

Joga: Learning Agile Humanoid Soccer Skills with Active Vision

<https://sites.google.com/view/joga-humanoid>



Fig. 1. Agile soccer skills with *Joga*, a full-stack system for humanoid dribbling, shooting, passing, and receiving. A custom 2-DoF actuated neck enables active vision. Our *real2sim2real* techniques on perception and dynamics help transfer fast, precise whole-body behaviors learned in simulation via reinforcement learning to the real world. In (a), the robot makes a “perfect” single first touch with the outside of the right foot to receive a fast incoming pass of 5.2 m/s. In (b), the robot dribbles from 10 m away and directly strikes the ball into the goal with a commanded ball speed of 10 m/s. Green boxes show detector outputs and orange arrows show ball joystick commands. **Nolan:** Need to show WB-UAN closes the sim-to-real gap. Current results only show it improves dribbling speed and ball control.

Abstract—Exhibiting human-like agility in object interaction is a long-standing goal in humanoid robotics. Soccer offers a demanding case study, requiring dynamic coordination of perception, locomotion, and foot-ball contacts. Reinforcement learning in simulation is a promising approach for learning agile object interaction because it can directly optimize task outcomes, but sim-to-real gaps in perception and dynamics make transfer to reality challenging. We introduce *Joga* (Joint Optimization of Gaze and Action), a full-stack system for humanoid dribbling, shooting, passing, and receiving that perceives the ball through an RGB camera mounted on a custom 2-DoF actuated neck. Our approach combines hardware design with data-driven model alignment to close the sim-to-real gap. The torso-mounted actuated neck enables rapid and stable active vision, while a learned perception alignment model learns the distributional bias and noise of onboard ball observations to achieve the precision needed for sharp turns and dribbling at high speeds. For dribbling, we further learn a whole-body residual action model from 10 minutes of hardware data and use the corrected simulator to finetune the policy, improving its speed. On hardware, *Joga* reaches a peak dribbling speed of 2.31 m/s while maintaining more ball controllability than domain-randomization and linear-perception-model baselines. The learned skills can be chained, transitioning from controlled dribbling to shooting or passing, and from receiving an incoming pass to dribbling. Together, these results show how limited real-world data and targeted modeling can preserve

agile humanoid object interaction learned entirely in simulation.

I. INTRODUCTION

Humans interact with objects with remarkable agility, adapting their movements and contact strategies to the object, task, and environment. Bringing comparable capability to humanoids requires more than replaying prescribed motions; the robot must coordinate whole-body dynamics, balance, and contact using feedback from both its own state and the surrounding scene. Robot soccer provides a demanding testbed for these capabilities [1–4]. We study four complementary skills: dribbling, which repeatedly redirects a moving ball while locomoting; shooting and passing, which impart directed impulses; and receiving, which brings an incoming ball under control. Beyond executing each skill in isolation, agile interaction requires transitioning between them while maintaining ball control. **Nolan:** previous sentence isn’t true.

Although the objectives of these skills can be straightforward to specify (e.g., dribble the ball at 2 m/s forward, pass the ball 30° to the right at 5 m/s, etc.), the whole-body movements and contact strategies needed to achieve them are difficult to design. Reinforcement learning (RL) offers a

natural approach: by optimizing task outcomes directly under the robot’s morphology and dynamics, it allows effective strategies to emerge through interaction rather than requiring predefined motions. Discovering these strategies, however, requires extensive trial and error. On a full-size humanoid, repeated failures are slow and risk hardware damage, motivating training in parallelized simulation [5, 6]. Simulation makes this exploration practical, but discrepancies in perception and dynamics can undermine the resulting behavior on hardware. The robot must keep the ball in view as it moves between close-range foot interaction and longer-range approaches, including while the body accelerates and turns. Even when the ball remains visible, onboard observations exhibit bias and noise that vary with both ball state relative to the robot and robot dynamics. On the actuation side, parallel-linkage mechanisms introduce configuration-dependent joint coupling and reflected inertia, while shared power constraints can make available torque depend on whole-body demand.

Nolan: *Measuring these coupled effects and how our approach addresses them would make the paper much stronger.*

These effects are difficult to model accurately, and errors in either the perceived ball position or the realized foot motion can disrupt precise foot-ball contacts at high speeds.

Domain randomization (DR) offers a way to tolerate these discrepancies without modeling each exactly by exposing policies to varied observations and dynamics during training [7–9]. However, overly broad randomization can favor slow, conservative strategies that work across many simulated systems over those that perform best on the real robot [10, 11]. Online adaptation can help policies adjust to the dynamics encountered at deployment [12–16], but designing training variations that capture coupled, state-dependent effects is challenging. The central challenge is therefore not merely to transfer a policy, but to preserve the speed, precision, and fluidity that simulation makes possible.

To preserve the agile object interaction learned in simulation, we pursue a complementary approach: improve the fidelity of the nominal system and use targeted randomization to accommodate remaining variability in the terrain, ball, and scene. This requires addressing both sides of the sensing and action loop. A policy must receive observations that reflect the real sensing pipeline, and the simulator must reproduce the physical consequences of its actions. Limited real-world data can help identify systematic discrepancies in both, reducing the need for randomization and allowing the policy to specialize to the real system.

We introduce *Joga*¹ (Joint Optimization of Gaze and Action), a full-stack humanoid soccer system. We combine hardware design with data-driven real2sim2real perception and dynamics alignment to reduce systematic modeling errors, retaining targeted randomization for remaining environmental variability. A custom, torso-mounted 2-DoF neck carrying an RGB camera enables rapid gaze changes while reducing mechanical jitter during fast locomotion. **Nolan:**

Could we run an experiment to measure the benefits of our neck design over TWIST2 neck? Each skill policy jointly controls the whole body and neck, coordinating viewpoint and physical interaction.

