

SkillX: Unified Multi-Skill Policy Learning for Humanoid Soccer

Zhangchen Ye^{1, 2*} Enxuan Ruan^{1*} Yifei Bao^{1*} Runhan Huang²
Jiankun Yang¹ Jiakang Jin¹ Yixiao Huo¹ Pengyuan Wang¹ Yinan Han¹
Huaxing Huang¹ Wenhao Cui¹ Yiming Li^{2†} Xiaoyu Tian^{1†}

¹ Noetix Robotics; ² Tsinghua University
* Equal Contribution † Corresponding Author

Project page: yzc0731.github.io/SkillX

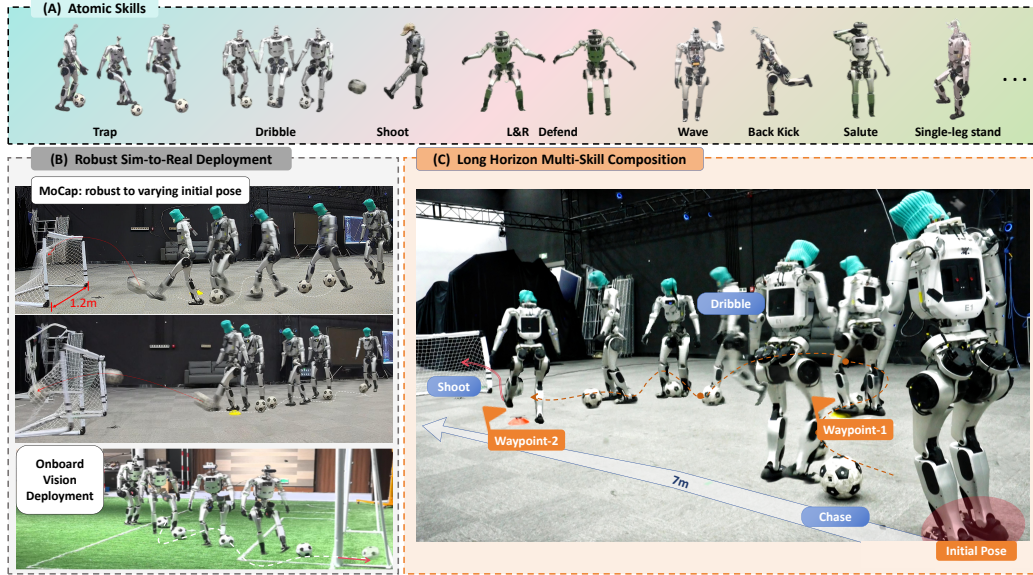


Figure 1: Our framework **SkillX** enables a humanoid robot to execute diverse soccer skills and compose them into long-horizon behaviors. (A) The robot learns *atomic skills* such as trapping, dribbling, shooting, and additional soccer skills. (B) The learned policy achieves robust *sim-to-real deployment* on the Noetix E1 robot using both the motion-capture system and onboard vision. (C) The robot composes atomic skills and performs *long-horizon multi-skill tasks*.

Abstract:

Humanoid soccer is a challenging testbed for dynamic whole-body control, requiring robots to coordinate balance, locomotion, object interaction, and skill switching over long horizons. Existing humanoid sports methods often rely on task-specific multi-stage pipelines, making it difficult to jointly learn and compose multiple object-interactive skills within a single deployable policy. To address this, we present SkillX, a unified reinforcement learning framework that learns and composes multiple atomic soccer skills through a single command-conditioned policy. SkillX integrates three core designs: skill-specific adversarial motion priors, skill-specific critics, and an object-aware temporal encoder, enabling the robot to execute atomic skills and transition among them such as dribbling, trapping, and shooting. Experiments in simulation and on a real Noetix E1 humanoid demonstrate robust multi-skill execution, long-horizon skill composition, and successful sim-to-real deployment.

Keywords: Humanoid Robots, Reinforcement Learning, Multi-Task Learning

1 Introduction

Humanoid soccer [1, 2, 3, 4, 5, 6] is a challenging testbed for dynamic whole-body control because it requires a robot to coordinate balance, locomotion, object interaction, and skill switching. A soccer robot must execute athletic skills such as dribbling, shooting, and trapping, while composing them into long-horizon behaviors in response to a moving ball and changing task objectives, going beyond conventional single-skill motion tracking, in which a policy primarily follows reference motions.

Recent progress in humanoid control has been largely driven by motion tracking, which enables agile locomotion and stylized motion imitation [7, 8, 9, 10, 11]. This paradigm has also been extended to athletic ball interaction tasks such as soccer [12], basketball [13], tennis [14], table tennis [15], and badminton [16, 17]. However, existing methods often rely on task-specific multi-stage pipelines that combine motion imitation with task-oriented refinement, making it difficult to jointly learn and compose multiple object-interactive skills with a single deployable policy.

As a representative instance of multi-skill object-interactive control, unified humanoid soccer further amplifies these difficulties through three coupled challenges. First, different skills exhibit *motion-style heterogeneity*, involving distinct kinematic patterns and contact behaviors. Naive joint training can cause cross-skill interference and unreliable transitions [18, 19, 20]. Second, these skills involve *skill-dependent task objectives*, making shared value estimation difficult across skill modes. Third, soccer interactions are subject to *partial observability*: ball velocity, contact events, and short-term interaction outcomes are noisy or latent and must be inferred from observation histories. Together, these challenges make it difficult for a single policy to preserve skill-specific motion quality while composing skills robustly over long horizons.

We present **SkillX**, a unified reinforcement learning framework for multi-skill humanoid soccer. As illustrated in Fig. 1, the learned policy supports diverse atomic skills, robust sim-to-real deployment, and long-horizon multi-skill composition. First, *skill-specific adversarial motion priors* separate motion regularization across skills to reduce style interference. Second, *skill-specific critics* address skill-dependent value estimation across different task objectives. Third, a transformer-based *object-aware temporal encoder* addresses partial observability by estimating hidden robot and object states from observation histories. Together with adaptive command sampling, a command-duration curriculum, and object domain randomization, these components enable stable command-driven skill composition with a single deployed policy. Although instantiated in humanoid soccer, these designs target general challenges in multi-skill object-interactive control.

We evaluate SkillX on humanoid soccer tasks in simulation and on the real Noetix E1 humanoid robot. In simulation, we test both atomic skills and composite tasks of varying difficulty levels, compare against representative baselines, and conduct ablations to quantify the contribution of each major component. On the hard long-horizon trap-dribble-shoot task, SkillX achieves **81.7%** success, compared with 27.4% for AMP. On hardware, we deploy the learned policy with a motion-capture (MoCap) system and further study onboard-vision deployment using a head-mounted camera. The real robot achieves a **72.5%** average success rate across four soccer tasks, demonstrating reliable object-interaction skill execution and command-driven skill transitions.

Our contributions are summarized as follows:

- We propose **SkillX**, a unified reinforcement learning framework for **multi-skill humanoid soccer**, enabling a single policy to execute multiple atomic skills and compose them into long-horizon tasks.
- We introduce multi-skill training mechanisms, including **skill-specific adversarial motion priors**, **skill-specific critics**, and a **transformer-based object-aware temporal encoder**, to address motion-style heterogeneity, skill-dependent task objectives, and partial observability in multi-skill object-interactive control.
- We validate SkillX in both simulation and real-world deployment, demonstrating robust sim-to-real deployment and **long-horizon multi-skill composition**.

2 Related Work

2.1 Multi-Task Reinforcement Learning in Locomotion

Multi-task reinforcement learning methods for locomotion aim to train a unified policy for diverse motion skills, but often face gradient interference across heterogeneous tasks [18, 21, 20]. Major solution categories include hierarchical policies, implicit skill disentanglement, and mixture-of-experts (MoE) architectures. Hierarchical methods pre-train individual skills and use high-level modules for skill selection and transitions [22, 23, 24, 25], but typically require multi-stage training and do not readily support end-to-end optimization. Implicit disentanglement methods regularize diverse motion styles in a shared latent space [26, 27, 19], but can still suffer from latent entanglement, making it difficult to fully separate skill-specific features. MoE methods use gating mechanisms to mitigate gradient conflicts in end-to-end multi-skill learning [18, 20], but can suffer from expert imbalance or unstable specialization. Inspired by multi-adversarial motion priors [28] and multi-critic reinforcement learning [29], our method decouples style regularization and value estimation across skills during training while maintaining a single unified policy for deployment, enabling command-driven skill execution and smooth transitions.

