C·ASE: Learning Conditional Adversarial Skill Embeddings for Physics-based Characters

Zhiyang Dou[*]
zhiyang0@connect.hku.hk
The University of Hong Kong
Hong Kong

Xuelin Chen[†]
xuelin.chen.3d@gmail.com
Tencent AI Lab
China

Qingnan Fan
fqnchina@gmail.com
Tencent AI Lab
China

Taku Komura
taku@cs.hku.hk
The University of Hong Kong
Hong Kong

Wenping Wang
wenping@tamu.edu
Texas A&M University
USA

Figure 1: Our framework enables physically simulated characters to master highly varied and extensive skills with high efficiency and effectiveness. Notably, it offers an explicit control handle for directly specifying the desired skill from a diverse and extensive set of skills. Here, a character is instructed by the user to perform a sequence of skills, including kick, jump attack, sword bash, shield bash, and finally, roaring.

ABSTRACT

We present C·ASE, an efficient and effective framework that learns Conditional Adversarial Skill Embeddings for physics-based characters. C·ASE enables the physically simulated character to learn a diverse repertoire of skills while providing controllability in the form of direct manipulation of the skills to be performed. This is achieved by dividing the heterogeneous skill motions into distinct subsets containing homogeneous samples for training a low-level conditional model to learn the conditional behavior distribution. The skill-conditioned imitation learning naturally offers explicit control over the character's skills after training. The training course incorporates the focal skill sampling, skeletal residual forces, and element-wise feature masking to balance diverse skills of varying complexities, mitigate dynamics mismatch to master agile motions and capture more general behavior characteristics, respectively. Once trained, the conditional model can produce highly diverse

and realistic skills, outperforming state-of-the-art models, and can be repurposed in various downstream tasks. In particular, the explicit skill control handle allows a high-level policy or a user to direct the character with desired skill specifications, which we demonstrate is advantageous for interactive character animation.

CCS CONCEPTS

• **Computing methodologies → Procedural animation**; *Control methods*; *Adversarial learning*.

KEYWORDS

physics-based character animation, motion control, conditional GAN, deep reinforcement learning

[*]Work done during an internship at Tencent AI Lab
[†]Corresponding author

1 INTRODUCTION

Humans possess a remarkable ability to acquire a wide range of skills through years of practice and can effectively utilize these skills to handle complex tasks. Generally, the motion of these diverse skills is *heterogeneous*. For instance, stationary standing and highly dynamic back-flipping exhibit significant differences in terms of their movement over time. In practice, humans can divide these

heterogeneous skill motions into homogeneous sets so as to focus on a specific skill during each training session. This divide-and-conquer idea is classic and has demonstrated efficacy in the field of physics-based character control [Liu and Hodgins 2018; Won et al. 2020].

However, this is at odds with recent state-of-the-art literature in learning a large-scale, reusable and unified embedding space for physics-based characters, where samples (i.e., state transitions) generated from diverse skills with heterogeneity are treated as *homogeneous* and learned collectively into latent representations [Peng et al. 2022; Won et al. 2022; Yao et al. 2022]. Apparently, transitions of an idle motion differ significantly from those in a sword-swinging motion. This inconsistency undermines the performance of existing models that assume learning on homogeneous samples, as evidenced by the severe *mode collapse* [Peng et al. 2022; Yao et al. 2022]. Moreover, these holistically embedded skill representations do not provide controllability in the form of direct specification of the desired skills, which is a highly desirable property of character animation for offering an interactive and immersive experience to the user. Although some have shown skill-level control with additional training to stimulate corresponding emergent behaviors from holistically embedded skills [Yao et al. 2022], they have not demonstrated the scalability to extensive skills.

In this work, we advocate that the heterogeneous nature of diverse skills, combined with the need for controllability over embedded skills, necessitates a novel training paradigm for efficient skill learning. To this end, we employ the classic divide-and-conquer strategy and present C·ASE, an efficient and effective framework that learns Conditional Adversarial Skill Embeddings for physics-based characters. While learning individual skills had been adopted in training physics-based character [Liu and Hodgins 2018; Won et al. 2020], a particular emphasis of this work is to scale it up to large-scale motion datasets. Specifically, given a large dataset containing diverse motion clips and labels, which can be annotated per clip manually or using a neural skill labeler (see Appendix A.2), a low-level latent-variable model is conditioned on skill labels to capture the conditional behavior distribution, and is trained via conditional adversarial imitation learning. The conditional model can learn extensive skills efficiently while also naturally offering explicit control over the character's skills.

That being said, we have to address unique challenges arising from training our characters. First, efficient training should distribute the learning resources, such as training time, over skills with different levels of complexity. For instance, imitating energetic sword swinging is apparently harder than idling. This is achieved by the *focal sampling strategy* in our framework, which encourages the training to dynamically mine hard skill samples and eventually leads to faster and balanced coverage of diverse reference skills. In addition, incorporating a vast array of heterogeneous skills leads to challenges or even failure in adversarial imitation learning, which has also been revealed and explained by the dynamics mismatch between the virtual character and real human in [Yuan and Kitani 2020]. Hence, we resort to the *skeletal residual forces* to augment the character's control policy when learning highly varied motions, improving the motion quality of particularly agile skills. Last, as samples under each skill are sparse, we further adopt an *element-wise feature masking*, which is simply realized by introducing dropout

layers inside the discriminator to avoid over-reliance on motion details and hence enable capturing general behavioral characteristics, leading to diversified transitions learned under each skill condition.

Once pre-trained, our conditional model can produce highly diverse and realistic skills and offers an explicit control handle for direct manipulation of the character's skills. Experiments show our model achieves state-of-the-art performance in capturing the reference motion distribution, outperforming competing methods by a significant margin (coverage: Ours - 91% vs. CALM [Tessler et al. 2023] - 71% vs. ASE [Peng et al. 2022] - 66%). Furthermore, we demonstrate that our conditional model can benefit the character animation by integrating it into an interactive authoring system by training deep RL-based high-level policies, that allows users to manipulate physically simulated characters with discrete skill label specifications and other control signals. This is similar to those in video games, but our model supports a much wider range of skills and produces physically plausible motions. Last, we showcase the use of our conditional model in various traditional high-level tasks, where policies learn to direct the low-level conditional model for completing different tasks (Appendix A.3).

2 RELATED WORK

With advancements in Deep Reinforcement Learning (DRL) and the accessibility of high-quality motion capture (mocap) datasets [CMU 2002; Harvey et al. 2020; Mahmood et al. 2019; SFU 2011; Tsuchida et al. 2019; Wang et al. 2020], data-driven methods have demonstrated impressive results in physics-based character animation. In the following, we mainly cover these data-driven methods that fall into two categories:

Tracking-based Methods. Bergamin et al. [2019]; Fussell et al. [2021]; Park et al. [2019]; Peng et al. [2018]; Won et al. [2020] train controllers to imitate reference motions by tracking target pose sequences from motion clips. DeepMimic [Peng et al. 2018], the pioneering work, trains a policy network with random state initialization and early termination for mimicking. The idea was later extended to track motion matching-generated reference motions for responsive characters [Bergamin et al. 2019] and improved by removing motion matching dependency [Park et al. 2019] with a recurrent neural network predicting future reference poses. Efforts have been made to control the diverse behaviors of physically simulated characters. Won et al. [2020] constructs a large motion graph from mocap data, groups graph nodes into clusters, and trains a mixture of expert networks for tracking. SuperTrack [Fussell et al. 2021] introduces a world model represented by a neural network trained to approximate physical simulation, enabling supervised policy network training and accelerating the process.

