SkillMimic: Learning Reusable Basketball Skills from Demonstrations

YINHUAI WANG*, The Hong Kong University of Science and Technology, China and Unitree Robotics, China

QIHAN ZHAO*, The Hong Kong University of Science and Technology, China

RUNYI YU*, The Hong Kong University of Science and Technology, China and Unitree Robotics, China

AILING ZENG†, Tencent, China

JING LIN, Tsinghua Shenzhen International Graduate School, China

ZHENGYI LUO, Carnegie Mellon University, USA

HOK WAI TSUI, The Hong Kong University of Science and Technology, China

JIWEN YU, Peking University Shenzhen Graduate School, China

XIU LI, Tsinghua Shenzhen International Graduate School, China

QIFENG CHEN, The Hong Kong University of Science and Technology, China

JIAN ZHANG†, Peking University Shenzhen Graduate School, China

LEI ZHANG, International Digital Economy Academy, China

PING TAN, The Hong Kong University of Science and Technology, China
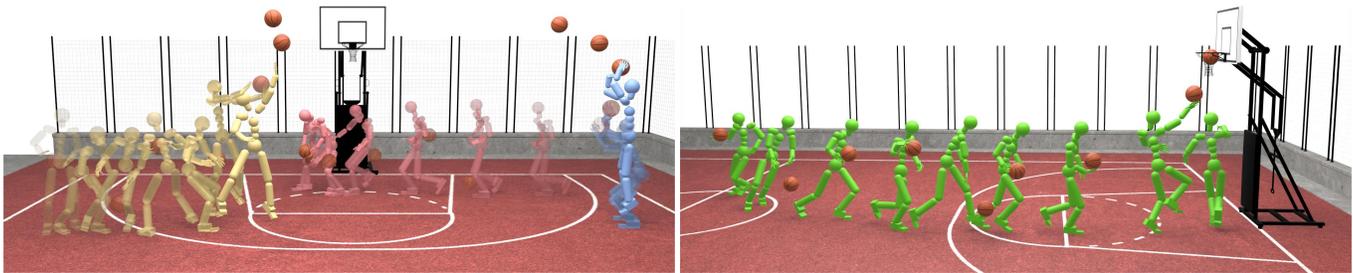
Fig. 1. We propose a novel approach that enables physically simulated humanoids to learn a variety of basketball skills from human-object demonstrations, such as shooting (blue), retrieving (red), and turnaround layup (yellow). Once acquired, these skills can be reused and combined to accomplish complex tasks, such as continuous scoring (green), which involves dribbling toward the basket, timing the dribble and layup to score, retrieving the rebound, and repeating.

*Equal contribution.

†Corresponding author.

Authors' addresses: Yinhuai Wang, The Hong Kong University of Science and Technology, China and Unitree Robotics, China, yinhuai.wang@connect.ust.hk; Qihan Zhao, The Hong Kong University of Science and Technology, China, qihan.zhao@outlook.com; Runyi Yu, The Hong Kong University of Science and Technology, China and Unitree Robotics, China, ingrid.yu@connect.ust.hk; Ailing Zeng, Tencent, China, ailingzengzzz@gmail.com; Jing Lin, Tsinghua Shenzhen International Graduate School, China, jinglin.stu@gmail.com; Zhengyi Luo, Carnegie Mellon University, USA, zluo2@cs.cmu.edu; Hok Wai Tsui, The Hong Kong University of Science and Technology, China, hwtsui@connect.ust.hk; Jiwen Yu, Peking University Shenzhen Graduate School, China, yujiwen@stu.pku.edu.cn; Xiu Li, Tsinghua Shenzhen International Graduate School, China, lixiu@mails.tsinghua.edu.cn; Qifeng Chen, The Hong Kong University of Science and Technology, China, cqf@ust.hk; Jian Zhang, Peking University Shenzhen Graduate School, China, zhangjian.sz@pku.edu.cn; Lei Zhang, International Digital Economy Academy, China, leizhang@idea.edu.cn; Ping Tan, The Hong Kong University of Science and Technology, China, pingtan@ust.hk.

Mastering basketball skills such as diverse layups and dribbling involves complex interactions with the ball and requires real-time adjustments. Traditional reinforcement learning methods for interaction skills rely on labor-intensive, manually designed rewards that do not generalize well across different skills. Inspired by how humans learn from demonstrations, we propose SkillMimic, a data-driven approach that mimics both human and ball motions to learn a wide variety of basketball skills. SkillMimic employs a unified configuration to learn diverse skills from human-ball motion datasets, with skill diversity and generalization improving as the dataset grows. This approach allows training a single policy to learn multiple skills, enabling smooth skill switching even if these switches are not present in the reference dataset. The skills acquired by SkillMimic can be easily reused by a high-level controller to accomplish complex basketball tasks. To evaluate our approach, we introduce two basketball datasets: one estimated through monocular RGB videos and the other using advanced motion capture equipment, collectively containing about 35 minutes of diverse basketball skills. Experiments show that our method can effectively learn various basketball skills included in the dataset with a unified configuration, including various styles of dribbling, layups, and shooting. Furthermore, by training a high-level controller to reuse the acquired skills, we can achieve complex basketball tasks such as layup scoring, which involves dribbling toward the basket, timing the dribble and layup to score, retrieving the rebound, and repeating the process. Project page and video demonstrations: https://ingrid789.github.io/SkillMimic/

1 INTRODUCTION

Basketball, as a collection of highly dynamic, complex, diverse, and precise object-interaction skills, exemplifies the pinnacle of human athleticism and control capabilities. Professional basketball players spend extensive time practicing fundamental skills such as shooting and dribbling, which they flexibly utilize to score during games. These basketball skills typically involve not only isolated body movements but also interactions with the ball, requiring real-time adjustments based on the state of the object to achieve precise control. Current state-of-the-art methods apply reinforcement learning to mimic isolated body movements, combining these motion priors with manually designed skill rewards to learn interaction skills, such as striking pillars [Peng et al. 2022], playing tennis [Zhang et al. 2023b], climbing ropes [Bae et al. 2023], and carrying boxes [Hassan et al. 2023]. However, designing these skill rewards is labor-intensive and difficult to generalize across different skills. Given the complexity and diversity of basketball skills, creating skill rewards for each basketball skill demands significant engineering effort. Consequently, existing methods fail to provide a unified learning framework for a single policy to learn diverse basketball skills, let alone the challenge of reusing and composing these skills to accomplish more complex tasks, such as continuous scoring.

How can a simulated humanoid learn a wide variety of reusable basketball skills in a simple and scalable manner? We can draw inspiration from the way humans learn basketball skills from demonstrations. Most basketball players, even without formal coaching, can master various skills by watching basketball videos and practicing diligently. During this learning process, individuals adjust their body movements to align with both the reference body motions and the reference ball movements they observe. This paradigm of learning skills from demonstrations allows a basketball player to acquire a set of reusable basketball skills and flexibly combine them for further purposes.

Inspired by these observations, we propose a data-driven method called SkillMimic, which simultaneously mimics human and object motions and their contacts to learn interaction skills. SkillMimic can learn various basketball skills purely from human-ball motion datasets, such as diverse styles of shooting, layups, and dribbling skills, and even a robust pickup skill that enables picking up balls in random locations and motions. Notably, SkillMimic uses the exact same configuration to learn different skills, with identical hyperparameters. This allows us to train a single policy to learn multiple skills and achieve smooth skill switching, even if these switches are not present in the reference dataset. By training a high-level policy to reuse these learned skills, we can accomplish challenging high-level tasks such as scoring, which requires the humanoid to composite diverse basketball skills to score accurately.

To evaluate our method, we introduce two basketball datasets. The first dataset, BallPlay-V, uses a purely visual approach, employing neural networks to estimate the motion of players and basketballs from RGB videos, and includes eight basic basketball skills. The second dataset, BallPlay-M, utilizes advanced optical motion capture equipment to record the movements of both players and balls, encompassing approximately 35 minutes of comprehensive basketball skills. Experiments on BallPlay-V demonstrate that our method exhibits robust performance against data inaccuracies. In BallPlay-M, our system, SkillMimic, successfully learns various reusable basketball skills within a single policy using a unified configuration. By additionally training a high-level controller to flexibly compose the skills acquired by SkillMimic, we can efficiently complete complex tasks such as directional dribbling and layup scoring. Compared to previous methods, our approach offers significant advantages in simplicity and efficiency in learning basketball skills and complex basketball tasks.

Specifically, our contributions are as follows:

- **SkillMimic**: A data-driven paradigm for learning reusable interaction skills, capable of learning diverse basketball skills in a unified manner. It supports joint learning and smooth switching of skills, with skill diversity and generalization improving as the dataset grows.
- **Contact Graph**: We propose a simple and general contact modeling method that applies to diverse skills, called the contact graph. A Contact Graph Reward (CGR) is designed to enable precise contact imitation, which proves to be critical for learning precise interaction skills.
- **Unified skill imitation reward**: We propose a set of important designs that form a unified reward configuration for imitation learning of various interaction skills.
- **A hierarchical solution for learning complex basketball tasks**: We propose training a high-level controller to flexibly compose the skills acquired by SkillMimic to accomplish challenging high-level tasks.
- **BallPlay Datasets**: We introduce two basketball datasets to facilitate research on basketball skill learning.