2.2 Humanoid Athletic Ball Interaction

While multi-skill locomotion studies focus on learning diverse motions, athletic ball interaction introduces an additional challenge of coordinating balance and whole-body motion under fast-changing object dynamics and contact-rich interactions. Recent learning-based methods have made considerable progress in humanoid ball interaction, yet many existing studies primarily target isolated skills, such as flat strikes in table tennis [30], soccer dribbling [31], and shooting [32, 33, 12]. To equip robots with broader athletic capabilities, prior methods often rely on multi-stage training pipelines, including progressive skill refinement [16, 17] and policy distillation [13, 14]. While effective, these designs can require careful training schedules and additional optimization procedures. In contrast, we study humanoid soccer as a multi-skill ball-interaction setting, in which the policy must compose multiple ball-control skills while inferring interaction-relevant object states from partial observations. Our framework addresses these requirements by integrating skill-specialized training signals and an object-aware temporal encoder into a unified policy.

3 Method

We design SkillX as a unified multi-skill reinforcement learning framework, instantiated here for humanoid soccer. The key principle is to keep deployment simple with a single command-conditioned actor, while using skill-specific training signals to reduce interference across heterogeneous skills. As shown in Fig. 2, the framework consists of three core components: **(A)** skill-specific adversarial motion priors for preserving distinct motion styles, **(B)** skill-specific critics for skill-dependent value estimation, and **(C)** an object-aware temporal encoder for estimating hidden robot and object states required for precise interaction. We detail these components in Sec. 3.2, Sec. 3.3, and Sec. 3.4, respectively.

3.1 Problem Formulation

We formulate multi-skill humanoid soccer as an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$ and train a policy π_θ with PPO [34] to maximize the expected discounted return. Since the actor receives only partial proprioceptive and object observations, the policy is conditioned on a history window of length K . The instantaneous observation at time t is $\mathbf{o}_t = [\mathbf{p}_t, \mathbf{c}_t, \mathbf{o}_t^{\text{ext}}]$, where $\mathbf{p}$ denotes proprioception, $\mathbf{c}$ the skill command, and $\mathbf{o}^{\text{ext}}$ external object information. In our soccer task, $\mathbf{o}^{\text{ext}}$ contains the robot-centric relative positions of the ball and goal. The command vector $\mathbf{c}_t \in [0, 1]^{N_{\text{skill}}}$ satisfies $\sum_i c_{t,i} = 1$: it is one-hot during single-skill execution and linearly interpolated during skill switches to produce soft transitions. The policy outputs target joint positions executed by low-level PD controllers. Detailed observation specifications are provided in Appendix C.1.

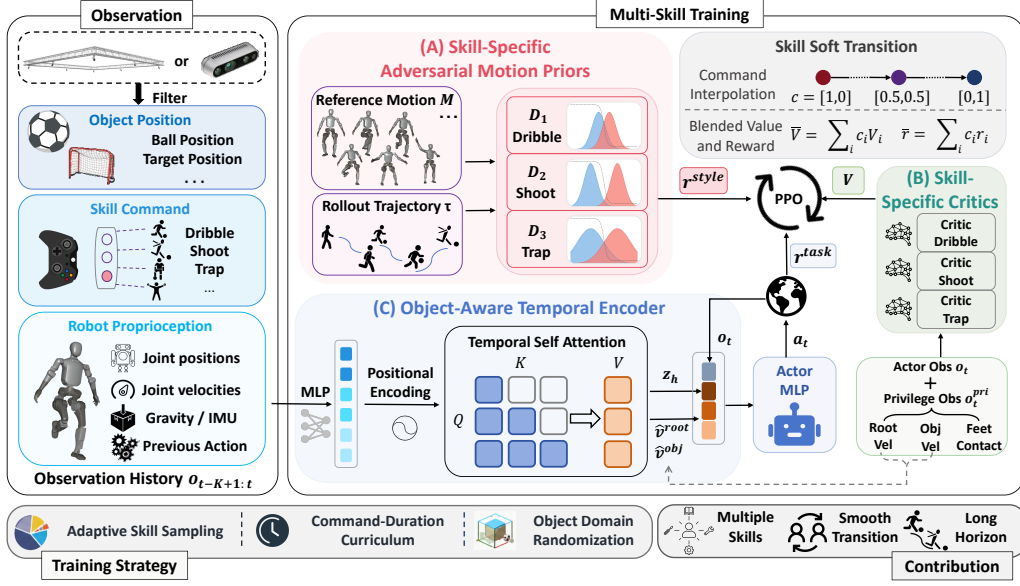


Figure 2: **Overview of SkillX.** The policy receives a history of robot proprioception, object states, and skill commands, and outputs joint-position targets. (A) *Skill-Specific Adversarial Motion Priors* compare policy rollouts with reference motions and provide style rewards for heterogeneous skills. (B) *Skill-Specific Critics* estimate skill-dependent values; during soft skill transitions, the command vector interpolates both rewards and values for PPO updates. (C) *Object-Aware Temporal Encoder* aggregates observation histories with temporal self-attention and provides compact history features for the actor.

3.2 Skill-Specific Adversarial Motion Priors

Different skills in a unified policy can exhibit distinct motion distributions and contact patterns. While AMP [35] uses an adversarial discriminator to provide motion-style rewards, a shared discriminator in a multi-skill setting can produce ambiguous style guidance and weaken skill-specific style separation. We address this issue with a set of skill-specific adversarial motion priors $\{D_i\}_{i=1}^{N_{\text{skill}}}$, where each discriminator is trained for one atomic skill while the actor remains shared across all skills.

Each discriminator D_i distinguishes expert motion segments of skill i from policy-generated segments under the same command mode. Its input is a compact observation consisting of motion-centric features, including joint states, key body positions, root velocities, and foot contact states. Following AMP [35], we train each D_i with a least-squares GAN objective and gradient penalty:

$$\mathcal{L}_{D_i} = \mathbb{E}_{\tau \sim d_i^M} [(D_i(\tau) - 1)^2] + \mathbb{E}_{\tau \sim d_i^\pi} [(D_i(\tau) + 1)^2] + \frac{w^{\text{GP}}}{2} \mathbb{E}_{\tau \sim d_i^M} \|\nabla_{\tau} D_i(\tau)\|_2^2, \quad (1)$$

where τ denotes a multi-step motion segment, d_i^M is the expert motion distribution for skill i , and d_i^π is the corresponding policy-generated distribution.

The style reward produced by discriminator D_i is defined as

$$r_i^{\text{style}}(t) = \max \left[0, 1 - 0.25 (D_i(\tau_t^\pi) - 1)^2 \right], \quad (2)$$

where τ_t^π is the policy motion segment ending at time t . For one-hot command segments, the policy receives the style reward from the active skill discriminator. During a soft transition, we compute a blended style reward according to the interpolated command vector $\mathbf{c}_t$:

$$\bar{r}^{\text{style}}(t) = \sum_{i=1}^{N_{\text{skill}}} c_{t,i} r_i^{\text{style}}(t). \quad (3)$$

We refer to this command-weighted interpolation as *command blending*.

Motion segments spanning command switches are excluded from the discriminator updates, since they contain mixed behaviors and should not be treated as clean samples of any single skill.

3.3 Multi-Critic PPO with Skill-Specific Critics

In standard PPO, a single critic estimates one value function for all training samples. This is insufficient for command-driven multi-skill control, where the same physical state can have different expected returns under different skill objectives. Fitting these skill-dependent returns with one shared critic can introduce value interference and biased advantage estimates. To address this issue, we maintain a set of skill-specific critics $\{V_i\}_{i=1}^{N_{\text{skill}}}$, while keeping a single shared actor.

Given the command vector $\mathbf{c}_t$, we compute a command-conditioned value estimate by selecting or blending the skill-specific critics:

$$\bar{V}(t) = \sum_{i=1}^{N_{\text{skill}}} c_{t,i} V_i(t). \quad (4)$$

For one-hot commands, $\bar{V}(t)$ selects the active skill critic; during soft transitions, it uses the same command blending rule used for the task and style rewards. PPO advantages are then computed using the command-conditioned reward and value estimates, followed by the standard PPO update.

