SkillMimic: Learning Basketball Interaction Skills from Demonstrations

Supplementary Material

1. The BallPlay Dataset

To address the scarcity of basketball HOI data and facilitate research on basketball skill learning, we introduce two datasets: one based on monocular vision estimation and the other using multi-view optical motion capture systems.

1.1. BallPlay-V

The BallPlay-V dataset applies a monocular annotation solution to estimate the high-quality human SMPL-X [5] parameters and object translations from RGB videos. However, annotating these videos with high-speed and dynamic movements and complex interactions in the 3D camera coordinate is quite challenging. Inspired by the whole-body annotation pipeline of Motion-X [3], our automatic annotation additionally introduces depth estimation [1], semantic segmentation [9], to obtain high-quality whole-body human motions and ball motions, as illustrated in Fig. 1. The BallPlay-V dataset contains eight basketball skills, including *back dribble, cross leg, hold, fingertip spin, pass, backspin, cross*, and *rebound*, as shown in Fig. 2.

1.2. BallPlay-M

Although BallPlay-V can acquire HOI data conveniently from RGB videos, its accuracy is limited due to errors of monocular depth estimation. Additionally, it struggles with occlusion issues, making it difficult to capture complex layup and dribbling data. To achieve more comprehensive and accurate basketball data, we create the BallPlay-M dataset using a optical motion capture system. During the capture process, optical markers are attached to both the player and the basketball to track body and ball movements. The player also wears gloves equipped with inertia sensors to estimate finger movements. Consequently, the player is parameterized as a skeleton with 52 joints (156 DOFs). We calculate and record the root rotation, root translation, joint positions, and joint rotations sequentially. The ball is parameterized as a sphere, with its rotation and center position recorded. All data are captured at 120 fps.

We collect a total of 251,656 frames of raw data at a frame rate of 120 fps, amounting to approximately 35 minutes of diverse basketball interactions. From these raw data, we extract and annotate a subset for learning basketball skills. To aid reader comprehension, we provide a coarse categorization of the labeled subset in Tab. 1. It is important to note that this coarse classification is intended for illustrative purposes only. In reality, we have annotated the skills with greater specificity. For example, the *dribble* category encompasses various distinct skills, such as *dribble forward*, *dribble right*, *back dribble*, and so on.

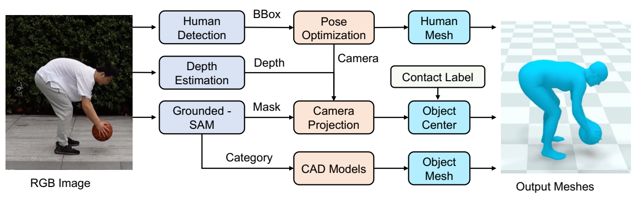


Figure 1. Annotation pipeline of BallPlay-V. Given an image, the pipeline estimates the human SMPL-X [5] parameters and object translations. The object is predefined as CAD models.

Coarse-Grained Classification	**Clips**	**Frames**	**FPS**
Pick Up	105	21,232	60
Catch	12	1,341	60
Rebound	31	2,175	60
Dribble	59	7,747	60
Shot	23	3,212	60
Pass	10	604	60
Layup	19	2,765	60
Getup	16	3,170	60
Misc.	24	4,812	60
Raw Data	-	251,656	120

Table 1. The composition of BallPlay-M dataset. We extracted and annotated a subset from approximately 35 minutes of raw data for skill learning purposes. We show the coarse categorization of the annotated subset in the above table.

2. Additional Results and Experiments

2.1. Additional Qualitative Results

We present comprehensive video results on our project page, demonstrating various aspects including basketball skill acquisition, high-level task execution, skill transitions, and comparative methods. Here we offer an in-depth comparison with variant methods, where detailed illustrations are provided in Fig. 3, Fig. 4, and Fig. 5.

2.2. Skill Robustness Against Physical Properties

To evaluate the robustness of the skills learned through SkillMimic, we conduct three perturbation tests on the dribble forward and pickup skills during inference: (1) varying the ball radius from 0.5× to 1.5× the default size; (2) altering the ball density from 0.1× to 6× the default; and (3) changing the ball restitution from 0.5× to 1.5× the default. The success rate was averaged across 1000 parallel environments. The quantitative results, presented in Tab. 2, demonstrate our method's robustness against variations in physical properties and external disturbances.