2 RELATED WORK

2.1 Human Motion Imitation and Skill Learning

Mimicking human motion for robot control is an efficient method of learning humanoid skills. DeepMimic [Peng et al. 2018], a pioneer in this field, uses imitation learning to perform a variety of highly dynamic skills, including backflips, cartwheels, and running. The introduction of Generative Adversarial Imitation Learning (GAIL) [Ho and Ermon 2016] into humanoid imitation learning by AMP [Peng et al. 2021] lessens constraints on data alignment, thus enhancing its versatility. ASE [Peng et al. 2022] further amplifies motion diversity in GAIL training, incorporates a pre-trained low-level policy to acquire locomotion skills, and deploys a high-level policy to repurpose these locomotion skills for specific tasks, such as striking pillars at random locations. This methodology is subsequently expanded in numerous efforts to augment condition and text control [Dou

et al. 2023; Juravsky et al. 2022; Pan et al. 2024; Ren et al. 2024b; Sun et al. 2023; Tessler et al. 2023; Zhu et al. 2023], as well as to design automated rewards [Cui et al. 2024]. This collection of studies employs imitation learning to broaden the scope of data-driven skill acquisition, enabling the control and reuse of these learned skills for high-level tasks. However, a limitation of these studies is their exclusive focus on learning isolated locomotion skills through imitation. This requires additional pipeline and reward design for learning interaction skills, even for relatively simple ones, such as moving boxes [Hassan et al. 2023] or striking pillars [Peng et al. 2022]. In contrast, our approach achieves unified learning of diverse basketball interaction skills without the need for skill-specific reward design and hyperparameter tuning.

2.2 Generating Human-Object Interactions

Owing to the complex dynamics involved in diverse interactions, the generation of interaction for simulated humanoids has long posed challenges in the fields of graphics and robotics. Early studies often relied on manually designed control structures. For example, Hodgins et al. [Hodgins et al. 1995] implement a variety of sports interactions, including running, bicycling, and vaulting, through the use of manually designed state machine controllers. Yin et al. [Yin et al. 2007] use finite-state machines to generate a large variety of gaits. Coros et al. [Coros et al. 2010] utilize an inverted pendulum model to control foot placement, thereby achieving a variety of gait controls. With the advancement of machine learning, deep reinforcement learning [Arulkumaran et al. 2017] has been widely employed in character control, giving rise to an array of methods based on policy network control. These methods have been applied to a range of challenging activities, such as basketball [Liu and Hodgins 2018], skateboarding [Liu and Hodgins 2017], and cycling [Tan et al. 2014], etc. The integration of imitation learning further simplifies the learning of interactions, and this data-driven learning paradigm significantly lessens the need for intricate expert design. Zhang et al. [Zhang et al. 2023b] propose a framework to learn diverse tennis skills from broadcast videos. Bae et al. [Bae et al. 2023] incorporate multiple part-wise motion priors for diverse whole-body interactions, such as climbing ropes and weightlifting. Similarly, Braun et al. [Braun et al. 2023] develop a framework for synthesizing whole-body motions for dexterous grasps.

A closely related work is [Zhang et al. 2023a], where an interaction graph [Ho et al. 2010] that represents the relative motions is proposed to learn the simulated multi-character interaction. However, its kinematic-only rewards fail to measure correct contacts and are unstable in learning diverse interactions. Instead, our proposed contact-aware imitation rewards prioritize contact learning and significantly improve the success rate. Additionally, a policy trained by [Zhang et al. 2023a] is limited to strictly mimicking a single reference clip during testing, which is insufficient for it to be considered a skill. In contrast, our method can train on multiple clips and learn generalizable and controllable skills beyond mere imitation. Another highly related work is [Braun et al. 2023], a framework for physically plausible whole-body grasping. Similar to previous kinematics-based grasping synthesis methods [Ghosh et al. 2023; Wu et al. 2022], they design skill-specific contact rewards as guidance for learning grasp skills. However, the method

contains multiple networks and training rounds with pre-trained modules dedicated to the grasp skills. Beyond grasp, our method covers diverse basketball interaction skills, within a simple unified framework, and without designing skill-specific rewards.

In addition to physics-based methods, a multitude of interaction generation approaches rooted in kinematics are flourishing [Jiang et al. 2023, 2024; Li et al. 2023a,b; Starke et al. 2020, 2021; Xu et al. 2023, 2024]. These methods typically use networks to predict HOI states without incorporating physical simulations. Consequently, the generated interactions often lack physical authenticity and demand large volumes of data.

3 PRELIMINARIES ON REINFORCEMENT LEARNING

Our method is based on reinforcement learning, where the agent interacts with the environment according to a policy to maximize the expected return. The basic task is formulated as a Markov Decision Process (MDP) defined by the tuple $\mathcal{M} = \{s, a, f, r, \gamma\}$ of states, actions, transition dynamics, reward function, and discount factor. At each time step t, the agent takes the system states s_t as input and outputs an action a_t by sampling the policy distribution $\pi(a_t|s_t)$. The action a_t will result in a new state s_{t+1} through physical simulation $f(s_{t+1}|a_t, s_t)$. Then a reward $r_t = r(s_t, a_t, s_{t+1})$ is calculated, and the goal is to learn a policy that maximizes the expected return $\mathcal{R}(\pi) = \mathbb{E}_{p_\pi(\tau)} \left[\sum_{t=0}^{T-1} \gamma^t r_t \right]$, where $\tau = \{s_0, a_0, r_0, ..., s_{T-1}, a_{T-1}, r_{T-1}, s_T\}$ represents the trajectory, $p_\pi(\tau)$ is the probability density function of the trajectory. T denotes the time horizon, and $\gamma \in [0, 1]$ is a discount factor. To enable skill control, we optionally provide the policy with extra conditions c, yielding a variant policy $\pi(a_t|s_t, c)$.

4 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.

4.1 BallPlay-V

The BallPlay-V dataset applies a monocular annotation solution to estimate the high-quality human SMPL-X [Pavlakos et al. 2019] 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 [Lin et al. 2023], our automatic annotation additionally introduces depth estimation [Bhat et al. 2023], semantic segmentation [Ren et al. 2024a], to obtain high-quality whole-body human motions and ball motions. The BallPlay-V dataset contains **eight** basketball skills, including *back dribble, cross leg, hold, fingertip spin, pass, backspin, cross*, and *rebound*.

4.2 BallPlay-M

Although BallPlay-V can acquire HOI data conveniently from RGB videos, its accuracy is limited due to significant depth estimation errors. Additionally, it struggles with occlusion issues, making it difficult to capture complex layup and dribbling data. To achieve

Fig. 2. Concept of SkillMimic. We define a skill as a specific set of Human-Object Interaction (HOI) state transitions that align with the intended skill semantics. These state transitions can be derived from capturing real-world skills into HOI motion clips. If a simulated humanoid can manipulate objects such that the resulting HOI state transitions closely match those of the reference, we consider the humanoid to have successfully learned the skill.

more comprehensive and accurate basketball data, we create the BallPlay-M dataset using optical motion capture equipment. This dataset comprises 35 minutes of diverse human-basketball interactions, including 39 dribble clips, 131 pickup clips, 8 layup clips, 13 shot clips, 10 pass clips, 16 getup clips, and 21 miscellaneous clips. 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.

5 SKILLMIMIC

In this section, we introduce a unified data-driven framework for simulated humanoids to learn various reusable basketball skills. We begin by introducing how we define skills with Human-Object Interaction (HOI) data (Sec. 5.1). Next, we explain how to learn diverse skills by imitating HOI data (Sec. 5.2) and discuss the designing of a unified HOI imitation reward (Sec. 5.3). Finally, we demonstrate how the learned skills can be reused to perform complex tasks (Sec. 5.4).

5.1 Defining Skills Using Data

We propose a data-driven approach to defining skills as collections of Human-Object Interaction (HOI) state transitions that align with the intended skill semantics. For instance, the skill of "picking up balls from various positions" can be encapsulated by various HOI motion clips, each capturing a unique instance of the action. If we consider each frame of HOI data as a state, these clips collectively form a collection of state transitions that represent the skill.

Unlike defining skills using explicit rules, our approach uses data to represent skills. This makes it easier to define skills that are difficult to describe with rules, such as various basketball skills. In addition, by collecting richer skill data samples, the skill coverage can be more completely represented. For example, a pickup skill defined by 100 different HOI motion clips will be more representative than one defined by a single clip.

5.2 Learning Skills by Imitation

Considering skills defined by a set of reference HOI state transitions, if a humanoid can manipulate objects such that its HOI state transitions closely resemble those of the reference, we consider the humanoid to have successfully learned the skill. Based on this concept, we propose a method for learning interaction skills by imitating HOI state transitions, which we call SkillMimic. Fig. 2 illustrates the concept of SkillMimic. Unlike previous approaches [Peng et al. 2018, 2022, 2021; Zhang et al. 2023b] that require manually designed skill rewards for each interaction skill, SkillMimic is fully data-driven, skill-agnostic, and scalable, making it capable of learning a wide range of basketball skills within a unified solution. Fig. 6 shows diverse basketball skills acquired using SkillMimic. Fig. 8 and Fig. 9 illustrate how the performance of the pickup skill improves as the amount of reference data increases.

Thanks to the exploratory nature of deep reinforcement learning, the skills learned through SkillMimic can quickly switch from minor out-of-domain states to learned state distributions, leading to skill generalization and zero-shot skill switching.