In practice, the critics share a common feature extractor and use separate skill-specific value heads. This parameter-efficient design allows each head to specialize in the value function of one skill, rather than forcing a single critic to represent all skill-dependent returns. As a result, the critics provide more accurate command-conditioned value estimates, leading to better advantage estimation and more stable multi-skill policy optimization.

3.4 Object-Aware Temporal Encoder

Dynamic object-interaction tasks require the policy to react to fast-changing robot-object states, while receiving partial and noisy observations. We therefore introduce an object-aware temporal encoder that aggregates observation histories and produces compact features for precise interaction.

Given the observation history $\mathbf{o}_{t-K+1:t}$, we first embed each observation with an MLP and add sinusoidal positional encoding. The resulting sequence is processed by a causal transformer with temporal self-attention:

$$\mathbf{E}_{t-K+1:t} = \phi_{\text{emb}}(\mathbf{o}_{t-K+1:t}) + \mathbf{PE}, \quad (5)$$

$$\mathbf{H}_{t-K+1:t} = \mathcal{T}_{\phi}(\mathbf{E}_{t-K+1:t}, \mathbf{M}_K), \quad (6)$$

where $\mathbf{PE}$ is the positional encoding and $\mathbf{M}_K$ is a causal attention mask that prevents future information leakage. The history latent $\mathbf{z}_h$ is obtained by applying layer normalization and average pooling over the temporal dimension of $\mathbf{H}_{t-K+1:t}$.

Following the hybrid optimization idea of HIM [36], we train the temporal encoder with two auxiliary objectives. First, we use a self-supervised temporal consistency loss based on Barlow Twins to regularize the history latent under temporally shifted observation windows. Second, we attach a lightweight estimation head to predict hidden states that are important for interaction:

$$\hat{\mathbf{v}} = h_{\text{est}}(\mathbf{z}_h), \quad \hat{\mathbf{v}} = [\hat{\mathbf{v}}^{\text{root}}, \hat{\mathbf{v}}^{\text{obj}}]. \quad (7)$$

Unlike purely proprioceptive internal models, the additional object-velocity target makes the latent representation sensitive to ball motion, which is available as privileged supervision in simulation but must be inferred from noisy position histories during deployment. The final encoder objective is $\mathcal{L}_{\text{enc}} = \mathcal{L}_{\text{est}} + \mathcal{L}_{\text{bt}}$, where $\mathcal{L}_{\text{est}}$ penalizes root- and object-velocity prediction errors and $\mathcal{L}_{\text{bt}}$ denotes the Barlow-Twins loss. The resulting latent $\mathbf{z}_h$ and estimated velocities $\hat{\mathbf{v}}$ are provided to the actor as compact history-based features for precise robot-object interaction under noisy or delayed observations.

Method	Medium				Hard				
	⌘ Dribble	⌘ Shoot	Overall↑	Time↓	# Trap	⌘ Dribble	⌘ Shoot	Overall↑	Time↓
Vanilla PPO	0.0	—	0.0	—	99.8	0.0	—	0.0	—
AMP	89.0	57.3	51.0	12.6	100.0	52.0	52.6	27.4	30.4
Conditional AMP	13.4	83.6	11.2	17.6	100.0	0.0	—	0.0	—
MoE-based AMP	58.0	73.4	42.6	16.6	100.0	5.0	100.0	5.0	32.6
MoE-Encoder AMP	100.0	67.2	67.2	13.6	100.0	95.7	47.6	45.6	34.7
Ours	100.0	88.0	88.0	12.0	100.0	100.0	81.7	81.7	28.4

Table I: **Composite Task Evaluation in Simulation.** Medium: DRIBBLE→SHOOT. Hard: TRAP→DRIBBLE×3 →SHOOT. Atomic skill evaluation is reported in Appendix B.4. Substage success rates are conditional on reaching the corresponding substage; “Overall” denotes full-task success. “—” indicates that no episode reaches the corresponding substage.

3.5 Training Strategy

To further improve stability, sample efficiency, and generalization for multi-skill training, we incorporate several practical training strategies. **Adaptive command sampling** allocates more rollouts to skills with higher recent failure rates. We track each skill’s failure rate using an exponential moving average and update the skill-level sampling probability. The **command duration curriculum** starts with long single-skill segments and progressively shortens command durations to encourage frequent skill switching. **Object domain randomization** perturbs ball mass, friction, and restitution to improve robustness to interaction variations. Details are provided in Appendix C.2.

4 Experiments

4.1 Implementation Details

We train all policies in NVIDIA Isaac Sim and Isaac Lab [37], enabling high-throughput parallel simulation. The policy runs at a control frequency of 50 Hz, while the simulation runs at 200 Hz. Human reference soccer motions are collected via a MoCap system and retargeted to the 25-DoF Noetix E1 humanoid using PHC [38]. Real-world experiments are conducted on the 25-DoF Noetix E1 humanoid robot. We evaluate two perception pipelines to obtain ball-related states. (1) A motion-capture system provides a 6-DoF base pose of the humanoid robot and the 3D positions of the ball and goal. (2) An onboard-vision setup uses a head-mounted ZED2i camera with visual-inertial odometry and YOLOv8-based ball detection [39]. Additional details are provided in Appendix A.

4.2 Simulation Results

Evaluation Protocol. We evaluate policies on both atomic soccer skills and skill-composition tasks. Atomic tasks include isolated shooting, dribbling, and trapping, with results reported in Appendix B.4. In the main text, we focus on two composite tasks with increasing difficulty. The *Medium* task requires DRIBBLE→SHOOT, while the *Hard* task requires TRAP→DRIBBLE×3 →SHOOT. In the composite tasks, skill commands are issued by a predefined event-based scheduler: command transitions are triggered when the current subtask objective is completed. All tasks use randomized initial conditions. The detailed environment parameters, switching conditions, success criteria, and termination conditions are provided in Appendix B.1.

Metrics. For composite tasks, we report the overall success rate and average completion time, where time is averaged over successful episodes. We also report substage success rates, which are computed over episodes that reach the corresponding substage. All composite-task results in simulation are evaluated over 1000 trials per method.

Baseline Setting. We select representative single-stage training methods as comparative baselines. (1) **AMP** [35] uses a single shared adversarial discriminator for all reference motions. (2) **Conditional AMP** [19, 27] is a latent-based multi-skill method. It maps the skill command to a target latent, while a skill discriminator predicts the latent from motion trajectories and provides a cosine-similarity skill reward. (3) **MoE-based AMP** [18] uses the same shared AMP discriminator as stan-

Method	Variant	Medium	Hard
Full Model	–	88.0	81.7
w/o Skill-Specific AMP	–	79.5	79.3
w/o Skill-Specific Critic	–	82.5	65.0
w/o Full Temporal Encoder	Raw Stacked History	49.0	20.0
	MLP History Encoder	61.9	47.6
	w/o Ball-Vel Estimation	83.6	44.4
	w/o Estimation	79.7	15.8
	Vanilla History Encoder	78.0	32.6
w/o Command-Duration Curriculum	–	85.5	30.4

Table II: **Ablation Study in Simulation.** Results are overall success rates. The top-level settings ablate the major components; the variants under ‘w/o Full Temporal Encoder’ further isolate the effects of each part.

dard AMP, but replaces the shared policy backbone with a mixture-of-experts architecture, where the number of experts matches the number of skill categories. (4) **MoE-Encoder AMP** [40] augments AMP with a mixture-of-experts history encoder using soft-gated dense routing. (5) **Vanilla PPO** [34] is trained only with task rewards, without adversarial motion priors. For a fair comparison, all methods use the same task rewards, maximum training budget, and evaluation protocol. Additional implementation details and hyperparameters are provided in Appendix B.2.

Results and Analysis. Table I evaluates command-driven skill composition in simulation. On the Medium DRIBBLE→SHOOT task, SkillX achieves **88.0%** overall success, outperforming the best baseline, MoE-Encoder AMP, by 20.8 percentage points. On the Hard task, involving trapping, three-waypoint dribbling, and final shooting, SkillX maintains **81.7%** overall success, compared with 45.6% for the best baseline, MoE-Encoder AMP.

The substage results further reveal the baselines’ failure modes. AMP completes some individual stages but suffers from error accumulation over longer sequences, especially during multi-waypoint dribbling and final shooting. Conditional AMP and MoE-based AMP show unstable composition despite occasional high substage success. MoE-Encoder AMP improves overall completion over these baselines, but its performance still degrades substantially on the Hard task. These results suggest that latent conditioning or expert routing alone does not reliably preserve reusable skills across transitions. In contrast, SkillX maintains consistently high substage success and substantially higher overall completion on both composition tasks. The Hard-task rollout in Fig. 3 further illustrates how our policy composes trapping, multi-waypoint dribbling, and shooting into a complete long-horizon behavior.

