DAVIS: A Depth-Only End-to-End Active-Vision Framework for HumanoId Soccer Skills

Jiakang Jin[1,*], Yixiao Huo[1,2,*], Pengyuan Wang[1,*], Yinan Han[1,*], Tingxuan Zhang[1], Zhuobing Zhao[1], Xuanxin Zhou[1], Zhangchen Ye[1,2], Enxuan Ruan[1], Yifei Bao[1], Jiankun Yang[1], Chenghao Sun[1], Wenhao Cui[1], Xiaoyu Tian[1,†], Yiming Li[2,†]

[1]Noetix Robotics [2]Tsinghua University

[*]Equal Contribution [†]Corresponding Author

Project Page: https://thusi-lab.github.io/DAVIS/

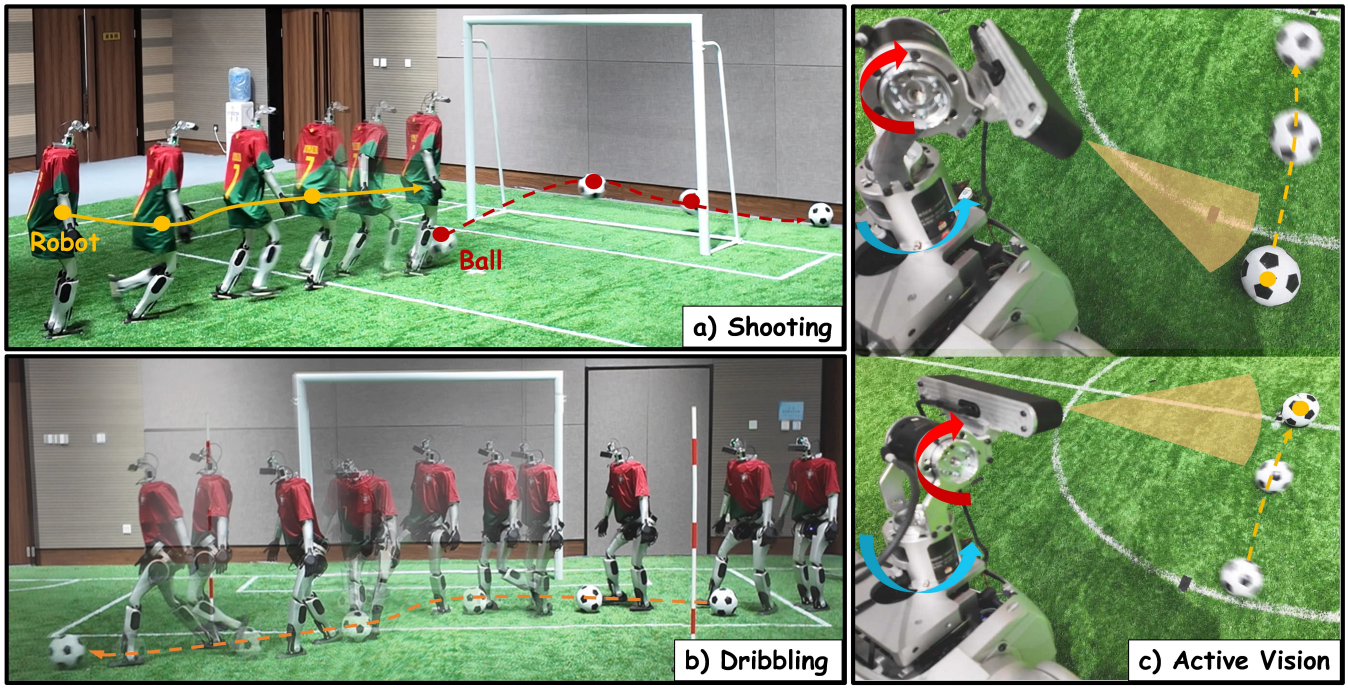

Fig. 1. **Overview of our humanoid soccer system.** (a) The robot performs goal-oriented shooting by approaching the ball and kicking it into the goal. (b) The robot demonstrates agile multi-directional dribbling with continuous ball control, including turning and weaving around poles. (c) Our active vision design enables the robot to actively reorient its head in yaw and pitch to keep the ball near the center of the field of view, thereby improving perception-action coordination.

***Abstract*— Humanoid soccer contact skills require more than producing high-impact foot–ball contacts: the robot must close the loop over perception, approach, alignment, impact, and recovery while its own motion induces substantial viewpoint changes, frequent loss of the ball from view, and uncertain contact outcomes. In this work, we ask a compact yet stricter question: can a humanoid learn soccer contact skills using only a head-mounted depth image, proprioceptive history, and an optional low-dimensional task command, and directly output 25-DoF joint PD targets without extra runtime perception or planning modules? To this end, we propose DAVIS, a depth-only end-to-end framework for humanoid soccer skills that learns visibility-aware auxiliary geometry during training, and combines GT-to-prediction annealing, task curricula, and AMP-style motion priors to smoothly bridge privileged supervision and real deployment. Built on this framework, we instantiate representative soccer contact skills, including goal-directed shooting and directional dribbling, through task-specific definitions of objects, commands, rewards, and curricula, and validate them through simulation, Noetix E1 real-robot experiments, and ablations.**

I. Introduction

Humanoid robots have made substantial progress in reinforcement-learning-driven dynamic locomotion [1]–[6], but soccer remains a harder setting because it requires the robot to perceive, approach, align, contact, and recover while the target and the robot itself are moving. Unlike standard locomotion benchmarks, soccer contact skills require the policy to reason not only about *where the ball is*, but also about *how to approach it*, *how to keep the target in view*, and *how contact should change the subsequent ball motion*.

On real humanoids, onboard sensing and whole-body control further amplify these difficulties: the head-mounted view changes with body motion, active head yaw and pitch determine what evidence the policy receives, and small perception or stance errors can change the contact outcome [7]–[11]. Existing soccer systems often manage this complexity with modular perception, localization, planning, and skill-switching interfaces, including RGB detectors, LiDAR/RGB-

D-based localization, or external state inputs. Because soccer contact skills depend on local ball and goal geometry, head-mounted depth provides a direct geometric observation from which to study control without externally supplied scene state. This motivates the question: **can a humanoid learn deployable soccer contact skills from only head-mounted depth, proprioceptive history, and low-dimensional task commands — without any external scene state?**

We propose **DAVIS**, a depth-only end-to-end reinforcement learning framework to answer this question. At deployment, the policy remains a closed-loop mapping from head-mounted depth, proprioceptive history, and low-dimensional commands to 25-DoF joint PD targets. This is achieved by placing structure inside the learning signal rather than in the deployment pipeline, following the idea of using privileged simulation information to guide deployable policies [12]–[14]. Auxiliary geometry and visibility predictions are learned from privileged simulation supervision, while the actor gradually transitions from stable geometric guidance to self-predicted features via GT-to-prediction annealing and visibility gating. With this interface fixed, different soccer behaviors can be learned by changing the local objective, geometry, and supervision that define the task.

DAVIS is designed to support a range of soccer skills through a shared learning framework: new skills are specified by task objects, commands, rewards, and curricula while retaining the depth-to-control interface. We evaluate goal-directed shooting and directional dribbling in detail as representative instances of one-shot striking and sustained ball control, using them to assess the feasibility of the framework across complementary contact regimes. Additional examples, including stopping the ball and obstacle-aware dribbling, are shown in Fig. 5; ball-loss recovery is demonstrated in the project-page video rather than as a panel in Fig. 5. Together, these instances provide evidence of the framework's extensibility through task-specific training.

In summary, our contributions are as follows: (1) We introduce a **depth-only end-to-end RL framework** for humanoid soccer contact skills, where the deployed policy maps head-mounted depth, proprioceptive history, and low-dimensional task commands directly to 25-DoF joint targets with active-vision head control. (2) We develop a **visibility-aware auxiliary geometry mechanism** that learns task-relevant object geometry from the depth latent, and combines visibility masking with GT-to-prediction annealing to stabilize visual learning without adding runtime perception modules. (3) We instantiate multiple soccer skills within the same framework and establish a complete pipeline from IsaacLab training to E1 real-robot deployment, including depth alignment, policy export, and 25-DoF PD control. Real-robot evaluations and additional skill demonstrations support the framework's practical feasibility and applicability across soccer tasks.

II. Related Work

A. Robot Soccer Skill Learning

Robot soccer has long served as a challenging benchmark for robot control, requiring the integration of visual perception, real-time decision making, agile locomotion, and ball-oriented skills [15]–[17]. RoboCup-style soccer systems often relied on modular perception-control pipelines, where sensing, state estimation, and soccer control were separately engineered rather than learned end-to-end [18]. Recent learning-based methods have enabled legged robots to acquire soccer skills through various paradigms, including reinforcement learning and imitation learning [19]–[26]. Yet most existing systems still depend on RGB-based object detection or multi-sensor perception pipelines [27]–[30].

In contrast, our work studies depth-driven humanoid soccer skill learning, where the robot learns to chase, dribble, and kick using only depth observations and proprioception.

B. Whole-Body Loco-Manipulation

Whole-body loco-manipulation on humanoid robots is inherently challenging, as the robot must simultaneously coordinate balance, locomotion, and task-directed manipulation within a high-dimensional and tightly coupled control space [31]–[33]. To make the problem tractable, prior systems often introduce decoupled or hierarchical interfaces, such as separate upper- and lower-body controllers r high-level task policies paired with low-level whole-body controllers [34]–[37]. Humanoid soccer further amplifies this difficulty: unlike manipulating static objects, the ball is a highly dynamic target, requiring the robot to approach from a suitable direction, maintain visibility, and generate precise foot–ball contacts for effective ball control.

To address these challenges, we propose a whole-body control framework, **DAVIS**, that jointly optimizes active visual tracking, approach, and foot–ball interaction for dynamic ball-centric loco-manipulation.

