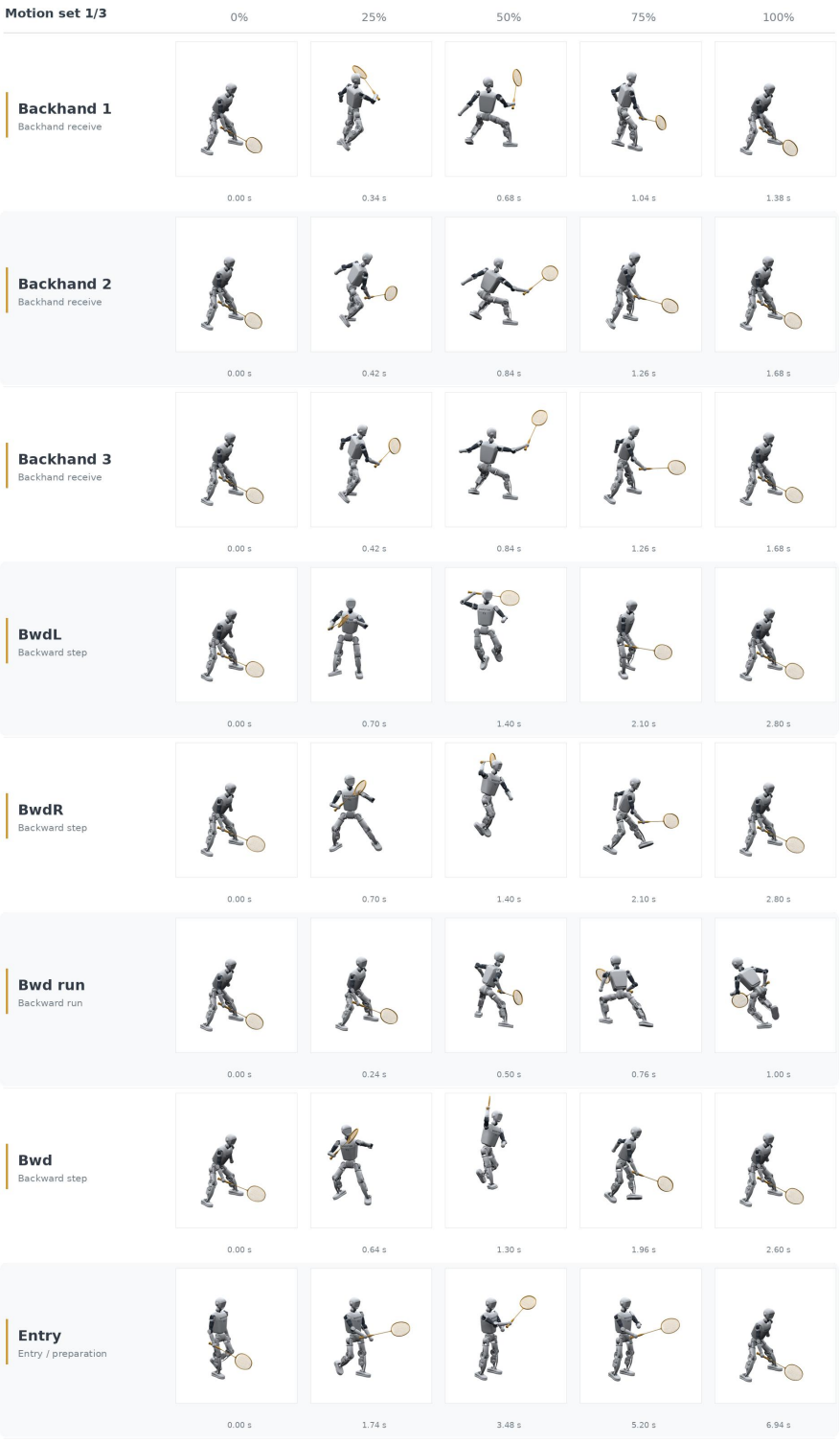

Figure 6: Complete retargeted reference-motion set, part 1. Each row is a source clip and columns sample 0%, 25%, 50%, 75%, and 100% of its duration.