4.3 Ablation Study

Table II evaluates the major framework components and training strategy, and further decomposes the full temporal-encoder design through five variants: (a) *Raw Stacked History* directly feeds observation history to the actor, without an encoder; (b) *MLP History Encoder* replaces the Transformer with an MLP while retaining the auxiliary objectives; (c) *w/o Ball-Vel Estimation* removes only the ball-velocity estimation objective; (d) *w/o Estimation* removes all velocity estimation objectives; and (e) *Vanilla History Encoder* uses a history MLP without auxiliary objectives. Full definitions and controlled settings are provided in Appendix B.3.

Removing the skill-specific adversarial motion priors reduces success on both medium and hard tasks, indicating that separated style regularization helps preserve distinct skill behaviors during composition. Removing the skill-specific critics causes a larger drop on the hard task, suggesting that skill-dependent value estimation becomes more important as the task horizon and number of transitions increase.

Raw Stacked History drops to 49.0% on Medium and 20.0% on Hard despite receiving the same historical inputs, showing that access to history alone is insufficient. *MLP History Encoder* recovers part of the performance but remains below the Full Model, particularly on Hard, indicating

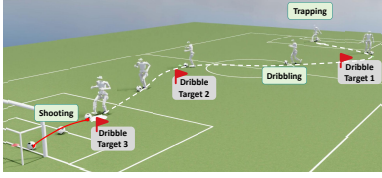


Figure 3: **Simulation Rollout.**

Method	Dribble	Shoot	Two-step Dribble	Dribble + Shoot
AMP [35]	1/10	1/10	0/10	0/10
Ours	8/10	8/10	7/10	6/10

Table III: **Real-World Performance.** Success rates over 10 hardware trials per task under the MoCap system.

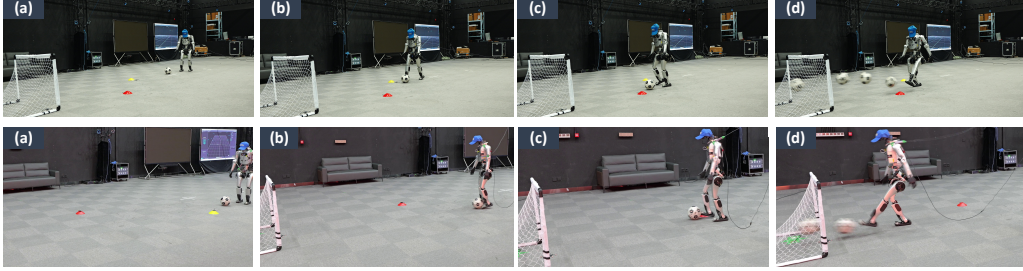


Figure 4: **Real-World Rollouts** on the Noetix E1 humanoid robot under the MoCap system.

the benefit of structured temporal modeling. Removing either all velocity estimation or only ball-velocity estimation also substantially degrades Hard-task performance. Together with the result of *Vanilla History Encoder*, these findings show that the gain arises from the combination of temporal architecture and interaction-relevant auxiliary supervision.

Removing command-duration curriculum causes only a modest decrease on Medium (88.0% to 85.5%) but a substantial drop on Hard (81.7% to 30.4%), indicating that the curriculum is especially important for long-horizon skill composition.

4.4 Real-World Experiments

Evaluation Setting. We evaluate real-world performance under the MoCap deployment setup described in Sec. A.1. The evaluation contains four tasks of varying difficulty: dribbling to a target region, shooting to the goal, two-step dribbling with a direction change, and dribble-then-shoot. The first two tasks evaluate atomic skill execution, while the latter two evaluate long-horizon skill composition and command-driven transitions. For the composite hardware tasks, skill commands are manually triggered when the corresponding subtask is completed.

Metrics and Baseline. We report success over 10 hardware trials per task. A trial is successful if the robot completes the task objective without falling and the ball remains within the valid field boundary. For composite tasks, all subtasks must be completed in the specified order. We compare SkillX with the standard AMP baseline using the same deployment interface and sensing pipeline.

Results. Table III shows that SkillX transfers reliably to the real robot and substantially outperforms AMP across all tasks. On atomic skills, our policy achieves **80%** success on both dribbling and shooting, whereas AMP succeeds in only 1 out of 10 trials for each task. The performance gap becomes larger on composite tasks: SkillX achieves **70%** and **60%** success, respectively, while AMP fails in all trials. These results indicate that separating skill-specific training signals improves not only individual skill execution but also real-world robustness during command-driven skill transitions. Qualitative rollouts in Fig. 4 further illustrate that the policy can execute robust soccer skill behaviors on hardware.

4.5 Onboard Vision Deployment

As an additional feasibility study beyond the MoCap evaluation, we deploy SkillX with onboard perception using a head-mounted ZED2i camera with visual-inertial odometry and YOLOv8-based ball detection. The estimated robot and ball states are converted into the same robot-centric policy

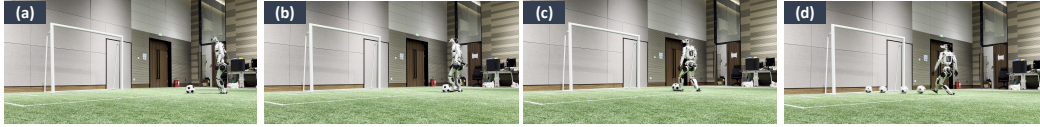


Figure 5: **Onboard vision deployment** of our policy on the dribble-then-shoot composite task.

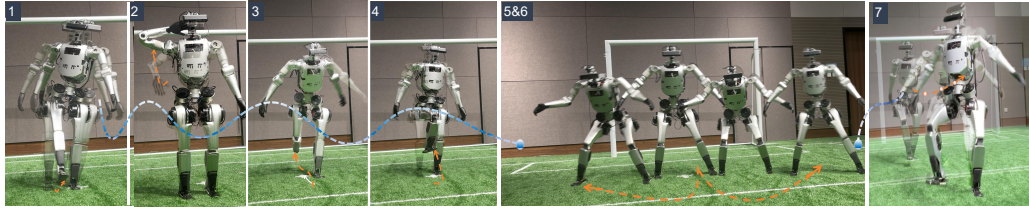


Figure 6: **Smooth Skill Transitions within One Policy.** A single command-conditioned policy executes and transitions among diverse whole-body skills, including (1) *kick*, (2) *salute*, (3) *back kick*, (4) *single-leg stand*, (5-6) *left/right defending motions*, and (7) *run-to-shoot*.

interface used in simulation and MoCap deployment. To account for perception noise, we train with degraded observations, including lower-rate updates and observation delays. Although onboard vision is noisier than MoCap, the policy can still execute ball-interaction behaviors, as shown in Fig. 5, suggesting improved sim-to-real robustness under practical sensing conditions. More qualitative results and deployment details are provided in Appendix A.2.

4.6 Skill Diversity Beyond Soccer Interaction

Beyond ball-interaction tasks, we further evaluate SkillX on multi-skill stylized motion imitation. We remove external object observations and object-interaction rewards, so the policy is driven only by proprioception and skill commands while retaining the same training framework. As shown in Fig. 6, a single policy learns diverse whole-body skills and transitions smoothly among them, including kicking, saluting, back kicking, single-leg standing, defending motions, and run-to-shoot. This result shows that the SkillX design is not tied to ball-interactive tasks and can support broader multi-skill humanoid motion learning.

5 Conclusion

We presented **SkillX**, a unified reinforcement learning framework for **multi-skill humanoid soccer**. SkillX uses a single command-conditioned policy for deployment while using **skill-specific adversarial motion priors**, **skill-specific critics**, and an **object-aware temporal encoder** to address style interference, skill-dependent value estimation, and partial observability. Experiments in simulation and on a real Noetix E1 humanoid demonstrate improved atomic skill execution, robust sim-to-real deployment, and **long-horizon multi-skill composition**. These results suggest that combining a unified deployable policy with skill-specialized training signals is a promising direction for multi-skill object-interactive control.