However, tracking-based methods typically struggle to imitate various skills from large, diverse motion datasets. Composing disparate skills often requires a dedicated motion planner to select appropriate clips for complex tasks, which eludes these methods.

Learning Skill Priors. The prevailing trend in physics-based character is to learn powerful skill priors from large and diverse motion datasets. By embedding distinct skills into a low-dimensional latent space, these models can reproduce versatile skills and be reused to learn high-level controllers for complex tasks. Merel et al. [2018]

distill skill expert networks into a latent space for high-level tasks, while Catch and Carry [Merel et al. 2020] incorporates vision signals for diverse full-body tasks. Won et al. [2020] utilize a mixture of experts to learn skills from a diverse set of behaviors, where the expert network for each skill needs to be trained individually, and subsequently, a gating network is trained to combine the experts. However, training a large number of experts can be laborious and costly, and learning the gating network to effectively integrate them can be challenging. Peng et al. [2019] propose a hierarchical controller with multiplicative compositional policies for more composable control. Recently, Won et al. [2022] employ a conditional variational auto-encoder (VAE) for embedding skills into a low-dimensional Gaussian distribution. Yao et al. [2022] further introduces ControlVAE to learn a state-conditioned motion prior.
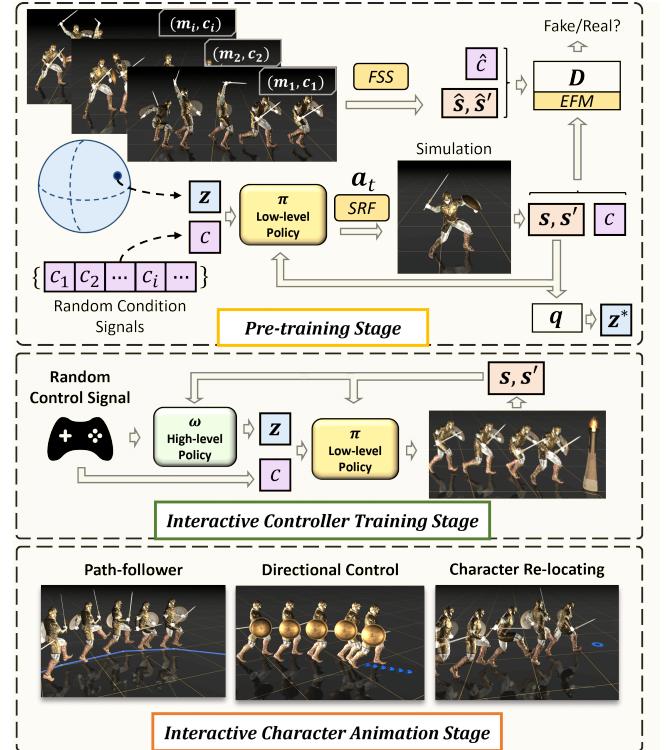
In the GAIL regime [Ho and Ermon 2016], Peng et al. [2021] present Adversarial Motion Prior (AMP) for goal-conditioned reinforcement learning with life-like motions. However, AMP's high-level objective is coupled with a low-level style reward, necessitating tedious fine-tuning for complex tasks. Later, Peng et al. [2022] introduce Adversarial Skill Embeddings (ASE) to learn reusable skill priors, with which high-level controllers learn to direct for complex tasks. [Juravsky et al. 2022] focuses on aligning skill embeddings with a pre-trained language latent space [Radford et al. 2021], allowing natural language-directed characters. CALM, a concurrent work that shares a similar setting to ours, introduces a framework for learning semantic motion representations from a large dataset and also demonstrates control over the skills with *known* semantic labels of reference motion clips. Their key is to heuristically align overlapping samples extracted from the same clip while pulling apart those from different clips. Hence, their success is still contingent upon well-semantically segmented clips. As a consequence, CALM's training is unstable and suffers from severe mode collapse, as recognized by the authors and evidenced in our comparisons.

In general, while these methods can generate various motions, they often suffer from mode collapse when handling highly varied and extensive skills. We attribute this issue to treating samples from distinct skills as homogeneous ones, which goes against the nature of human motions, and propose decomposing the whole repertoire into homogeneous subsets for learning skill-conditioned behavior distributions with several crucial training techniques. Last, our low-level conditional model provides effective controllability over the embedded skills, allowing a high-level policy or user to direct characters to perform desired skills.

3 METHOD

Our framework consists of three stages (See Figure 2): 1) Pre-training stage, where a low-level conditional policy is trained to imitate reference skills; 2) Interactive controller training stage, where more high-level policies are trained to allow more interactive controls of the character; and 3) Interactive character animation stage, where users can interactively animate the character in various ways.

During the pre-training stage, a reference dataset $\mathcal{M} = \{(m^i, c^i)\}$ with annotated motion clips m^i and skill labels c^i is used for learning conditional adversarial skill embeddings. Each motion clip $m^i = \{s_t^i\}$ is represented as a sequence of states that depicts a particular

Figure 2: Our framework contains three stages: the pre-training, interactive controller training, and interactive character animation stages. During pre-training, a low-level policy π learns conditional adversarial skill embeddings from a diverse and extensive motion dataset, followed by more high-level policies π trained to allow interactive control of the character. Last, during the interactive character animation stage, users can interactively animate the character in various manners, possibly with desired skills.

skill. Note that different m^i can correspond to an identical skill label c^i in the dataset. Then, a low-level conditional policy $\pi(a|s, z, c)$[1] is trained through conditional adversarial imitation learning, mapping latent variables z to behaviors resembling motions specified by c. At the interactive controller training stage, we train additional policies to attain more controls for interactive character animation, such as directional control, path-follower, etc. At the interactive animation stage, trained policies are fixed, and then users can interactively animate the physics-based character by specifying the desired skills, moving directions/paths or target location.

3.1 Learning Conditional Adversarial Skill Embeddings

C·ASE divides the dataset into sub-sets that each contain homogeneous samples, from which a conditional low-level policy learns a conditional action distribution. We assume that a transition under a skill category c is represented by a latent variable z sampled from a prior hypersphere distribution $\mathcal{Z}$, i.e., $z = \bar{z}/\|\bar{z}\|, \bar{z} \sim \mathcal{N}(0, \mathbf{I})$.

---

[1]For simplicity, we ignore the superscript and subscript from now on

Specifically, a low-level policy $\pi(\mathbf{a}|\mathbf{s}, \mathbf{z}, c)$ takes as input the character's current state $\mathbf{s}$, a latent variable $\mathbf{z}$, and, more importantly, a skill label c, and then learns to output an action $\mathbf{a}$ that eventually leads to motions conforming to behavioral characteristics specified by motions sampled from the skill indicated by c.