To address the sensing sim-to-real gap, we learn a *perception alignment model* (PAM) from 10 minutes of paired onboard detections and motion-capture labels to reproduce observation errors during policy training. Unlike prescribed motion-dependent noise [4] or distance-dependent noise models [17], PAM learns how bias and variability depend jointly on ball geometry and robot motion.

Building on Contact-UAN’s contact-consistent replay formulation [18], we use a single residual network conditioned on whole-body joint states and commands. This parameterization allows each correction to depend on other joints, rather than restricting it to a joint-local history. We evaluate the resulting residual-model fitting and policy-finetuning procedure through downstream dribbling performance.

Independently trained skills may encounter unfamiliar robot–ball configurations when chained. To address this mismatch, we perform distributional skill-chaining by finetuning the shooting and passing policies with initial states sampled from dribbling rollouts.

Together, *Joga* combines hardware for active vision, data-driven perception and dynamics alignment, and finetuning for compatible skill transitions into a system for agile humanoid ball interaction. On a Unitree G1, we demonstrate all four agile skills coordinating all 31 actuators using only onboard sensing at deployment. With PAM and WB-UAN, our policy reaches a sustained dribbling speed of 1.7 m/s and an estimated peak speed of 2.3 m/s. It consistently completes sharp 135° turns and allows human operators to joystick-control with high precision. Hardware demonstrations of chaining dribbling to shooting demonstrates remarkable stability and foot contact accuracy. These results show how improving the fidelity of what the policy sees and the dynamics under which it learns can translate task-driven RL into fast, precise, and chained object interactions on hardware shown in Fig. 1.

II. METHOD

Fig. 2 summarizes our system. Below, we describe perception alignment, whole-body actuator modeling, and skill-chaining. Implementation details for the neck design and calibration, YOLO detector finetuning, ball contact and friction identification, and skill-specific rewards, observations, actions, and DR are provided on our [project website](#).

A. Perception Alignment Model

The skill policies observe the ball position in the camera frame, back-projected from the onboard detector’s 2D bounding box using a pinhole camera model. This estimate carries systematic, state-dependent errors absent in simulation despite both intrinsic calibration and hand-eye calibration for the extrinsics: depth is inaccurate because the RGB camera is not a perfect pinhole camera, noise grows with range and extrinsics mismatches, and the detector drops out at a

¹“Joga bonito” translates to “play beautifully” in Portuguese and is a famous philosophy of creativity, joy, and flair in soccer.

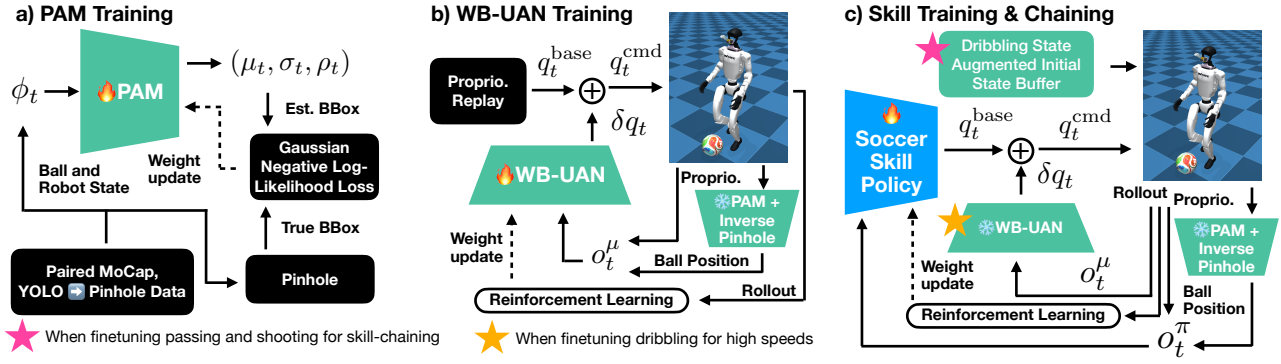


Fig. 2. **Joga**. a) Train PAM on paired YOLO bounding box, proprioception, and MoCap data to align simulated ball observations with real onboard ball observations. b) Train WB-UAN on proprioceptive replay data to align simulated and real actuation. c) Train soccer skill policies in the aligned simulator. Dribbling policy is finetuned with WB-UAN to improve performance at high speeds. Passing and shooting skills are finetuned with initial states collected from the dribbling policy to enable skill-chaining.

range-dependent rate. Rather than cover this gap with hand-tuned DR, we train a *perception alignment model* (PAM) on paired observed hardware data with ground-truth MoCap and freeze it as the observation model during task-policy training, replacing perception DR entirely. Let $p = (X, Y, Z)$ be the true ball position in the camera frame (+X forward, +Y left, +Z up), (f_x, f_y, c_x, c_y) the pinhole intrinsics, and R the ball radius. The PAM is designed to work in bounding-box space rather than raw xyz space as it replicates the actual vision pipeline. The analytic silhouette box, parametrized by center and log-size, is

$$b(p) = \left(\underbrace{c_x - f_x \frac{Y}{X}}_u, \underbrace{c_y - f_y \frac{Z}{X}}_v, \log \frac{2f_x R}{X}, \log \frac{2f_y R}{X} \right), \quad (1)$$

so the center encodes bearing and the log-size encodes range. The PAM predicts a residual $r = \hat{b} - b(p)$; adding it back and back-projecting through a pinhole camera model, π , yields the point $\hat{p} = \pi^{-1}(\hat{b})$ the policy sees, with depth recovered from apparent size,

$$\hat{X} = \frac{2f_x R}{e^{\hat{b}_3}}, \quad \hat{Y} = -\frac{\hat{u} - c_x}{f_x} \hat{X}, \quad \hat{Z} = -\frac{\hat{v} - c_y}{f_y} \hat{X}, \quad (2)$$

so a residual in $\hat{b}_3$ maps directly to a relative depth error.