6 Limitations

SkillX focuses on low-level multi-skill execution and command-conditioned transitions under externally provided high-level skill commands. Therefore, it does not address strategic decision-making, task planning, or autonomous skill sequencing in open-ended soccer scenarios. Our experiments also focus on single-robot tasks; extending the framework to multi-agent soccer with teammates and opponents would introduce additional challenges such as physical disturbances, coordination, and strategic skill selection. Future work may integrate SkillX with hierarchical planning or multi-agent coordination to support broader soccer scenarios.

Acknowledgments

We thank Beijing Virtual Point Tech Co., Ltd. for providing motion-capture support for the real-world experiments.

References

- [1] 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*, pages 340–347, 1997.
- [2] R. Gerndt, D. Seifert, J. H. Baltes, S. Sadeghnejad, and S. Behnke. Humanoid robots in soccer: Robots versus humans in robocup 2050. *IEEE Robotics & Automation Magazine*, 22(3):147–154, 2015.
- [3] S.-J. Yi, S. McGill, D. Hong, and D. Lee. Hierarchical motion control for a team of humanoid soccer robots. *International Journal of Advanced Robotic Systems*, 13(1):32, 2016.
- [4] I. J. Da Silva, D. H. Perico, T. P. D. Homem, and R. A. da Costa Bianchi. Deep reinforcement learning for a humanoid robot soccer player. *Journal of Intelligent & Robotic Systems*, 102(3):69, 2021.
- [5] D. Tirumala, M. Wulfmeier, B. Moran, et al. Learning robot soccer from egocentric vision with deep reinforcement learning. In *The Eighth Conference on Robot Learning*, 2024.
- [6] T. Haarnoja, B. Moran, G. Lever, et al. Learning agile soccer skills for a bipedal robot with deep reinforcement learning. *Science Robotics*, 9(89):eadi8022, 2024.
- [7] Y. Ze, Z. Chen, J. P. Araújo, Z. ang Cao, X. B. Peng, J. Wu, and C. K. Liu. Twist: Teleoperated whole-body imitation system, 2025.
- [8] W. Xie, J. Han, J. Zheng, H. Li, X. Liu, J. Shi, W. Zhang, C. Bai, and X. Li. Kungfubot: Physics-based humanoid whole-body control for learning highly-dynamic skills. *Advances in Neural Information Processing Systems*, 38:62406–62433, 2026.
- [9] Q. Liao, T. E. Truong, X. Huang, 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.
- [10] Z. Luo, Y. Yuan, T. Wang, C. Li, S. Chen, F. Castañeda, Z.-A. Cao, J. Li, D. Minor, Q. Ben, X. Da, R. Ding, C. Hogg, L. Song, E. Lim, E. Jeong, T. He, H. Xue, W. Xiao, Z. Wang, S. Yuen, J. Kautz, Y. Chang, U. Iqbal, L. Fan, and Y. Zhu. Sonic: Supersizing motion tracking for natural humanoid whole-body control. *arXiv preprint arXiv:2511.07820*, 2025.
- [11] Y. Ma, H. Yu, J. Xie, C. Lv, Q. Luo, C. Zhang, Y. Yin, B. Xing, X. Ren, and D. Zheng. Robust and generalized humanoid motion tracking. *arXiv preprint arXiv:2601.23080*, 2026.
- [12] 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. *arXiv preprint arXiv:2602.05310*, 2026.
- [13] Y. Wang, Q. Zhao, Y. F. Lau, R. Yu, H. W. Tsui, Q. Chen, J. Wang, J. Pang, and P. Tan. Humanx: Toward agile and generalizable humanoid interaction skills from human videos. *arXiv preprint arXiv:2602.02473*, 2026.
- [14] Z. Zhang, H. Lu, Y. Lian, Z. Chen, Y. Liu, C. Lin, H. Xue, Z. Zeng, Z. Qi, S. Zheng, et al. Learning athletic humanoid tennis skills from imperfect human motion data. *arXiv preprint arXiv:2603.12686*, 2026.

- [15] J. Ren, Y. Li, K. Zhang, P. Fu, H. Jiang, Y. Pan, G. Zeng, T. Huang, W. Guo, P. Lu, et al. Smash: Mastering scalable whole-body skills for humanoid ping-pong with egocentric vision. *arXiv preprint arXiv:2604.01158*, 2026.
- [16] C. Liu, L. Jiang, Y. Wang, K. Yao, J. Fu, and X. Ren. Humanoid whole-body badminton via multi-stage reinforcement learning. *arXiv preprint arXiv:2511.11218*, 2025.
- [17] Y. Chen, S. Dong, X. Ji, J. Sun, Z. Luo, L. Zhao, J. Zhang, W. Li, J. Ma, B. Xu, et al. Learning human-like badminton skills for humanoid robots. *arXiv preprint arXiv:2602.08370*, 2026.
- [18] R. Huang, S. Zhu, Y. Du, and H. Zhao. Moe-loco: Mixture of experts for multitask locomotion. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 14218–14225. IEEE, 2025.
- [19] N. Huang, Z. Xie, and Q. Li. Learning multi-skill legged locomotion using conditional adversarial motion priors. *arXiv preprint arXiv:2509.21810*, 2025.
- [20] S. Ma, H. Chen, Z. Xu, Y. Zhao, K. Wu, R. Yang, L. Zou, Z. Gan, and W. Ding. Cmoe: Contrastive mixture of experts for motion control and terrain adaptation of humanoid robots. *arXiv preprint arXiv:2603.03067*, 2026.
- [21] D. Wang, X. Wang, X. Liu, J. Shi, Y. Zhao, C. Bai, and X. Li. More: Mixture of residual experts for humanoid lifelike gaits learning on complex terrains. *arXiv preprint arXiv:2506.08840*, 2025.
- [22] N. Shah, K. Tiwari, and A. Bera. Mtac: Hierarchical reinforcement learning-based multi-gait terrain-adaptive quadruped controller. *arXiv preprint arXiv:2401.03337*, 2023.
- [23] Y. Kuang, H. Geng, A. Elhafsi, T.-D. Do, P. Abbeel, J. Malik, M. Pavone, and Y. Wang. Skillblender: Towards versatile humanoid whole-body loco-manipulation via skill blending. *arXiv preprint arXiv:2506.09366*, 2025.
- [24] Y. Wang, Q. Zhao, R. Yu, H. W. Tsui, A. Zeng, J. Lin, Z. Luo, J. Yu, X. Li, Q. Chen, J. Zhang, L. Zhang, and P. Tan. Skillmimic: Learning basketball interaction skills from demonstrations. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 17540–17549, June 2025.
- [25] W. Yu, F. Acero, V. Atanassov, C. Yang, I. Havoutis, D. Kanoulas, and Z. Li. Discovery of skill-switching criteria for learning agile quadruped locomotion. *Frontiers in Robotics and AI*, 13:1697159, 2026.
- [26] X. B. Peng, Y. Guo, L. Halper, S. Levine, and S. Fidler. Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Transactions On Graphics (TOG)*, 41(4):1–17, 2022.
- [27] C. Tessler, Y. Kasten, Y. Guo, S. Mannor, G. Chechik, and X. B. Peng. Calm: Conditional adversarial latent models for directable virtual characters. In *ACM SIGGRAPH 2023 conference proceedings*, pages 1–9, 2023.
- [28] E. Vollenweider, M. Bjelonic, V. Klemm, N. Rudin, J. Lee, and M. Hutter. Advanced skills through multiple adversarial motion priors in reinforcement learning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5120–5126. IEEE, 2023.
- [29] S. Mysore, G. Cheng, Y. Zhao, K. Saenko, and M. Wu. Multi-critic actor learning: Teaching rl policies to act with style. In *International Conference on Learning Representations*, 2022.
- [30] Z. Su, B. Zhang, N. Rahmanian, Y. Gao, Q. Liao, C. Regan, K. Sreenath, and S. S. Sastry. Hitter: A humanoid table tennis robot via hierarchical planning and learning. *arXiv preprint arXiv:2508.21043*, 2025.