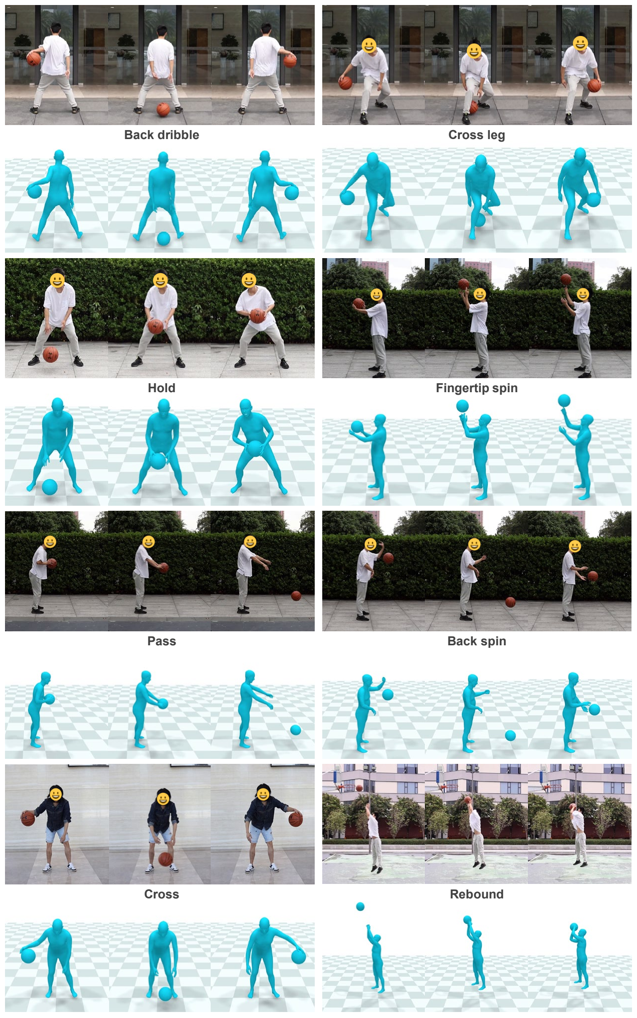


Figure 2. The BallPlay-V dataset. We show the eight HOI demonstrations of high-dynamic basketball skills. For each skill, the **upper** rows show the real-life videos, and the **lower** rows give the estimated whole-body SMPL-X human model and object mesh.

2.3. Ablation on Mixed Skill Training.

We conduct the following experiment to analyze the effect between different skills when learned together. From BallPlay-M, we select one clip for the layup skill, one clip for the dribble left skill, one clip for the dribble right skill, and two clips for the dribble forward skill. These skills are commonly used and combined in real-world basketball games. We first train four individual policies for each skill independently, with each policy trained for around 0.65 billion samples. As a control, we then train these four skills using a single policy, and report the results when trained for around 0.65 billion samples (the same as the individual training) and 2.6 billion samples (4 times of the individual training, but the average sampling number for each skill is the same as in the individual training). During testing, we compare the success rates of executing each skill independently and transitioning between skills. Specifically, for testing each skill, we use the skill clips for Reference State Initialization (RSI) and execute the corresponding skill. For testing skill switching, we use the source skill clips for RSI and execute the target skill.