a) *Data collection*: We collect paired onboard-detector and motion-capture measurements while the robot performs locomotion and dribbling, covering the range of ball and robot poses seen at deployment. MoCap tracks the ball, the camera, and the body, yielding the true point p (Eq. (1)) and the ball in the base frame, paired with the detector box $\hat{b}$ at the same instant. This gives a dataset $\{(\phi_i, r_i, s_i)\}_{i=1}^N$ of contexts ϕ , residuals r , and visibility labels $s \in \{0, 1\}$.

b) *Architecture and observation space*: A network f_θ maps a conditioning vector ϕ to a heteroscedastic Gaussian over the residual plus a visibility probability ρ ,

$$f_\theta(\phi) = (\mu(\phi), \sigma(\phi), \rho(\phi)), r \sim \mathcal{N}(\mu(\phi), \text{diag} \sigma(\phi)^2), \quad (3)$$

with $\mu, \sigma \in \mathbb{R}^4$ and $\rho \in [0, 1]$. We choose ϕ so the model can express state-dependent extrinsics offsets and a depth correction rather than a single global noise level,

$$\phi = [p, \omega_{\text{base}}, \dot{q}_{\text{neck}}, \dot{q}_{\text{waist}}, g_{\text{proj}}, p^B], \quad (4)$$

comprising the true point (carrying range, for depth bias), the ego-motion (base angular velocity and neck and waist joint velocities, for motion-blur growth in the box center), and the pose (projected gravity and the ball in the base frame p^B , for a configuration-dependent bearing offset). We model r as Gaussian because the map from ϕ to r is inherently stochastic: mocap label noise and detector error are not resolvable from ϕ alone, since the PAM never sees the RGB image. The predicted $\sigma(\phi)$ absorbs this, and the fit is qualitatively *unlike isotropic DR*: the depth bias is $\approx +5$ cm at dribbling range and changes sign beyond ~ 2 m, $\|\sigma\|$ grows with range and $\|\omega_{\text{base}}\|$, and $1 - \rho$ rises from a few percent at close range to over 30% at 2.5 m.

At injection time we sample the residual and visibility and reproject,

$$\begin{aligned} \hat{b} &= b(p) + \mu(\phi) + \text{diag} \sigma(\phi) \varepsilon, \\ \varepsilon &\sim \mathcal{N}(0, I), \quad s \sim \text{Bernoulli}(\rho(\phi)), \end{aligned} \quad (5)$$

emitting $\pi^{-1}(\hat{b})$ when the ball is visible ($s = 1$). When $s = 0$, we hold the last ball position observation. Because the bearing bias is consistent with residual mocap calibration error rather than a real deployment error, we zero out such bias, whereas the depth bias is a true monocular effect the policy must compensate. This state dependence is what separates the PAM from our linear perception baseline [17], which conditions only on range $\rho = \|p\|$ with affine per-axis moments and a constant dropout, $\mu(\phi) = \alpha + \beta \rho$, $\sigma(\phi) = \gamma + \delta \rho$, $\rho(\phi) = \bar{\rho}$. It only captures the mean depth bias but not the motion- and pose-dependent structure of Eq. (4), so a policy trained against it can struggle during fast motions.

c) *Training*: We fit θ by minimizing a Gaussian negative log-likelihood on the residual, gated to visible frames, plus a binary cross-entropy term on the visibility,

$$\begin{aligned} \mathcal{L}(\theta) &= \frac{1}{N} \sum_{i=1}^N \left\{ s_i \sum_{k=1}^4 \frac{1}{2} \left[\frac{(r_{i,k} - \mu_k)^2}{\sigma_k^2} + \log \sigma_k^2 \right] \right. \\ &\quad \left. - \lambda [s_i \log \rho_i + (1 - s_i) \log (1 - \rho_i)] \right\}, \end{aligned} \quad (6)$$

with μ_k, σ_k, ρ_i evaluated at ϕ_i . To avoid overfitting on the limited dataset, we regularize with weight decay and dropout. The frozen model is used only during downstream task training, mapping $b(p) \mapsto \hat{b}$ via Eq. (5) at every control step.

B. Whole-Body Unsupervised Actuator Network

Accurate perception alone is insufficient for agile soccer skills if the simulator mispredicts the motion produced by the policy’s commands. Previous work has considered actuation models that make joint-level corrections [19, 20], but actuation discrepancies can depend on the configuration, motion, and commands of multiple joints [10], motivating corrections that use information beyond an individual actuator’s history. We therefore extend the joint-local residual formulation of Contact-UAN [18] to a *whole-body unsupervised actuator net* (WB-UAN). The model retains the simulator’s rigid-body and contact dynamics while learning joint-position command corrections from hardware trajectories, without torque supervision. We collect 10 minutes of data using a pretrained dribbling policy, fit the WB-UAN to improve agreement between simulated and recorded motion, and then finetune the same policy in the corrected simulator.

a) Residual dynamics: Let s_t denote the state and $q_t^{\text{base}} \in \mathbb{R}^n$ the uncorrected joint-position target. The residual model μ predicts a correction δq_t from observations o_t^μ :

$$\begin{aligned} \delta q_t &= \mu(o_t^\mu), \\ s_{t+1} &= f_{\text{sim}}(s_t, q_t^{\text{base}} + \delta q_t), \end{aligned} \quad (7)$$

where f_{sim} includes the nominal actuator controllers, rigid-body dynamics, contact solver, and numerical integration. During residual fitting, q_t^{base} is the recorded hardware command; during downstream finetuning, it is the task policy’s output. The residual thus corrects the actuator input rather than directly modifying simulated states or contact forces. Fig. 2(c) includes a block diagram of the WB-UAN.