- [31] Z. Wang, J. Zhou, and Q. Wu. Dribble master: Learning agile humanoid dribbling through legged locomotion. *arXiv preprint arXiv:2505.12679*, 2025.
- [32] 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. *arXiv preprint arXiv:2511.03996*, 2025.
- [33] Z. Xu, M. Seo, D. Lee, H. Fu, J. Hu, J. Cui, Y. Jiang, Z. Wang, A. Brund, J. Biswas, et al. Learning agile striker skills for humanoid soccer robots from noisy sensory input. *arXiv preprint arXiv:2512.06571*, 2025.
- [34] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [35] 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 (ToG)*, 40(4): 1–20, 2021.
- [36] 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 *International Conference on Learning Representations*, volume 2024, pages 14084–14100, 2024.
- [37] M. Mittal, P. Roth, J. Tigue, A. Richard, O. Zhang, P. Du, A. Serrano-Muñoz, X. Yao, R. Zurbrugg, N. Rudin, L. Wawrzyniak, M. Rakhsha, A. Denzler, E. Heiden, A. Borovicka, O. Ahmed, I. Akinola, A. Anwar, M. T. Carlson, J. Y. Feng, A. Garg, R. Gasoto, L. Gulich, Y. Guo, M. Gussert, A. Hansen, M. Kulkarni, C. Li, W. Liu, V. Makoviychuk, G. Malczyk, H. Mazhar, M. Moghani, A. Murali, M. Noseworthy, A. Poddubny, N. Ratliff, W. Rehberg, C. Schwarke, R. Singh, J. L. Smith, B. Tang, R. Thaker, M. Trepte, K. V. Wyk, F. Yu, A. Millane, V. Ramasamy, R. Steiner, S. Subramanian, C. Volk, C. Chen, N. Jawale, A. V. Kurutukulam, M. A. Lin, A. Mandlekar, K. Patzwaldt, J. Welsh, H. Zhao, F. Anes, J.-F. Lafleche, N. Moënné-Loccoz, S. Park, R. Stepinski, D. V. Gelder, C. Amevor, J. Carius, J. Chang, A. H. Chen, P. de Heras Ciechowski, G. Daviet, M. Mohajerani, J. von Muralt, V. Reutsky, M. Sauter, S. Schirm, E. L. Shi, P. Terdiman, K. Vilella, T. Widmer, G. Yeoman, T. Chen, S. Grizan, C. Li, L. Li, C. Smith, R. Wiltz, K. Alexis, Y. Chang, D. Chu, L. J. Fan, F. Farshidian, A. Handa, S. Huang, M. Hutter, Y. Narang, S. Pouya, S. Sheng, Y. Zhu, M. Macklin, A. Moravanszky, P. Reist, Y. Guo, D. Hoeller, and G. State. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025.
- [38] Z. Luo, J. Cao, A. W. Winkler, K. Kitani, and W. Xu. Perpetual humanoid control for real-time simulated avatars. In *International Conference on Computer Vision (ICCV)*, 2023.
- [39] R. Varghese and M. Sambath. Yolov8: A novel object detection algorithm with enhanced performance and robustness. In *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, pages 1–6. IEEE, 2024.
- [40] T. Wu, H. Guo, Y. Wang, J. Yang, X. Sui, J. Xie, X. Chen, Z. Liu, and X. Lan. Toward reliable sim-to-real predictability for moe-based robust quadrupedal locomotion. In *Proceedings of Robotics: Science and Systems*, 2026.
- [41] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 3803–3810. IEEE, 2018.

A Deployment Details

Our real-world experiments are conducted on the 25-DoF Noetix E1 humanoid robot. The learned policy runs on the onboard computer and outputs target joint positions at 50 Hz, which are tracked by the low-level motor controller. For deployment safety, policy actions are clipped before being sent to the low-level controller. We implement two real-world deployment backends: a motion-capture backend for controlled evaluation and an onboard-vision backend for infrastructure-free execution.

Unified deployment interface. Both deployment backends use the same robot-centric policy interface. Raw measurements from MoCap, onboard vision, or simulation are converted into a common observation format containing ball position, target position, proprioception, previous action, and skill command. The observation is appended to a short history buffer and fed to the exported ONNX policy. This design separates the learned policy from the sensing backend and reduces deployment-specific changes.

A.1 Motion-Capture Deployment

For MoCap deployment, rigid-body poses from the Motive system are received through VRPN and exposed to the controller as ROS topics. The MoCap system provides global measurements of the robot, ball, and goal. We attach a rigid 3D-printed marker mount to the robot torso and calibrate its fixed transform to the robot `base_link`, allowing the measured rigid-body pose to be converted into the robot base frame. Reflective markers are also attached to the ball and goal to estimate their global positions.

The controller transforms all global object measurements into robot-centric observations. Let $\mathbf{p}_r^w$ and $\mathbf{R}_r^w$ denote the robot position and orientation in the world frame, and let $\mathbf{p}_o^w$ denote the world-frame position of an external object such as the ball or target.

$$\mathbf{p}_o^r = (\mathbf{R}_r^w)^\top (\mathbf{p}_o^w - \mathbf{p}_r^w), \quad (8)$$

Only the xy-plane components are used for ball and target observations.

The MoCap system runs at 120 Hz, while the policy runs at 50 Hz. Although this setup provides repeatable closed-loop evaluation, it is not equivalent to a perfect simulator state: relative object observations can be affected by tracking jitter, temporary occlusion, calibration error, and latency, especially for fast ball motion. We therefore apply exponential low-pass filtering to relative object observations and treat stale rigid-body measurements as invalid to avoid using outdated data. The marker mount is positioned to reduce occlusion during dynamic motions and improve tracking reliability.

Fig. 4 illustrates representative deployment rollouts under the MoCap setup. Utilizing high-precision, real-time object state estimations from the tracking infrastructure, the robot robustly switches between distinct skills. This continuous feedback loop facilitates seamless closed-loop object interactions, ensuring high-fidelity coordination and successful completion of the final tactical task.

A.2 Onboard Vision Deployment

For onboard-vision deployment, we use a head-mounted ZED2i camera. Robot odometry is estimated from the camera’s visual-inertial odometry module, while the soccer ball is detected using a YOLOv8-based detector. The detector runs at approximately 30 Hz with GPU acceleration. The 2D detection is converted into a robot-centric 3D ball observation using depth measurements and camera calibration. Since the target location is fixed in the field, its relative position is obtained from the estimated odometry. The resulting ball and target observations are converted into the same robot-centric policy interface used in MoCap deployment.

To improve robustness to onboard perception, we train the policy with a perception-degraded observation model rather than directly using perfect object states. During training, the object observation

is updated at a lower rate, perturbed with noise and random delay, and temporarily frozen when the ball is outside the simulated camera field of view. We also simulate basic geometric visibility constraints, including camera range and field-of-view limits, to approximate the failure modes of the physical detector. These perturbations encourage the policy to handle delayed, noisy, and intermittent object observations during deployment.

Fig. 5 shows representative onboard-vision deployment rollouts. The robot can localize the ball from the head-mounted camera, approach it, and execute closed-loop ball-interaction behaviors without relying on an external MoCap system. Although this setup is noisier and less reliable than MoCap-based sensing, the same policy interface can accommodate both sensing backends, demonstrating the robustness of our deployment design under practical perception conditions.

B Simulation Experiment

B.1 Evaluation Protocol

We evaluate the proposed method in NVIDIA Isaac Lab [37] using a soccer-field environment. All policies are evaluated with a single environment, and each episode is executed sequentially. For each task setting, we report the success rate over independent evaluation episodes and the average completion time over successful episodes. During evaluation, all domain randomization and observation corruption are disabled to ensure reproducibility.

The evaluation field is a $0.25\times$ scaled standard soccer field with a length of 25.0 m and a width of 15.0 m. The goal width and height are 2.745 m and 1.22 m, respectively. The ball radius is 0.11 m and its mass is fixed to 0.45 kg during evaluation.

We consider three atomic-skill tasks and two composite tasks. Atomic-skill tasks are designed to isolate a single soccer primitive at a time. In these tasks, the high-level skill command is fixed throughout the episode, and the policy is only required to complete one behavior. This setting provides a controlled evaluation of the learned motion primitives before they are composed into longer and more complex soccer tasks.

Trap. The trap task evaluates the robot’s ability to stop an incoming ball. The robot is initialized at the origin and faces the positive x direction. The ball is initialized 1.0–2.0 m away from the robot with a random bearing within ± 0.785 rad and an initial speed of 0.5–1.5 m/s toward the robot. The skill command is fixed to TRAP. The task is considered successful when the ball’s planar velocity becomes lower than 0.3 m/s within 20 s.