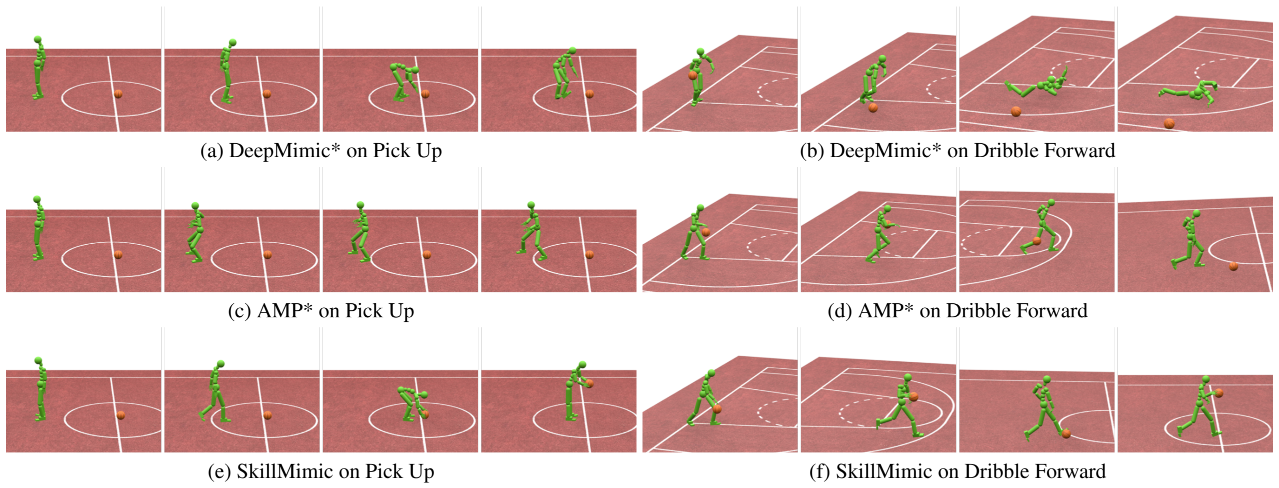
Success rates are calculated as described in Sec. 3.4. Tab. 3 presents the quantitative results. Despite equal sampling for each skill in both sets of experiments, mixed training shows a significant improvement in the success rates of individual skills and an even more substantial improvement in skill switching. It should be noted that while reward convergence is slightly faster in individual training, this approach is susceptible to overfitting. For example, the DL skill demonstrates excellent convergence of its reward, but due to the inadequate state cycle in the reference data, the DL skill trained independently tends to fall after a few dribbling steps, resulting in a zero success rate in sustained operations. Conversely, mixed training allows for cross-learning from other skills, thereby significantly enhancing the success rate of the dribble left skill. A similar phenomenon is observed in skill switching, where the reference data lacks examples of skill switches. Mixed training enables the policy to adapt to the state distributions of all skills, facilitating zero-shot skill switching during tests. These findings not only demonstrate that SkillMimic can support a single policy to learn diverse skills but also underscore the importance of mixed training in enhancing skill generalization and robustness.