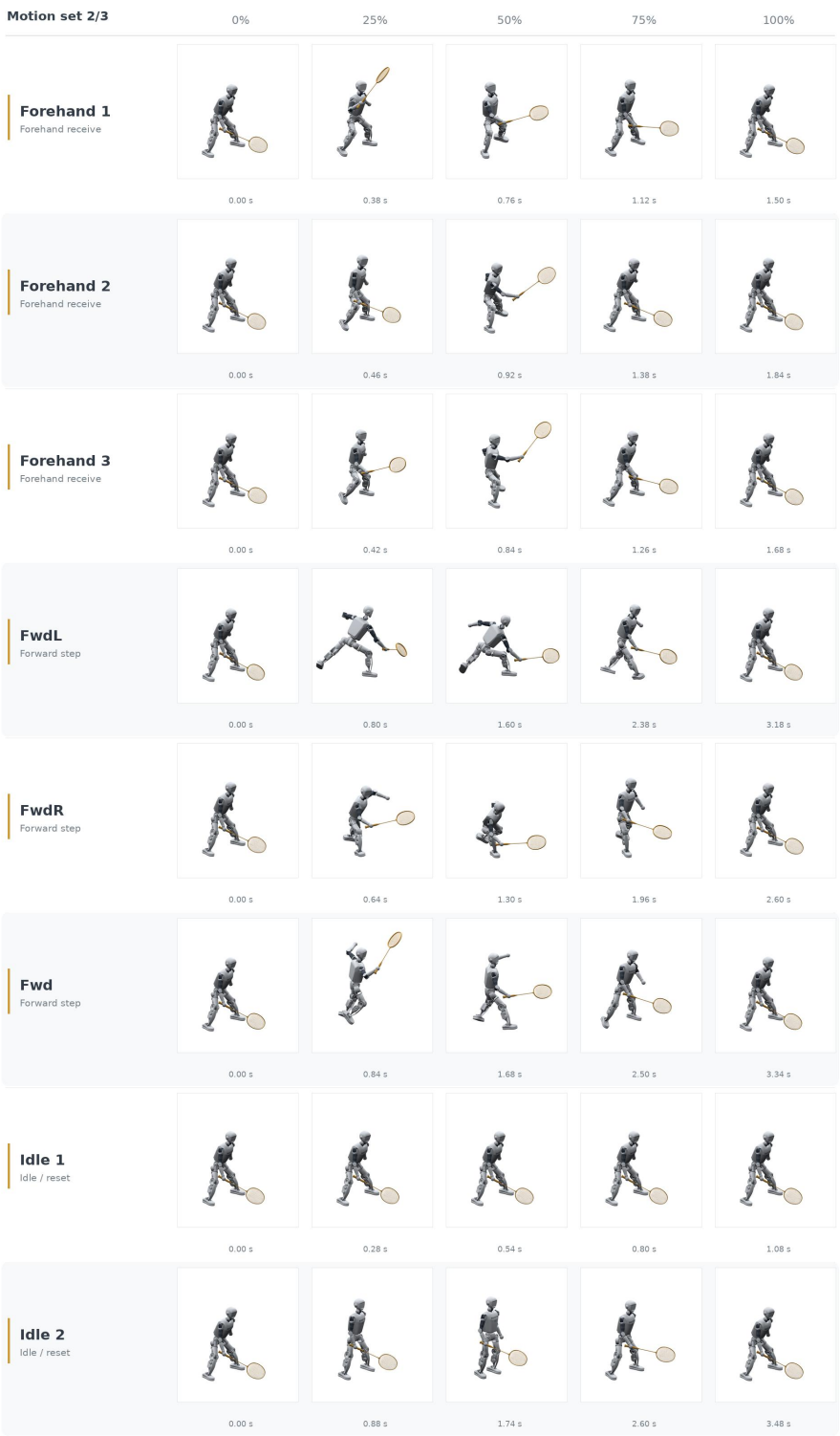

Figure 7: Complete retargeted reference-motion set, part 2.