5.2.1 Training. Fig. 3 (b) shows the training pipeline of SkillMimic. Given an HOI dataset with diverse skill-labeled clips (Fig. 3 (a)), SkillMimic trains by randomly selecting a frame from a clip corresponding to a chosen skill to initialize the humanoid and object states. The state s_t (Sec. 5.2.2) and skill label c_j are input into the policy, which predicts actions (Sec. 5.2.3) that are then simulated to produce the next state s_{t+1}. A unified HOI imitation reward (Sec. 5.3) is designed to measure the consistency between the simulated state transitions and the reference transitions. When the simulation triggers specific conditions, such as falling or reaching the maximum simulation length, the environment resets, and the process repeats. After training converges, the policy enables the humanoid to execute skills that closely mirror the reference demonstrations. Moreover, since it does not rely on reference data during testing, skills can operate continuously with robustness and generalization capabilities.

Our implementation is based on Isaac Gym [Makoviychuk et al. 2021], which supports the parallel execution of numerous environments. To quickly adapt to the diverse states encapsulated in the dataset, we apply the reference state initialization (RSI) [Peng et al. 2018] independently for each environment. Since clips vary in length, the rewards at convergence can differ significantly, potentially causing imbalanced skill learning. To ensure more balanced skill learning, we set a fixed maximum simulation length, making the upper limit of imitation rewards similar across all skills.

5.2.2 Observation. The state observed by the skill policy is not under direct supervision and may theoretically encompass any information that is available in the simulation environment. Importantly, this information does not necessarily have to be included in the reference HOI data. Similar to prior arts [Peng et al. 2022], we transform all coordinates into the root local coordinate of the humanoid, which aligns the data distribution and benefits the generalization performance. For the humanoid, we observe its global root height, local body position, rotation, position velocity, and angular velocity. These representations form the humanoid proprioception o_t^{prop}. In addition, we detect net contact forces o_t^f for all fingertips, which
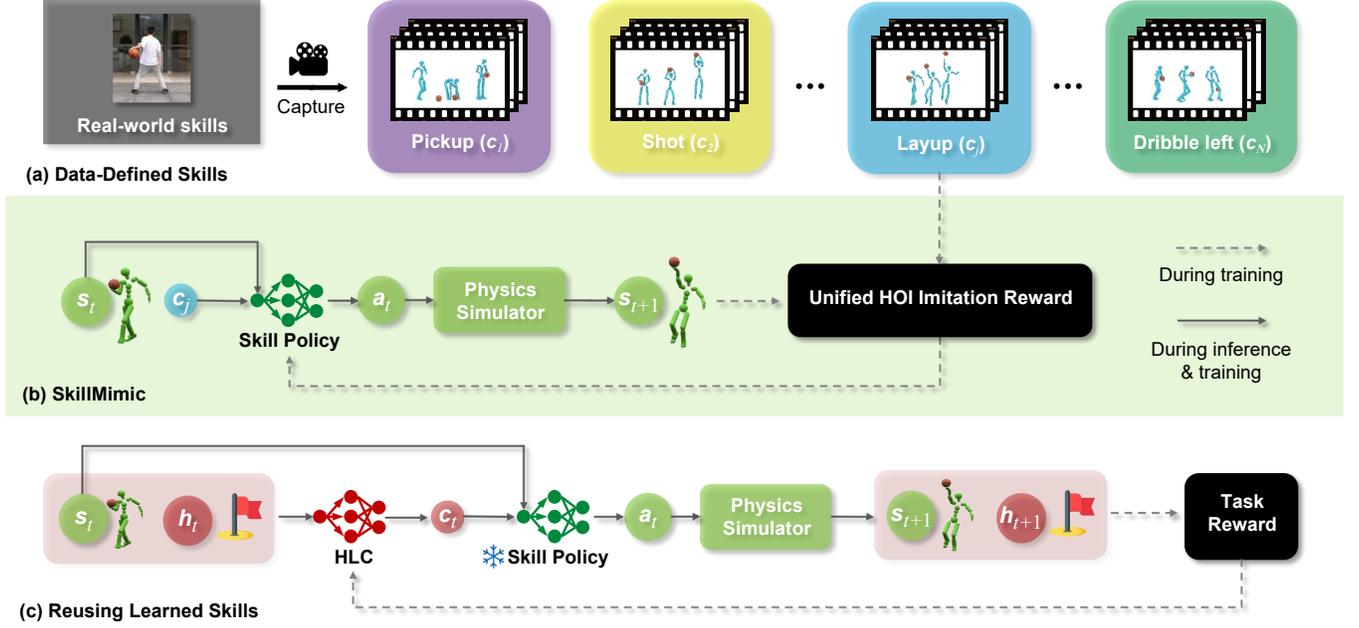
Fig. 3. Our system consists of three parts. (a) First, we capture real-world basketball skills to create a large Human-Object Interaction (HOI) motion dataset. (b) Second, we train a skill policy to learn interaction skills by imitating the corresponding HOI data. Specifically, the policy takes as input the HOI state s_t and skill label c_j and predicts the action a_t. The new state s_{t+1} is calculated by the simulator. A unified HOI imitation reward is designed to imitate diverse HOI state transitions. (c) The third part involves training a High-Level Controller (HLC) to reuse the learned skills for complex tasks. The HLC takes as input s_t and extra task observations h_t, e.g., the basket position, and predicts the skill label c_t to drives a pre-trained skill policy.

helps to sense contact and accelerate training. For the object, we observe its local position, rotation, velocity, and angular velocity, which form the object observation o_t^{obj}. Finally, the state perceived by the skill policy is

$$s_t = \{o_t^{prop}, o_t^f, o_t^{obj}\}. \tag{1}$$

Skill labels c_j are also used as inputs to the policy to differentiate between various skills, which is crucial for skill reuse and switching. We use one-hot encodings to represent the skill label.

5.2.3 Policy and Action. The policy output is modeled as a Gaussian distribution with dimensions equal to the DOF number of the humanoid robot, featuring constant variance. The mean is modeled by a three-layer MLP consisting of [1024, 512, 512] units with ReLU activations. We use the action a_t sampled from the policy as the target joint rotations for a full set of PD controllers. The PD controllers adjust and output the joint torques to reach the target rotations.

5.2.4 Physical Simulation of the Humanoid, Ball, and Environment. We use Isaac Gym [Makoviychuk et al. 2021] as the physics simulation platform. For GRAB [Taheri et al. 2020] and BallPlay-V, the whole-body humanoid follows the SMPL-X [Pavlakos et al. 2019] 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 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/m³ 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.

5.3 Unified HOI Imitation Reward

The design of the unified HOI imitation reward in Fig. 3 (b) is crucial to the performance of learning interaction skills. This reward function is pivotal as it must not only accurately gauge the efficacy of skill acquisition but also avoid the incorporation of skill-specific settings to maintain robust cross-skill generalizability.

Traditional imitation rewards often focus solely on imitating human motions [Peng et al. 2018, 2022, 2021] or neglect interactive contacts [Zhang et al. 2023a], leading to suboptimal imitation performance and necessitating individualized hyperparameter tuning. Recognizing the interlocking relationship between interaction and contact, we propose a straightforward and universal contact modeling approach called Contact Graph. A corresponding CG imitation reward r_t^{cg} is designed to enhance the precision of interaction imitation. Beyond the standard body motion imitation reward r_t^b and object motion imitation reward r_t^o, we incorporate a relative

position reward r_t^{rel} to further improve interaction accuracy. Additionally, we propose an adaptive velocity regularization term r_t^{reg} to suppress high-frequency jitters. These sub-rewards are multiplied to encourage balanced learning, thereby avoiding local optima, as justified in Tab. 1. The combination of these innovations forms a unified reward function, enabling the learning of diverse basketball skills in BallPlay-M with the same configuration. To conclude, the complete HOI imitation reward is

$$r_t = r_t^b * r_t^o * r_t^{rel} * r_t^{reg} * r_t^{cg}. \tag{2}$$

Next, we will explore the methodology involved in crafting these reward functions.

5.3.1 Contact Graph. We observe that existing imitation rewards [Peng et al. 2022, 2021; Zhang et al. 2023a] only measure kinematic properties and are insufficient to measure precise contacts. To tackle this problem, we propose the Contact Graph (CG) to model the contact in general interactions and design a CG imitation reward.

As illustrated in Fig. 4, the CG is a complete graph where every pair of distinct nodes is connected by a unique edge, defined as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} \in \{0, 1\}^k$ is the set of k nodes and $\mathcal{E} \in \{0, 1\}^{k(k-1)/2}$ is the set of edges. A CG node stores a binary value indicating whether it contacts other nodes. Each CG edge stores a binary label that denotes the contact between two nodes, where 1 represents contact, and 0 means no contact. CG node and edge values are calculated frame by frame, explicitly describing the mutual contact relationship between CG nodes at different moments. Contacts inside a node are not considered. In practice, we can use the nodes set $\mathcal{V}$ or edge set $\mathcal{E}$ depending on the needs.

The definitions of nodes are flexible. For example, a node can be a single part (e.g., a fingertip), or an aggregation of multiple parts (e.g., the whole left hand). Hinged objects can also be broken up into finer nodes [Fan et al. 2023; Geng et al. 2023; Zhang et al. 2024]. The definition of CG is unified for a certain scene and shared between diverse interaction skills. For example, in basketball scenarios, we can aggregate two hands as a node, the rest of the bodies as a node, and the ball as a node, as illustrated in Fig. 4. This simple CG is effective enough for learning all the basketball skills covered in BallPlay-M. Fig. 6 shows a subset of these skills.

Theoretically, the reference CG edge values can be extracted from accurate kinematic HOI data by calculating the mesh collision or distances. A CG node value can be yielded by the logic "OR" operation on all its connected edges.