III. Method

A. Problem Formulation

We model humanoid soccer contact skills as a POMDP problem [38], denoting the process as $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma \rangle$. The simulator maintains a full state $\mathbf{s}_t \in \mathcal{S}$ containing robot, ball, goal, and contact-related physical variables; during training, privileged variables derived from this state, such as ball pose, goal pose, visibility, contact state, and reference motion, are used as critic inputs, reward-computation terms, and auxiliary-supervision labels. The deployed policy instead observes only a head-mounted depth image $\mathbf{D}_t$, a proprioceptive history $\mathbf{p}_{t-H:t}$ of length H, and an optional low-dimensional task variable $\mathbf{c}_t$; for instances without an external command, $\mathbf{c}_t$ is omitted. We write the deployed observation and action as $\mathbf{o}_t = (\mathbf{D}_t, \mathbf{p}_{t-H:t}, \mathbf{c}_t)$ and $\mathbf{a}_t = \pi_\theta(\mathbf{o}_t) \in \mathbb{R}^{25}$. The proprioceptive history is encoded inside the policy to infer latent body dynamics, while all explicit task geometry used by the actor must be produced from deployable inputs. The low-level controller converts the

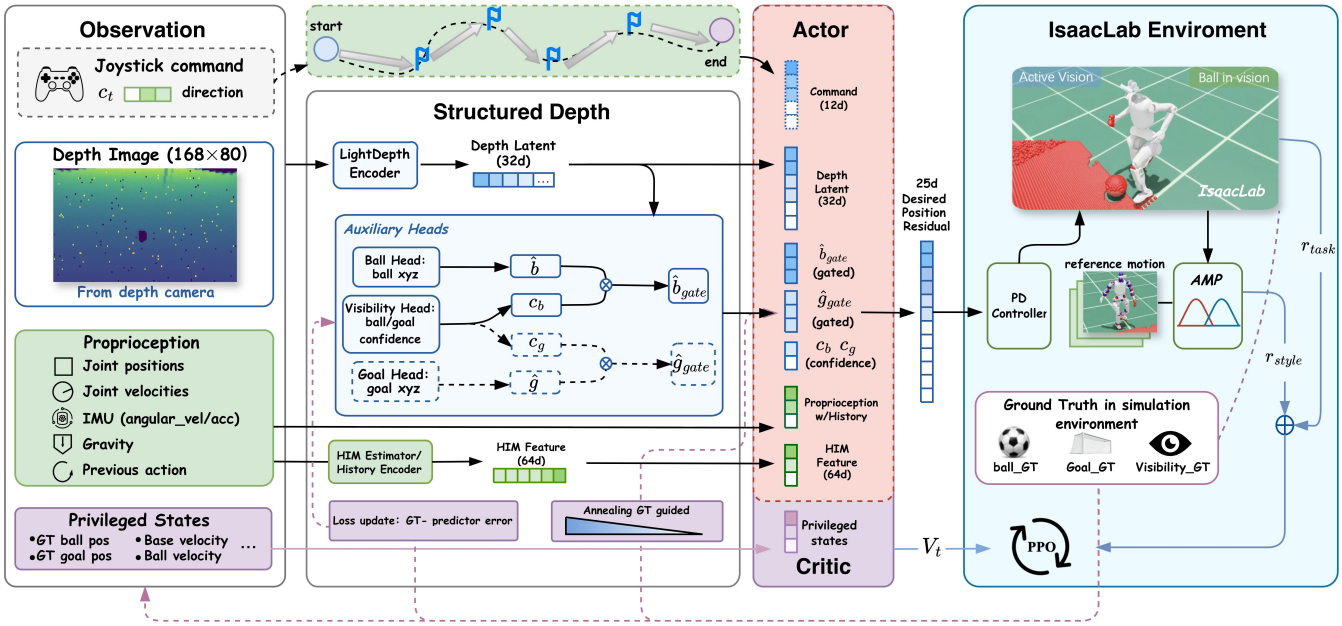

Fig. 2. **Overview of our DAVIS framework.** The observation stream contains the command c_t, **a single** 168×80 **depth image**, proprioceptive history, and privileged simulation states. The structured-depth module encodes the depth input into a 32-D latent, predicts ball/goal geometry and visibility confidence through **auxiliary heads**, forms confidence-gated geometry features, and combines them with a 64-D HIM/history feature. The actor receives the command, depth latent, gated geometry/confidence features, proprioceptive history, and HIM feature, and outputs a 25-D position residual for the PD controller. The auxiliary losses, GT-to-prediction annealing, PPO update, task reward, and style reward use simulation-only signals from IsaacLab rollouts, including ball, goal, and visibility ground truth. Dashed boxes in the structured-depth module indicate optional task-specific modules.

action into joint PD targets $\mathbf{q}_t^{\text{des}} = \mathbf{q}^0 + \mathbf{s} \odot \mathbf{a}_t$, where $\mathbf{q}^0$ is the nominal joint configuration and $\mathbf{s}$ is the per-joint action scale. The E1 humanoid consists of 23 body DoFs and 2 head DoFs, so we decompose the action as $\mathbf{a}_t = [\mathbf{a}_t^{\text{body}}, \mathbf{a}_t^{\text{head}}]$ with $\mathbf{a}_t^{\text{body}} \in \mathbb{R}^{23}$ and $\mathbf{a}_t^{\text{head}} \in \mathbb{R}^2$. Because the depth camera is mounted on the actuated head, the head action is part of the sensing loop: it directly changes the next depth observation. We optimize the asymmetric actor-critic policy with PPO [39] by maximizing the expected discounted return

$$\max_{\theta} \mathbb{E}_{\pi_\theta} \left[\sum_{t=0}^{T} \gamma^t \left(r_t^{\text{task}} + \lambda_{\text{amp}} r_t^{\text{amp}} - p_t^{\text{reg}} \right) \right], \qquad (1)$$

where r_t^{task} denotes task reward, r_t^{amp} encourages motion near a reference prior, and p_t^{reg} penalizes joint limits, action changes, unstable posture, collisions, and unsafe contact.

B. Depth-Only End-to-End Structured Soccer RL Framework

We build **DAVIS** around a deployment interface that remains deliberately narrow: the actor receives head-mounted depth, proprioceptive history, and an optional low-dimensional command, and outputs 25-DoF joint target residuals for the PD controller, as illustrated in Fig. 2. During training, privileged simulation states provide dense supervision for perception, value learning, and motion regularization; at deployment, the policy keeps only the depth/proprioception-to-control path.

To keep this deployment path compact, we introduce task structure only through training-time supervision and regularization. The depth latent feeds auxiliary geometry and visibility heads, whose predictions are confidence-gated before entering the actor. Simulation ground truth supplies geometry and visibility labels, but only as auxiliary supervision and as early-stage geometric guidance. A GT-to-prediction annealing schedule gradually replaces privileged features with predicted features, so the actor first learns how local geometry should affect whole-body motion and then operates with geometry inferred from depth. In parallel, PPO optimizes task return, the history estimator learns proprioceptive state inference, and AMP-style motion priors regularize full-body motion. These components make long-horizon soccer contact learning tractable while preserving a single deployable policy graph [40]–[42].

The next subsection specifies how task objects, variables, rewards, curricula, and optional contact conditioning instantiate individual skills while preserving this deployment interface.

C. Visibility-Aware Auxiliary Geometry

Learning task geometry from depth only through task returns is inefficient: task objects can occupy a small image region, while the gradient from PPO returns must pass through the actor and the depth encoder before shaping visual features. We therefore attach a visibility-aware auxiliary geometry module to the depth latent and supervise it with privileged simulation labels during training. As shown in Fig. 3, the module predicts object-level geometry and visibility inside the policy, so the actor receives direct object-level supervision without adding a runtime perception module.

Let $\mathbf{z}_t^D = f_D(\mathbf{D}_t) \in \mathbb{R}^{32}$ be the depth latent. For each task-relevant object $\text{obj}_i \in \mathcal{O}$, the auxiliary module predicts its 3-D center and visibility confidence from the depth latent,

$$(\hat{\mathbf{y}}_t^i, \hat{q}_t^i) = h_{\text{aux}}^i(\mathbf{z}_t^D), \qquad \hat{p}_t^i = \sigma(\hat{q}_t^i), \qquad (2)$$

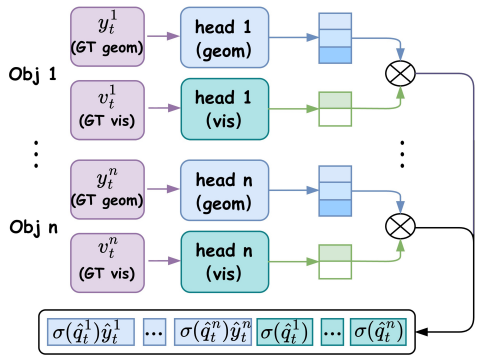

Fig. 3. Scalable auxiliary geometry. Each task object is assigned a geometry head and a visibility head supervised by privileged labels during training; their outputs are visibility-gated and concatenated into the actor feature, enabling the same mechanism to extend from one object to n objects.

where $\mathbf{y}_t^i = [x_t^i, y_t^i, z_t^i]^\top$ denotes the object center in a right-handed head-yaw-aligned frame, with x along the head-yaw forward direction, y to the left, and z upward. Both geometry and visibility predictions depend only on the current 32-D depth latent; head states are provided separately to the actor through proprioception. The same construction extends to $\mathcal{O} = \{\text{obj}_i\}_{i=1}^n$ by assigning each object an indexed 3-D center prediction $\hat{\mathbf{y}}_t^i$, visibility prediction $\hat{q}_t^i$, ground-truth center $\mathbf{y}_t^i$, and visibility label v_t^i.

Visibility is learned jointly with geometry because a head-mounted camera can temporarily lose task-relevant objects. Geometry is supervised only when the corresponding object is visible, while visibility is trained with a binary objective,

$$\mathcal{L}_{\text{aux}} = \sum_{i=1}^{n} \lambda_i v_t^i \ell_i(\hat{\mathbf{y}}_t^i, \mathbf{y}_t^i) + \sum_{i=1}^{n} \lambda_{v_i} \text{BCE}(\hat{q}_t^i, v_t^i). \tag{3}$$

Visibility labels are computed using the full camera transform together with valid field-of-view and depth-range constraints; for multi-point objects, visibility is positive if any predefined structural point is valid, while the geometry target always remains the single 3-D object center.

To avoid forcing the actor to rely on unreliable predictions early in training, we gradually replace privileged geometry with predicted geometry. Let $\beta_k \in [0, 1]$ denote the prediction weight at policy update k. For object i,

$$\mathbf{y}_{t,k}^{e,i} = (1 - \beta_k)\mathbf{y}_t^i + \beta_k \hat{\mathbf{y}}_t^i. \tag{4}$$

Using the correspondingly annealed visibility confidence $p_{t,k}^{e,i}$, we define the detached confidence gate

$$c_{t,k}^i = \text{clip}\left(\text{stopgrad}(p_{t,k}^{e,i}), 0.05, 1\right). \tag{5}$$

The actor-side auxiliary representation is then

$$\mathbf{u}_{t,k}^{\text{aux}} = \text{Concat}_{i=1}^n \left[c_{t,k}^i \mathbf{y}_{t,k}^{e,i}, c_{t,k}^i\right]. \tag{6}$$

As β_k increases from 0 to 1, the actor transitions from privileged geometric guidance to depth-based predictions. At deployment, all auxiliary features are predicted solely from depth.

D. Skill Instantiation

Once the deployment interface in Sec. III-B is fixed, a soccer skill is specified by a task definition rather than by switching runtime modules. This separation makes the interface reusable: new skills retain the depth-to-control path and are introduced by changing task objects, commands, objectives, and curricula. We denote such a definition by $\mathcal{T} = (\mathcal{O}, \mathbf{c}_t, \mathbf{y}_t, r_t, \mathcal{K})$, where $\mathcal{O}$ is the set of task objects, $\mathbf{c}_t$ is the optional command, $\mathbf{y}_t$ is the task geometry supervised through Sec. III-C, r_t defines the contact objective, and $\mathcal{K}$ is the curriculum. We use goal-directed shooting and directional dribbling as representative instances to verify the framework across complementary contact regimes; following details illustrate how the shared framework is instantiated for these examples.

1) Goal-Directed Shooting: Shooting is defined by the ball–goal geometry. Let $\mathbf{b}_t$ and $\mathbf{g}_t$ denote the ball position and goal center in the head-yaw-aligned frame, and let $\mathbf{d}_t^{bg} = \frac{\mathbf{g}_t - \mathbf{b}_t}{\|\mathbf{g}_t - \mathbf{b}_t\|_2 + \epsilon}$ be the normalized ball-to-goal direction. A behind-ball target can then be written as $\mathbf{x}_t^{\text{behind}} = \mathbf{b}_t - \rho \mathbf{d}_t^{bg}$, which turns chasing, alignment, and contact into one geometric target family. In this task, the object set is $\mathcal{O}_{\text{shoot}} = \{\text{ball}, \text{goal}\}$, the task geometry is $\mathbf{y}_t^{\text{shoot}} = (\mathbf{b}_t, \mathbf{g}_t)$, and the objective is to strike the ball so that its exit motion moves toward the goal. The reward is organized around run-up, contact, terminal success, visual maintenance, and safety regularization, while the curriculum gradually broadens the initial ball placement and approach difficulty. Active vision is maintained through the head DoFs: the head keeps the ball within view, and the body gradually follows the viewing direction.