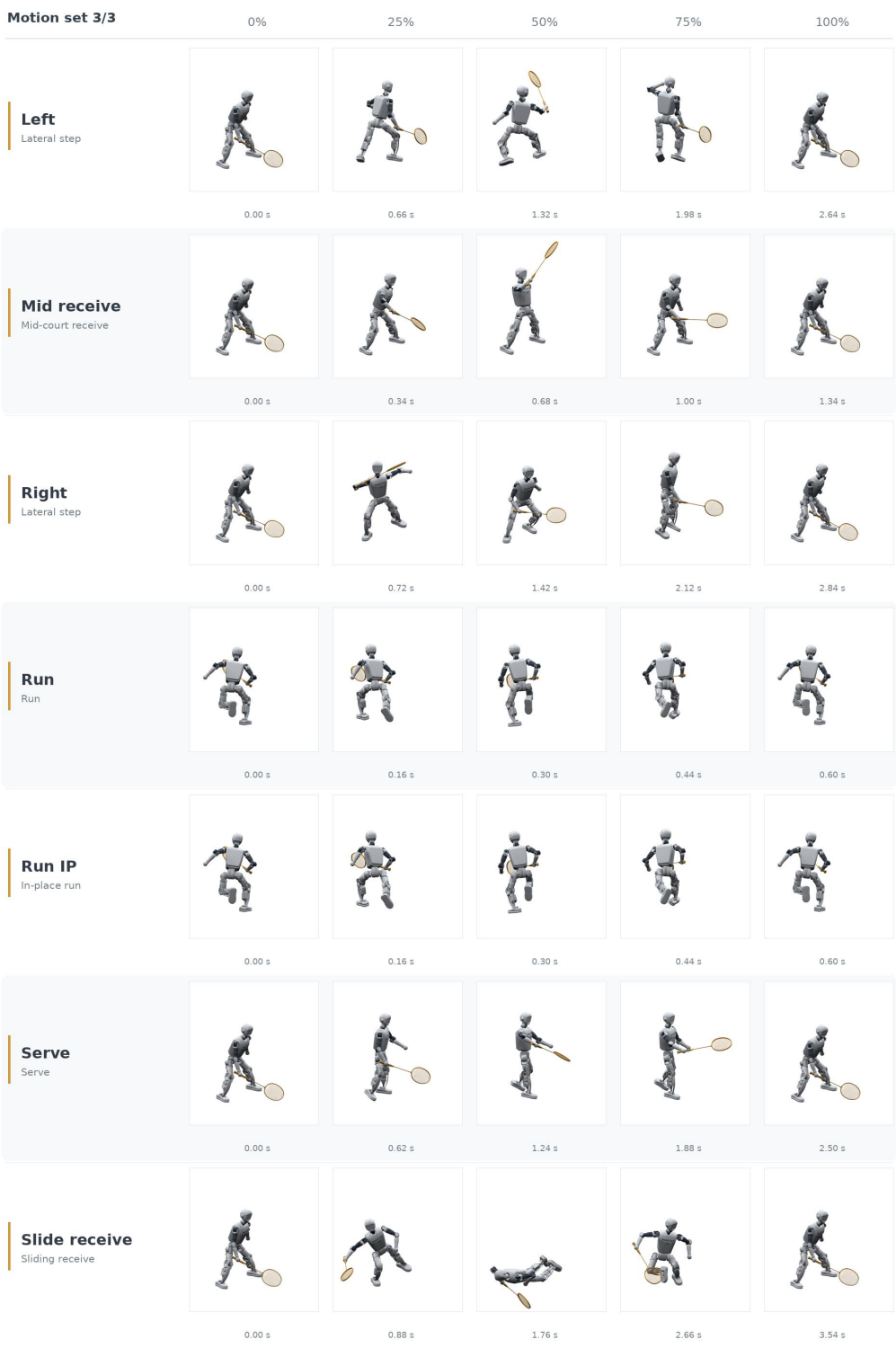

Figure 8: Complete retargeted reference-motion set, part 3.