5.3.2 Contact Graph Reward. Kinematic imitation rewards [Peng et al. 2018, 2021; Zhang et al. 2023a] fall short in measuring contacts and thus yield poor performance in HOI imitation. For example, during the toss skill training (Fig. 5 (a)), the contact between the right hand and the ball will in most cases cause the ball to drop, making the kinematic rewards drastically smaller. In this case, the policy may learn a local-optimal solution that uses the head and left hands to stabilize the ball, as demonstrated in Fig. 5 (b). To tackle this problem, we propose the Contact Graph Reward (CGR) as a critical complement of kinematic imitation rewards to learn precise contact imitation. The CG error is defined as $e_t^{cg} = |s_t^{cg} - \hat{s}_t^{cg}|$, where $|\cdot|$ calculates element-wise absolute value, s_t^{cg} and $\hat{s}_t^{cg}$ denotes the
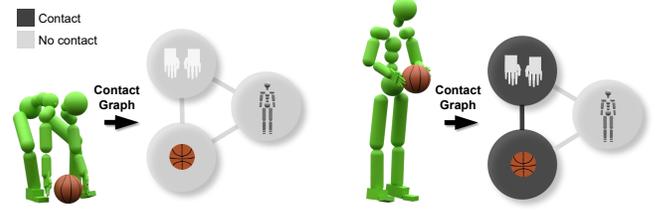
Fig. 4. We propose the Contact Graph (CG) to model general contacts within an explicitly defined scene. The node stores a binary value that denotes whether it contacts other nodes. Each edge stores a binary value indicating whether the two connected nodes are in contact. The node definition is unified for a certain scene and shared between diverse interactive skills. For example, we define three nodes: hands, hands-exclusive body, and ball, to form a simple CG to model contacts for diverse basketball skills.
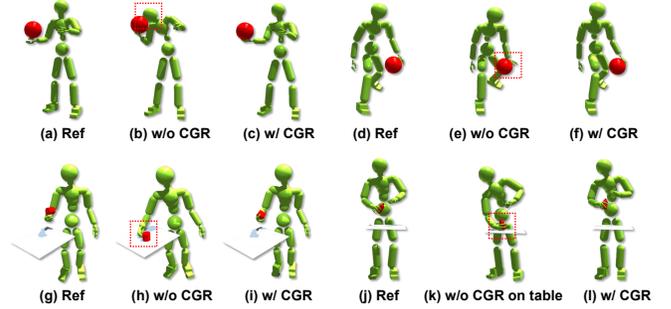
Fig. 5. The HOI imitation falls into kinematic local-optimal solutions without Contact Graph Reward (CGR): (**b**) use the head to help control the ball; (**e**) use the wrist to contact the ball; (**h**) fail to catch the object; (**k**) support the table to keep balance. In comparison, the guidance of CGR effectively yields precise interactions, as shown in (**c, f, i, l**).

simulated and reference CG state respectively, which can be the nodes set $\mathcal{V}_t \in \{0, 1\}^k$ or the edge set $\mathcal{E}_t \in \{0, 1\}^{k(k-1)/2}$ depending on the needs. The CGR is measured by CG error, with independent weights on different elements:

$$r_t^{cg} = \exp(- \sum_{j=1}^{J} \lambda^{cg}[j] * e_t^{cg}[j]), \tag{3}$$

where $e_t^{cg}[j]$ is the jth element of $e_t^{cg} \in \{0, 1\}^J$, a binary label representing a contact error; $\lambda^{cg}[j]$ is the jth element of $\lambda^{cg} \in \mathbb{R}^J$, a hyperparameter controls the sensitivity of a contact. Fig. 5 and Tab. 1 justify the effectiveness of CGR, where experiments without CGR show kinematic local-optimal solutions while using CGR effectively eliminates such problems.

5.3.3 Kinematic Rewards. Kinematic imitation rewards form the core 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:

$$r_t^b = r_t^p * r_t^r * r_t^{pv} * r_t^{rv}, \qquad (4)$$

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:

$$r_t^p = \exp(-\lambda^p * e_t^p), \quad e_t^p = \mathrm{MSE}(s_t^p, \hat{s}_t^p), \qquad (5)$$

where $\hat{s}_t^p$, is the reference humanoid body positions, 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:

$$r_t^o = r_t^{op} * r_t^{or} * r_t^{opv} * r_t^{orv}, \qquad (6)$$

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. 5.

The relative motion is represented as a vector group, obtained by subtracting the object's position from the key body positions. The Relative Motion Reward r_t^{rel} is also calculated following Eq. 5 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:

$$r_t^{reg} = \exp(-\lambda^{reg} * e_t^{acc}), \quad e_t^{acc} = \mathrm{mean}\left(\frac{||s_t^{acc}||^2}{||\hat{s}_t^{vel}||^2 + \lambda^{reg}}\right), \quad (7)$$

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

5.4 Reusing Acquired Skills for Complex Tasks

Although the definition of skills can be quite broad, employing more complex data to define more intricate skills introduces additional demands for data collection. Instead, we consider leveraging the skills acquired through SkillMimic to accomplish long-term, complex tasks. Specifically, we propose training a High-Level Controller (HLC) to reuse acquired skills for implementing complex tasks. As depicted in Fig. 3 (c), the HLC takes as input the current HOI state s_t and task observation h_t, and outputs a discrete skill embedding c_t that serves as inputs to a pre-trained SkillMimic policy, which subsequently generates actions for the humanoid. Compared to previous hierarchical approaches that train a High-Level Controller (HLC) to reuse motion priors [Peng et al. 2022; Zhang et al. 2023b], our method does not require rewards designed for interactions. Instead, simple goal-related rewards can be highly effective. This significantly simplifies the design of task rewards and greatly accelerates the convergence speed of training. In the following sections, we present four representative complex tasks and our detailed solutions. Results in Tab. 4 and Fig. 11 justify the superiority of our method.

5.4.1 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:

$$r_t^{throwing} = \exp(-|h_t^{ball} - 2.5|), \qquad (8)$$

where h_t^{ball} is the ball height.

5.4.2 Heading. This task aims to dribble the ball to approach the target position. The task observation h_t contains the target position. We simply define the task reward as:

$$r_t^{heading} = \exp(-||x_t^{ball} - x_t^{target}||^2), \qquad (9)$$

where x_t^{ball} is the ball position while x_t^{target} is the target position.

5.4.3 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 h_t contains the target position and radius. The task reward can be defined as:

$$r_t^{circling} = r_t^v * \exp(-|d^{target} - ||x_t^{ball} - x_t^{center}||^2|), \qquad (10)$$

where d^{target} is the target radius and 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:

$$r_t^v = \begin{cases} 1, & \text{if } ||v_t^{ball}||^2 > 0.5 \\ 0.1, & \text{else} \end{cases} \qquad (11)$$

where v_t^{ball} is the ball velocity.

5.4.4 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 h_t contains the basket position. The reward function consists of four parts:

$$r_t^{scoring} = r_t^v * (r_t^{heading} + r_t^{bonus} + 0.2 * r_t^{throwing}), \qquad (12)$$

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:

$$r_t^{bonus} = \begin{cases} 1, & \text{if scored} \\ 0, & \text{else} \end{cases} \qquad (13)$$

After only three hours of training (on a Nvidia RTX 4090 GPU), the HLC is capable of controlling a humanoid to dribble the ball towards the basket, execute layups at the right moments to score, retrieve the rebound, dribble away, and repeat the scoring sequence.

6 EXPERIMENT

Our experiments are divided into two main parts. The first part evaluates skill learning (Sec. 6.3). This includes comparisons with existing methods, robustness testing of the acquired skills, and ablation studies on the reward function designs and data volume. The second part assesses learning complex tasks by reusing acquired skills (Sec. 6.4), including comparisons with existing methods and a demo of a two-player competition.