2) Directional Dribbling: Dribbling is defined by a direction command and sustained ball regulation. Let $c_t \in \mathcal{C}$ be the commanded direction angle, where $\mathcal{C} = \{k\pi/6 : k = -6, \ldots, 5\}$ is a 12-way ring uniformly spaced over 360°, and $\mathbf{d}_t^{\text{cmd}} = [\cos c_t, \sin c_t]$ the corresponding horizontal direction. The task object set is $\mathcal{O}_{\text{dribble}} = \{\text{ball}\}$, and the auxiliary geometry is the 3-D ball center $\mathbf{y}_t^{\text{ball}} = \mathbf{b}_t$. The commanded direction $\mathbf{d}_t^{\text{cmd}}$ and the robot–ball horizontal distance d_t are task variables used to define the dribbling objective rather than outputs of the auxiliary geometry head. Unlike shooting, the objective is not a single high-speed strike; the policy must keep the ball within a controllable band and generate progress along $\mathbf{d}_t^{\text{cmd}}$. We express the controllable-band condition by $\chi_t^{\text{band}} = \mathbb{I}\big(|d_t - d_0| < \Delta_d,\, v_{lo} < \|\mathbf{v}_t^b\| < v_{hi},\, \mathbf{v}_t^b \cdot \mathbf{d}_t^{\text{cmd}} > 0\big)$. The reward covers commanded progress, re-approach, heading, visual maintenance, and out-of-band penalties. Continuous dribbling further uses contact-intent-conditioned AMP and phase masks so approach, touch, follow-through, and recovery emphasize different motion states.

E. Training and Deployment Pipeline

The deployment policy uses a single-frame head-mounted depth image, a five-step proprioceptive history, and task variables to output 25-DoF joint PD targets. Since this depth image is the main sim-to-real interface, we align simulation

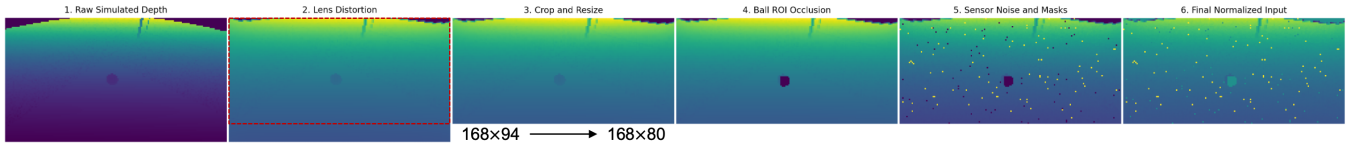

Fig. 4. Depth alignment transforms the raw depth frame into the aligned policy input through distortion correction, cropping from 168×94 to 168×80, ball ROI occlusion, sensor noise, random masks and normalization, so the same single-frame depth interface can be used both in sim and real.

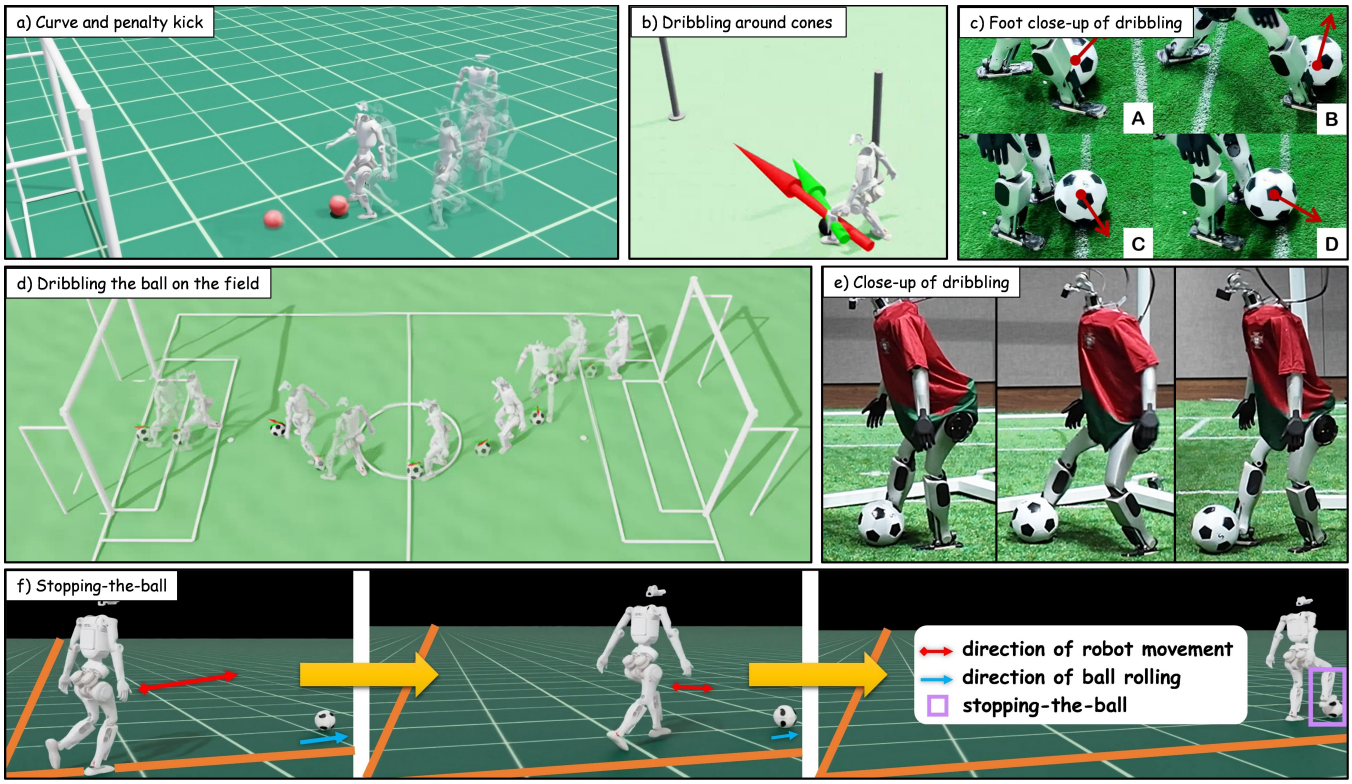

Fig. 5. **Qualitative examples of learned soccer skills.** In simulation: (a) curved and penalty kicks, (b) cone-avoidance dribbling, (d) dribbling across a soccer field, and (f) stopping a rolling ball, where red and blue arrows indicate robot and ball motion, respectively, and the purple box marks the stopped ball. On the real robot: (c) foot–ball contact during dribbling, with arrows indicating ball motion, and (e) a close-up dribbling sequence.

and deployment before it enters the actor. As shown in Fig. 4, a raw frame $\mathbf{D}_t$ is processed to $\tilde{\mathbf{D}}_t$. During training, the aligned frame is further perturbed to reduce dependence on clean depth and serve as domain randomization for sim-to-real transfer [43], [44]. Both simulated and real depth frames are cropped to 168×80, clipped to 0.3–5.0 m, normalized, and zero-filled at invalid pixels. Policies are trained in Isaac Lab for 20,000 PPO iterations with AMP and a history-informed latent estimator (HIM) [45], then exported to ONNX. On the robot, the actor runs at 50 Hz, the low-level PD controller at 200 Hz, and the depth stream at 15 Hz; no external localization, object detector, state estimator, or runtime soccer planner is used.

IV. Experiments

We evaluate whether **DAVIS** supports both simulated and real-world soccer contact skills, as illustrated in Fig. 5. And we answer the following questions: **Q1.** Can a depth-only end-to-end policy perform soccer contact skills? **Q2.** Which training structures are necessary? **Q3.** Does the learned policy transfer to the real Noetix E1 robot?

A. Simulation Experiments

To answer Q1, we extensively evaluate **DAVIS** in IsaacLab simulator on two representative skill categories in soccer: shooting and dribbling.

Shooting and directional dribbling are trained as separate task-specific checkpoints that share the same depth-to-control interface; they are not a single multi-skill policy. At deployment, the shooting actor receives no operator command, while the dribbling actor receives only the intended direction from a joystick, quantized to the same 12-way ring defined in Sec. III-D.2.

Shooting: The robot is required to perform shooting with randomly initialized robot or ball positions. In the penalty kick task, the robot starts within an angle of $[-100°, 100°]$ behind the ball and distances of 2–3.5 m. In the free kick task, the robot is initialized 5 m in front of the goal, and is required to kick the ball within the region $x \in [0, 4]$ m and $y \in [-3, +3]$ m. The difficulty level categories are shown in Fig. 6. We record the success rate of the task. Experimental results in Table I demonstrate strong positional generalization capability. By analyzing the kinematic pattern during the shooting process in Fig. 6, we observe a smooth gait and

TABLE I: Quantitative results for goal-directed shooting and directional dribbling in simulation and on the real robot. Shooting entries and real-robot dribbling entries report success rate (SR) on a 0–1 scale; simulation dribbling reports SR, velocity error (E_{vel}), and ball-in-band ratio (R_{bib}) over 900 repeated-S slalom trials. Arrows indicate the preferred direction, and bold marks the best result within each fixed-N simulation-dribbling group. For shooting, a success requires the ball to cross the goal line in one kick. For dribbling, a success requires the robot to remain upright throughout execution and both the robot and ball to finish within 1 m of the target.