3. Implementation Details

3.1. Simulation Settings.

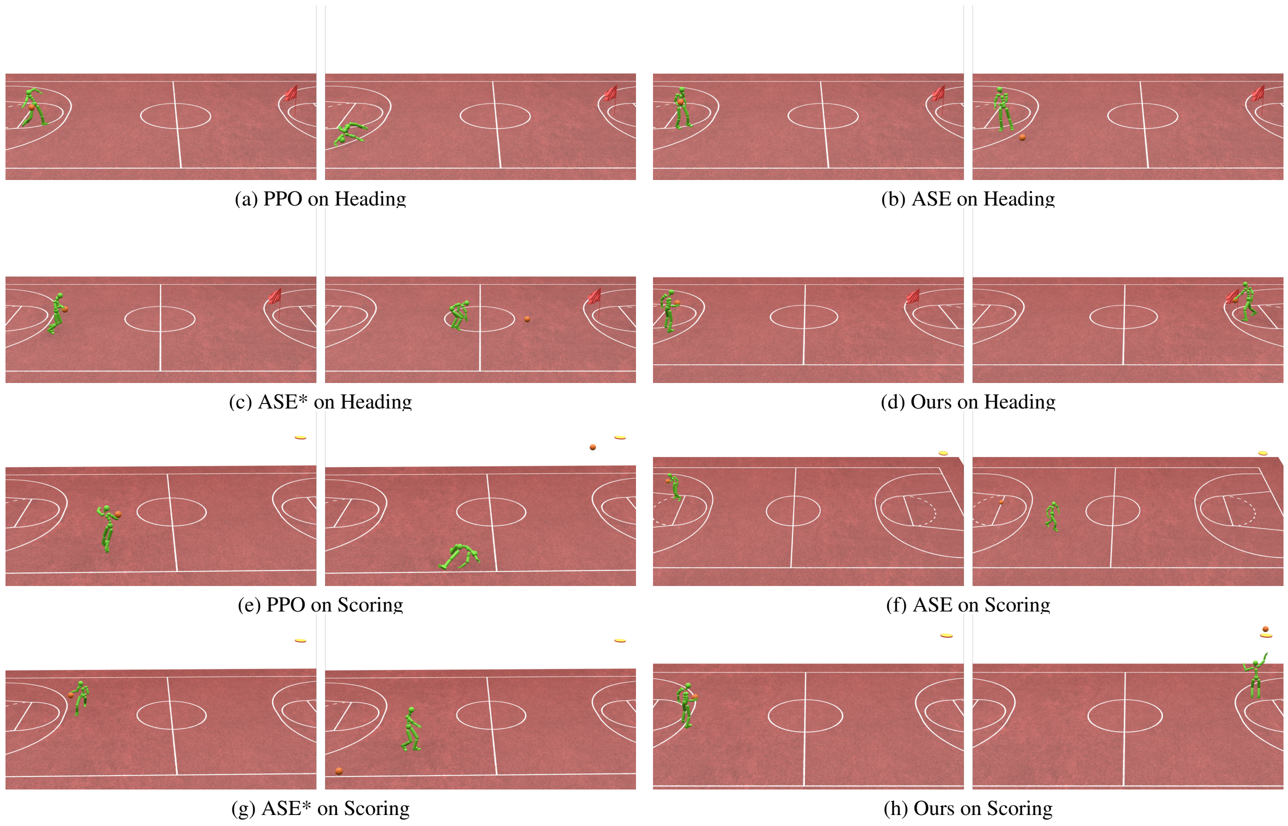
We use Isaac Gym [4] as the physics simulation platform. All experiments are trained on a single Nvidia RTX 3090 or 4090 GPU, with 2048 parallel environments. For GRAB and BallPlay-V, both the simulation and PD controller operate at 60 Hz, while the skill policy is sampled at 30 Hz. For BallPlay-M, the simulation and PD controller run at 120 Hz, with the skill policy sampled at 60 Hz. We resample the reference HOI clips to match the skill policy frequency, and the high-level policy is sampled at 20 fps. All neural networks are implemented using PyTorch and trained using Proximal Policy Optimization [10]. We use the edge set $\mathcal{E}$ of the CG and calculate the CG edge values by judging the contact force of each CG node. The setting of hyperparameters is fixed for all experiments and can be found in Sec. 3.5.

For GRAB [11] and BallPlay-V, the whole-body humanoid follows the SMPL-X [5] kinematic tree and has a total of 52 body parts and 51×3 DOF actuators where 30×3 DOF is for the hands and 21×3 DOF for the rest of the body. For BallPlay-M, the humanoid model consists



(a) DeepMimic* on Pick Up
(b) DeepMimic* on Dribble Forward
(c) AMP* on Pick Up
(d) AMP* on Dribble Forward
(e) SkillMimic on Pick Up
(f) SkillMimic on Dribble Forward

Figure 3. Comparisons on imitation learning of Interaction Skills. Both DeepMimic* and AMP* demonstrate difficulties in simultaneously managing object and body motion learning. For instance, while AMP* can roughly replicate limb movements during dribbling, it struggles to precisely control ball movement. Additionally, AMP* suffers from mode collapse, manifesting as prolonged hesitation during ball pickup attempts. In contrast, our unified HOI imitation reward successfully addresses these limitations, demonstrating superior performance in HOI imitation and establishing the first solution for purely data-driven interaction skill learning.



(a) PPO on Heading
(b) ASE on Heading
(c) ASE* on Heading
(d) Ours on Heading
(e) PPO on Scoring
(f) ASE on Scoring
(g) ASE* on Scoring
(h) Ours on Scoring

Figure 4. Comparisons on high-level basketball tasks. Left: start status. Right: end status. The high-level task rewards are extremely sparse as they only depend on object states, making convergence challenging for training from scratch (denoted as PPO). Even with locomotion skill priors, achieving convergence remains difficult (denoted as ASE). Instead, our approach first trains an IS policy through SkillMimic to acquire basic basketball interaction skills, then learns a HLC to effectively compose these interaction skills for high-level tasks. We also adapt ASE to enable direct imitation of human-object interactions in its LLC (denoted as ASE*). However, the coarse-grained nature of GAIL rewards prevents the policy from learning precise interactions, consequently hindering HLC convergence.