Building upon the success in [Peng et al. 2022], a pioneer work learns large-scale adversarial skill embeddings in a GAN-like framework, we train the low-level policy network with a conditional adversarial imitation learning procedure, where the low-level policy π learns to fool a discriminator $D(\mathbf{s}, \mathbf{s}', c)$, that is also conditioned on skill labels c and learns to distinguish reference motions from generated ones. We train with the imitation objective and diversity objective to capture the conditional behavior distribution implicitly. Specifically, the conditional discriminator is trained to minimize:

$$
\begin{aligned}
\min_D = & -\mathbb{E}_{d^{\mathcal{M}}(\mathbf{s}, \mathbf{s}', c)} \left[\log(D(\mathbf{s}, \mathbf{s}', c) \right] \\
& - \mathbb{E}_{d^{\pi}(\mathbf{s}, \mathbf{s}', c)} \left[\log(1 - D(\mathbf{s}, \mathbf{s}', c) \right] \\
& + w_{\text{gp}} \mathbb{E}_{d^{\mathcal{M}}(\mathbf{s}, \mathbf{s}', c)} \left[\|\nabla_\phi D(\phi)|_{\phi \in (\mathbf{s}, \mathbf{s}', c)} \|^2 \right],
\end{aligned} \quad (1)
$$

where $d_{\mathcal{M}}(\mathbf{s}, \mathbf{s}', c)$ and $d_\pi(\mathbf{s}, \mathbf{s}', c)$ denote state transitions $(\mathbf{s}, \mathbf{s}')$ drawn from the reference skill c and ones generated by the conditional policy π, respectively. The last term is a gradient penalty regularization for stabilizing the training.

To facilitate training π, we first employ a conditional motion encoder q to enforce the mapping between state transitions $(\mathbf{s}, \mathbf{s}')$ and the latent $\mathbf{z}$ under the c label-conditioned distribution. Since the conditional latent space under c is modeled as a hypersphere, q is modeled as a von Mises-Fisher distribution: $q(\mathbf{z}|\mathbf{s}, \mathbf{s}', c) = \frac{1}{Z} e^{\kappa, \mu_q(\mathbf{s}, \mathbf{s}', c)^T \mathbf{z}}$, where $\mu_q(\mathbf{s}, \mathbf{s}', c)$ is the mean of the distribution, which is further normalized by requiring $\|\mu_q(\mathbf{s}, \mathbf{s}', c)\| = 1$. Z and κ represent a normalization constant and a scaling factor, respectively. The encoder network q for each skill condition c is then trained by maximizing the log-likelihood:

$$
\max_q \mathbb{E}_{p(\mathbf{z})} \mathbb{E}_{d^{\pi}(\mathbf{s}, \mathbf{s}', c|\mathbf{z})} \left[\kappa \mu_q(\mathbf{s}, \mathbf{s}', c)^T \mathbf{z} \right] \quad (2)
$$

Then, the reward for policy π at each simulation time step t is given by: $r_t = -\log(1 - D(\mathbf{s}_t, \mathbf{s}_{t+1}, c)) + \beta \log q(\mathbf{z}_t|\mathbf{s}_t, \mathbf{s}_{t+1})$, where β is a balancing factor. We further add the diversity term to the total objective of π in a buffer of T^{sim} simulation time steps:

$$
\begin{aligned}
& \arg\max_\pi = \mathbb{E}_{p(Z)} \mathbb{E}_{p(\tau|\pi, Z, C)} \left[\sum_{t=0}^{T^{sim}-1} \gamma^t(r_t) \right] \\
& - \lambda_D \mathbb{E}_{d^{\pi}(\mathbf{s})} \mathbb{E}_{\mathbf{z}_1, \mathbf{z}_2 \sim p(\mathbf{z})} \left[\left(\frac{D_{KL}(\pi(\cdot|\mathbf{s}, \mathbf{z}_1, c), \pi(\cdot|\mathbf{s}, \mathbf{z}_2, c))}{0.5(1 - \mathbf{z}_1 \mathbf{z}_2)} - 1 \right)^2 \right]
\end{aligned} \quad (3)
$$

The above description provides a conceptually valid formulation for learning the conditional distribution. We further elaborate on algorithmic designs that are crucial for training the low-level policy in the following sections.

Focal Skill Sampling (FSS). Large motion datasets exhibit a notable characteristic wherein the skills contained within vary in terms of their difficulty to learn. This would inevitably cause unbalanced development, i.e., missing a considerable number of skills, if all skills were treated equally. To overcome this issue, we devise a focal skill sampling strategy that naturally fits into the conditional

imitation learning course, improving the training via adaptively adjusting the sampling across different reference skills.

At each training step, let $\overline{w}_c$ denote the sampling probability of skill c into the reference motion buffer, and $b_c \in (0, 1)$ represent the average score (the probability that the sample is real, i.e., from the reference data) output by the conditional discriminator D on the generated motions under skill category c. The sampling probability is then updated online at each training step as follows:

$$
w_c = (1 - \alpha)w_c + \alpha\sigma(b_c), \quad \sigma(b_c) = 1 - b_c / \sum_{c \in C} b_c, \quad (4)
$$

where C denotes the set of skill category labels, α control the update rate, and w_c is the evenly initialized sampling weights. Note that we normalize $\overline{w}_c = \frac{w_c}{\sum_{c \in C} w_c}$ to serve as the final sampling probability. Empirically, this strategy is applied after approximately 2500 training steps, allowing D to gain discrimination ability first. We demonstrate that this approach enables the policy to effectively cover more skills with high efficiency.

Skeletal Residual Forces (SRF). To augment the control policy to effectively imitate complex and agile motions, such as the jump sidekick, we apply a torque computed from PD target control signals to each joint and require the policy network to predict the residual force at each joint position, effectively compensating for the dynamics mismatch between the virtual character and the real human. Note that such skeletal residual forces are applied to the character in both training and inference stages. The equation of motion for multi-body systems with residual forces is given by:

$$
B(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) + g(\mathbf{q}) = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\tau} \end{bmatrix} + \underbrace{\sum_i J_{v_i}^T \mathbf{h}_i}_{\text{Contact Forces}} + \underbrace{\sum_{j=1}^{J-2} J_{e_j}^T \boldsymbol{\xi}_j}_{\text{Residual Forces}} \quad (5)
$$

On the left-hand side, $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, B, C$, and g represent degrees of freedom (DoFs) of joints, joint velocities, joint accelerations, the inertial matrix, the vector of Coriolis and centrifugal terms, and the gravity vector, respectively. On the right-hand side, the first term comprises the torques $\boldsymbol{\tau}$ computed from the PD target control signals applied to the non-root joint DoFs, whereas $\mathbf{0}$ denotes the non-actuated root DoFs. The second term describes the contact forces $\mathbf{h}_i$ on the humanoid (typically exerted by the ground plane) and the contact points $\mathbf{v}_i$ of $\mathbf{h}_i$, ascertained by the simulation environment. The Jacobian matrix $J_{v_i} = dv_i/d\mathbf{q}$ delineates the manner in which the contact point $\mathbf{v}_i$ varies with respect to the joint DoFs $\mathbf{q}$. The Jacobian matrix $J_{e_j} = de_j/d\mathbf{q}$ defines the change of the contact point e_i of the residual force $\boldsymbol{\xi}_j$ w.r.t. the joint DoFs $\mathbf{q}$. In the last term, similar to [Yuan and Kitani 2020], we ask the policy network to predict the residual forces $\boldsymbol{\xi}_j$ at contact point e_j. We set the contact points e_j to be $J - 2$ body joints j, excluding the sword and shield. In this paper, the joint number J of the avatar is 17. We apply regularization to the residual forces, ensuring that the policy employs these forces when necessarily required: $r_f = \exp(-\sum_j^{J-2} \boldsymbol{\xi}_{j=1})$.