Dribble. The dribble task evaluates the robot’s ability to move the ball to a randomly sampled target position. The robot is initialized at the origin, and the ball is initialized 0.5–1.0 m away with an angular offset within $\pm\pi/3$. The target is sampled with $x \in [4.5, 5.5]$ m and $y \in [-2.0, 2.0]$ m. The skill command is fixed to DRIBBLE. The task is successful if the ball reaches within 0.5 m of the target within 20 s.

Shoot. The shoot task evaluates the robot’s ability to kick a stationary ball into the goal. The robot is initialized 4–6 m in front of the goal with a heading perturbation within ± 0.3 rad. The ball is initialized 0.5–1.2 m away from the robot with an angular offset within ± 0.3 rad. The skill command is fixed to SHOOT. An episode is successful if the ball crosses the goal line, and it fails after 15 s if the goal is not reached.

We further evaluate the policy on composed tasks that explicitly require multi-stage skill execution.

Medium. The medium task evaluates a composed behavior consisting of dribbling followed by shooting. The robot starts near the midfield with the ball initialized 0.6–1.2 m away. The initial skill command is DRIBBLE. After the ball reaches the waypoint at the penalty point, the command is automatically switched to SHOOT. The episode is successful if the ball enters the goal within 30 s.

Hard. The hard task evaluates the full composition of trapping, dribbling, and shooting. The task starts with the TRAP skill, where the ball moves toward the robot with an initial speed of

0.5–1.0 m/s. Once the ball speed falls below 0.3 m/s, the skill command switches to DRIBBLE. The robot then needs to guide the ball through three waypoints: $(-6.25, 1.145)$, $(0, -1.145)$, and $(9.75, 0)$ along an S-shaped route. After all waypoints are reached, the command switches to SHOOT. The task is successful if the ball enters the goal within 40 s.

B.2 Baseline Settings

We compare our method with five baselines: Vanilla PPO, AMP, Conditional AMP, MoE-based AMP, and MoE-Encoder AMP. Conditional AMP, MoE-based AMP, and MoE-Encoder AMP are self-implemented in our framework. All methods use the same robot model, simulation settings, observation space, task reward design, reference motion dataset, and evaluation protocol.

Vanilla PPO. [34] Vanilla PPO is trained only with task rewards and does not use adversarial motion priors. The actor and critic are both multilayer perceptrons with hidden dimensions [2048, 1024, 1024, 512, 512] and ELU activations.

AMP. [35] AMP uses a standard adversarial motion prior with a single discriminator shared by all skills. The policy and value networks follow the same architecture as Vanilla PPO. The discriminator is an MLP with hidden dimensions [1024, 512]. The AMP reward coefficient is set to 1.0.

Conditional AMP. [19] Conditional AMP extends AMP with latent-based skill conditioning. It maps the skill command g to a target latent $z = E(g)$. In addition to the shared AMP discriminator, a skill discriminator with hidden dimensions [512, 256] predicts $\hat{z} = f_\psi(\tau)$ from motion trajectory τ and provides the cosine-similarity reward $r^{\text{skill}} = \cos(\hat{z}, z)$. The total reward is $r = r^{\text{task}} + \lambda^{\text{style}} r^{\text{style}} + \lambda^{\text{skill}} r^{\text{skill}}$. The AMP style reward coefficient is 1.0, and the skill reward coefficient is 0.5.

MoE-based AMP. [18] MoE-based AMP combines a mixture-of-experts policy with AMP. It uses three policy experts, corresponding to the number of skills, and a gating network with a hidden dimension of 128. Top-1 routing is used, meaning that only one expert is activated at each step. Each expert has its own actor and critic networks.

MoE-Encoder AMP. [40] MoE-Encoder AMP augments AMP with a mixture-of-experts history encoder using soft-gated dense routing. The encoder contains three experts and a soft gating network, and maps a five-step observation history to a 32-dimensional ℓ_2 -normalized latent representation. The resulting latent is concatenated with the current observation and provided to the actor. We adopt the load-balancing objective from the original paper [40] for expert utilization, but do not use CTS training or RoboGauge evaluation protocol.

Shared hyperparameters. All baseline methods are trained under the same PPO hyperparameter configuration, except for method-specific network structures and adversarial modules. The shared training hyperparameters are summarized in Table IV.

B.3 Details of Ablation Experiments

We conduct simulation-based ablation studies to quantify the contribution of each core component in our framework.

The framework-level variants are defined as follows: w/o Skill-Specific AMP replaces the skill-specific discriminators with a single discriminator, w/o Skill-Specific Critic uses a single shared critic head instead of skill-specific value heads, and w/o Command-Duration Curriculum samples the final command-duration range from the beginning of training instead of progressively shortening skill segments.

The five variants under w/o Full Temporal Encoder further isolate the source of the complete encoder design’s gains. **(a)** *Raw Stacked History* preserves the full five-frame observation history but flattens it and feeds it directly to the actor, without any encoder or encoder-side auxiliary objective. **(b)** *MLP History Encoder* replaces the Transformer with an MLP while retaining the same auxiliary

Table IV: **Shared training hyperparameters.** These hyperparameters are kept the same for all baseline methods and our method unless otherwise specified.

Hyperparameter	Value
Learning epochs	5
Mini-batches	4
Learning rate	1.0×10^{-3} , adaptive schedule
Discount factor γ	0.99
GAE parameter λ	0.95
Entropy coefficient	0.01
Target KL divergence	0.01
Gradient clipping	1.0
PPO clipping coefficient	0.2
Value loss coefficient	1.0
Action clipping	18.0
Initial noise standard deviation	1.0, scalar

objectives. **(c)** *w/o Ball-Vel Estimation* removes the ball-velocity prediction target while retaining the remaining encoder objectives. **(d)** *w/o Estimation* removes all velocity-estimation objectives while retaining the other parts of the temporal encoder, whereas **(e)** *Vanilla History Encoder* uses a history MLP without auxiliary objectives. These comparisons distinguish the effects of an explicit history representation, temporal architecture, and interaction-relevant supervision.

B.4 Additional Results

In addition to the composite task results reported in the main text, we further evaluate all methods on isolated atomic skill tasks. These results serve to verify whether the learned policy can reliably execute each primitive skill. Each atomic-skill result is evaluated over 100 trials per method.

Table V shows that our method achieves the best overall performance across the three atomic skills. Vanilla PPO performs well on trapping but fails on shooting and dribbling, indicating that task rewards alone are insufficient for highly dynamic, contact-rich ball skills. Although Standard AMP improves these skills with motion priors, its shared discriminator still underperforms our method. Conditional AMP and MoE-based AMP also fail to consistently learn all skills, suggesting that skill conditioning or expert routing cannot effectively resolve interference among heterogeneous skills. In contrast, our method achieves **95.0%** success on shooting and **100.0%** success on both trapping and dribbling, demonstrating more balanced and robust primitive-skill learning.

Table V: **Atomic-Skill tasks for baseline experiments.**

Method	Shoot	Trap		Dribble	
	Succ.(%)	Succ.(%)	Time(s)	Succ.(%)	Time(s)
Vanilla PPO	9.0	97.0	3.90	5.0	7.26
AMP	87.0	99.0	5.22	94.0	6.27
Conditional AMP	1.0	100.0	5.84	8.0	6.70
MoE-based AMP	1.0	99.0	5.77	36.0	5.90
Ours	95.0	100.0	5.31	100.0	5.62

Table VI further reports ablation experiments on atomic-skill tasks. Removing skill-specific AMP mainly degrades shooting performance, showing that skill-specific adversarial supervision is particularly important for dynamic contact behaviors. Removing the skill-specific critics also reduces shooting success, indicating that skill-specific value estimation helps stabilize optimization across heterogeneous skills. Raw Stacked History causes a substantial drop in shooting and dribbling performance despite retaining the five-frame inputs, suggesting that explicitly encoding temporal context is important for stable ball control. Overall, these supplementary results confirm that the

components of our method not only improve the complex multi-stage tasks in the main text, but also strengthen the underlying atomic skills required for successful skill composition.