b) Whole-body observations and architecture: WB-UAN receives a causal history of joint positions, joint velocities, and uncorrected position targets. Unlike Contact-UAN’s joint-local tracking-error histories, this observation includes absolute joint positions and commands across the entire body. A single network predicts all corrections jointly, allowing each joint’s correction to depend on the configuration, motion, and commands of other joints.

c) Contact-aware trajectory replay: As in Contact-UAN, we train WB-UAN through short simulator rollouts driven by recorded command sequences, where each window is initialized using the contact-consistent reset procedure [18]. Frozen proprioceptive estimators provide base velocity and contact-relevant root pose, while joint states and base angular velocity are taken from the recorded observations. This procedure accounts for the stance-foot penetration and tilt required by the simulator’s compliant contacts, reducing initialization-induced errors that could otherwise be absorbed by the learned residual.

d) Rollout-based model fitting: We optimize μ with proximal policy optimization [21], treating δq_t as the action and agreement with recorded joint trajectories as the reward [18]. Optimization requires neither torque labels nor differentiation through the simulator.

e) Policy finetuning and deployment: We freeze μ and finetune the dribbling policy in the augmented simulator

using the original task reward. WB-UAN conditions on the evolving simulated joint history and current policy commands, without access to recorded reference trajectories. We initialize downstream finetuning with a small learning rate and use KL-adaptive scheduling to limit abrupt policy changes when introducing the learned dynamics. WB-UAN remains exclusively a simulator-side model; only the finetuned task policy is deployed on hardware.

C. Distributional Skill-Chaining

Independently trained skills need not produce compatible transitions. A passing or shooting policy trained from a stationary reset may succeed in isolation yet fail when invoked during dribbling, when the robot is mid-stride and the ball is already moving. We address this mismatch by finetuning the passing and shooting policies from states encountered by a pretrained dribbling policy.

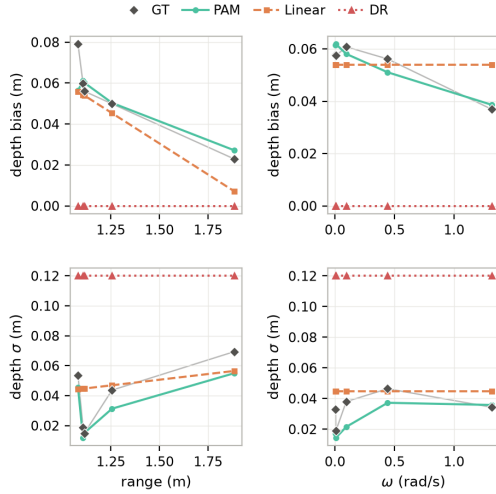
a) Collecting the state bank: We roll out the dribbling policy in simulation across its commanded speed and direction distribution and collect a bank of robot–ball states, excluding the initial reset transient of each episode. Each entry stores joint positions and velocities, pelvis height and orientation relative to its heading, base linear and angular velocities, and the ball’s position and linear and angular velocities. We express the ball state and base velocities in the pelvis frame so that recorded configurations can be restored at different locations and headings. Recording velocities alongside poses preserves the motion associated with each configuration, including the robot’s gait phase and the ball’s ongoing motion.

b) Fine-tuning from dribbling states: At each episode reset, we sample an entry uniformly from the bank and restore the corresponding robot and ball state. We retain the global position and heading sampled by the target task’s reset procedure and transform the recorded quantities into that frame. Let $\mathcal{B}_d = \{b_i\}_{i=1}^{N_d}$ denote the state bank and $\mathcal{T}_\xi$ the reconstruction at a sampled placement ξ . The resulting initialization is

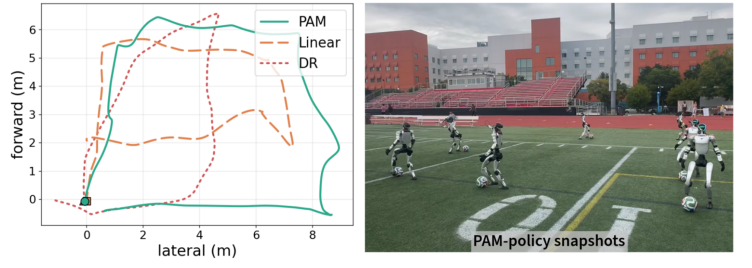
$$i \sim \text{Uniform}\{1, \dots, N_d\}, \quad s_0 = \mathcal{T}_\xi(b_i). \quad (8)$$

Starting from their pretrained parameters, we finetune the passing and shooting policies using their respective task rewards and observations. The dribbling policy remains fixed. This exposes each successor policy to the configurations from which it will be invoked, allowing it to learn how to initiate a directed strike with the robot and ball in motion.

c) Executing transitions: At deployment, control switches from the dribbling policy to the selected passing or shooting policy while the physical robot and ball continue from their current state. The state bank is used only to initialize simulation episodes during training; the deployed policies continue to use onboard observations. Skill compatibility is therefore learned through the successor policy’s initial-state distribution, without requiring the dribbler to stop in a prescribed pose before initiating a pass or shot.



(a) **Perception alignment vs. dynamic motion.** On held-out hardware data, PAM captures the GT bias and noise distribution of onboard ball detection against both range and ego-motion, while Linear and DR baselines fall short, under-or-over-estimating it.



(b) **Real-world square steering.** A human operator, blind to the policy, is asked to joystick the robot to dribble along a square (success 4/4 PAM, 3/4 DR, 1/4 Linear); the PAM-policy is more steerable than the baselines.