	Ball Radius						Ball Density						Ball Restitution					
Skill	0.5×	0.7×	0.9×	1.1×	1.3×	1.5×	0.1×	0.4×	0.7×	2×	3×	4×	0.6×	0.8×	1.2×	1.4×	1.6×	1.8×
Dribble Forward	0.0%	29.0%	84.2%	85.5%	57.2%	0.0%	0.1%	60.1%	79.5%	92.0%	33.3%	0.0%	7.0%	87.6%	87.0%	85.8%	76.1%	3.64%
Pickup	2.2%	58.7%	78.7%	79.7%	64.1%	0.4%	12.2%	78.3%	79.1%	79.1%	75.3%	17.4%	79.0%	79.6%	78.6%	79.6%	79.2%	78.9%

Table 2. Impact of varying physical properties on success rates. Models trained with fixed physical attributes were tested by scaling these attributes by a factor. The results show that the interaction skills learned by SkillMimic are robust against minor physical property changes.

	Success Rate on Individual Skills				Success Rate on Skill Switching			
Training	Dribble Forward	Dribble Left	Dribble Right	Layup	Dribble Forward-Left	Dribble Forward-Right	Dribble Forward-Layup	Dribble Left-Forward
Ind.-1×	41.3%	0.0%	81.0%	95.5%	0.0%	5.14%	8.2%	0.09%
Mixed-1×	62.8%	4.1%	48.14%	**100.0**%	1.7%	8.8%	40.5%	13.5%
Mixed-4×	**87.3**%	**67.9**%	**92.6**%	99.9%	**60.5**%	**14.5**%	**40.6**%	**46.3**%

Table 3. Success rates of skills trained independently versus jointly. Ind. denotes individual training. 1× denotes 0.65 billion training samples while 4× denotes 4 times of that. Mixed training significantly improves both individual skill execution and skill switching, demonstrating the effectiveness of SkillMimic in handling diverse interaction skills at once.

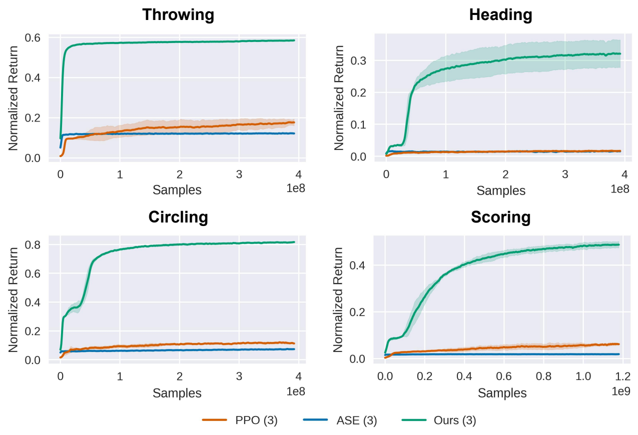


Figure 5. Learning curves of different methods on 4 high-level basketball tasks. On these challenging tasks, sparse task rewards struggle to converge when training from scratch (PPO) or using locomotion priors (ASE), whereas our method achieves rapid convergence by using interaction skill prior.

of 53 body parts and 52×3 DOF actuators, the hands having 30×3 DOF and the rest of the body having 22×3 DOF. The basketball is modeled as a sphere with a radius of 12 cm, which is close to the size of a real-world basketball. The restitution coefficients for the plane and the ball are set to 0.8 and 0.81, respectively, to ensure the ball's bounce closely resembles real-world basketball behavior. The humanoid's mass is set to match that of a real player. We set the ball's density to 1000 kg/m^3 to enhance stability and accelerate training convergence, while other physical parameters remain at their default settings. Despite being trained with fixed physical properties, our method can withstand a wide range of changes in physical properties during inference, such as variations in the ball's density, radius, and restitution, as shown in Tab. 2.

3.2. Kinematic Imitation Rewards.

Kinematic imitation rewards form the basis of the HOI imitation. We design these rewards in four distinct parts: the Body Kinematics Reward r_t^b, the Object Kinematics Reward r_t^o, the Relative Motion Reward r_t^{rel}, and a Velocity Regularization term r_t^{reg}.