Table VI: **Atomic-Skill tasks for ablation experiments.**

Method	Shoot	Trap		Dribble	
	Succ.(%)	Succ.(%)	Time(s)	Succ.(%)	Time(s)
w/o Skill-Specific AMP	87.0	100.0	2.77	100.0	5.60
w/o Skill-Specific Critic	85.0	100.0	5.32	100.0	6.66
Raw Stacked History	16.0	98.0	5.77	91.0	10.80
Ours	95.0	100	5.31	100.0	5.62

Stricter trapping criterion. To further evaluate trapping under a more challenging criterion, we measure success from the first robot–ball contact, defined by a planar distance of at most 0.4 m. A trial is successful only if, within 3 s after contact, the ball maintains a speed below 0.25 m/s and remains within 0.5 m of the robot for at least 0.5 s. Over 100 trials per method, SkillX achieves a success rate of **63.0%**, compared with 31.0% for AMP, 24.0% for Conditional AMP, and 50.0% for MoE-based AMP, retaining the best performance among all compared methods under this stricter evaluation.

C Implementation Details

C.1 Observation Details

We use three observation groups during training: actor observations, critic observations, and discriminator observations. The actor observations are available during both training and deployment, while the critic and discriminator observations are used only during training.

Actor observation. The actor is fed with a historical observation window of length $K = 5$. Each timestep frame integrates proprioceptive states, external object observations, and the skill command, formulated as:

$$\mathbf{o}_t = [\mathbf{p}_t, \mathbf{c}_t, \mathbf{o}_t^{\text{ext}}], \quad (9)$$

$$\mathbf{p}_t = [\boldsymbol{\omega}_t^{\text{base}}, \mathbf{g}_t^{\text{body}}, \mathbf{q}_t, \dot{\mathbf{q}}_t, \mathbf{a}_{t-1}], \quad (10)$$

$$\mathbf{o}_t^{\text{ext}} = [\mathbf{p}_{t,xy}^{\text{ball}}, \mathbf{p}_{t,xy}^{\text{target}}]. \quad (11)$$

Here, $\boldsymbol{\omega}_t^{\text{base}} \in \mathbb{R}^3$ denotes the base angular velocity, $\mathbf{g}_t^{\text{body}} \in \mathbb{R}^3$ represents the gravity vector projected onto the body frame, $\mathbf{q}_t, \dot{\mathbf{q}}_t \in \mathbb{R}^{25}$ correspond to joint positions and velocities respectively, $\mathbf{a}_{t-1} \in \mathbb{R}^{25}$ is the prior-step action, $\mathbf{p}_{t,xy}^{\text{ball}} \in \mathbb{R}^2$ defines the horizontal-plane ball position relative to the robot, $\mathbf{p}_{t,xy}^{\text{target}} \in \mathbb{R}^2$ specifies the horizontal-plane target position relative to the robot, and $\mathbf{c}_t \in \mathbb{R}^{N_{\text{skill}}}$ stands for the skill command.

Accordingly, the dimensionality of the per-frame actor observation is calculated as: $d_\pi = 3 + 3 + 25 + 25 + 25 + 2 + 2 + N_{\text{skill}} = 85 + N_{\text{skill}}$. For our core soccer experiments with $N_{\text{skill}} = 3$, the per-frame dimension reduces to $d_\pi = 88$. Combined with the history window $K = 5$, the actor input features a shape of 5×88 , amounting to 440 scalar values when flattened. During training, observation noise is injected into all components of the actor observation except the skill command.

Critic observation. Following the standard asymmetric actor-critic paradigm, the critic adopts a privileged observation design that differs from the actor. The critic takes the uncorrupted actor observation sequence as the base input, and further incorporates additional privileged state information exclusive to the training phase:

$$\mathbf{o}_t^V = [\mathbf{o}_t, \mathbf{v}_t^{\text{base}}, \mathbf{v}_t^{\text{ball}}, \mathbf{f}_t^{\text{feet}}]. \quad (12)$$

Here, $\mathbf{v}_t^{\text{base}} \in \mathbb{R}^3$ denotes the base linear velocity of the robot, $\mathbf{v}_t^{\text{ball}} \in \mathbb{R}^3$ represents the ball velocity expressed in the robot-centric frame, and $\mathbf{f}_t^{\text{feet}} \in \mathbb{R}^2$ refers to the binary contact state of the robot’s two feet. The dimensionality of the per-frame critic observation is thus formulated as: $d_V = d_\pi + 3 + 3 + 2 = 93 + N_{\text{skill}}$. In our soccer experimental setup with $N_{\text{skill}} = 3$, the critic observation dimension yields $d_V = 96$. Consistent with the asymmetric actor-critic framework, these privileged observational features are solely utilized for value function estimation during model training, and are completely discarded in the deployed policy, incurring no extra inference overhead for the actor.

Discriminator observation. The AMP discriminator adopts a set of motion-centric observations distinct from the actor and critic modalities, which focus on capturing full-body motion characteristics for adversarial training. The discriminator observation is formulated as:

$$\mathbf{o}_t^D = [\mathbf{q}_t, \mathbf{x}_t^{\text{body}}, \mathbf{v}_t^{\text{base}}, \boldsymbol{\omega}_t^{\text{base}}, \dot{\mathbf{q}}_t, \mathbf{f}_t^{\text{feet}}].$$

Here, $\mathbf{q}_t \in \mathbb{R}^{25}$ and $\dot{\mathbf{q}}_t \in \mathbb{R}^{25}$ represent the joint positions and joint velocities, respectively. $\mathbf{x}_t^{\text{body}} \in \mathbb{R}^{3N_{\text{body}}}$ encodes the positional information of all robot rigid bodies, where N_{body} denotes the total number of robot bodies. For the 25-DoF Noetix E1 robot adopted in our experiments, $N_{\text{body}} = 26$. In addition, $\mathbf{v}_t^{\text{base}} \in \mathbb{R}^3$ is the base linear velocity, $\boldsymbol{\omega}_t^{\text{base}} \in \mathbb{R}^3$ is the base angular velocity, and $\mathbf{f}_t^{\text{feet}} \in \mathbb{R}^2$ corresponds to the binary foot-contact states of the robot’s two feet.

The dimensionality of the per-frame discriminator observation is calculated as $d_D = 25 + 3 \times 26 + 3 + 3 + 25 + 2 = 136$.

The discriminator is trained on sequential multi-step motion segments constructed from the above motion-centric observations. Discriminator observations are leveraged for adversarial motion prior training during the training phase, without being involved in policy deployment.

C.2 Training Strategy

Adaptive Command Sampling. We employ an adaptive command sampling strategy to prioritize underperforming skills and alleviate imbalanced learning. The failure rate f_i of each skill is tracked via exponential moving average (EMA) with a smoothing coefficient of 0.99. The sampling probability for skill i is computed as a weighted mixture between the failure-aware distribution and a uniform prior:

$$p_i = (1 - \alpha) \cdot \frac{f_i}{\sum_j f_j} + \alpha \cdot \frac{1}{N_{\text{skill}}}, \quad (13)$$

where the mixing weight α is set to 0.5 to balance adaptive focusing and uniform exploration. An initial warm-up phase of 5000 iterations uses uniform random sampling for all skills. After the warm-up period, the adaptive sampling strategy is activated to allocate more updates to skills with higher failure rates.

Command Duration Curriculum. A command duration curriculum is designed to gradually improve the policy’s ability to perform frequent and smooth skill transitions. Before iteration 9000, the command duration is uniformly sampled from a fixed range (5.0, 10.0) seconds to encourage stable single-skill execution. From iteration 9000 to 12000, the lower bound of the duration range is linearly annealed from 5.0 seconds to 0.2 seconds, while the upper bound remains fixed at 10.0 seconds. After iteration 12000, the duration range is fixed at (0.2, 10.0) seconds for the remainder of training. This curriculum forces the policy to first master stable behaviors and then learn agile skill switching.

Object Domain Randomization. To improve the policy’s robustness against physical variations during real-world deployment and reduce the sim-to-real gap in ball-interactive tasks, we apply domain randomization [41] to the material and mass properties of the soccer ball at each environment

reset. Specifically, we randomize the static friction coefficient within $(0.30, 0.70)$, the dynamic friction coefficient within $(0.10, 0.55)$, and the restitution within $(0.55, 0.95)$. The ball mass is randomized in the range of $(0.40, 0.48)$ kg. All physical properties are sampled independently and clamped to the specified ranges. This domain randomization improves generalization to varying object interaction dynamics without modifying the core learning framework.

Command Blending. Command blending is applied only during the 5-frame (0.1 s) transition window, accounting for approximately 2% of training steps. Therefore, the blended value is used as a local approximation to smooth value changes near skill transitions; it does not assume that the returns of arbitrarily interpolated policies are globally linear.