Policy	easy (%)	medium (%)	hard (%)
DR	77.5 $\pm$ 0.5	53.0 $\pm$ 0.4	29.9 $\pm$ 0.5
Linear	76.9 $\pm$ 0.3	54.5 $\pm$ 0.5	33.5 $\pm$ 0.3
PAM	86.7 $\pm$ 0.4	69.2 $\pm$ 0.2	57.4 $\pm$ 0.4
Oracle	89.4 $\pm$ 0.3	70.9 $\pm$ 0.3	58.0 $\pm$ 0.3

(c) **Sim-to-sim course tracking.** Under a large target-environment perception gap, the PAM-policy nearly matches an Oracle trained directly in that environment, while DR and Linear-policy baselines degrade (higher is better, 5 seeds).

Fig. 3. **PAM closes the perception sim-to-real gap for precise dribbling.** We fit PAM using MoCap and onboard hardware data, and freeze it as the simulator’s observation corrector, then compare it against Linear and DR baselines in both fitting quality (a) and downstream dribbling controllability (b, c).

D. Actuated Neck Design

We designed a custom 2-DoF yaw-pitch neck to enable active perception of the ball. The neck mounts directly to the torso and uses two DYNAMIXEL XC330-M288-T’s along with a U2D2+ Hub. The mechanism is kept simple, avoiding external linkages, gears, and bearings. Current open-source designs utilize more complicated transmission mechanisms, which introduce backlash, compliance, slippage, and calibration error. This reduces speed and reliability while increasing the sim-to-real gap [22]. In contrast, our design provides the high range of motion and speed for reliable, rapid gaze changes due to ball and robot movements. Most importantly, the reduced mechanical complexity eliminates neck DR after simple SysID for policy training.

III. RESULTS

We build *Joga* on the Unitree G1. We seek to demonstrate the performance benefits of principled modeling, and data-driven perception and dynamics alignment on agile soccer skills. Most of our tasks are the first to be demonstrated at this level of athletic performance on a real humanoid platform. *Joga* will be publicly available upon acceptance.

A. PAM Enables Precise Dribbling

1) **Perception alignment:** We train our PAM and a Linear Model baseline [17] on MoCap data and evaluate them on a held-out test set. Because PAM is conditioned on range, ego-motion, and bearing, and its injection procedure is similar to the actual deployment, it is highly expressive in capturing both intrinsic and extrinsic perception misalignment. Fig. 3a shows that PAM better models the real bias and noise distribution against the agility-relevant factors than other baselines. In particular, PAM fits the GT very closely in the regime that matters most for agile dribbling, since the data is

collected on-policy, where close range and fast ego-motion dominate. In contrast, the Linear baseline overestimates the depth noise at close range and stays flat against ego-motion.

2) **Dribbling controllability:** To demonstrate steerability of our method, we performed 4 hardware runs following a square trajectory of 5 meters and measured success rate. We defined a success with three conditions: (1) ball controlled by multiple kicks, (2) robot did not step on the ball, and (3) robot did not fall. The operator had no knowledge of the running policy and their goal was to track the trajectory to the best of their ability.

While the PAM, DR and Linear policies were all able to achieve the square trajectory, the PAM-policy had 100% success rate compared with 75% and 25% respectively. Additionally, from leg odometry estimates of successful runs in Fig. 3b, it is shown that the operator was more easily able to control the movement of the PAM-policy in the real world.

Furthermore, we stress test the downstream task benefits of PAM in a sim-to-sim setup where all policies are trained on a base environment and deployed on a target environment with significant perception gap. The target camera differs from the base by a miscalibrated pinhole model, a 5% focal-length error and a principal-point shift, compounded by lens distortion and a small camera extrinsic offset (~ 1 cm translation and 4° rotation). As a result, the analytic ball projection that the base policy implicitly relies on is systematically biased at deployment, mimicking the real onboard perception gap. PAM and Linear model are trained using target data to close the gap. Oracle policy is trained directly on the target environment to provide a performance upper-bound. We predefine command sequences of three difficulty levels: easy with slow 45° turns, medium with 90° turns, and hard with fast 180° turns. As can be seen from Fig. 3c, PAM-policy can reach the performance level of Oracle policy while

Approach	train-set RMSE	test-set RMSE
No-Res	0.08788	0.08852
Contact-UAN	0.04949	0.04722
WB-UAN	0.03787	0.04953

(a) **Dynamics alignment.** WB-UAN best fits recorded hardware trajectories (lowest training rollout RMSE), improving the fidelity of the augmented simulator.



Policy	success	dir. error (°)
DR	3/5	-18.1 ± 04.5
Linear	4/5	-16.3 ± 12.0
PAM	4/5	-15.6 ± 13.7
PAM+WB-UAN	5/5	-14.5 ± 03.4

(b) **Sharp 135° turns.** Over five turns at 1.2 m/s, PAM+WB-UAN is the most stable and precise: highest success and lowest ball-direction error.

Fig. 4. **WB-UAN improves simulator fidelity and real-world agile dribbling.** (a) WB-UAN attains the lowest rollout error when replaying hardware trajectories, better aligning the augmented simulator’s dynamics. (b) On sharp 135° turns, policy trained with PAM+WB-UAN attains the highest success and the smallest ball-direction error. (c) In high-speed dribbling, corrected actuator dynamics push the ball’s sustained and peak speed above the PAM-only policy while keeping it controllable. [Khai: add new WB-UAN results](#)

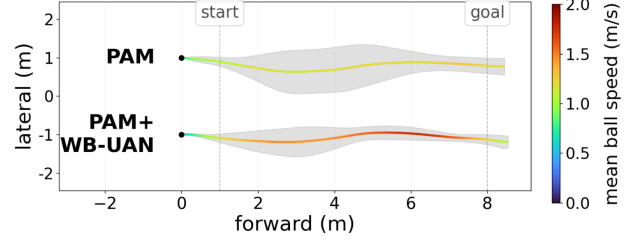
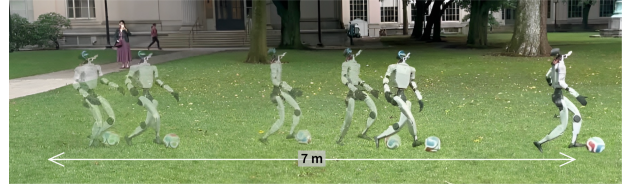
the DR and Linear policies degrade due to their conservatism or less accurate vision corrections.