(a) Jump Shot	(b) Rebound	(c) Layup	(d) Turnaround Layup
(e) Dribble Left	(f) Dribble Right	(g) Dribble Forward	(h) Dribble Turn
(i) Pass	(j) Pickup	(k) Cross Leg	(l) Back Dribble
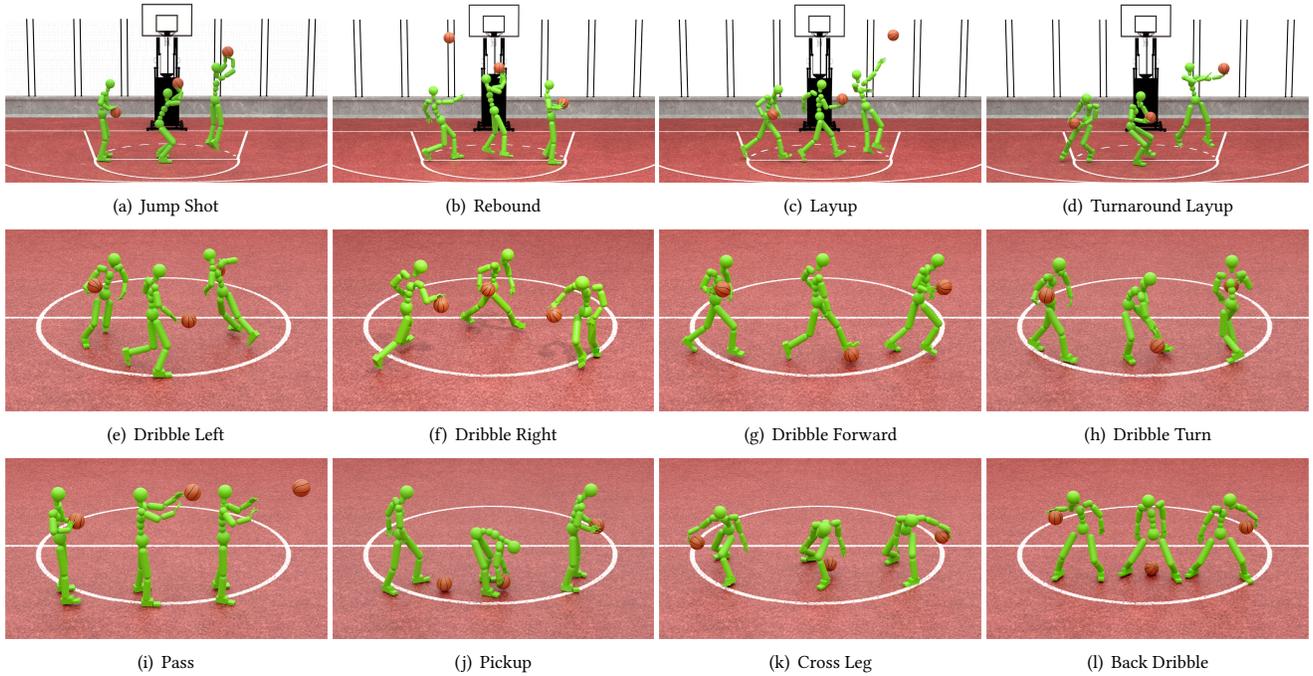
Fig. 6. Simulated humanoids exhibit comprehensive basketball skills. SkillMimic can teach humanoids a wide range of basketball skills using the same configuration in a purely data-driven manner, covering almost all fundamental basketball skills. Keyframes are placed in chronological order from left to right.

6.1 Experimental Setup

We use Isaac Gym [Makoviychuk et al. 2021] 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 [Schulman et al. 2017]. 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 the appendix.

6.2 Metrics

To evaluate the experimental performance, we consider the following quantitative metrics:

6.2.1 Accuracy. The overall accuracy of HOI imitation, abbreviated as *Acc.*, is defined per frame, and deems imitation accurate when the object position and body position errors are both under the thresholds and the contacts are correct. The object threshold is defined as $0.2m$. The body threshold is defined as $0.1m$. The *Acc* is calculated by averaging the success values of all frames.

6.2.2 Position Error. The Mean Per-Joint Position Error (MPJPE) of the body ($E_{\text{b-mpjpe}}$) and object ($E_{\text{o-mpjpe}}$) is used to evaluate the positional imitation performance (in mm) following [Luo et al. 2023].

6.2.3 Contact Error. The contact error E_{cg}, ranging from 0 to 1, is defined as $\frac{1}{N}\sum_{t=1}^{N}\text{MSE}(s_t^{cg}, \hat{s}_t^{cg})$, where N is the total frames of the reference HOI data.

6.2.4 Success Rate. We introduced a set of skill-specific rules to determine the success or failure and skill executions:

- **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, and the distance between the ball and the target position is less than 0.3 m.
- **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.

Table 1. Ablation study on imitation reward design. We experiment on two HOI datasets, including GRAB, which contains multiple objects, and BallPlay-V, which has significant data inaccuracies. The result highlights the importance of the CG reward and multiplication in achieving precise HOI imitation.

	SkillMimic w/o Multiplication				SkillMimic w/o CGR				SkillMimic			
Dataset	$Acc.\uparrow$	$E_{\text{b-mpjpe}}\downarrow$	$E_{\text{o-mpjpe}}\downarrow$	$E_{\text{cg}}\downarrow$	$Acc.\uparrow$	$E_{\text{b-mpjpe}}\downarrow$	$E_{\text{o-mpjpe}}\downarrow$	$E_{\text{cg}}\downarrow$	$Acc.\uparrow$	$E_{\text{b-mpjpe}}\downarrow$	$E_{\text{o-mpjpe}}\downarrow$	$E_{\text{cg}}\downarrow$
GRAB	27.0%	**44.7**	180.2	0.724	38.6%	91.0	180.1	0.337	**95.4%**	71.1	**78.0**	**0.026**
BallPlay-V	7.5%	**40.1**	1662.5	0.306	13.6%	88.5	155.3	0.412	**82.4%**	56.8	**82.9**	**0.087**

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.

	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%

6.3 Evaluating Skill Learning

In this section, we comprehensively evaluate the performance of SkillMimic in skill learning. We not only compare it with baseline methods but also conduct robustness tests on the skills acquired by SkillMimic. Additionally, we perform ablation studies and analyses from multiple dimensions, including reward function design and data volume. These experiments demonstrate the superior performance of SkillMimic, the rationality of its design, and its potential for further scaling up.

6.3.1 Comparing with Baselines.

Baseline Methods. We compare our proposed SkillMimic with closely related imitation-based skill-learning methods, including DeepMimic [Peng et al. 2018] and AMP [Peng et al. 2021], which imitate body motion. We also create diverse versions of SkillMimic following the concept of reward designs in DeepMimic and AMP, dubbed SkillMimic-DM and SkillMimic-AMP, respectively. This allows for a more straightforward comparison of the differences in our reward function design. Detailed implementation of these methods can be found in the appendix.

Datasets. We conduct experiments using data from three datasets: GRAB, BallPlay-V, and BallPlay-M. From the GRAB S8 subset, we select 5 cases, including grasping a cube, cylinder, flashlight, flute, and bottle. All data from BallPlay-V are used in the experiments. For BallPlay-M, we selected 40 pickup clips, one layup clip, one shot clip, and one dribble forward clip. These datasets were chosen for their representativeness and diversity, providing a comprehensive evaluation of our method's overall performance.

Training Details. Due to significant object and robot penetration issues in the GRAB and BallPlay-V data, using Reference State Initialization (RSI) often causes objects to bounce away. Therefore, for GRAB and BallPlay-V, we initialize the humanoid and object using the first frame of the reference clip and set the maximum simulation length to the clip length. For GRAB and BallPlay-V, we train 1 billion samples for all experiments, use the reference state as the condition, and only evaluate the imitation performance. For BallPlay-M, we apply RSI with a maximum simulation length of 60. We train approximately 0.65 billion samples for all experiments on
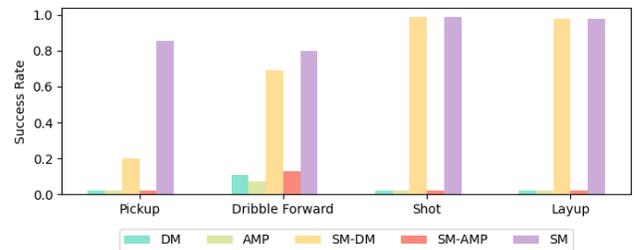
Fig. 7. A comparison of success rates across four typical basketball skills. Our method (SM) significantly outperforms skill-learning methods that only imitate body-only motions (DM and AMP). Additionally, our final reward design (SM) achieves superior overall performance compared to other variants (SM-DM and SM-AMP).

BallPlay-M, each requiring around 10 hours on a single Nvidia RTX 4090 GPU. The exception is the pickup skill, which includes 40 data samples and is trained with about 3.2 billion samples.

Quantitative Results. In Fig. 7, we present the success rate on 4 typical basketball skills from BallPlay-M. We can see that DeepMimic and AMP exhibit very low success rates in all skills. This is because they solely imitate body movements without object perception and constraints. When object perception and object reward functions are introduced, the re-implementation of DeepMimic-style and AMP-style rewards within our SkillMimic framework (SkillMimic-DM and SkillMimic-AMP) show significant performance improvements over their original versions, though the performance is still unstable. This instability is because they totally rely on kinematic imitation rewards, which do not adequately reflect the correctness of contact, leading to kinematic local optima. In contrast, our final reward design achieves stable performance across all skills. This stability is primarily because our reward function not only considers kinematics but also explicitly models contact and uses multiplicative methods to encourage balanced learning among various sub-rewards. Additionally, Tab. 1 presents detailed metrics of HOI imitation across two diverse datasets, GRAB and BallPlay-V, demonstrating that our method (SkillMimic) performs well even on datasets with kinematic