Goal-Directed Shooting						**Directional Dribbling**										
Setting	Metric	Easy	Med.	Hard	Total	Setting	Metric	$N=3$			$N=5$			$N=7$		
								60°	90°	120°	60°	90°	120°	60°	90°	120°
Sim.	Penalty SR ↑	0.95	0.96	0.56	0.85	Sim.	SR ↑	0.71	**0.77**	0.65	0.66	**0.72**	0.55	**0.68**	0.64	0.43
	Free-kick SR ↑	1.00	0.88	0.71	0.85		E_{vel} ↓	**0.69**	0.74	0.91	0.74	**0.73**	0.80	**0.71**	0.81	0.97
	–	–	–	–	–		R_{bib} ↑	0.73	**0.74**	0.64	**0.76**	0.75	0.70	**0.76**	0.72	0.60
Real	Penalty SR ↑	0.68(26/38)	0.57(28/49)	0.55(33/60)	–	Real	SR ↑	Easy: 0.79(22/28)			Medium: 0.68(19/28)			Hard: 0.61(17/28)		

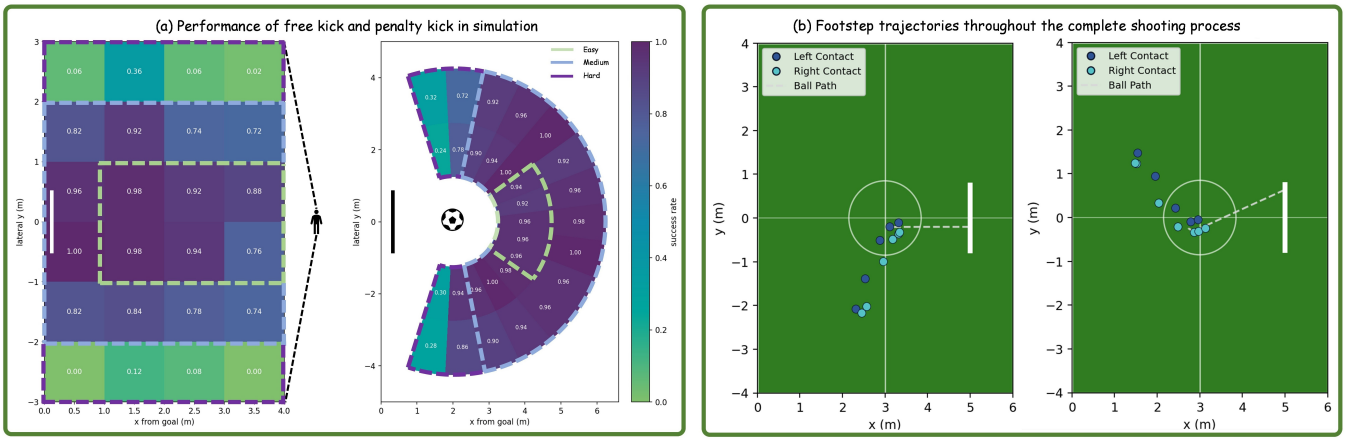

Fig. 6. (a) Free-kick and Penalty-kick performance in simulation. Each region uses over 50 trials. Darker colors indicate higher success rates, and dashed lines mark difficulty regions. (b) Footstep trajectories during shooting. The policy shows a smooth gait and consistent speed pattern.

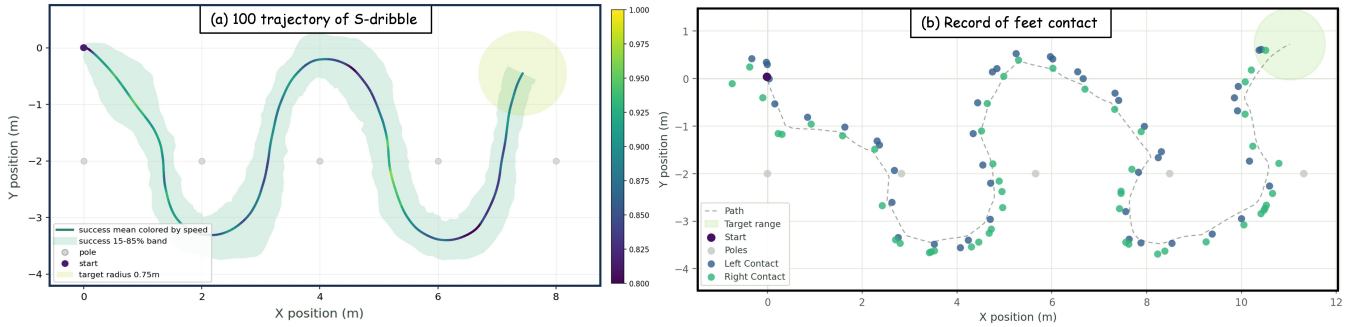

Fig. 7. Performance of repeated S-slalom in simulation. (a) Statistical results of 100 S-slalom trajectories. (b) Foot-contact record from a representative successful trajectory.

consistent speed pattern.

Dribbling: The robot is required to complete a repeated-S slalom task. N is the number of poles, and α is the turning angle of instruction. A task is considered successful if the robot passes through all waypoints in the predefined order. To ensure consistency, we use a fixed instruction sequence for all experiments. The policy achieves an average success rate of 0.65 across 900 trials. Experimental results in Table I demonstrate that more than half of the task configurations obtain success rates above 0.65, showing robustness in long trajectory and extreme angles. The visualization in Fig. 7 shows the policy's dynamic behavior.

Head-control comparison: We compare end-to-end learned head control with a heuristic that gazes at the predicted ball (Table II, last row). On static objects such as penalty kicks, the heuristic is comparable to learned active vision. On highly dynamic dribbling, the heuristic fails while the learned controller remains effective. The gap confirms that head motion must be learned jointly with locomotion and contact, rather than as a detached look-at-ball module, motivating active vision within the end-to-end policy.

Comparison with an external protocol: Dribble Master [27] is the closest published system to our setting. Both use full-size humanoids with 2-DoF active heads and onboard sensing only; however, its policy consumes detector-derived ball coordinates, whereas DAVIS consumes a depth image. We evaluate DAVIS on its two core dribbling tasks with 50 trials per condition. At the 1 m tolerance, DAVIS achieves target-reaching and obstacle-avoidance success rates of 0.88 and 0.96, compared with the reported Dribble Master rates of 0.87 and 0.93.

B. Ablation Study

To verify the necessity of key structures and answer Q2, we conduct ablation studies on three tasks in simulation and record different metrics in Table II. All methods share the

TABLE II: Simulation ablations and comparison. (a) Ablations. Penalty-kick and free-kick results use over 900 shooting trajectories, and repeated-S slalom uses over 100 dribbling trajectories. Metrics are success rate (SR), body-to-goal angular error (E_{ang}), successful ball-contact ratio (r_{contact}), velocity error (E_{vel}), and time in the control band (r_{bib}). The full method provides the best overall balance across tasks and metrics. (b) Comparison between our end-to-end learned head control (Active vision) and a heuristic head controller (Heuristic) that gazes at the predicted ball. The heuristic performs comparably to active vision in static-object tasks (e.g., penalty kicks), while active vision shows a clear advantage in highly dynamic dribbling scenarios.

Method	Penalty kick			Free kick			Repeated-S slalom		
	SR ↑	E_{ang} ↓	r_{contact} ↑	SR ↑	E_{ang} ↓	r_{contact} ↑	SR ↑	E_{vel} ↓	r_{bib} ↑
DAVIS	0.84	**23.29**	**99.75**	**0.80**	**30.18**	**99.27**	**0.70**	0.79	**0.74**
(a) Ablation									
DAVIS-w/o-curriculum	0.62	27.40	32.53	0.55	34.42	87.67	0.51	**0.77**	0.69
DAVIS-auxiliary head-w/o-anneal	0.32	54.79	84.87	0.22	76.20	87.92	0.31	0.83	0.66
DAVIS-auxiliary head-w/o-geometry	0.20	53.53	87.33	0.15	69.47	78.13	0.34	0.91	0.54
DAVIS-auxiliary head-w/o-supervision	0.13	81.87	93.33	0.14	93.66	91.67	0.30	0.95	0.48
DAVIS-auxiliary head-w/o-visibility gate	0.34	51.97	91.33	0.29	68.44	92.08	0.27	1.06	0.64
DAVIS-w/o-auxiliary head	0	–	0	0	–	0	0	1.45	0.05
DAVIS-w/o-active vision	0	–	0	0.36	97.81	1.71	0	1.25	0.03
(b) Comparison									
DAVIS-Head controller	**0.88**	26.66	93.67	0.76	37.96	88.13	0.0	0.99	0.13

same training parameters and evaluation settings.

Ablation on curriculum. We remove the ramp-up and scaling curriculum. All reward terms are activated with full weights from the beginning. Without the progressive curriculum, the policy requires more training iterations to converge, and performance is significantly worse than baseline.

Ablation on auxiliary head. Our policy incorporates an object position prediction head. To validate effectiveness, we mask the predicted positions fed into the actor, while retaining the network architecture and input dimension. We disable the aux loss and stop backpropagation. As expected, without the explicit predictions, the model struggles to localize objects to make contact and success drops to zero. To further isolate each switch of the full auxiliary head, we disaggregate this into four finer variants: w/o anneal keeps $\beta_k = 1$ throughout (always predicted auxiliary features; no GT→pred schedule); w/o geometry zeros the actor's auxiliary input while keeping auxiliary losses; w/o supervision drops the goal/ball/visibility losses but still feeds predicted features; w/o vis. gate skips visibility×geometry masking.

Ablation on active vision. Active vision is another key design. Since its components all act through the same two head-joint control dimensions and cannot be ablated independently, We fix the 2-DoF head joints, replace the original head-based rewards with body alignment rewards, and adjust the curriculum. After fixing the head joints, the ball frequently moves out of the FOV or enters blind spots, causing collapse. Another notable observation is that the robot is unable to locate the gate. The performance drops nearly to zero, with a high likelihood of falling.

C. Real-World Deployment

To answer Q3, we deploy our policies on the Noetix E1 humanoid robot equipped with a ZED2i depth camera mounted on a 2-DoF head. For the shooting experiments, we evaluate 12 predefined positions in one penalty-kick protocol; the Easy, Medium, and Hard settings use progressively more distant and more oblique positions, with 38, 49, and 60 repeated trials, respectively. For dribbling experiments, we design three tasks including reaching target position, obstacle traversal, and slalom dribbling. To better reflect real-world soccer scenarios, all experiments are conducted on a grass surface. Safety protection uses a tether or staff member; emergency-stop and projected-gravity fall triggers terminate unsafe rollouts, which are restarted from their initial conditions. If the ball is lost without an active recovery behavior, the contact policy exits to the default walk–run policy. Fig. 1 and Fig. 5 show that our policy can perform dynamic soccer tasks with precise actions. Quantitative results in Table I demonstrate that our policy maintains a high success rate in real deployment. Shooting records 26/38, 28/49, and 33/60 successes for the Easy, Medium, and Hard settings, respectively; a success requires the ball to cross the goal line in one kick. Dribbling records 22/28, 19/28, and 17/28 successes for target reaching, obstacle traversal, and slalom, respectively; a successful dribbling trial requires the robot to remain upright throughout execution and both the robot and ball to finish within 1 m of the target. All dribbling trials were conducted by the same operator using a joystick controller.

V. Conclusion

We presented a depth-only end-to-end active-vision framework **DAVIS** for humanoid soccer skills. The main idea is simple: keep the deployment interface minimal, but place enough structure inside training so that the policy can learn soccer-specific geometry, visibility, and whole-body contact behavior from depth and proprioception alone.

Within the shared framework, goal-directed shooting and directional dribbling serve as representative instances of one-shot striking and sustained ball control, retaining the observation-to-control interface while changing task-specific objects, commands, rewards, curricula, and motion-prior conditioning. Across simulation, ablations, and the complete Noetix E1 deployment pipeline, real-robot execution confirms that this depth-only interface transfers across contact regimes; together with additional skill demonstrations, these results support DAVIS as an extensible framework for humanoid soccer skills through task-specific training.

VI. Limitations and Future Work

Two limitations remain. The first is perceptual: depth-only sensing keeps the interface compact but ties the policy to a single head-mounted camera, inheriting its occlusions, dropouts, and imperfectly simulated depth—a residual sim-to-real gap that limits real-world accuracy. The second sits upstream: the task-definition interface unifies shooting and dribbling, yet each skill needs hand-tuned rewards, curricula, and contact conditioning, leaving the pipeline a recipe, not an automatic one.

Closing the perceptual gap is largely engineering—sensor-noise modeling and real-world fine-tuning. Automating skill construction is the more open problem, where language-conditioned specification and automatic curricula could generate new behaviors on the same interface.

References

[1] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Proceedings of the 5th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 164. PMLR, 2022, pp. 91–100.

[2] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.

[3] Z. Zhuang, S. Yao, and H. Zhao, "Humanoid parkour learning," 2024.