Element-wise Feature Masking (EFM). Learning conditional distributions exacerbates data scarcity under each skill, leading to overfitting and reduced motion diversity. Hence, we further adopt an element-wise feature masking strategy, which is simply realized by introducing dropout layers inside the discriminator, to

significantly alleviate this issue. This stochastic operation avoids over-reliance on motion details, enabling the capture of general behavioral characteristics from sparse samples and resulting in diversified transitions under each skill.

3.2 Interactive Character Animation

The learned low-level conditional model can capture extensive and diverse skills and provides explicit control over the character skills. Moreover, we train additional deepRL-based high-level policies to support interactively animation of the character in various ways.

Directional Control. To enable directional control, a high-level policy takes as input the control signals (c_t, d_t^*, h_t^*) where c_t, d_t^*, h_t^* represents the user desired skill, target local facing direction of the root and the target moving direction, respectively; $*$ stands for the character's local coordinate frame. The objective of the interactively directional control is given by

$$r_t^D = 0.7 \exp\left(-0.25\|\mathbf{v}_t^* - \mathbf{d}_t^* \cdot \dot{\mathbf{x}}_t^{\text{root}}\|^2\right) + 0.3\mathbf{h}^* \cdot \mathbf{h}_t^{\text{root}}, \quad (6)$$

where $\mathbf{h}_t^{\text{root}}$ and $\dot{\mathbf{x}}_t^{\text{root}}$ represent the heading direction and velocity of the character root under specified skill label c_t. We vary the desired velocity $v^* \in [0, 5]$ m/s during training for velocity control. Instead of training the interactive controller to learn to switch skill labels, our low-level model allows for skill switching by explicitly assigning c_t at the t-th time step, and the interactive controller thus only needs to predict a configuration of latent codes $\mathbf{z}$ to complete the task under skill c. During training, we randomize the skill label every five execution steps to simulate user interactive control. After training, users can interactively control the character by specifying the desired skills while dynamically controlling the moving direction akin to those in video games.

Target Location Control. We also support re-locating the character to a target location. The inputs to the high-level policy are (c_t, x_t^*), where c_t, x_t^* represents the user desired skill and target location in the character's local frame, respectively. The objective is given by

$$r_t^L = -0.5\|\mathbf{x}_t^* - \mathbf{x}_t^{\text{root}}\|^2. \quad (7)$$

Here, $\mathbf{x}_t^{\text{root}}$ is the character root location. The high-level controller predicts latent codes $\mathbf{z}$ to navigate the character through the low-level controller. Skill labels are randomized every five execution steps. Once trained, users can dynamically specify a target location and a desired skill label to re-locate the character.

In addition to these interactive controllers, we also evaluate C·ASE in various representative high-level tasks, such as *Reach*, *Steering*, *Location*, and *Strike* as in [Peng et al. 2022]. In these tasks, no user-specified labels are used, and the high-level policies learn to predict the configuration of the skill label and skill latent code for completing the task. For details, please refer to Appendix A.3.

4 EXPERIMENTS

We evaluate the efficacy of our framework by training skilled-conditioned control policies for a 3D simulated humanoid character. Please refer to the supplementary video for more qualitative results.

Dataset. We conduct evaluations on two datasets, including: 1) *Sword&Shield* dataset from [Peng et al. 2022] containing 87 clips [2] and each with a corresponding skill label; 2) *Composite Skills* dataset, that has 265 types of skills, including 87 motion clips from [Peng et al. 2022] and 691 clips of 178 manually annotated skills from the CMU Mocap dataset [CMU 2002]. The character, equipped with a sword and shield, has 37 degrees of freedom. We retarget the motions from the CMU Mocap dataset to an avatar with a sword and a shield; more details are in Appendix A.5. Unless specified, we conduct experiments on the Sword&Shield dataset for a fair comparison with baselines. In addition, we also demonstrate the scalability of our method on the Composite Skills dataset in Appendix A.1.
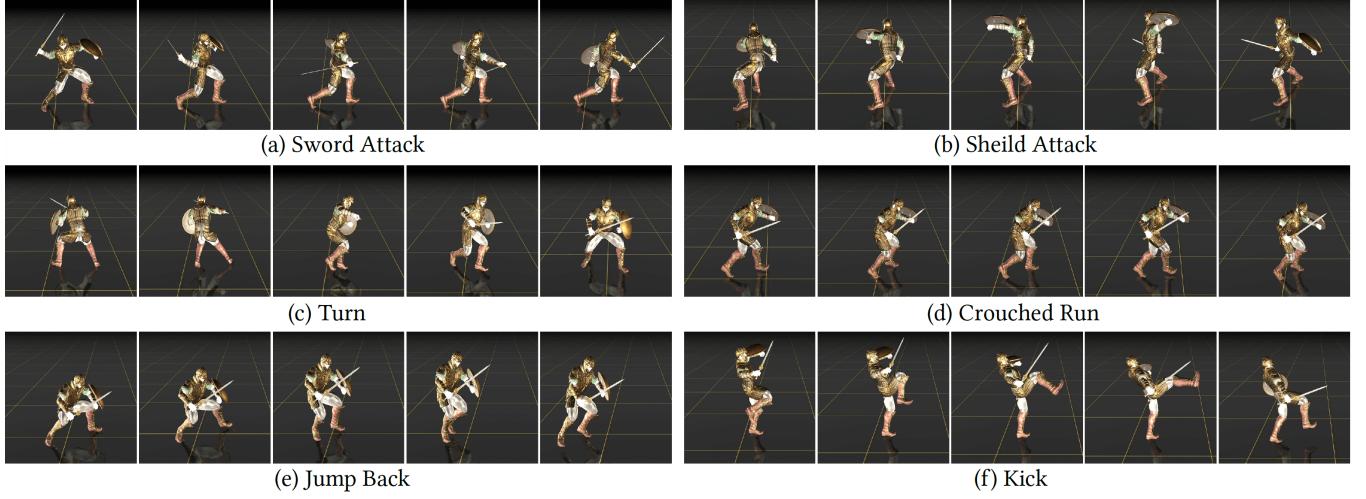
Training. We train the character in IsaacGym [Makoviychuk et al. 2021] with a simulation frequency of 120 Hz and policy frequency of 30Hz. The policies, value functions, encoder and discriminator, are modeled using separate multi-layer perceptions, and the policy networks π and ω are trained with the proximal policy optimization [Schulman et al. 2017]. Policies are trained on a single A100 GPU, with about 1.5 billion samples, corresponding to approximately 1.5 years of simulated time, taking 1.5 days, and high-level policies taking one day. The final animation is retargeted to a rigged avatar. Curriculum learning and joint masking are employed during training; See more details in Appendix A.5.
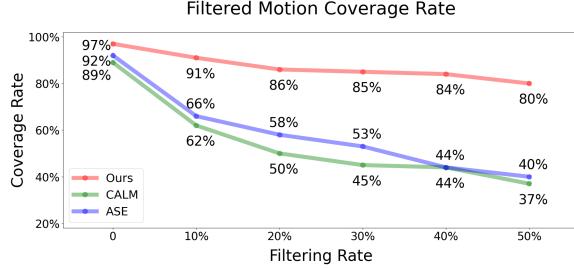
4.1 Low-level Conditional Policy

We first train the low-level policy alone to evaluate its ability to reproduce skills in the motion dataset, particularly when directed by a specified skill label. The policy is able to follow a *random* skill label presented to it, such as left sword swing, right shield bash, etc. Although the Sword&Shield dataset contains only one clip of each skill label, the policy is able to perform corresponding skills with *local variations*. Examples of behaviors produced by the policy when given various skill labels are shown in Figure 3. Next, we present more quantitative evaluations of the low-level policy. More results are presented in the supplementary video.