| (a) 1 Clip, 0.5% *Succ.* | (b) 10 Clips, 12.2% *Succ.* | (c) 40 Clips, 76.5% *Succ.* | (d) 131 Clips, 85.6% *Succ.* |
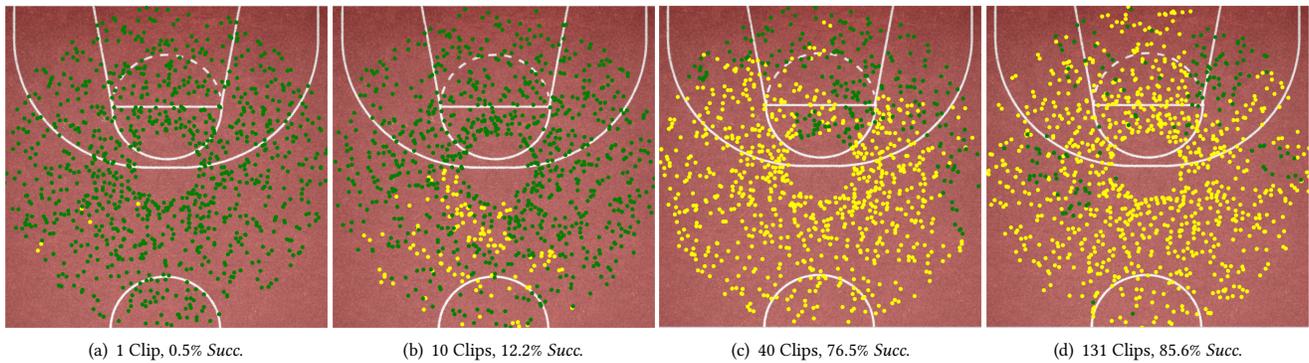
Fig. 8. Pickup performance with different scales of training data. We collect 1, 10, 40, and 131 pickup clips from BallPlay-M to train pickup skills using SkillMimic. 1000 humanoids are initialized at the central, and 1000 balls are initialized at random locations 1 to 5 meters away from the center. The yellow dots indicate successfully picked balls and the green dots represent failures. Results demonstrate improved pickup skill performance with larger datasets.
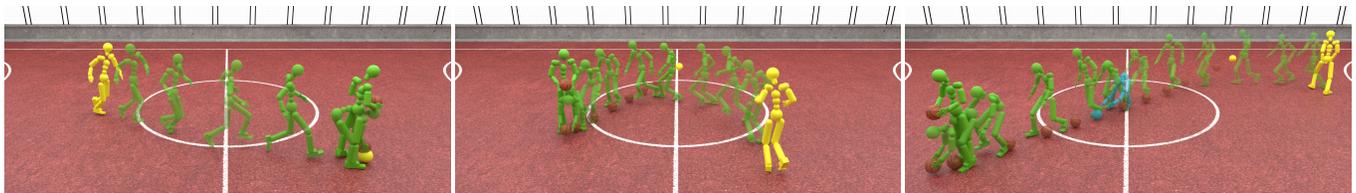
Fig. 9. Demonstration of the pickup skill learned from 40 HOI motion clips. Yellow denotes the initial frame. Left: The humanoid picks up a stationary ball effortlessly. Middle: The humanoid intercepts a ball with random velocity. Right: The humanoid adjusts after missing the ball initially (the frame in blue) and successfully retrieves it on the second attempt, showcasing the potential for learning robust and generalizable skills through extensive data collection.

biases. Comparatively, the absence of multiplicative rewards or CGR leads to noticeable performance drops (SkillMimic w/o multiplication and SkillMimic w/o CGR). This is because data with kinematic biases are more prone to kinematic local optima. Through these comprehensive evaluations, our method shows state-of-the-art performance in HOI imitation.

6.3.2 Evaluating Skill Robustness. 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.

6.3.3 Ablation on Contact Graph Reward. We conduct ablation experiments on diverse datasets to comprehensively evaluate the impact of Contact Graph Reward (CGR) on skill learning. Fig. 5 illustrates the qualitative results of ablating CGR on the GRAB and BallPlay-V datasets. It can be clearly observed that without CGR, the humanoid often resorts to incorrect contacts to achieve more stable object control, such as using its head to push the ball, clamping the ball with its hands and legs, or using its hands to support itself on a table for balance. These examples illustrate typical kinematic local optima. By incorporating CGR, these local optima are effectively

eliminated, resulting in accurate interactions. Additionally, the variant SkillMimic w/o CGR in Tab. 1 presents the quantitative results of ablating CGR on two datasets. The results show that, without CGR, the contact error (E_{cg}) significantly increases, leading to kinematic local optima and causing a noticeable decline in overall imitation accuracy. These results underscore the critical importance of CGR for learning correct interactions.

6.3.4 Ablation on Data Volume. To analyze the impact of data scale on the performance of skills acquired through SkillMimic, we conduct the following ablation experiments: (1) increasing the data for a single skill to evaluate the effect of data scale on its performance; (2) comparing the effectiveness of learning skills through independent training versus mixed training of multiple skills.

Data Scale Effect on a Single Skill. We select the pickup skill for experiments due to its extensive semantic state transitions, as the ball can be in various conditions, leading to diverse state transitions for picking it up. We randomly select pickup clips from BallPlay-M to create four pickup skill collections: one with a single clip, one with 10 clips, one with 40 clips, and one with 131 clips. For each pickup skill collection, we train a single policy using SkillMimic for around 3.2 billion samples. During testing, balls are randomly distributed within a circular area with a radius of 1 to 5 meters centered around a humanoid. This setup is designed to evaluate the generalization performance of the learned pickup skills. Fig. 8 presents the results, where yellow dots indicate successfully picked-up balls and green dots represent failures. The results reveal a substantial improvement

in pickup performance with increasing data scale. This improvement is attributed to the more comprehensive coverage of the state transitions inherent in the pickup skill provided by larger datasets. In Fig. 9, we also visualize how the pickup skill learned from 40 clips handles diverse ball states with diverse behavior. In the left graph, a ball is placed in a random stationary position, allowing a humanoid to run directly towards it and pick it up effortlessly. The middle graph introduces a scenario where the ball, starting from a random position, is given a random velocity. The humanoid dynamically alters direction to intercept and retrieve the moving ball. In the right graph, the ball's velocity is increased, causing the humanoid to miss the ball on the first attempt. Then the humanoid quickly stands up, continues to pursue the ball, and successfully picks it up on the second attempt. These scenarios illustrate how extensive data collection enables the development of complex and adaptive ball retrieval skills. These experiments demonstrate the potential of SkillMimic to extend to learning more complex and diverse skills by enriching the dataset. Unlike previous approaches that required carefully designed skill-specific rewards, our purely data-driven method achieves a high success rate with diverse and adaptive behavior, which is particularly noteworthy.

Mixed Skill Training vs. Individual 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 (DL) skill, one clip for the Dribble Right (DR) skill, and two clips for the Dribble Forward (DF) 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. 6.2. 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 DL 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

Table 3. Success rates of skills trained independently versus jointly. DF, DL, and DR denote Dribble Forward, Left, and Right. 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.

Training	*Succ.* on Individual Skills				*Succ.* on Skill Switching			
	DF	DL	DR	Layup	DF-DL	DF-DR	DF-Layup	DL-DF
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%**

during tests. These findings not only demonstrate that SkillMimic can support a single policy to learn various basketball skills but also underscore the importance of mixed training in enhancing skill generalization and robustness.

6.4 Evaluating Complex Basketball Tasks

By training High-Level Controllers (HLC) to leverage skills learned via SkillMimic, humanoids can rapidly acquire complex basketball skills with simple task rewards. We conduct experiments on the four complex tasks described in Sec. 5.4. We employ a single skill policy as the skill prior, trained using SkillMimic for around 4.5 billion samples across seven skills: pickup, layup, turnaround layup, dribble left, dribble right, dribble forward, and shot. During training, the skill policy is kept fixed, and only the HLC is trained. The HLC operates at one-third the control frequency of the skill policy, i.e., 20 Hz. For tasks such as throwing, heading, and circling, we train for around 0.4 billion samples. For the more challenging scoring task, we train for around 1.2 billion samples. To objectively evaluate our method's performance, we compared it against two baseline methods: (1) learning from scratch using PPO, and (2) learning with motion priors using ASE [Peng et al. 2022]. The low-level controller of ASE is trained using the same data and training samples as our skill policy. For task training, all methods were trained using identical task rewards and simulation steps for fair comparisons.

Fig. 10 (a) showcases an example of manually controlling the pretrained skill policy to perform skill switching. Specifically, we first execute the shooting skill, then switch to the pickup skill as the ball falls, and finally switch to the turnaround layup skill after retrieving the ball. Manual control, however, often fails to achieve precise goals, such as scoring a layup. Besides, since the success rate of skill switching is not 100%, effective skill switching requires switching within appropriate states, making manual control prone to potential failure. However, training a HLC to manage skill switching addresses these issues effectively.

Fig. 10 (b) shows several trajectories of the HLC performing skill controls to accurately deliver the ball to the target basket. Tab. 4 shows that our method achieves a layup success rate of up to 80.25%. Fig. 10 (c) presents a top-down view of the HLC controlling the humanoid to dribble the ball to target locations. Fig. 10 (d) depicts a top-down view of the HLC guiding the humanoid to dribble around a point with varying radii. As shown in Tab. 4, the success rates for the heading and circling tasks are 93.04% and 79.92%, respectively.