[4] T. He, J. Gao, W. Xiao, Y. Zhang, Z. Wang, J. Wang, Z. Luo, G. He, N. Sobanbabu, C. Pan, Z. Yi, G. Qu, K. Kitani, J. Hodgins, L. Fan, Y. Zhu, C. Liu, and G. Shi, "Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills," 2025.

[5] S. Zhu, Z. Zhuang, M. Zhao, K.-Y. Lee, and H. Zhao, "Hiking in the wild: A scalable perceptive parkour framework for humanoids," 2026.

[6] Z. Wu, X. Huang, L. Yang, Y. Zhang, K. Sreenath, X. Chen, P. Abbeel, R. Duan, A. Kanazawa, C. Sferrazza, G. Shi, and C. K. Liu, "Perceptive humanoid parkour: Chaining dynamic human skills via motion matching," 2026.

[7] H. Xiong, X. Xu, J. Wu, Y. Hou, J. Bohg, and S. Song, "Vision in action: Learning active perception from human demonstrations," in *Proceedings of The 9th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 305. PMLR, 2025, pp. 5450–5463.

[8] Y. Liu, S. Mu, X. Chao, Z. Li, Y. Mu, T. Chen, S. Li, C. Lyu, X.-P. Zhang, and W. Ding, "Avr: Active vision-driven robotic precision manipulation with viewpoint and focal length optimization," 2025.

[9] X. Cheng, J. Li, S. Yang, G. Yang, and X. Wang, "Open-television: Teleoperation with immersive active visual feedback," 2024.

[10] K. Nakagawa, Y. Ohmura, and Y. Kuniyoshi, "Imitation learning for active neck motion enabling robot manipulation beyond the field of view," 2025.

[11] Y. Fu, F. Xie, C. Xu, J. Xiong, H. Yuan, and Z. Lu, "Demohlm: From one demonstration to generalizable humanoid loco-manipulation," 2025.

[12] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," 2021.

[13] H. Wang, H. Luo, W. Zhang, and H. Chen, "Cts: Concurrent teacher-student reinforcement learning for legged locomotion," 2024.

[14] H. Zhang, H. Yu, L. Zhao, A. Choi, Q. Bai, Y. Yang, and W. Xu, "Slim: Sim-to-real legged instructive manipulation via long-horizon visuomotor learning," 2025.

[15] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa, "Robocup: The robot world cup initiative," in *Proceedings of the First International Conference on Autonomous Agents*, 1997, pp. 340–347.

[16] R. Gerndt, D. Seifert, J. Baltes, S. Sadeghnejad, and S. Behnke, "Humanoid robots in soccer: Robots versus humans in robocup 2050," *IEEE Robotics & Automation Magazine*, vol. 22, no. 3, pp. 147–154, 2015.

[17] M. Beukman, B. Ingram, G. Nangue Tasse, B. Rosman, and P. Ranchod, "Robocupgym: A challenging continuous control benchmark in robocup," 2024.

[18] S.-J. Yi, S. McGill, D. Hong, and D. D. Lee, "Hierarchical motion control for a team of humanoid soccer robots," *International Journal of Advanced Robotic Systems*, vol. 13, no. 1, p. 32, 2016.

[19] L. Leottau, C. Celemin, and J. Ruiz-del Solar, "Ball dribbling for humanoid biped robots: A reinforcement learning and fuzzy control approach," in *RoboCup 2014: Robot World Cup XVIII*, ser. Lecture Notes in Computer Science, vol. 8992. Springer, 2015, pp. 549–561.

[20] I. J. da Silva, D. H. Perico, T. P. D. Homem, and R. A. C. Bianchi, "Deep reinforcement learning for a humanoid robot soccer player," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 3, p. 69, 2021.

[21] M. Abreu, L. P. Reis, and N. Lau, "Designing a skilled soccer team for robocup: Exploring skill-set-primitives through reinforcement learning," *Neural Computing and Applications*, 2025.

[22] Y. Ji, Z. Li, Y. Sun, X. B. Peng, S. Levine, G. Berseth, and K. Sreenath, "Hierarchical reinforcement learning for precise soccer shooting skills using a quadrupedal robot," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 1479–1486.

[23] 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," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 2715–2722.

[24] Y. Ji, G. B. Margolis, and P. Agrawal, "Dribblebot: Dynamic legged manipulation in the wild," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[25] T. Haarnoja, B. Moran, G. Lever, S. H. Huang, D. Tirumala, J. Humplik, M. Wulfmeier *et al.*, "Learning agile soccer skills for a bipedal robot with deep reinforcement learning," *Science Robotics*, vol. 9, no. 89, p. eadi8022, 2024.

[26] D. Tirumala, M. Wulfmeier, B. Moran, S. Huang, J. Humplik, G. Lever, T. Haarnoja, L. Hasenclever, A. Byravan, N. Batchelor, N. Sreendra, K. Patel, M. Gwira, F. Nori, M. Riedmiller, and N. Heess, "Learning robot soccer from egocentric vision with deep reinforcement learning," in *Proceedings of The 8th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 270, 2025, pp. 165–184.

[27] Z. Wang, J. Zhou, and Q. Wu, "Dribble master: Learning agile humanoid dribbling through legged locomotion," 2025.

[28] J. Kong, X. Liu, Y. Lin, J. Han, S. Schwertfeger, C. Bai, and X. Li, "Learning soccer skills for humanoid robots: A progressive perception-action framework," 2026.

[29] Z. Ye, E. Ruan, Y. Bao, R. Huang, J. Yang, J. Jin, Y. Huo, P. Wang, Y. Han, H. Huang *et al.*, "Skillx: Unified multi-skill policy learning for humanoid soccer," *arXiv preprint arXiv:2609.06718*, 2026.

[30] 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," 2025.

[31] L. Sentis and O. Khatib, "A whole-body control framework for humanoids operating in human environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 2641–2648.

[32] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti, "Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid," *Autonomous Robots*, vol. 40, no. 3, pp. 473–491, 2016.

[33] T. He, W. Xiao, T. Lin, Z. Luo, Z. Xu, Z. Jiang, C. Liu, G. Shi, X. Wang, L. Fan, and Y. Zhu, "Hover: Versatile neural whole-body controller for humanoid robots," *arXiv preprint arXiv:2410.21229*, 2024.

[34] Y. Li, Y. Zhang, W. Xiao, C. Pan, H. Weng, G. He, T. He, and G. Shi, "Hold my beer: Learning gentle humanoid locomotion and end-effector stabilization control," 2025.

[35] Q. Ben, F. Jia, J. Zeng, J. Dong, D. Lin, and J. Pang, "Homie: Humanoid loco-manipulation with isomorphic exoskeleton cockpit," 2025.

[36] Y. Xue, W. Dong, M. Liu, W. Zhang, and J. Pang, "A unified and general humanoid whole-body controller for versatile locomotion," 2025.

[37] W. Sun, L. Feng, B. Cao, Y. Liu, Y. Jin, and Z. Xie, "Ulc: A unified and fine-grained controller for humanoid loco-manipulation," 2025.

[38] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, no. 1–2, pp. 99–134, 1998.

[39] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017.

[40] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 1–14, 2018.
[41] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, "Amp: Adversarial motion priors for stylized physics-based character control," *ACM Transactions on Graphics*, vol. 40, no. 4, pp. 1–20, 2021.
[42] A. Escontrela, X. B. Peng, W. Yu, T. Zhang, A. Iscen, K. Goldberg, and P. Abbeel, "Adversarial motion priors make good substitutes for complex reward functions," 2022.
[43] 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," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 23–30.
[44] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1–8.
[45] J. Long, Z. Wang, Q. Li, L. Cao, J. Gao, and J. Pang, "Hybrid internal model: Learning agile legged locomotion with simulated robot response," in *The Twelfth International Conference on Learning Representations*, 2024.

APPENDIX

For reproducibility, policies are trained in Isaac Lab (built on Isaac Sim 5.0.0) on eight NVIDIA RTX 4090D GPUs for 20,000 PPO iterations before ONNX export. The 25-dimensional action vector is preserved across training, export, and deployment; hardware rates, sensor inputs, and excluded runtime modules are specified in Sec. IV-C.

Table III defines the metrics used by the simulation, ablation, and real-world result tables. The main paper gives the task-level success definitions; this appendix records the protocol details that expand those definitions for each evaluation setting.

TABLE III: Metric definitions used throughout the evaluation.

Metric	Definition
Success rate (SR)	The main paper defines task-level success: shooting requires the ball to cross the goal line in one kick, while dribbling requires the robot to remain upright and both the robot and ball to finish within 1 m of the target. This appendix expands that definition for evaluation protocols: simulated dribbling additionally requires visiting all waypoints in order (Sec. IV-A), whereas the real-world protocol applies the same endpoint and uprightness criteria to each trial.
Valid contact ratio (r_{contact})	Fraction of shooting trials with a valid foot–ball contact and goal-directed ball exit.
Angular error (E_{ang})	Heading error between the robot body and the robot-to-goal direction.
Velocity error (E_{vel})	Mean error between commanded and actual ball velocity during dribbling.
Ball-in-band ratio (R_{bib})	Fraction of engaged dribbling time in which the ball stays inside the controllable band.
Turn bias (B_{turn})	$\lvert\bar{\theta}_{\text{achieved}} - \theta_{\text{cmd}}\rvert/\theta_{\text{cmd}}$ over scored rounds under the Dribble Master turning protocol; the statistic behind its reported turning error.
Reach rate (RR_τ)	Fraction of rounds whose minimum ball–goal distance falls below tolerance τ under the Dribble Master reaching protocol.

Each main-paper result table follows either Table III or the task-specific protocol described in this appendix. When a metric is computed differently for simulation and real-world deployment, the difference is reported with the corresponding result table.

A. Directional Shooting Simulation Protocol

The shooting simulation results in Table I and Fig. 6 are evaluated on two tasks: penalty kick and free kick. In the free-kick task, the robot starts from a fixed position 5 m in front of the goal center and faces the goal, while the ball is randomly placed within $x \in [0, 4]\,\text{m}$ and $y \in [-3, +3]\,\text{m}$. In the penalty-kick task, the ball is fixed 1.5 m in front of the goal center, while the robot is initialized behind the ball within an angular range of $[-100°, 100°]$ and a distance range of 2–3.5 m, facing the ball. The difficulty level diagram of the shooting task is shown as Figure 8.

To evaluate spatial generalization, we use a stratified random initialization scheme. The penalty-kick sector is divided into 2×15 regions, and the free-kick ball-placement area is discretized into $1\,\text{m} \times 1\,\text{m}$ grid cells. We sample the same number of trials from each region to reduce sampling bias and ensure uniform coverage.

We report SR, E_{ang}, and r_{contact} following the definitions in Table III. The angular error is computed as

$$E_{ang} = \arccos\left(\frac{\mathbf{v}_1^\top \mathbf{v}_2}{\|\mathbf{v}_1\|\|\mathbf{v}_2\|}\right) \tag{7}$$

where $\mathbf{v}_1$ is the robot body-heading vector and $\mathbf{v}_2$ points from the robot base to the goal center. All shooting simulation evaluations use the same policy checkpoint, IsaacLab environment configuration, 5 m depth range, and depth preprocessing pipeline described in Sec. III-E.