B. WB-UAN Enables Agile Dribbling

1) *Sharp turning:* We further seek to push the agility boundary of our PAM-policy by adding the WB-UAN. We test this through five 135° turns at 1.2 m/s on each policy measuring both ball velocity direction error ($\arctan(\hat{v}_y^{\text{ball}}/\hat{v}_x^{\text{ball}}) - \arctan(\hat{v}_y^{\text{cmd}}/\hat{v}_x^{\text{cmd}})$) and success rate, where a success is defined as not falling with the ball in control. We optimize for both metrics based on our training rewards, so they provide insight on our policies’ ability to transfer to real. We do not have an operator for this experiment and instead hold the 1.2 m/s command towards 135° for 3 s in order to evaluate command tracking performance. We estimate the ball velocity direction in world frame at the end of the command period by combining the robot’s IMU yaw reading, ball detection on the camera frame, and joint positions over a 1 s period of 50 Hz logs.

As shown in Fig. 4b, the success rate increases with the expressivity of the perception models as well as with corrected actuator dynamics. Along with success rate, we find that real-to-sim alignment allows for behaviors encouraged by the rewards in sim to appear on real deployments.

2) *High-speed dribbling:* We evaluate our policies through a real-world ablation study on the speed of the ball during controlled dribbling. We define controlled dribbling using three factors: (1) ball in front of the robot throughout the trajectory, (2) ball controlled by multiple kicks, and (3) crossing the finish line without falling. Our trajectory had a length of 7 meters, and the operator’s goal was to minimize the time over the 7 meters while ensuring



Policy	success	average (m/s)	peak (m/s)
PAM	05/11	1.35 ± 0.20	1.56
PAM+WB-UAN	05/05	1.71 ± 0.16	2.31

(c) **High-speed dribbling.** With corrected actuator dynamics, PAM+WB-UAN sustains 1.7 m/s (peak 2.3 m/s) with higher success and a tighter trajectory spread than PAM alone.

controlled dribbling. Beside timing the trials, we also use leg odometry to estimate the ball velocity profiles. The operator did not have knowledge of the running policy, and each policy was run until it had 5 successful trials.

As shown in Fig. 4c, WB-UAN significantly improves both average speed and peak speed of the ball, highlighting the impact of corrected actuator dynamics. Additionally, the policy was easier for the operator to control as shown by the tighter spread of trajectories and higher success rate.

C. Skill-Chaining Enables Superior Skill Transitions

We evaluate our skill-chaining method by ablating the finetuning state for both dribbling to passing and dribbling to shooting. We run 4096 environments that sample a command for the dribbling stage and another for the striking stage. The length of the dribbling stage is uniformly chosen between 4 and 6 seconds with a dribbling warmup period to ensure the robot is able to dribble before switching to a striking policy. We measure four metrics: (1) missed strike rate, (2) fall rate, (3) strike speed error (m/s), and (4) strike heading error (degrees). A strike is defined as the ball crossing above $0.7 \cdot \|v_{\text{cmd}}\|$ after being within 0.5 meters of the robot. Speed and heading error are calculated at the moment the ball reaches its peak speed.

In Fig. 5 (top), we see notable improvements in missed strike rate, fall rate, speed error, and heading error (lower is better) as a result of the finetuning stage. The improvement in missed strike rate highlights that the finetuning stage allows the policy to learn how to start a kick in more diverse states that are visited in dribbling.

We further validate skill-chaining on hardware, running 5 real-world trials for each transition. As shown in Fig. 5

(bottom), skill-chaining transfers to the real robot but not perfectly, reaching 4/5 on dribble-to-pass and 3/5 on harder dribble-to-shoot. Combining it with our real2sim2real techniques (PAM and WB-UAN) raises both transitions to 5/5, showing that all three components – chaining, perception, and dynamics alignment – jointly enable reliable agile skill transitions on a humanoid robot in the real world.

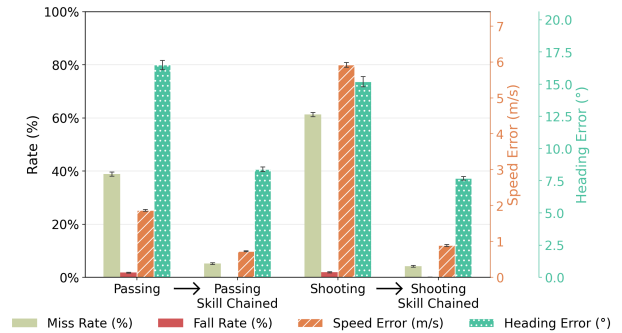
IV. RELATED WORK

Soccer and athletic skills. Reinforcement learning has enabled quadrupedal dribbling [1] and humanoid soccer through several perception and control formulations. These include policies using explicit state estimates [2, 23], learned reconstructions of privileged state [17, 24], modular RoboCup pipelines [25], and image-conditioned policies [26–28]. Xu et al. [4] develop agile directional kicking through a staged teacher–student pipeline. Like Dribble Master [3], we use active vision for humanoid dribbling, but we consider more skills, chain them together, and achieve higher dribbling speeds through whole-body control and real-to-sim alignment. Related athletic systems coordinate visual feedback and rapid object interception in badminton [29] and table tennis [30]. Our receiving objective differs from shot-blocking in robotic goalkeeping [31]: the first touch should retain the ball for subsequent play.