Filtered Motion Coverage Rate. We evaluate our model in reproducing various motions in the dataset when given random skill labels. Moreover, we compare our model to SOTA methods – ASE and CALM, which also train a low-level policy to reproduce skills in the dataset. Note CALM also trains a low-level conditional policy. All models are trained with the Sword&Shield dataset released by [Peng et al. 2022; Tessler et al. 2023]. While there are other important prior works [Juravsky et al. 2022] , we were not able to compare exhaustively with them as they have not released the source code. Specifically, the quantitative comparison is conducted with the metric motion coverage rate, following [Peng et al. 2022]. The trajectories of our model are generated using random skill labels and latent codes, whereas the trajectories of ASE and CALM are obtained with random skill latent codes. Furthermore, we propose to compute *filtered* motion coverage to factor out stochastic factors built upon the motion coverage rate proposed by [Peng et al. 2022]. For each state transition $(\hat{\mathbf{s}}_t, \hat{\mathbf{s}}_{t+1})$ produced from the policy

---

[2] Due to permission issues, the released dataset contains only 87 clips instead of the 187 described in [Peng et al. 2022], as confirmed by the authors.

(a) Sword Attack

(b) Sheild Attack

(c) Turn

(d) Crouched Run

(e) Jump Back

(f) Kick

Figure 3: C·ASE enables the physically simulated character to perform skills specified by skill labels and transition latent codes.

Figure 4: Comparison of the motion coverage. The coverage rate of ASE and CALM falls dramatically with an increasing filtering rate, implying a serious imbalance of the coverage, whereas ours consistently produces high coverage rates.

π with a skill label c and a latent code $\mathbf{z}$, we find the closest motion clip m^* in the reference motion dataset $\mathcal{M} = \{(m^i, c^i)\}$:

$$m^* = \arg\min_{m^i \in \mathcal{M}} \min_{(\mathbf{s}_t, \mathbf{s}_{t+1}) \in m_i} ||\hat{\mathbf{s}}_t - \mathbf{s}_t||_2 + ||\hat{\mathbf{s}}_{t+1} - \mathbf{s}_{t+1}||_2. \quad (8)$$

We repeat it for every transition in randomly generated trajectories, and the reference motion clip that contains the best-matched transition will be marked as the one that best matches the trajectory.
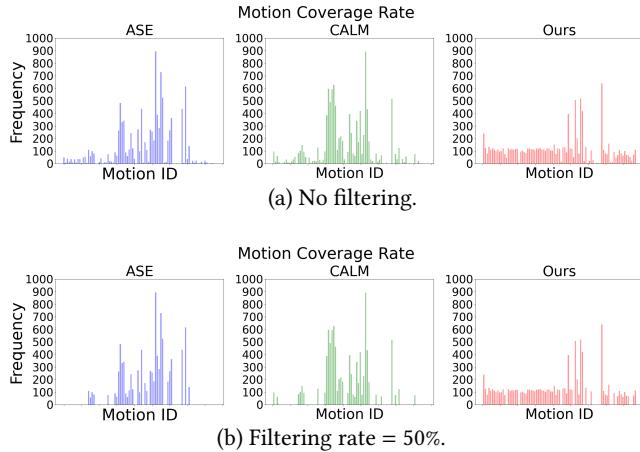
Let l_i denote the number of matched motions under the i-th skill category. The *expected* number of samples under each category is given by $\frac{N}{K}$, where N is the number of generated trajectories and K is the number of skill categories. Then, a reference motion skill is identified as covered only if $l_i > \gamma\frac{N}{K}$, where γ denotes the filtering rate. Then, the filtered motion coverage rate is given by:

$$\text{coverage}(\mathcal{M}, \pi, \gamma) = \frac{1}{K}\mathcal{I}_{i \in \{1, \cdots, K\}}(l_i > \gamma\frac{N}{K}). \quad (9)$$

Figure 4 presents the motion coverage rate under different filtering rates. We can see although ASE achieves a competitive coverage rate to ours when no filtering is applied, its coverage dramatically drops to 66%, 57%, and 40% at the filtering rate of 10%, 20%, and 50%,

respectively. The coverage rate of CALM drops from 84% to 71%, and 43% at the filtering rate of 10%, 20%, and 50%. These results indicate a serious unbalance of the motion coverage, i.e., many motion clips are matched only a few times, possibly due to stochastic factors existing in the randomly generated trajectories. In contrast, our model produces consistently high motion coverage rates at different filtering rates, indicating that all motion clips are matched rather evenly by the randomly generated trajectories. This is further evidenced by Figure 5, which records the frequencies at which π produces trajectories that match each motion clip in the dataset across 10,000 trajectories. We also evaluate on the larger Composite Skills dataset. For a fair comparison, we conducted the evaluation on the Composite Skills dataset while maintaining consistent SRF settings for baseline comparisons. The results are as follows: a) With SRF, our approach achieved a filtered coverage rate of 82%, outperforming CALM with 58% and ASE with 51%. b) Without SRF, our approach achieved a filtered coverage rate of 80%, surpassing CALM with 55% and ASE with 44%. We observed that the inclusion of SRF positively impacted the performance of all models in terms of motion coverage. Notably, the increase in motion coverage was particularly evident for highly dynamic, agile, and stylized motions such as ballet, sidekick, and zombie walks. Despite these improvements, our model continues to excel in learning extensive and complex skills compared with CALM and ASE, which underscores the effectiveness of learning conditional skill embeddings and other key design elements in our framework. See more details in Appendix A.1.

Fréchet Inception Distance. Following [Hassan et al. 2021; Wang et al. 2022], we further measure the similarity between the distribution of generated motions and that of reference motions using Fréchet Inception Distance. The distance is computed using the character state of each frame. We report FID scores computed at three different levels: per frame, per transition (2 frames), and per clip (30 frames). As shown in Table 1, our model achieves lower FID, indicating motions produced from C·ASE are closer to the distribution of reference motions.

(a) No filtering.
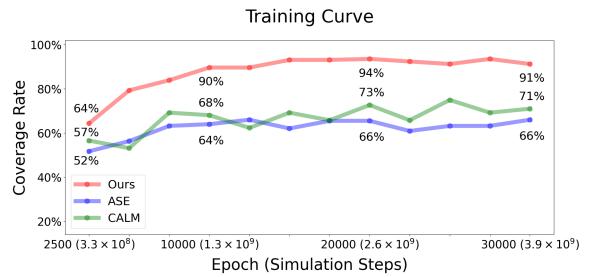

(b) Filtering rate = 50%.

Figure 5: Frequencies at which the low-level policy produces motions that match all 87 individual clips. We show distributions produced by the filtering rate of 0% and 50% here. Compared to ASE and CALM, our method produces diverse motions that much more evenly cover all reference clips.

Table 1: Fréchet Inception Distance (lower is better) comparison. A lower FID indicates that generated motions are closer to the reference distribution.