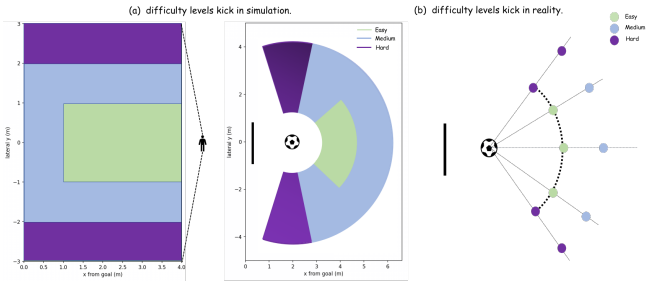

Fig. 8. Multi-position Shooting Simulation and Real Machine Difficulty Level Classification

B. Directional Dribbling Simulation Protocol

The dribbling simulation results in Table I are evaluated by replaying a fixed policy checkpoint in the same IsaacLab environment and control configuration used for training. The benchmark uses a parametrized repeated-S slalom fixture (Table IV) over $N \in 3, 5, 7$ trajectory turns and $\alpha \in 60, 90, 120°$ turning angles, resulting in nine evaluation tasks and 900 trials in total. The ball is initialized 0.6 m in front of the robot, while the robot pose and AMP reference-state reset vary across rounds.

Following Table III, we report SR, E_{vel}, and R_{bib} for dribbling. All metrics are computed over the post-engagement window, i.e., after the robot first takes over the ball, so the initial approach phase is excluded. Specifically, E_{vel} is the mean L2 error between the commanded ball velocity and the measured ball velocity over the engaged steps, where

TABLE IV: Directional dribbling simulation fixture.

Field	Definition	Value
Task grid	Repeated-S slalom with trajectory turns N and turn angle α	$N \in 3, 5, 7$, $\alpha \in 60, 90, 120^\circ$
Waypoints	$N{+}2$ turning waypoints plus $N{+}1$ midline gate points	$2N{+}3$ total
Geometry	Fixed arm length for waypoint generation	$L = 4.0\,\mathrm{m}$
Command	Waypoint attraction and pole repulsion, snapped to 12 direction bins	$k_{\mathrm{att}} = 1.0$, $k_{\mathrm{rep}} = 0.6$, $d_0 = 1.2\,\mathrm{m}$
Ball spawn	Fixed point directly in front of the robot	0.6 m
Engagement	Metrics start after first ball contact	threshold 0.15 m/s; timeout 5 s
Waypoint arrival	Radius for advancing to the next waypoint	0.75 m
Segment timer	Time budget refreshed after each waypoint arrival	5 s
Failure cases	Fall, timeout, or ball leaving the controllable band	—

the commanded velocity points along the current dribbling command with a peak speed of $1.0\,\mathrm{m/s}$. R_{bib} is accumulated over the same engaged window and is undefined for rounds that never enter the controllable band.

The in-band predicate follows the deployed termination settings: the robot–ball distance must lie within 0.10 m of the 0.42 m target, the ball speed must lie in $[0.20, 1.50]$ m/s, and the ball velocity must not point against the command direction. A round fails if the ball remains out of band for more than 3.5 s or exceeds the 0.80 m far-field scope. For fair aggregation, every policy is evaluated with the same checkpoint selection rule, fixed ball spawn, slalom grid, band thresholds, and reset distribution; only the policy weights differ across ablations.

a) Trajectory visualization.: Figure 9 visualizes the repeated-S slalom trajectories. Each panel corresponds to a task configuration (N, α) and shows engagement-anchored paths in a local forward-left frame, the mean successful trajectory, its success band, ball-speed coloring, the start point, and the target region. The figure uses the same checkpoint and evaluation protocol as the metric tables.

C. Dribble Master Protocol Reproduction

We reproduce the dribbling evaluation of Dribble Master [27] on DAVIS with the checkpoint and simulator of Table I. Table V aligns the two setups and Tables VI and VII report the full results; the main-text figures are the $d{=}3$ m reaching and $d{=}5$ m obstacle cells at the 1 m tolerance. Turning is reported here only. The achieved turn is the angle between the ball's mean travel directions over 0.6 s before the command switch and over a 1.5 s window starting 2.0 s after it; its bias of the mean, the statistic behind their published percentages, averages 7.1% over the eight conditions against their 2.2–3.2%. The gap is consistent with the command interface—they track a ball-velocity vector, whereas DAVIS commands direction only and leaves lateral drift inside the controllable band uncorrected—rather than with perception, and the off-grid 45° conditions track no worse than the neighbouring on-grid $30^\circ/60^\circ$ controls.

D. Ablation Experiment Protocol

For all variant in ablation experiments, We follow the identical test settings, evaluation protocols and metrics mentioned in Appendix A and B. Meanwhile, we adopt identical learning hyperparameters, training steps and parallel environments as those used in simulation experiments to guarantee consistent experimental variables. For safety concerns, no ablation studies are conducted on real-world robots.

TABLE V: Dribble Master's published dribbling protocol and our reproduction of it on DAVIS.

Field	Dribble Master [27]	DAVIS reproduction
Robot / policy input	Booster T1; detector ball position + in-view flag	Noetix E1; depth image
Command	global ball-velocity vector, issued by a human operator	direction only, scripted (potential field for reaching, fixed heading for turning)
Reaching / obstacle	unreported distance; success = final error < 1 m	goal at $d \in \{3, 5, 8\}$ m; success = minimum ball–goal distance $< \tau$, $\tau \in \{0.5, 1.0, 1.5\}$ m; round ends at 0.5 m or after $\max(20, 6d)$ s
Obstacle	physical; not perceived by the policy	virtual pole at the path midpoint; not perceived by the policy
Turning	$\pm 45^\circ/90^\circ$ on entering a 0.4 m circle at $(1.5, 0)$ m	$\pm 30/45/60/90^\circ$ on 1.1 m of forward progress (where a straight path enters that circle)
Trials	5 rollouts per turn condition (sim); 15 per task (real)	50 rounds per condition (sim), Wilson 95% CI

Ablation on curriculum. The ramp-up of head and body rewards groups is removed. From step 0 onward, body and head reward scale are fixed to 1. No staged scaling is applied to ball-chasing and gaze-related terms. All task rewards and ball-chasing components take full effect from the beginning of training, eliminating the staged curriculum of "movement first, head activation later".

Ablation on auxiliary head. In the full architecture, depth maps are encoded into latent features via a CNN network. Three prediction heads (goal, ball, visibility) then generate geometric and visibility features, which are concatenated to the actor input. An auxiliary loss is constructed using privileged ground truth to update the encoder and prediction heads. In this ablation, we retain the network structure of the prediction heads and the dimension of the auxiliary input to the actor, but mask all auxiliary features fed into the actor to 0, disable the auxiliary loss, and cut off its backpropagation. This renders the prediction head fully non-functional during both training and inference without changing the overall model structure.

Ablation active vision. The reward structure for ball chasing remains unchanged, but the control objective is switched from "where to orient the head" to "where to rotate the body". The two original terms relying on head joint angles are replaced with body yaw/pitch to align with the ball and target line. Accordingly, the curriculum learning is adjusted

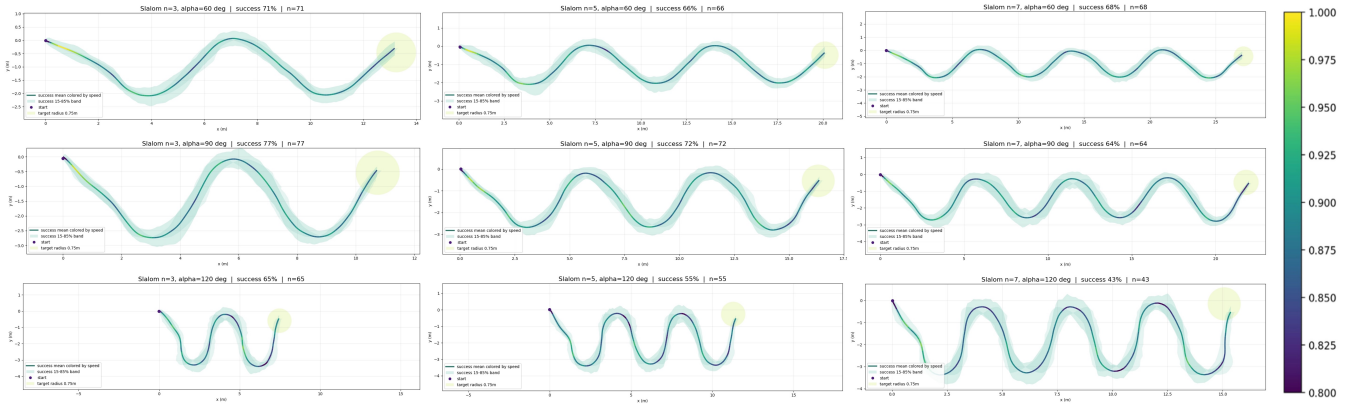

Fig. 9. Trajectory visualization for the repeated-S slalom dribbling benchmark. Each subplot corresponds to one task configuration (N, α), where N is the number of turns and α is the turning angle. The colored curve shows the mean successful ball trajectory with speed-based coloring, and the shaded region indicates the 15–85% success band. The start point and target region are marked for reference. The title of each subplot reports the success rate and the number of successful rounds.

TABLE VI: Target-reaching and single-obstacle results under the Dribble Master protocol (DAVIS simulation, n=50 rounds per row, Wilson 95% CI). Success is reported at three tolerances, together with the time from first ball contact to the 1 m tolerance (mean ± SD over the rounds that got there, count in parentheses); a round ends when the ball first comes within 0.5 m. Dribble Master's reaching and obstacle values come from real-robot trials with n=15 at an unreported distance.

Task	d (m)	$\mathrm{RR}_{0.5}$	$\mathrm{RR}_{1.0}$	$\mathrm{RR}_{1.5}$	time (s)
Target reaching	3	0.88 [0.76, 0.94]	0.88 [0.76, 0.94]	0.90 [0.79, 0.96]	3.0 ± 0.6 (44)
	5	0.82 [0.69, 0.90]	0.84 [0.71, 0.92]	0.84 [0.71, 0.92]	5.7 ± 0.5 (42)
	8	0.78 [0.65, 0.87]	0.80 [0.67, 0.89]	0.80 [0.67, 0.89]	9.4 ± 1.1 (40)
Single obstacle	3	0.92 [0.81, 0.97]	0.92 [0.81, 0.97]	0.92 [0.81, 0.97]	4.8 ± 1.1 (46)
	5	0.94 [0.84, 0.98]	0.96 [0.87, 0.99]	0.96 [0.87, 0.99]	6.4 ± 1.1 (48)
	8	0.94 [0.84, 0.98]	0.94 [0.84, 0.98]	0.94 [0.84, 0.98]	9.9 ± 1.1 (47)
Dribble Master (real, n=15)	–	–	0.87 [0.62, 0.96] (reach) / 0.93 [0.70, 0.99] (obstacle)	–	22.0 / 31.4

TABLE VII: Turning results under the Dribble Master protocol (DAVIS simulation, n=50 rounds per condition). Achieved heading change is reported as mean ± SD over scored rounds, together with the bias of the mean. Dribble Master's turning values come from MuJoCo with 5 rollouts per condition.