Real-to-sim alignment. Real-to-sim methods use hardware experience to refine the environment in which policies are trained. In legged sports, Ma et al. [29] fit perception error and detection probability as functions of range and camera angular velocity, and Wang et al. [17] fit distance-dependent perception noise from hardware measurements. Our perception alignment model follows this data-driven approach but learns a nonlinear, geometry- and motion-conditioned distribution of bounding-box errors and visibility, reproducing the detector interface used to construct the policy’s ball observations.

On the dynamics side, learned actuator networks have been shown to close the sim-to-real gap on robotic actuators. Supervised actuator networks predict output torque from a history of joint position errors and joint velocities but require torque sensing [19]. UAN [20] learns actuator-input corrections without torque supervision, and Contact-UAN [18] enables this formulation on floating-base humanoids through contact-consistent replay initialization. ASAP [32] learns residual actions from hardware trajectories recorded with MoCap and finetunes humanoid motion-tracking policies in the augmented simulator. Our whole-body actuator model builds on Contact-UAN’s replay formulation to avoid use of MoCap, replacing joint-local residuals with jointly conditioned corrections that can represent cross-joint effects.

Skill-chaining. Prior work addresses skill compatibility by aligning the states produced by one policy with those from which the next can succeed. Adversarial Skill-Chaining [33] uses a discriminator to regularize predecessor terminal states toward successor initiation sets. Sequential Dexterity [34] combines forward initialization from predecessor terminal states with backward finetuning based on learned transition



Policy	dribble to pass	dribble to shoot
Vanilla	4/5 (07/09)	3/5 (4/6)
PAM+WB-UAN	5/5 (12/12)	5/5 (7/7)

Fig. 5. **Skill chaining.** Finetuning passing and shooting over the dribbling state distribution improves striking accuracy and stability in simulation (top; lower is better). This alone does not transfer perfectly to hardware; adding our real2sim2real techniques (PAM and WB-UAN) further raises real-world success to 5/5 for both transitions (bottom).

feasibility. For humanoid soccer, SkillX [35] instead learns multiple skills within a single command-conditioned policy using skill-specific motion priors. Our approach is effective but intentionally simple: we keep dribbling fixed and finetune shooting and passing from states sampled during ongoing dribbling, exposing the policies to mid-stride transitions.

V. CONCLUSIONS

Agile humanoid soccer, comprising dribbling, shooting, passing, and receiving, requires a system that can perceive a fast-moving ball and react with precise whole-body control. Achieving this is challenging due to sim-to-real gaps in perception and actuation, and skills trained in isolation struggle when composed together. We present *Joga*, a system that closes these gaps through targeted real2sim2real modeling. A perception alignment model corrects the vision gap by learning the depth- and motion-dependent errors of onboard ball detection, while a whole-body unsupervised actuator network corrects the dynamics gap in agile dribbling by aligning simulated and real trajectories. We further chain these dribbling to passing and shooting together with a finetuning stage that trains each skill to start from the state distribution of another skill. As a result, *Joga* achieves state-of-the-art agile and controllable soccer dribbling, directed passing and shooting from dribbling, and uniquely demonstrates single-touch receiving. *Joga* establishes a new level of athletic intelligence for humanoid soccer.

ACKNOWLEDGEMENT

We acknowledge the use of coding agents including Codex, Claude Code, and Cursor for code development and when formatting figures.

REFERENCES

- [1] Y. Ji, G. B. Margolis, and P. Agrawal, “Dribblebot: Dynamic legged manipulation in the wild,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 5155–5162.