The Body Kinematics Reward r_t^b encourages the alignment of the body's movements with the reference data:

$$r_t^b = r_t^p * r_t^r * r_t^{pv} * r_t^{rv}, \tag{1}$$

where r_t^p, r_t^r, r_t^{pv}, r_t^{rv} are the humanoid position reward, rotation reward, position velocity reward, and angular velocity reward. Each sub-reward is calculated by computing the Mean Squared Error (MSE) with the reference data, followed by a negative exponential normalization. For instance, the calculation for r_t^p is as follows:

$$r_t^p = \exp(-\lambda^p * e_t^p), \quad e_t^p = \text{MSE}(\boldsymbol{s}_t^p, \hat{\boldsymbol{s}}_t^p), \tag{2}$$

where $\hat{\boldsymbol{s}}_t^p$, is the reference humanoid body positions, $\boldsymbol{s}_t^p$ is the simulated humanoid body positions, λ^p is a hyperparameter that conditions the sensitivity.

The Object Kinematics Reward r_t^o ensures the object's movements are consistent with the reference:

$$r_t^o = r_t^{op} * r_t^{or} * r_t^{opv} * r_t^{orv}, \tag{3}$$

where r_t^{op}, r_t^{or}, r_t^{opv}, r_t^{orv} are the object position reward, rotation reward, position velocity reward, and angular velocity reward, respectively. The calculation of these sub-rewards resembles Eq. 2.

The relative motion is represented as a vector group, obtained by subtracting the object's position from the key body positions. The calculation of Relative Motion Reward r_t^{rel} is also similar to Eq. 2 and is effective in constraining the relative motion between the object and key body points to be consistent with the reference.

Lastly, a Velocity Regularization term is employed to suppress high-frequency jitter in the humanoid when it is supposed to be stationary:

$$r_t^{reg} = \exp(-\lambda^{reg} * e_t^{acc}), \tag{4}$$

$$e_t^{acc} = \text{mean}\left(\frac{||\boldsymbol{s}_t^{acc}||^2}{||\hat{\boldsymbol{s}}_t^{vel}||^2 + \lambda^{reg}}\right), \tag{5}$$

where λ^{reg} is a hyperparameter adjusts the sensitivity, $\boldsymbol{s}_t^{acc}$ is the simulated DOF accelerations of the humanoid, and $\hat{\boldsymbol{s}}_t^{vel}$ is the reference DOF velocities.

3.3. High-level Task Rewards

3.3.1. Throwing

In this task, the objective is to throw the ball to approach a certain height, grab the rebound, and keep on throwing the ball. The goal-related task reward can be simply defined as:

$$r_t^{throwing} = \exp(-|h_t^{ball} - 2.5|), \tag{6}$$

where h_t^{ball} is the ball height.

3.3.2. Heading

This task aims to dribble the ball to approach the target position. The task observation $\boldsymbol{h}_t$ contains the target position. We simply define the task reward as:

$$r_t^{heading} = \exp(-||\boldsymbol{x}_t^{ball} - \boldsymbol{x}_t^{target}||^2), \tag{7}$$

where $\boldsymbol{x}_t^{ball}$ is the ball position while $\boldsymbol{x}_t^{target}$ is the target position.

3.3.3. Circling

In the circling task, the objective is for the humanoid to dribble the ball around the target position following a target radius. The task observation $\boldsymbol{h}_t$ contains the target position and radius. The task reward can be defined as:

$$r_t^{circling} = r_t^v * \exp(-|d^{target} - ||\boldsymbol{x}_t^{ball} - \boldsymbol{x}_t^{center}||^2|), \tag{8}$$

where d^{target} is the target radius and $\boldsymbol{x}_t^{center}$ is the center point around which the humanoid is required to circle. r_t^v is a speed constraint that prevents the ball from staying still, defined as:

$$r_t^v = \begin{cases} 1, & \text{if} \quad ||\boldsymbol{v}_t^{ball}||^2 > 0.5 \\ 0.1, & \text{else} \end{cases} \tag{9}$$

where $\boldsymbol{v}_t^{ball}$ is the ball velocity.