Input	#Frame	ASE	CALM	Ours
Per-frame	1	28.8	30.1	**16.5**
Per-transition	2	72.3	69.1	**47.4**
Per-clip	30	1969.8	1874.6	**1742.5**

Skill Transition Coverage. It is important that the low-level policy can learn to transition between various skills to perform composed and sequenced skills in complex tasks. To evaluate the model's capability to transition between different skills, we generate transition trajectories by conditioning on two pairs of random condition signals $p_1 = (c, \mathbf{z})$ and $p_2 = (c', \mathbf{z}')$ per trajectory. A transition trajectory is generated by first conditioning on p_1 for 200 time steps, then is conditioned on p_2 for another 200 time steps. Then, these two sub-trajectories are used to separately match in the dataset, using Equation 9, to identify a source motion (denoted as source motion m_S) and a destination motion (denoted as destination motion m_D). We repeat this process for 10, 000 transition trajectories and record the transition coverage, as well as the probability between each pair of motion clips. We compare our model with ASE and CALM, where a transition trajectory is generated using two random skill codes. The transition coverage and probability result is shown in Figure 7, where C·ASE produces a denser connection of each possible transition and the transition coverage is distributed more balanced compared with ASE and CALM. Furthermore, we report the *transition coverage rate* = $\frac{\text{\#transitions from model}}{\text{\#all possible transitions}}$, on which our model (44.3%) outperforms ASE (25.4%) and CALM (28.4%) by a margin of 74.4% and 55.9%, respectively.

Motion Diversity. We evaluate the diversity of motions produced by the low-level model. Following [Hassan et al. 2021; Lu et al. 2022;

Figure 6: Comparison of training effectiveness and efficiency. We plot the filtered coverage rate (filtering rate = 10%) w.r.t the training course.

Wang et al. 2022] We adopt the Average Pairwise Distance (APD) to measure the diversity of a set of generated motion sequences. Specifically, given a set of generated motion sequences $\mathcal{M} = \{m_i\}$ where each motion clip m_i contains L frames, the APD is computed as:

$$APD(\mathcal{M}) = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j \neq i}^{N} (\sum_{t=1}^{L} (\|\mathbf{s}_t^i - \mathbf{s}_t^j\|^2))^{\frac{1}{2}}, \quad (10)$$

where $s_t^i \in M_i$ is a state in a motion clip M_i and N is the number of generated sequences. A larger APD indicates a more diverse set of motion sequences. We compare with ASE and CALM on the mean and standard deviation of this metric 10 times to investigate the diversity across all generated motions. For each time, we test with $N = 10, 000$ sequences that are generated by ASE and CALM conditioned on random latent codes and by our model conditioned on random skill labels and latent codes. As a result, our model produces a higher APD score (160.4 ± 1.77), which indicates higher diversity of generated motions compared to ASE (145.4 ± 2.11) as well as CALM (152.7 ± 1.86).

Moreover, we conduct qualitative evaluations of the motion diversity of our model: (i) *Global root trajectory.* We visualize the behaviors produced by random motions. Figure 9 illustrates the root trajectories produced by different skill labels and latent codes. All motions are generated from the same initial idle state. We generated 100 trajectories, with each containing 300 time steps, for each top-10 skill (ranked by APD of the motion within the skill category). (ii)*Local motion diversity.* We investigate the diversity of motions under each skill category by fixing the skill label c and randomizing the latent code $\mathbf{z}$. As demonstrated in Figure 8, the motions produced with each skill label exhibit local variations while still conforming to the general characteristics of each skill. More qualitative results are presented in the supplementary video.

Learning Efficiency and Effectiveness. We show that learning a structured latent space via conditional adversarial imitation learning not only offers an explicit skill control handle but also greatly improves the effectiveness and efficiency of the training course. In Figure 6, our model can effectively cover around 91% of the reference motions in the dataset within just 30,000 epochs, whereas CALM and ASE converge to only a coverage rate of 71% and 66% and barely improves with more epochs. We conjecture that the instability of CALM's coverage rate may be caused by the dynamically changing latent codes produced by the encoder during training.

4.2 Ablation Study

Focal Skill Sampling. We compare with the baseline model that is trained without the focal skill sampling and those trained with different update rates α. Figure 10 (a) shows the motion coverage rate that these models achieve during the training course. In general, models with focal skill sampling outperform the one where the module is ablated, converging to higher coverage rates. In our experiments, the update rate is set to $\alpha = 20\%$ by default.

Skeletal Residual Force. We investigate the impact of Skeletal Residual Forces (SRF) on the embedding of skills associated with agile movements. A qualitative comparison is presented in the supplementary video, demonstrating that SRF facilitates learning more agile motions. We noticed that, while SRF is important in learning agile movements, it can compromise physical accuracy. So we introduced regularization ($r_f = \exp(- \sum^{J-2} j\xi j = 1)$) to residual forces, ensuring their utilization only when necessary, for which we investigate in the following. We found the average residual forces (L2-norm of the force magnitude) across 17 joints in 200-time steps of 1024 trajectories generated by randomizing the skill label c and the latent code $\mathbf{z}$ during test time amount to only 0.842% of the internal force derived from PD target control. This represents a small percentage and causes only minor deviations from physical correctness.

Element-wise Feature Masking. Empirically, we found that this simple yet effective element-wise feature masking not only improved the motion diversity produced under each skill category with APD increased from 150.4 ± 1.37 to 160.2 ± 1.23 (measured by 10 times) but also improved training efficiency, which can be observed in Figure 10 (b). We found setting the random probability ρ to large values leads to jittering; See the supplementary video. Thus, by default, we use $\rho = 20\%$, which is a good trade-off in practice.

4.3 Interactive Character Animation

We evaluate the efficacy of our framework in interactive character animation. Since the Sword&Shield dataset contains only two loco-motion skills, i.e., walking and running, we pre-train the low-level conditional model and those interactive controller policies on the larger Composite Skills dataset containing richer skills. We present interactive character animations in various ways realized with the learned conditional model and interactive controller policies, including *path-follower*, *directional control* and *character re-locating* with the desired skill explicitly specified by the user. Figure 11 depicts characters faithfully following user-specified paths under various specified skills. Figure 12 demonstrates dynamic user control over the moving direction with desired skills akin to those in video games, while Figure 13 displays relocating the character to a specified location with desired skills. Additional results can be found in the supplementary video. We believe these features are valuable for video game and animation production.

5 DISCUSSION AND CONCLUSION

In this work, we introduce C·ASE, an efficient and effective framework for learning Conditional Adversarial Skill Embedding for physics-based characters. The key idea is dividing the repertoire into homogeneous sub-sets and conquering them for learning conditional behavior distribution. Consequently, C·ASE outperforms state-of-the-art methods, enabling characters to master diverse motor skills efficiently. Notably, skill-conditioned imitation learning naturally offers explicit control over the embedded skills. We demonstrate the application of such explicit control handles in controllable character animation in various ways, showing its superior practical value.

Despite its remarkable advantages, we note a few shortcomings. We are aware of some artifacts remaining in reproducing some CMU skills, as shown in the supplementary video. This is due to two main reasons: First, GAN-based models often suffer from mode-collapse issues, although we have shown that conditional distribution learning could significantly alleviate this problem. Exploring other generative models like the diffusion model [Shi et al. 2023; Song et al. 2020; Tevet et al. 2022] and VQ-GAN [Esser et al. 2021] may be beneficial in the future. Second, practical operations performed during experiments can contribute to the artifacts. For example, our simplified skeleton with 17 joints, compared to the original CMU skeleton with 31 joints, may limit the expression of agile motions, resulting in stiffness. Last, our framework is highly sample-intensive. The artifacts may imply unsaturated sampling during the PPO training and could be mitigated by more sufficient training. This can be supported by the increasing motion coverage rate over longer training time.