(a) Manual Control

(b) Scoring

(c) Heading

(d) Circling
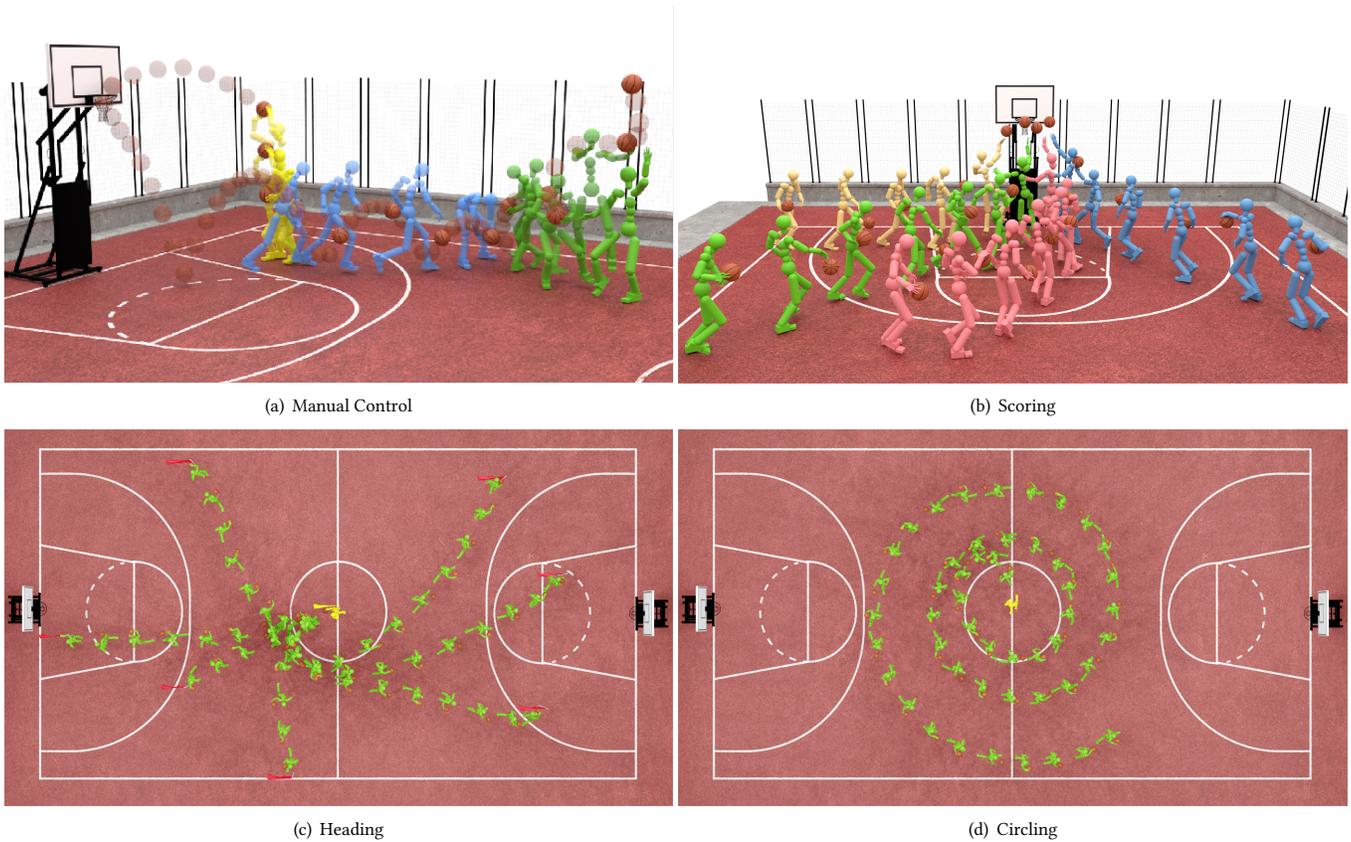
Fig. 10. We train a single policy under a unified configuration to acquire various skills. These skills can be flexibly switched, as illustrated in (a), where yellow denotes shot, blue denotes pickup, and green denotes turnaround layup. Complex high-level tasks can be easily achieved by training a high-level controller (HLC) to manage skill switching: (b) scoring from random positions; (c) dribbling to target locations; and (d) dribbling with different radii.
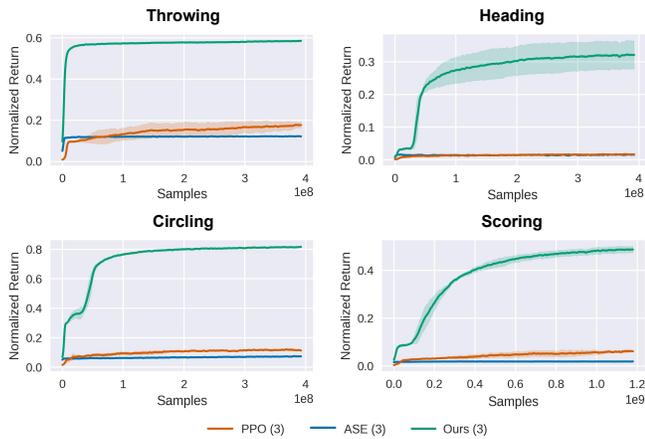
Fig. 11. Learning curves of different methods on 4 complex basketball tasks. On these challenging tasks, simple goal-related task rewards struggle to converge when training from scratch (PPO) or using motion priors (ASE), whereas our method achieves rapid convergence by using skill prior.

Table 4. Success rates on 4 complex basketball tasks. Both PPO (learn from scratch) and ASE (using motion prior) fail to converge on these four challenging tasks. In contrast, leveraging skill priors acquired through SkillMimic, our method effectively learns these difficult tasks.

	PPO	ASE	Ours
Task	$Succ \uparrow$	$Succ \uparrow$	$Succ \uparrow$
Heading	0.70%	0.19%	**93.04%**
Circling	11.14%	4.37%	**79.92%**
Throwing	0.00%	0.00%	**93.40%**
Scoring	0.00%	0.00%	**80.25%**

These results highlight the extensive applicability of skill priors acquired through SkillMimic.

As shown in Fig. 11 and Tab. 4, both PPO and ASE exhibit poor performance in terms of both success rate and reward convergence. This is because the simple goal-defined task rewards are too difficult to learn for these complex basketball tasks. Like how human athletes, after mastering a wide array of basketball skills, can quickly adapt to new tasks, utilizing the pre-trained skill policy as a skill prior

greatly simplifies the learning of complex tasks and results in rapid training convergence.

7 DISCUSSION AND FUTURE WORK

We presented a novel framework that learns various basketball skills in a purely data-driven manner. The proposed Contact Graph (CG) reward plays a critical role in human-object interaction (HOI) imitation, yielding significant improvements over traditional kinematic imitation rewards, particularly on data with kinematic errors. Thanks to a series of innovations, our approach is applicable to various basketball skills, enabling a single policy to master multiple skills and achieve zero-shot skill switching. Additionally, our method demonstrates a clear data scale effect, where the capability of the learned skills improves as the amount of data increases. We have also provided two basketball datasets to facilitate further research within the community. We believe this work paves an exciting new direction for humanoid skill learning and has the potential to extend to more general interactive skills.

Despite these merits, our method faces several limitations. Firstly, exploring more general skill learning within a single policy, such as household tasks or other types of sports, presents challenges in handling generalization across different objects, like varying object shapes. Moreover, perceiving multiple objects simultaneously requires a more general environmental perception approach, rather than directly obtaining privileged information for each object. Secondly, similar to previous works [Peng et al. 2022, 2021], our method predicts the next action based solely on the current moment without considering historical information. This can lead to convergence issues when learning from data with conflicting state transitions, such as a person standing for a long time before picking up a ball, where the data includes both standing-to-standing and standing-to-picking transitions, making it impossible for the policy to learn both transitions. In this work, we address this issue by dividing the conflicting data into different skills. In the future, we hope for more effective solutions, such as incorporating historical observations or learning automatic data segmentation and encoding.

REFERENCES

Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. 2017. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine* 34, 6 (2017), 26–38.

Jinseok Bae, Jungdam Won, Donggeun Lim, Cheol-Hui Min, and Young Min Kim. 2023. Pmp: Learning to physically interact with environments using part-wise motion priors. In *ACM SIGGRAPH 2023 Conference Proceedings*. 1–10.

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

Jona Braun, Sammy Christen, Muhammed Kocabas, Emre Aksan, and Otmar Hilliges. 2023. Physically plausible full-body hand-object interaction synthesis. *arXiv preprint arXiv:2309.07907* (2023).

Stelian Coros, Philippe Beaudoin, and Michiel Van de Panne. 2010. Generalized biped walking control. *ACM Transactions On Graphics (TOG)* 29, 4 (2010), 1–9.

Jieming Cui, Tengyu Liu, Nian Liu, Yaodong Yang, Yixin Zhu, and Siyuan Huang. 2024. Anyskill: Learning openvocabulary physical skill for interactive agents. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 3.

Zhiyang Dou, Xuelin Chen, Qingnan Fan, Taku Komura, and Wenping Wang. 2023. C-ase: Learning conditional adversarial skill embeddings for physics-based characters. In *SIGGRAPH Asia 2023 Conference Papers*. 1–11.

Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J Black, and Otmar Hilliges. 2023. ARCTIC: A Dataset for Dexterous Bimanual Hand-Object Manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12943–12954.

Haoran Geng, Helin Xu, Chengyang Zhao, Chao Xu, Li Yi, Siyuan Huang, and He Wang. 2023. Gapartnet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7081–7091.

Anindita Ghosh, Rishabh Dabral, Vladislav Golyanik, Christian Theobalt, and Philipp Slusallek. 2023. IMoS: Intent-Driven Full-Body Motion Synthesis for Human-Object Interactions. In *Eurographics*.

Mohamed Hassan, Yunrong Guo, Tingwu Wang, Michael Black, Sanja Fidler, and Xue Bin Peng. 2023. Synthesizing physical character-scene interactions. In *ACM SIGGRAPH 2023 Conference Proceedings*. 1–9.

Edmond SL Ho, Taku Komura, and Chiew-Lan Tai. 2010. Spatial relationship preserving character motion adaptation. In *ACM SIGGRAPH 2010 papers*. 1–8.

Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. *Advances in neural information processing systems* 29 (2016).

Jessica K Hodgins, Wayne L Wooten, David C Brogan, and James F O'Brien. 1995. Animating human athletics. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. 71–78.

Nan Jiang, Tengyu Liu, Zhexuan Cao, Jieming Cui, Zhiyuan Zhang, Yixin Chen, He Wang, Yixin Zhu, and Siyuan Huang. 2023. Full-body articulated human-object interaction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9365–9376.