3.3.4. Scoring

To further validate our method's capability to combine a diverse set of skills for precise operations, we consider the scoring task. In this task, the objective is to shot the ball precisely into a randomly positioned basket. The task observation $\boldsymbol{h}_t$ contains the basket position. The reward function consists of four parts:

$$r_t^{scoring} = r_t^v * (r_t^{heading} + r_t^{bonus} + 0.2 * r_t^{throwing}), \tag{10}$$

where $r_t^{throwing}$ rewards the ball height, $r_t^{heading}$ encourages the ball to move close to the basket, r_t^v prevents the ball from staying still, and r_t^{bonus} is a bonus for a score, defined as:

$$r_t^{bonus} = \begin{cases} 1, & \text{if scored} \\ 0, & \text{else} \end{cases} \tag{11}$$

3.4. Success Rate

To evaluate the success rate of skills, we introduced a set of skill-specific rules to determine the success or failure of skill execution:

- **Pickup**: When testing the pickup skill, we determine success by checking if the ball is lifted above 1 m after 10 seconds.
- **Dribble**: We have the humanoid dribble for 10 seconds, and if the root height of the humanoid is greater than 0.5 m and the distance between the ball and the humanoid root is less than 1.5 m, we consider the frame to be valid. The success rate is calculated as the proportion of valid frames to the total number of frames in 10 seconds.
- **Layup & Shot**: We consider a success if the distance between the ball's maximum height and the target height is less than 0.1 m, and the body root height is above 0.5 m.
- **Throwing**: We evaluate success by checking if the ball remains above 0.3 m within 10 seconds after the first throw.
- **Heading**: We determine success if the distance between the ball and the target position is less than 0.5 m.
- **Scoring**: We consider a success if the ball's maximum height is above 2.5 m, the distance between the ball and target position is less than 0.3 m, and there is no contact between the ball and the humanoid.
- **Circling**: A frame is considered valid if the distance between the ball and the target point differs from the set radius by less than 0.5 m and the ball's speed exceeds 0.5 m/s. The success rate is calculated as the proportion of valid frames to the total number of frames in the run.

3.5. Hyperparameters

The hyperparameter configurations employed during the pre-training phase of the skill policy are detailed in Tab. 4, while the hyperparameters utilized for the training of the high-level controller are presented in Tab. 5. Additionally, Tab. 6 displays the hyperparameter settings for all sub-rewards involved.

3.6. Details of Compared Methods

3.6.1. Variants in Skill Learning

Since our approach is the first to learn interaction skills from demonstration, there are no direct benchmarks for comparison. Therefore, we adapt reward strategies commonly used in locomotion imitation, i.e., DeepMimic [6] and AMP [7], to an object-inclusive setting. Specifically, we adapt our reward to the styles of DeepMimic [6] and AMP [7] while keeping the other components unchanged for a fair comparison. We denote these variant versions as DeepMimic* and AMP*. We will next delineate the specific differences in the reward functions of these variants. The sub-rewards below are consistent to that defined in Sec. 3.2. Unless specified, the hyperparameters of these sub-rewards are the same, as shown in Tab. 6.

DeepMimic* The reward function is

$$r_t = r_t^p + r_t^r + r_t^{rv} + r_t^o. \tag{12}$$

The hyperparameters of these sub-rewards are shown in Tab. 6. Unlike DeepMimic [6], we do not incorporate phase information as policy input. This decision is made because phase-based methods face several practical limitations: they cannot operate continuously when reference data fails to form a complete cycle, and they show limited resistance to interference. All methods presented in this paper deliberately avoid using phase information.

AMP* The reward function is

$$r_t = -\log\left(1 - D(\boldsymbol{s}_t, \boldsymbol{s}_{t+1})\right), \tag{13}$$

where $\boldsymbol{s}$ represents the HOI state which includes body and object state. D denotes the discriminator.

SkillMimic w/o Multiplication The reward function is

$$r_t = r_t^b + r_t^o + r_t^{rel} + r_t^{reg} + r_t^{cg}. \tag{14}$$

SkillMimic w/o CGR The reward function is

$$r_t = r_t^b * r_t^o * r_t^{rel} * r_t^{reg}. \tag{15}$$

3.6.2. Variants in High-Level Tasks

To evaluate the performance of our method on high-level tasks, we established three sets of experiments for comparison: one involving training from scratch and the other two utilizing ASE [8], which first train a Low-Level Controller (LLC) using GAIL [2] then train a high-level controller to control the LLC. For fair comparison, we construct two versions of LLC for ASE. The first LLC follows the original style of ASE [8] which learns locomotion skills without

Parameter	Value
$\dim(\boldsymbol{c})$ Skill Embedding Dimension	64
Σ_π Action Distribution Variance	0.055
Samples Per Update Iteration	65536
Policy/Value Function Minibatch Size	16384
γ Discount	0.99
Adam Stepsize	2×10^{-5}
GAE(λ)	0.95
TD(λ)	0.95
PPO Clip Threshold	0.2
T Episode Length	60

Table 4. Hyperparameters for training skill policy.