- [2] T. Haarnoja, B. Moran, G. Lever, S. H. Huang, D. Tirumala, J. Humplik, M. Wulfmeier, S. Tunyasuvunakool, N. Y. Siegel, R. Hafner *et al.*, “Learning agile soccer skills for a bipedal robot with deep reinforcement learning,” *Science Robotics*, vol. 9, no. 89, p. eadi8022, 2024.
- [3] Z. Wang, J. Zhou, and Q. Wu, “Dribble Master: Learning agile humanoid dribbling through legged locomotion,” 2025.
- [4] Z. Xu, M. Seo, D. Lee, others, J. Biswas, and P. Stone, “Learning agile striker skills for humanoid soccer robots from noisy sensory input,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2026.
- [5] K. Zakka, Q. Liao, B. Yi, L. L. Lay, K. Sreenath, and P. Abbeel, “mjlab: A lightweight framework for gpu-accelerated robot learning,” *arXiv preprint arXiv:2601.22074*, 2026.
- [6] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, “Isaac gym: High performance gpu-based physics simulation for robot learning,” *arXiv preprint arXiv:2108.10470*, 2021.
- [7] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, “Domain randomization for transferring deep neural networks from simulation to the real world,” 2017.
- [8] F. Muratore, F. Ramos, G. Turk, W. Yu, M. Gienger, and J. Peters, “Robot learning from randomized simulations: A review,” *Frontiers in Robotics and AI*, vol. 9, p. 799893, 2022.
- [9] Z. Xie, X. Da, M. Van de Panne, B. Babich, and A. Garg, “Dynamics randomization revisited: A case study for quadrupedal locomotion,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4955–4961.
- [10] J. P. Sleiman, H. Li, A. Adu-Bredu, R. Deits, A. Kumar, K. Bergamin, M. Bhardwaj, S. Biddlestone, N. Burger, M. A. Estrada *et al.*, “Zest: Zero-shot embodied skill transfer for athletic robot control,” *arXiv preprint arXiv:2602.00401*, 2026.
- [11] Q. Liao, T. E. Truong, X. Huang, Y. Gao, G. Tevet, K. Sreenath, and C. K. Liu, “Beyondmimic: From motion tracking to versatile humanoid control via guided diffusion,” *arXiv preprint arXiv:2508.08241*, 2025.
- [12] A. Kumar, Z. Fu, D. Pathak, and J. Malik, “RMA: Rapid motor adaptation for legged robots,” in *Robotics: Science and Systems (RSS)*, 2021.
- [13] M. Liu, D. Pathak, and A. Agarwal, “Locoformer: Generalist locomotion via long-context adaptation,” in *Conference on Robot Learning*. PMLR, 2025, pp. 532–546.
- [14] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal, “Rapid locomotion via reinforcement learning,” 2022.
- [15] G. B. Margolis, X. Fu, Y. Ji, and P. Agrawal, “Learning to see physical properties with active sensing motor policies,” 2023.
- [16] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath, “Real-world humanoid locomotion with reinforcement learning,” 2023.
- [17] Y. Wang, C. Luo, P. Chen, J. Liu, W. Sun, T. Guo, K. Yang, B. Hu, Y. Zhang, and M. Zhao, “Learning vision-driven reactive soccer skills for humanoid robots,” *Science Robotics*, 2026.
- [18] N. Fey, K. Nguyen, Y. Prabhu, G. B. Margolis, and P. Agrawal, “Contact-UAN: Unsupervised actuator networks as a residual world model for humanoid control,” in *Robotics: Science and Systems (RSS) Workshop on Robot World Models*, 2026.
- [19] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, “Learning agile and dynamic motor skills for legged robots,” *Science Robotics*, vol. 4, no. 26, Jan. 2019.
- [20] N. Fey, G. B. Margolis, M. Peticco, and P. Agrawal, “Bridging the sim-to-real gap for athletic locomanipulation,” in *Robotics: Science and Systems (RSS)*, 2025.
- [21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” 2017.
- [22] Y. Ze, S. Zhao, W. Wang, A. Kanazawa, R. Duan, P. Abbeel, G. Shi, J. Wu, and C. K. Liu, “Twist2: Scalable, portable, and holistic humanoid data collection system,” *arXiv preprint arXiv:2511.02832*, 2025.
- [23] Y. Zhong, Y. Lu *et al.*, “RoboNaldo: Accurate, stable and powerful humanoid soccer shooting via motion-guided curriculum reinforcement learning,” 2026.
- [24] J. Kong, X. Liu, others, and C. Bai, “Learning soccer skills for humanoid robots: A progressive perception-action framework,” 2026.
- [25] A. Labiosa, Z. Wang, S. Agarwal, W. Cong, G. Hemkumar, A. N. Harish, B. Hong, J. Kelle, C. Li, Y. Li, Z. Shao, P. Stone, and J. P. Hanna, “Reinforcement learning within the classical robotics stack: A case study in robot soccer,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.
- [26] D. Tirumala, M. Wulfmeier, B. Moran, S. Huang, J. Humplik, G. Lever, T. Haarnoja, L. Hertweck, N. Siegel, M. Wulfmeier *et al.*, “Learning robot soccer from egocentric vision with deep reinforcement learning,” in *Conference on Robot Learning (CoRL)*, 2024.
- [27] H. Xue, T. He, Z. Wang, Q. Ben, W. Xiao, Z. Luo, F. Da, F. Castañeda, G. Shi, S. Sastry, L. Fan, and Y. Zhu, “Opening the sim-to-real door for humanoid pixel-to-action policy transfer,” 2025.
- [28] W. Sun, Y. Su, L. Huang, A. Zhang, D. Wei, M. San, D. Tian, E. Cao, B. Cao, Y. Liu *et al.*, “Now you see that: Learning end-to-end humanoid locomotion from raw pixels,” *arXiv preprint arXiv:2602.06382*, 2026.
- [29] Y. Ma, A. Cramariuc, F. Farshidian, and M. Hutter, “Learning coordinated badminton skills for legged manipulators,” *Science Robotics*, 2025, doi 10.1126/scirobotics.adu3922.
- [30] D. B. D’Ambrosio, S. Abeyruwan, L. Graesser, A. Is-

- cen, H. Ben Amor, A. Bewley, B. J. Reed, K. Reymann, L. Takayama, Y. Tassa *et al.*, “Achieving human level competitive robot table tennis,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.
- [31] X. Huang, Z. Li, Y. Xiang, Y. Ni, Y. Chi, Y. Li, L. Yang, X. B. Peng, and K. Sreenath, “Creating a dynamic quadrupedal robotic goalkeeper with reinforcement learning,” 2022.
- [32] T. He, J. Gao, W. Xiao, Y. Zhang, Z. Wang, J. Wang, Z. Luo, G. He, N. Sobanbab, C. Pan *et al.*, “Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills,” in *Robotics: Science and Systems (RSS)*, 2025.
- [33] Y. Lee, J. J. Lim, A. Anandkumar, and Y. Zhu, “Adversarial skill chaining for long-horizon robot manipulation via terminal state regularization,” 2021.
- [34] Y. Chen, C. Wang, L. Fei-Fei, and C. K. Liu, “Sequential dexterity: Chaining dexterous policies for long-horizon manipulation,” 2023.
- [35] Z. Ye, E. Ruan, Y. Bao, R. Huang, J. Yang, J. Jin, Y. Huo, P. Wang, Y. Han, H. Huang, W. Cui, Y. Li, and X. Tian, “Skillx: Unified multi-skill policy learning for humanoid soccer,” 2026.