Nan Jiang, Zhiyuan Zhang, Hongjie Li, Xiaoxuan Ma, Zan Wang, Yixin Chen, Tengyu Liu, Yixin Zhu, and Siyuan Huang. 2024. Scaling up dynamic human-scene interaction modeling. *arXiv preprint arXiv:2403.08629* (2024).

Jordan Juravsky, Yunrong Guo, Sanja Fidler, and Xue Bin Peng. 2022. Padl: Language-directed physics-based character control. In *SIGGRAPH Asia 2022 Conference Papers*. 1–9.

Jiaman Li, Alexander Clegg, Roozbeh Mottaghi, Jiajun Wu, Xavier Puig, and C Karen Liu. 2023a. Controllable human-object interaction synthesis. *arXiv preprint arXiv:2312.03913* (2023).

Jiaman Li, Jiajun Wu, and C Karen Liu. 2023b. Object motion guided human motion synthesis. *ACM Transactions on Graphics (TOG)* 42, 6 (2023), 1–11.

Jing Lin, Ailing Zeng, Shunlin Lu, Yuanhao Cai, Ruimao Zhang, Haoqian Wang, and Lei Zhang. 2023. Motion-X: A Large-scale 3D Expressive Whole-body Human Motion Dataset. *Advances in Neural Information Processing Systems* (2023).

Libin Liu and Jessica Hodgins. 2017. Learning to schedule control fragments for physics-based characters using deep q-learning. *ACM Transactions on Graphics (TOG)* 36, 3 (2017), 1–14.

Libin Liu and Jessica Hodgins. 2018. Learning basketball dribbling skills using trajectory optimization and deep reinforcement learning. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–14.

Zhengyi Luo, Jinkun Cao, Kris Kitani, Weipeng Xu, et al. 2023. Perpetual humanoid control for real-time simulated avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10895–10904.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. 2021. 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)*. https://openreview.net/forum?id=fgFBtYgJQX_

Liang Pan, Jingbo Wang, Buzhen Huang, Junyu Zhang, Haofan Wang, Xu Tang, and Yangang Wang. 2024. Synthesizing physically plausible human motions in 3d scenes. In *2024 International Conference on 3D Vision (3DV)*. IEEE, 1498–1507.

Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. 2019. Expressive Body Capture: 3D Hands, Face, and Body from a Single Image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 10975–10985.

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. 2018. DeepMimic. *ACM Transactions on Graphics* (Aug 2018), 1–14. https://doi.org/10.1145/3197517.3201311

Xue Bin Peng, Yunrong Guo, Lina Halper, Sergexuey Levine, and Sanja Fidler. 2022. ASE: Large-scale Reusable Adversarial Skill Embeddings for Physically Simulated Characters. *ACM Trans. Graph.* 41, 4, Article 94 (July 2022).

Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. 2021. AMP: Adversarial Motion Priors for Stylized Physics-Based Character Control. *ACM Transactions on Graphics* (Aug 2021), 1–20. https://doi.org/10.1145/3450626.3459670

Jiawei Ren, Mingyuan Zhang, Cunjun Yu, Xiao Ma, Liang Pan, and Ziwei Liu. 2024b. Ins-Actor: Instruction-driven Physics-based Characters. *Advances in Neural Information Processing Systems* 36 (2024).

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

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

Sebastian Starke, Yiwei Zhao, Taku Komura, and Kazi Zaman. 2020. Local motion phases for learning multi-contact character movements. *ACM Trans. Graph.* 39, 4

(aug 2020). https://doi.org/10.1145/3386569.3392450

Sebastian Starke, Yiwei Zhao, Fabio Zinno, and Taku Komura. 2021. Neural animation layering for synthesizing martial arts movements. *ACM Trans. Graph.* 40, 4 (jul 2021). https://doi.org/10.1145/3450626.3459881

Jingkai Sun, Qiang Zhang, Yiqun Duan, Xiaoyang Jiang, Chong Cheng, and Renjing Xu. 2023. Prompt, plan, perform: Llm-based humanoid control via quantized imitation learning. *arXiv preprint arXiv:2309.11359* (2023).

Omid Taheri, Nima Ghorbani, Michael J. Black, and Dimitrios Tzionas. 2020. GRAB: A Dataset of Whole-Body Human Grasping of Objects. In *European Conference on Computer Vision (ECCV)*. https://grab.is.tue.mpg.de

Jie Tan, Yuting Gu, C Karen Liu, and Greg Turk. 2014. Learning bicycle stunts. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–12.

Chen Tessler, Yoni Kasten, Yunrong Guo, Shie Mannor, Gal Chechik, and Xue Bin Peng. 2023. Calm: Conditional adversarial latent models for directable virtual characters. In *ACM SIGGRAPH 2023 Conference Proceedings*. 1–9.

Yan Wu, Jiahao Wang, Yan Zhang, Siwei Zhang, Otmar Hilliges, Fisher Yu, and Siyu Tang. 2022. SAGA: Stochastic Whole-Body Grasping with Contact. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Sirui Xu, Zhengyuan Li, Yu-Xiong Wang, and Liang-Yan Gui. 2023. Interdiff: Generating 3d human-object interactions with physics-informed diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14928–14940.

Sirui Xu, Ziyin Wang, Yu-Xiong Wang, and Liang-Yan Gui. 2024. InterDreamer: Zero-Shot Text to 3D Dynamic Human-Object Interaction. *arXiv preprint arXiv:2403.19652* (2024).

KangKang Yin, Kevin Loken, and Michiel Van de Panne. 2007. Simbicon: Simple biped locomotion control. *ACM Transactions on Graphics (TOG)* 26, 3 (2007), 105–es.

Hui Zhang, Sammy Christen, Zicong Fan, Luocheng Zheng, Jemin Hwangbo, Jie Song, and Otmar Hilliges. 2024. ArtiGrasp: Physically plausible synthesis of bi-manual dexterous grasping and articulation. In *2024 International Conference on 3D Vision (3DV)*. IEEE, 235–246.

Haotian Zhang, Ye Yuan, Viktor Makoviychuk, Yunrong Guo, Sanja Fidler, Xue Bin Peng, and Kayvon Fatahalian. 2023b. Learning Physically Simulated Tennis Skills from Broadcast Videos. *ACM Trans. Graph.* (2023), 14 pages. https://doi.org/10.1145/3592408

Yunbo Zhang, Deepak Gopinath, Yuting Ye, Jessica Hodgins, Greg Turk, and Jungdam Won. 2023a. Simulation and retargeting of complex multi-character interactions. In *ACM SIGGRAPH 2023 Conference Proceedings*. 1–11.

Qingxu Zhu, He Zhang, Mengting Lan, and Lei Han. 2023. Neural categorical priors for physics-based character control. *ACM Transactions on Graphics (TOG)* 42, 6 (2023), 1–16.

A HYPERPARAMETERS

The hyperparameter configurations employed during the pre-training phase of the skill policy are detailed in Tab. 5, while the hyperparameters utilized for the training of the high-level controller are presented in Tab. 6. Additionally, Tab. 7 displays the hyperparameter settings for all sub-rewards involved.

B DETAILS OF VARIANT MODELS

B.1 Variants in Skill Learning

In the main paper, we developed multiple variants to enable thorough ablation studies and comparative analysis in skill learning. As a control measure, these variants are identical in all respects except for the variations in their reward functions. We will next delineate the specific differences in the reward functions of these variants. The symbols used for the sub-reward functions below are consistent with those in the main text. Unless specified, the hyperparameters of these sub-rewards are the same, as shown in Tab. 7.

SkillMimic-DM. The reward function is

$$r_t = r_t^p + r_t^r + r_t^{rv} + r_t^o. \tag{14}$$

The hyperparameters of these sub-rewards are shown in Tab. 7.

SkillMimic-AMP. The reward function is

$$r_t = -\log\left(1 - D(s_t, s_{t+1})\right), \tag{15}$$

where s represents the HOI state, D denotes the discriminator.

SkillMimic w/o Multiplication. The reward function is

$$r_t = r_t^b + r_t^o + r_t^{rel} + r_t^{reg} + r_t^{cg}. \tag{16}$$

SkillMimic w/o CGR. The reward function is

$$r_t = r_t^b * r_t^o * r_t^{rel} * r_t^{reg}. \tag{17}$$

B.2 Variants in High-Level Tasks

To evaluate the performance of our method on high-level tasks, we established two sets of experiments for comparison: one involving training from scratch and the other utilizing ASE. For ASE, we trained a Low-Level Controller (LLC) using the same dataset as our skill policy, employing the following reward function:

$$r_t = -\log\left(1 - D(s_t, s_{t+1})\right) + \beta \log q\left(z_t | s_t, s_{t+1}\right). \tag{18}$$

where s represents the body-only state (object is not considered), D denotes the discriminator, q denotes the encoder and z represents the latent code. The LLC can effectively generate body motions similar to those in the dataset.

Subsequently, we trained a High-Level Controller (HLC) using the same task reward as our method. The network architecture of this HLC is identical in size to that used in our approach; however, the HLC of ASE outputs continuous latent variables, whereas our HLC outputs discrete skill conditions.

Table 5. Hyperparameters for training skill policy.

Parameter	Value
dim(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 6. Hyperparameters for training high-level controller.

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

Parameter	SM	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	–
$\lambda^{cg}[0]$ Sensitivity of Ball-Hands Contact Error	5	–
$\lambda^{cg}[1]$ Sensitivity of Ball-Body Contact Error	5	–
$\lambda^{cg}[1]$ Sensitivity of Body-Hands Contact Error	5	–
λ^{reg} Sensitivity of Velocity Regularization	10^{-12}	–