Parameter	Value
Σ_π Action Distribution Variance	0.055
Samples Per Update Iteration	65536
Policy/Value Function Minibatch Size	16384
γ Discount	0.99
Adam Stepsize	2×10^{-5}
GAE(λ)	0.95
TD(λ)	0.95
PPO Clip Threshold	0.2
T Episode Length	800

Table 5. Hyperparameters for training high-level controller.

Parameter	SM	DM*
λ^p Sensitivity of Key Body Position Error	20	20
λ^r Sensitivity of DOF Rotation Error	20	2
λ^{pv} Sensitivity of Key Body Velocity Error	0	–
λ^{rv} Sensitivity of DOF Rotation Velocity Error	0	0.1
λ^{op} Sensitivity of Object Position Error	20	20
λ^{or} Sensitivity of Object Rotation Error	0	–
λ^{opv} Sensitivity of Object Velocity Error	0	–
λ^{orv} Sensitivity of Object Angular Velocity Error	0	–
λ^{rel} Sensitivity of Relative Position Error	20	–
$\boldsymbol{\lambda}^{\boldsymbol{cg}}[0]$ Sensitivity of Ball-Hands Contact Error	5	–
$\boldsymbol{\lambda}^{\boldsymbol{cg}}[1]$ Sensitivity of Ball-Body Contact Error	5	–
$\boldsymbol{\lambda}^{\boldsymbol{cg}}[1]$ Sensitivity of Body-Hands Contact Error	5	–
λ^{reg} Sensitivity of Velocity Regularization	10^{-12}	–

Table 6. Hyperparameters of Sub-Rewards. SM denotes SkillMimic, and DM* denotes SkillMimic with DeepMimic-style rewards.

considering object motion. We denote this method as ASE in our experiment. The second LLC follows the style of SkillMimic which learns interaction skills and considers object motion. We denote this object-inclusive LLC as ASE* in our experiment. The reward of both LLCs share the same formulation:

$$r_t = -\log\left(1 - D(\boldsymbol{s}_t, \boldsymbol{s}_{t+1})\right) + \beta \log q\left(\boldsymbol{z}_t | \boldsymbol{s}_t, \boldsymbol{s}_{t+1}\right), \tag{16}$$

The difference is in the representations of $\boldsymbol{s}$. For ASE*, $\boldsymbol{s}$ contains the body and object state. For ASE, $\boldsymbol{s}$ represents the body-only state (object is not considered). D denotes the discriminator, q denotes the encoder and $\boldsymbol{z}$ represents the latent code.

Subsequently, we trained High-Level Controllers (HLC) to reuse these pre-trained LLCs for high-level tasks. For fair comparison, the task rewards are the same and the network size of the HLC is identical to that used in our approach. The HLC of ASE and ASE* outputs continuous latent variables, whereas our HLC outputs discrete skill conditions.

References

[1] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 1

[2] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016. 6

[3] Jing Lin, Ailing Zeng, Shunlin Lu, Yuanhao Cai, Ruimao Zhang, Haoqian Wang, and Lei Zhang. Motion-x: A large-scale 3d expressive whole-body human motion dataset. *Advances in Neural Information Processing Systems*, 2023. 1

[4] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance GPU based physics simulation for robot learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 2

[5] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019. 1, 2

[6] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. Deepmimic. *ACM Transactions on Graphics*, page 1–14, 2018. 6

[7] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics*, page 1–20, 2021. 6

[8] Xue Bin Peng, Yunrong Guo, Lina Halper, Sergexuey Levine, and Sanja Fidler. Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Trans. Graph.*, 41(4), 2022. 6

[9] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 1

[10] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 2

[11] Omid Taheri, Nima Ghorbani, Michael J. Black, and Dimitrios Tzionas. GRAB: A dataset of whole-body human grasping of objects. In *European Conference on Computer Vision (ECCV)*, 2020. 2