Turn command	DAVIS				Dribble Master	
	scored	$\bar{\theta} \pm \sigma$ (°)	bias (%)	per-round (%)	$\bar{\theta}$ (°)	bias (%)
30° L	40	$+28.6 \pm 20.5$	4.8	52.9	–	–
30° R	39	-25.5 ± 15.3	15.1	41.5	–	–
45° L	44	$+39.3 \pm 14.9$	12.6	27.6	+43.58	3.16
45° R	44	-44.8 ± 18.2	0.5	32.1	-46.44	3.20
60° L	43	$+60.4 \pm 19.5$	0.6	26.3	–	–
60° R	41	-53.7 ± 17.2	10.6	23.3	–	–
90° L	42	$+84.0 \pm 25.4$	6.7	18.8	+88.06	2.16
90° R	43	-84.9 ± 22.6	5.6	19.7	-87.98	2.24
Mean bias			7.1			2.7

from head reward scale to body reward scale, allowing the constraint to be gradually activated together with the body-following reward, instead of using an independent head-oriented curriculum. Figure 10 visualizes this active-vision ablation.

E. Real World Experiment Protocol

Figures 11 and 12 show real-world shooting and dribbling examples, respectively.

All real-world experiments follow the settings detailed in Appendix , use safety protection on a grass field, and report the task success rate defined in Table III.

Real World Shooting Experiment. For the real-world shooting experiments showed in Table I, we adopt the penalty-kick setting. The ball is placed 1.5 m in front of the

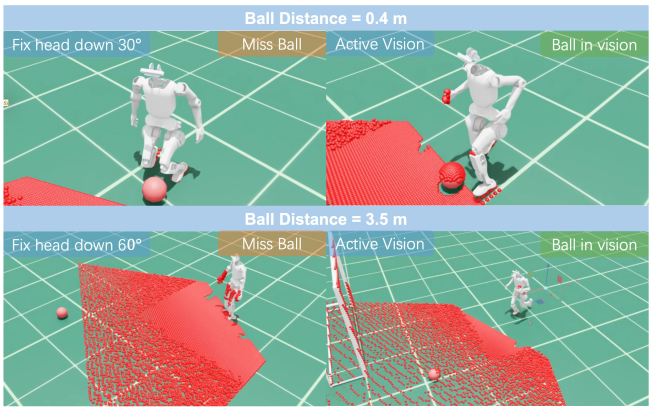

Fig. 10. Ablation on active vision.

TABLE VIII: Target point configurations and difficulty levels used in the real-world shooting evaluation.

Difficulty	Distance (m)	Target Angles
Easy	2.5	$-30°$, $-5°$, $+5°$, $+30°$
Medium	3.0	$-30°$, $-5°$, $+5°$, $+30°$
Hard	2.5–3.0	$-45°$, $+45°$

goal, and the robot performs penalty kicks from different initial positions. The goal dimensions match those in simulation: half-width $1.20\,\mathrm{m}$, height $1.80\,\mathrm{m}$, and depth $1.00\,\mathrm{m}$.

Fig. 11. Multi-directional shooting real-world example

To ensure experimental consistency, we evaluate the policy from 12 predefined positions within the test area. The Easy, Medium, and Hard settings were categorized according to both the target distance and the required heading change, as shown in Table VIII. In all experiments, the robot was initialized facing the goal. A successful trial required the robot to score by kicking the ball into the goal with a single kick attempt. Score is defined as the ball completely crossing the goal line.

Real World Dribbling Experiment. For the real-world dribbling experiments showed in Table I, we design three experimental settings with increasing difficulty. The Easy difficulty level corresponds to the **Reaching Target Position** task. In this task, the robot was required to dribble the ball to a designated target location. The target was randomly generated for each trial at a distance of 4–5 m from the robot and within an angular range of $\pm 30°$ relative to its initial heading. The Medium difficulty level corresponds to the **Obstacle Traversal** task. In this experiment, the robot was required to dribble the ball to a designated target location while avoiding a predefined obstacle. The obstacle was positioned 3–4 m away from the robot, and the target point was typically located approximately 3 m beyond the obstacle. Depending on the obstacle configuration, successful traversal required the robot to execute a heading change of approximately $60°$–$120°$ during obstacle avoidance. The hard difficulty level corresponds to the **Slalom Dribbling** task. In this experiment, the robot was required to complete a continuous dribbling task by navigating through a sequence of target locations. Consecutive target points were separated by 2–3 m, and transitioning between targets required a heading change of approximately $60°$–$120°$, consistent with the previous experimental setting. For each trial, the robot was required to visit 2–4 target points sequentially while maintaining control of the ball. For all experiments described above, the ball spawn distance was fixed at 0.6 m.

For each of the three experimental settings, a trial was counted as successful when the robot remained upright throughout execution and both the robot and the ball finished within a 1 m radius of the target point. The real-world experiments use the fixed deployment checkpoint. To ensure consistency, all trials were performed by the same operator using a joystick controller. This concrete end-state test supplements the task-level definition in the main-paper results table.

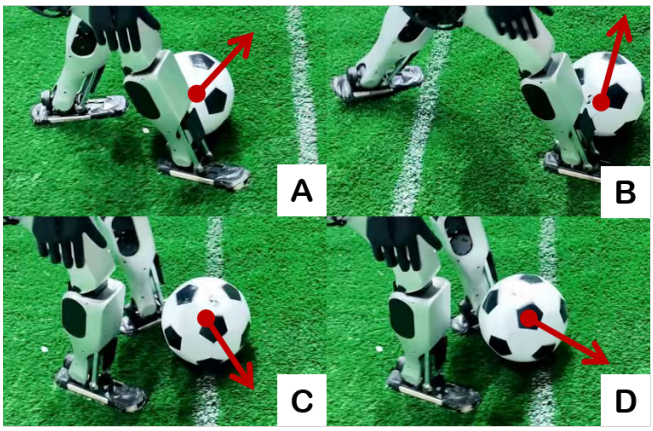

Fig. 12. Foot snapshot in dribbling experiment.

This section provides implementation details for the deployable observation interface, policy architecture, PPO / AMP-HIM optimization, and the auxiliary geometry and visibility losses used by DAVIS.

F. Observation Space and Policy Architecture

The actor input consists of a single aligned depth frame, a 5-step proprioceptive history, and an optional low-dimensional task command. The shooting policy uses an 81-D proprioceptive vector, while the dribbling policy uses an 84-D proprioceptive vector. The proprioceptive vector includes base angular velocity, projected gravity, relative joint positions, relative joint velocities, and the previous action. The depth image is encoded by a `LightDepthEncoder` into a 32-D latent feature. Both the actor and critic MLPs use hidden layers $[512, 256, 128]$ with ELU activations and empirical observation normalization.

During training, the critic additionally receives privileged quantities, including base linear velocity, foot contact states, terrain height scan, ball and goal geometry, auxiliary labels, and curriculum variables. These signals are used only for training-time value learning, reward computation, or auxiliary supervision, and are not actor inputs at deployment. The HIM estimator uses the proprioceptive history to produce a 3-D latent estimate of hidden robot state, which is concatenated with the actor features.

The depth latent is supervised by auxiliary geometry and visibility heads. The geometry heads predict local 3-D positions of the ball and goal, and the visibility head predicts whether each object is visible in the current depth frame. Ground-truth geometry and visibility are mixed with their predictions under the scheduled transition defined below.

Table IX summarizes the observation and network configuration, and Table X lists the optimization settings.

TABLE IX: Observation and network configuration.

Item	Configuration
Actor sensory input	Single-frame depth image + 5-step proprioceptive history
Optional command	Low-dimensional task command if enabled
Shooting proprioception	81-D
Dribbling proprioception	84-D
Depth encoder	`LightDepthEncoder`
Depth latent	32-D
HIM estimate output	3-D
Actor MLP	$[512, 256, 128]$, ELU
Critic MLP	$[512, 256, 128]$, ELU
Actor normalization	Enabled
Critic normalization	Enabled
Auxiliary geometry heads	Goal 3-D, ball 3-D
Auxiliary visibility head	2-D: goal visible, ball visible

TABLE X: PPO and AMP-HIM optimization settings.

Item	Value
Rollout length	24 steps per environment
PPO epochs	5
Mini-batches	4
Learning rate	1×10^{-3}, adaptive schedule
Discount factor	$\gamma = 0.99$
GAE parameter	$\lambda = 0.95$
PPO clip	0.2
Entropy coefficient	0.01
Value loss coefficient	1.0
Max gradient norm	1.0
Discriminator LR	5×10^{-6}
Discriminator loss	Wasserstein loss
Discriminator hidden layers	$[1024, 512]$
AMP replay buffer	200000
AMP reward coefficient	0.8
AMP reward interpolation	0.8
Auxiliary geometry loss	goal: 1.0, ball: 1.0
Visibility loss	goal: 0.2, ball: 0.2

The AMP discriminator uses a 136-D single-frame state composed of joint position, key-body positions, base linear velocity, base angular velocity, joint velocity, and foot-contact masks. We mask the head-related dimensions in the discriminator input because head yaw and pitch are optimized for active visual tracking rather than locomotion-style imitation.

G. Auxiliary Geometry and Visibility Losses

Let z_t^d denote the 32-D depth latent produced by the `LightDepthEncoder`. For each task object i (the ball, and the goal in the shooting task), a geometry head predicts the object geometry $\hat{\mathbf{y}}_t^i$, while a separate visibility head predicts a visibility logit $\hat{v}_t^i$ from the shared latent,

$$\begin{gathered}\hat{\mathbf{y}}_t^i = h_{\text{geo}}^i(z_t^d), \qquad \hat{v}_t^i = h_{\text{vis}}^i(z_t^d), \\ p_t^i = \sigma(\hat{v}_t^i).\end{gathered} \tag{8}$$

The geometry is expressed in the head-yaw-aligned frame defined in Sec. III-C, with x along the head-yaw forward direction, y to the left, and z upward. Both auxiliary heads use only the current depth latent, while head states are provided separately to the actor through proprioception. Simulator labels provide a target geometry $\mathbf{y}_t^{i*}$ and a binary visibility label v_t^{i*}. Let $\mathcal{P}_i$ denote the predefined structural points used for visibility evaluation. The visibility label is defined as

$$v_t^{i*} = \mathbb{I}\left[\exists\, \mathbf{s} \in \mathcal{P}_i : \text{valid}\left(\mathbf{T}_{C \leftarrow R,t}\mathbf{s}\right)\right], \tag{9}$$

where $\text{valid}(\cdot)$ indicates that the transformed point lies within the valid camera field of view and depth range. For a single-point object, $\mathcal{P}_i$ contains only its reference point; for a multi-point object, the object is considered visible if any predefined structural point is valid. These structural points are used only for visibility evaluation, while the geometry target remains the object center. Geometry is supervised only on visible frames, while visibility is trained by binary cross-entropy on the logits:

$$\begin{aligned}\mathcal{L}_{\text{geo}} &= \sum_i v_t^{i*}\, \ell_{\text{geo}}\left(\hat{\mathbf{y}}_t^i, \mathbf{y}_t^{i*}\right), \\ \mathcal{L}_{\text{vis}} &= \sum_i \lambda_v\, \text{BCEWithLogits}\left(\hat{v}_t^i, v_t^{i*}\right), \\ \mathcal{L}_{\text{aux}} &= \mathcal{L}_{\text{geo}} + \mathcal{L}_{\text{vis}}.\end{aligned} \tag{10}$$

The visibility BCE is the direct supervision for the visibility head; the detached actor-side gate defined below prevents the policy gradient from propagating through the visibility confidence.

During training, the actor consumes geometry through a scheduled ground-truth-to-prediction mixture, measured in policy updates k (in code, world-size-normalized environment steps with $k \approx \text{step}/24$). Let β_k be the prediction weight and $\alpha_k = 1 - \beta_k$ the ground-truth weight. Rather than a linear decay, β_k follows a monotone, concave $\sqrt{\cdot}$ ramp from a warm-up update k_0 to an end update k_1,

$$\begin{gathered}\beta_k = \max\Big(\beta_{k-1}, \sqrt{\text{clip}\big((k - k_0)/(k_1 - k_0),\, 0,\, 1\big)}\Big), \\ \alpha_k = 1 - \beta_k.\end{gathered} \tag{11}$$

That is, the actor uses pure ground-truth guidance for $k \le k_0$ and gradually transitions to pure prediction at k_1. The ramp is ratcheted (non-decreasing) and is frozen whenever the auxiliary-loss EMA exceeds a threshold before a grace update, so the actor is shifted onto its own predictions only as they become reliable. Geometry and visibility are mixed under the same training schedule:

$$\begin{aligned}\mathbf{y}_t^{e,i} &= (1 - \beta_k)\mathbf{y}_t^{i*} + \beta_k \hat{\mathbf{y}}_t^i, \\ p_t^{e,i} &= (1 - \beta_k) v_t^{i*} + \beta_k p_t^i.\end{aligned} \tag{12}$$

The mixed visibility confidence is converted into a soft, detached actor-side gate,

$$c_t^i = \text{clip}\left(\text{stopgrad}\left(p_t^{e,i}\right), 0.05, 1\right), \tag{13}$$

and the auxiliary actor feature is

$$\mathbf{a}_t^{\text{aux}} = \left[c_t^i \mathbf{y}_t^{e,i},\ c_t^i \right]_i . \tag{14}$$

The stop-gradient operation prevents the policy objective from modifying the visibility head through the actor-side gating path. At deployment, the GT-to-prediction mixture is disabled ($\beta_k = 1$, equivalently $\alpha_k = 0$), and all auxiliary actor features are predicted from the depth branch. Figure 13 shows the training curve of GT-annealing.

Fig. 13. Loss curve of GT-annealing and w/o GT-annealing

H. Goal-Directed Shooting

Shooting is trained as a goal-directed contact task. Task-specific training uses 1024 parallel environments, a 20 s episode length, a 0.005 s physics step, and the same control decimation used by the shared training and deployment configuration in Appendix .

Let d_t denote the robot–ball distance, and define the ball-to-goal direction as

$$\mathbf{u}_t^{bg} = \frac{\mathbf{g}_t - \mathbf{b}_t}{\|\mathbf{g}_t - \mathbf{b}_t\|_2}. \tag{15}$$

Near-ball alignment and contact terms are modulated by a smooth distance gate,

$$G(d_t; d_0, \tau) = \sigma\left(\frac{d_0 - d_t}{\tau}\right), \tag{16}$$

which gives larger weights to these terms as the robot approaches the ball. The shooting reward is organized into approach, active vision, ball–goal alignment, contact, terminal success, and safety regularization terms, as summarized in Table XI.

I. Directional Dribbling

For reproducibility, the checkpoint uses the 12-way command ring and controllable-band definitions in Sec. III-D. The dribbling episode lasts 60 s with a 0.005 s physics step and the shared control decimation; training tightens the band half-width from 0.15 m to 0.10 m and the out-of-band dwell from 6.0 s to 3.5 s.

This section records the remaining sim-to-real details beyond the deployment interface and policy settings already given in Appendices and E: depth sensing and robot–ball–ground physical variation.

The simulated camera is mounted on the active head and renders `distance_to_image_plane`. Both simulation and real deployment use a single 168×80 aligned depth frame, clipped to 0.3–5.0 m, normalized as $2D/5.0 - 1$, and zero-filled at invalid pixels. In simulation, this tensor is obtained by rendering a 168×94 pinhole depth image with calibrated ZED2i intrinsics and cropping it to the actor input size. The real controller applies the same cropping, range filtering, and normalization before actor inference. Temporal information is provided by proprioceptive history rather than by stacking depth frames.

During training, we randomize visual observations and physical interaction parameters. Depth perturbations are applied to the rendered tensor, while robot, ball, ground, and contact variations are applied through IsaacLab event terms at startup, reset, or fixed intervals. Table XIII summarizes the perturbation groups used for sim-to-real transfer.

This section contains additional empirical results that complement the main paper.

J. Powerful Shot.

We additionally study a close-range powerful-shot skill as another task-specific instantiation. This skill uses the same depth-only actor interface and auxiliary ball-geometry prediction as the base skills; only the reset distribution, reward terms, and motion-prior schedule are changed.

We use the same near-ball reward gating as in the shooting task. The reward follows a two-stage curriculum: the first stage rewards near-ball leg effort through a distance-weighted foot-speed term, and the second stage adds ball-outcome terms, including horizontal ball speed and displacement from the reset position. The corresponding reward scales are ramped in sequentially during training.

An AMP kick-motion prior is activated once a sufficient fraction of parallel environments reaches the near-ball zone, regularizing the explosive leg swing while keeping the actor input unchanged. This experiment tests whether the same depth-only framework can support high-speed striking from a close pre-kick stance without adding a separate runtime ball-localization or shooting controller. Figure 14 shows the real-world result.

Fig. 14. Power shoot real-world example

TABLE XI: Shooting reward summary.

Reward Terms	Components	Weight
Approach and chase	Approach ball reward Adaptive velocity reward	3.0 2.0
Active vision	Head track ball reward Keep in horizon reward Body follow head reward	3.0 -0.5 2.0
Ball–goal alignment	Robot behind ball reward Lateral kick line penalty Head yaw align reward	1.0 1.0 1.5
Ball reward to goal	Ankle toward ball reward Ball velocity reward Ball velocity direction reward	2.0 4.0 1.5
Sparse goal scored	One-shot score reward	10.0
Safety and style regularization	Unstable posture; Foot sliding; Excessive contact force; Joint limits; Joint acceleration; Action-rate changes; Upper body deviations;	task-dependent penalties
AMP motion prior	AMP style reward.	coefficient 0.8

TABLE XII: Dribbling reward and regularization summary.

Group	Terms	Weight / setting
Commanded progress & heading	Ball command reward Body command reward	5.0 0.4
Ball control	Dense re-approach reward Out-of-band penalty	2.0 -1.0
Active vision	Ball in view reward Body follow head reward	1.0 0.6
Safety and style regularization	Unstable posture; Foot sliding; Excessive contact force; Joint limits; Joint acceleration; Action-rate changes; Upper body deviations;	task-dependent penalties
Termination	Falls penalty; Bad-orientation penalty; Lose ball control penalty;	-1.0
AMP motion prior	Head-masked Wasserstein AMP style reward	coefficient 0.8, interpolation 0.8

TABLE XIII: Representative training randomization ranges for sim-to-real transfer.

Perturbation	Range / value
Valid depth range	0.3–5.0 m
Distance-proportional depth noise	0.01
Gaussian depth-noise coefficient	0.005
Salt-and-pepper probability	0.01
Stripe dropout probability	0.005
Edge-drag probability	0.01
Rectangular mask probability	0.1
Rectangular mask size	10–25%
Ball-region occlusion probability	0.2
Ball-region occlusion radius	2–10 px
Partial occlusion ratio	45–85%
Border near-field noise probability	0.15
Camera intrinsic perturbation	± 0.3–± 0.5 px
Camera translation perturbation	0.02–0.04 m
Camera rotation perturbation	0.025–0.060 rad
Joint-position offset	± 0.02 rad
Base COM perturbation	± 0.05 m
Base mass perturbation	± 5 kg
Actuator stiffness / damping scale	0.8–1.2
Ball mass randomization	0.36–0.56 kg
External push interval	3–15 s
External linear velocity perturbation	$x, y \in [-1, 1]$ m/s; $z \in [-0.2, 0.2]$ m/s
External angular velocity perturbation	$[-0.78, 0.78]$ rad/s

K. Dribbling while Avoiding Obstacles.

This skill highlights the extensibility of our depth-predicted auxiliary heads: extending directional dribbling to avoid obstacles requires no architectural change. Following the scalable visibility-aware auxiliary geometry of Sec. III-C, the obstacle simply enters the task object set as one more object, adding a single indexed geometry / visibility head that predicts its relative position and visibility from the shared depth latent—with no privileged obstacle input and no change to the actor interface or control loop.

Avoidance is induced through the reward rather than a separate controller: an artificial potential field deflects the commanded direction $\mathbf{d}_t^{\mathrm{cmd}}$ by a tangential repulsion from the nearest in-path obstacle,

$$\tilde{\mathbf{d}}_t^{\mathrm{cmd}} = \frac{\mathbf{d}_t^{\mathrm{cmd}} + \kappa\, w_t\, a_t\, \mathbf{t}_t}{\left\| \mathbf{d}_t^{\mathrm{cmd}} + \kappa\, w_t\, a_t\, \mathbf{t}_t \right\|}, \tag{17}$$

where $\mathbf{t}_t$ is the unit normal to the command on the obstacle-clearing side ($\mathbf{t}_t \mathbf{u}_t \leq 0$, with $\mathbf{u}_t$ the ball-to-obstacle bearing), $a_t = \max(\mathbf{u}_t \cdot \mathbf{d}_t^{\mathrm{cmd}}, 0)$ activates the repulsion only for obstacles ahead, $w_t \in [0, 1]$ is a proximity weight that vanishes beyond an influence radius, and κ is the steering gain. Every direction-aware dribble reward tracks $\tilde{\mathbf{d}}_t^{\mathrm{cmd}}$ in place of $\mathbf{d}_t^{\mathrm{cmd}}$, so obstacle avoidance emerges from the same velocity-direction tracking objective with no change to the reward structure. We show detailed loss curve of the aux loss in Figure 15.

L. Recovery upon Losing the Ball.

We also test whether the chase policy can recover after the ball temporarily leaves the head-mounted field of view. The recovery behavior is not implemented as a separate search controller. Instead, the same active-vision policy is shaped by a visibility flag and an invisible-ball timer: when

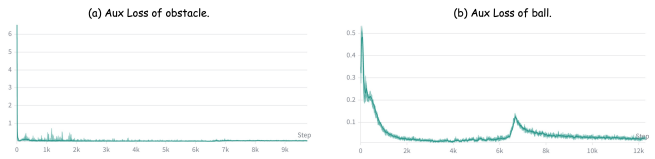

Fig. 15. Auxiliary loss for the ball and obstacles. The prediction heads for both objects converge during training.

the ball is visible, the robot continues pursuit; when it is lost, the policy is penalized for prolonged invisibility and for moving too fast without reliable visual evidence. In the submitted configuration, the search penalty starts after $0.5\,\mathrm{s}$ of continuous invisibility with weight 0.2, and the invisible-speed penalty limits body-frame forward speed around $0.5\,\mathrm{m/s}$ with weight 0.6.

Recovery is driven by the active head and body-following rewards. The head reward tracks the desired yaw/pitch angles toward the ball in the robot body frame, while the body-follow-head reward encourages the torso to realign with the recovered viewing direction. These rewards are introduced gradually: the head reward starts at 12k steps and ramps over 36k steps, and the body-following reward starts at 48k steps and ramps over 48k steps. This produces a closed-loop search-and-reacquire behavior while preserving the same depth-only actor interface used by the main soccer skills.