While the incorporation of SRF enhances motion quality and is compatible with various simulation platforms [Coumans 2015; Makoviychuk et al. 2021; Todorov et al. 2012], it may not be applied to real-world setups. We believe SRF is not the optimal solution for learning highly agile and complex motions in the simulation environment, which demands more research effort into innovative solutions that guarantee physical correctness. Another limitation of C·ASE is the reliance on the skill label of the motion clips. Although we have shown that an action recognition network could help with motion segmentation to a large extent (see Appendix A.2), developing a fully automatic framework for embedding extensive motions from unstructured data remains challenging.

Last, a more powerful low-level model could, in turn, pose challenges to the learning of high-level strategies for more complex tasks since the effective action space has become larger. Thus, it would be worth exploring training high-level policies that can leverage diverse and extensive skills embeddings more effectively and efficiently for empowering the simulated character with the intelligence to undertake more complex tasks in more complicated environments. For example, training warriors that can make full use of all diverse and extensive skills to win a contest.
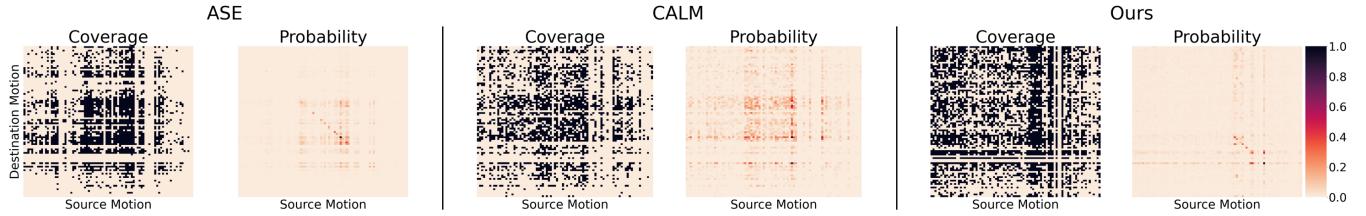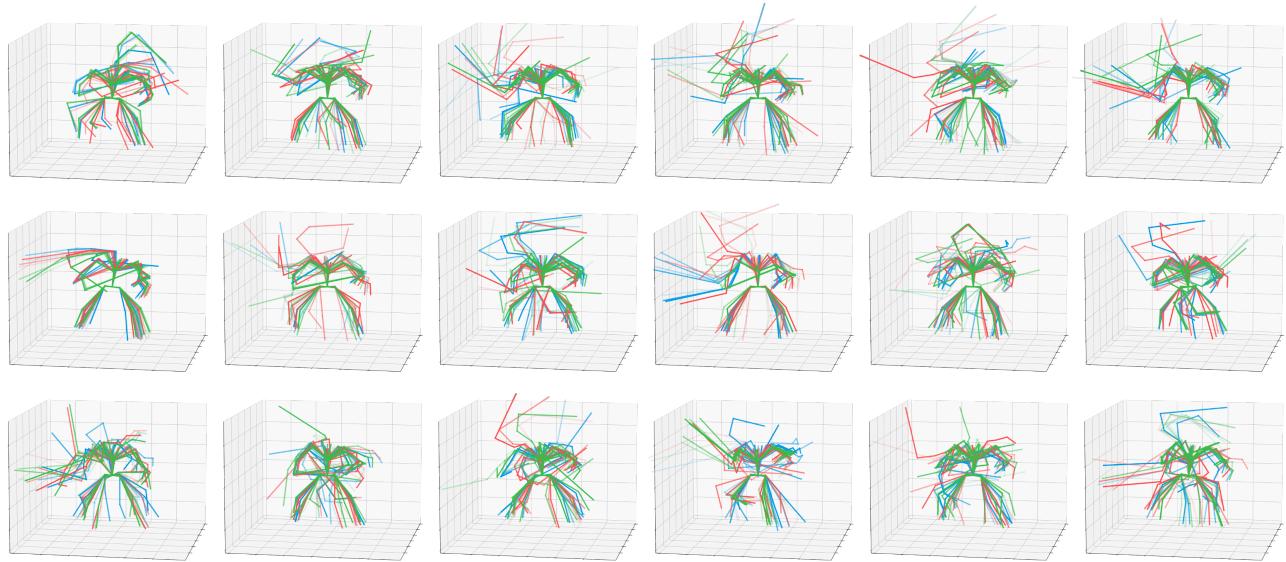
REFERENCES

Kevin Bergamin, Simon Clavet, Daniel Holden, and James Richard Forbes. 2019. DReCon: data-driven responsive control of physics-based characters. *ACM Transactions On Graphics (TOG)* 38, 6 (2019), 1–11.

Zhongang Cai, Daxuan Ren, Ailing Zeng, Zhengyu Lin, Tao Yu, Wenjia Wang, Xiangyu Fan, Yang Gao, Yifan Yu, Liang Pan, et al. 2022. Humman: Multi-modal 4d human dataset for versatile sensing and modeling. In *European Conference on Computer Vision*. Springer, 557–577.

CMU. 2002. CMU Graphics Lab Motion Capture Database. http://mocap.cs.cmu.edu/.

Erwin Coumans. 2015. Bullet physics simulation. *ACM SIGGRAPH 2015 Courses* (2015).

Zhiyang Dou, Qingxuan Wu, Cheng Lin, Zeyu Cao, Qiangqiang Wu, Weilin Wan, Taku Komura, and Wenping Wang. 2022. TORE: Token Reduction for Efficient Human Mesh Recovery with Transformer. *arXiv preprint arXiv:2211.10705* (2022).

Haodong Duan, Jiaqi Wang, Kai Chen, and Dahua Lin. 2022. Pyskl: Towards good practices for skeleton action recognition. In *Proceedings of the 30th ACM International Conference on Multimedia*. 7351–7354.

Patrick Esser, Robin Rombach, and Bjorn Ommer. 2021. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 12873–12883.

Zhou Fan, Rui Su, Weinan Zhang, and Yong Yu. 2019. Hybrid actor-critic reinforcement learning in parameterized action space. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 2279–2285.

Levi Fussell, Kevin Bergamin, and Daniel Holden. 2021. Supertrack: Motion tracking for physically simulated characters using supervised learning. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–13.

Félix G Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. 2020. Robust motion in-betweening. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 60–1.

Mohamed Hassan, Duygu Ceylan, Ruben Villegas, Jun Saito, Jimei Yang, Yi Zhou, and Michael J Black. 2021. Stochastic scene-aware motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 11374–11384.

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

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144* (2016).

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.

Boyan Li, Hongyao Tang, YAN ZHENG, HAO Jianye, Pengyi Li, Zhen Wang, Zhaopeng Meng, and LI Wang. 2021a. HyAR: Addressing Discrete-Continuous Action Reinforcement Learning via Hybrid Action Representation. In *International Conference on Learning Representations*.

Jiefeng Li, Siyuan Bian, Qi Liu, Jiasheng Tang, Fan Wang, and Cewu Lu. 2023. NIKI: Neural Inverse Kinematics with Invertible Neural Networks for 3D Human Pose and Shape Estimation. *arXiv preprint arXiv:2305.08590* (2023).

Jiefeng Li, Chao Xu, Zhicun Chen, Siyuan Bian, Lixin Yang, and Cewu Lu. 2021b. Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3383–3393.

Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. 2020. NTU RGB+D 120: A large-scale benchmark for 3D human activity understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 10 (2020), 2684–2701.

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.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. 2015. SMPL: A skinned multi-person linear model. *ACM transactions on graphics (TOG)* 34, 6 (2015), 1–16.

Qiujing Lu, Yipeng Zhang, Mingjian Lu, and Vwani Roychowdhury. 2022. Action-conditioned On-demand Motion Generation. In *Proceedings of the 30th ACM International Conference on Multimedia*. 2249–2257.

Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. 2019. AMASS: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*. 5442–5451.

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 *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, Joaquin Vanschoren and Sai-Kit Yeung (Eds.). https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/28dd2c7955ce926456240b2ff0100bde-Abstract-round2.html

Josh Merel, Leonard Hasenclever, Alexandre Galashov, Arun Ahuja, Vu Pham, Greg Wayne, Yee Whye Teh, and Nicolas Heess. 2018. Neural probabilistic motor primitives for humanoid control. *arXiv preprint arXiv:1811.11711* (2018).

Josh Merel, Saran Tunyasuvunakool, Arun Ahuja, Yuval Tassa, Leonard Hasenclever, Vu Pham, Tom Erez, Greg Wayne, and Nicolas Heess. 2020. Catch & Carry: reusable neural controllers for vision-guided whole-body tasks. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 39–1.

Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. 2018. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 6292–6299.

Soohwan Park, Hoseok Ryu, Seyoung Lee, Sunmin Lee, and Jehee Lee. 2019. Learning predict-and-simulate policies from unorganized human motion data. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–11.

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel Van de Panne. 2018. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)* 37, 4 (2018), 1–14.

Xue Bin Peng, Michael Chang, Grace Zhang, Pieter Abbeel, and Sergey Levine. 2019. Mcp: Learning composable hierarchical control with multiplicative compositional policies. *Advances in Neural Information Processing Systems* 32 (2019).

Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. 2022. ASE: Large-Scale Reusable Adversarial Skill Embeddings for Physically Simulated Characters. *ACM Transactions on Graphics (TOG)* 41, 4, Article 94 (jul 2022), 17 pages. https://doi.org/10.1145/3528223.3530110

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 (TOG)* 40, 4 (2021), 1–20.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. 2017. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087* (2017).

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *CoRR* abs/1707.06347 (2017). arXiv:1707.06347 http://arxiv.org/abs/1707.06347

NUS SFU. 2011. SFU Motion Capture Database. https://mocap.cs.sfu.ca/.

Dana Sharon and Michiel van de Panne. 2005. Synthesis of controllers for stylized planar bipedal walking. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2387–2392.

Mingyi Shi, Kfir Aberman, Andreas Aristidou, Taku Komura, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. 2020. MotioNet: 3D Human Motion Reconstruction from Monocular Video with Skeleton Consistency. *arXiv preprint arXiv:2006.12075* (2020).

Yi Shi, Jingbo Wang, Xuekun Jiang, and Bo Dai. 2023. Controllable Motion Diffusion Model. *arXiv preprint arXiv:2306.00416* (2023).

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*.

Chen Tessler, Yoni Kasten, Yunrong Guo, Shie Mannor, Gal Chechik, and Xue Bin Peng. 2023. CALM: Conditional Adversarial Latent Models for Directable Virtual Characters. *arXiv preprint arXiv:2305.02195* (2023).

Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. 2022. Human motion diffusion model. *arXiv preprint arXiv:2209.14916* (2022).

Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 5026–5033.

Shuhei Tsuchida, Satoru Fukayama, Masahiro Hamasaki, and Masataka Goto. 2019. AIST Dance Video Database: Multi-Genre, Multi-Dancer, and Multi-Camera Database for Dance Information Processing.. In *ISMIR*, Vol. 1. 6.

Ziniu Wan, Zhengjia Li, Maoqing Tian, Jianbo Liu, Shuai Yi, and Hongsheng Li. 2021. Encoder-decoder with multi-level attention for 3D human shape and pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 13033–13042.

Jingbo Wang, Yu Rong, Jingyuan Liu, Sijie Yan, Dahua Lin, and Bo Dai. 2022. Towards Diverse and Natural Scene-aware 3D Human Motion Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20460–20469.

Tingwu Wang, Yunrong Guo, Maria Shugrina, and Sanja Fidler. 2020. Unicon: Universal neural controller for physics-based character motion. *arXiv preprint arXiv:2011.15119* (2020).

Wenjia Wang, Yongtao Ge, Haiyi Mei, Zhongang Cai, Qingping Sun, Yanjun Wang, Chunhua Shen, Lei Yang, and Taku Komura. 2023. Zolly: Zoom Focal Length Correctly for Perspective-Distorted Human Mesh Reconstruction. *arXiv preprint arXiv:2303.13796* (2023).

Jungdam Won, Deepak Gopinath, and Jessica Hodgins. 2020. A scalable approach to control diverse behaviors for physically simulated characters. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 33–1.

Jungdam Won, Deepak Gopinath, and Jessica Hodgins. 2022. Physics-based character controllers using conditional VAEs. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–12.
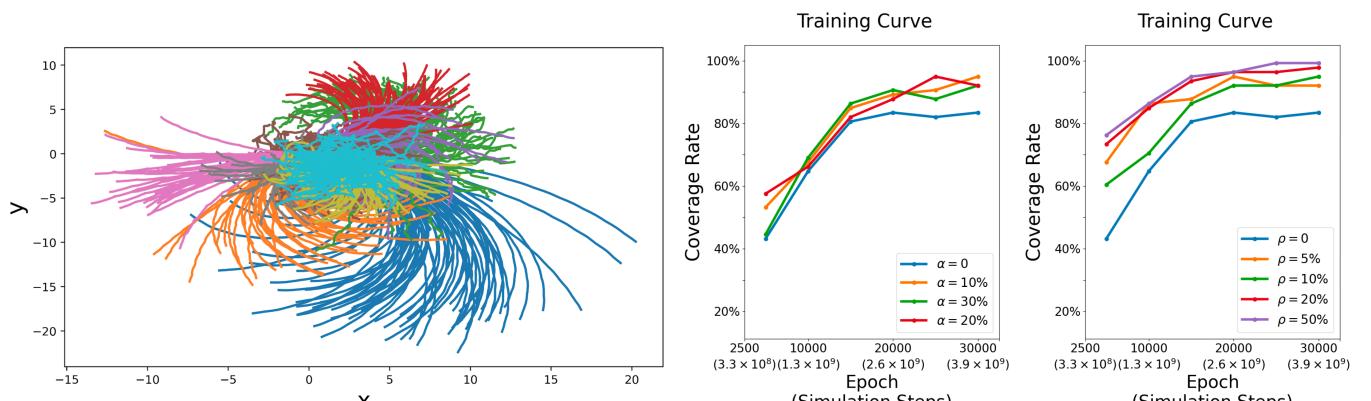
Heyuan Yao, Zhenhua Song, Baoquan Chen, and Libin Liu. 2022. ControlVAE: Model-Based Learning of Generative Controllers for Physics-Based Characters. *ACM Transactions on Graphics (TOG)* 41, 6 (2022), 1–16.

Ye Yuan and Kris Kitani. 2020. Residual force control for agile human behavior imitation and extended motion synthesis. *Advances in Neural Information Processing Systems* 33 (2020), 21763–21774.
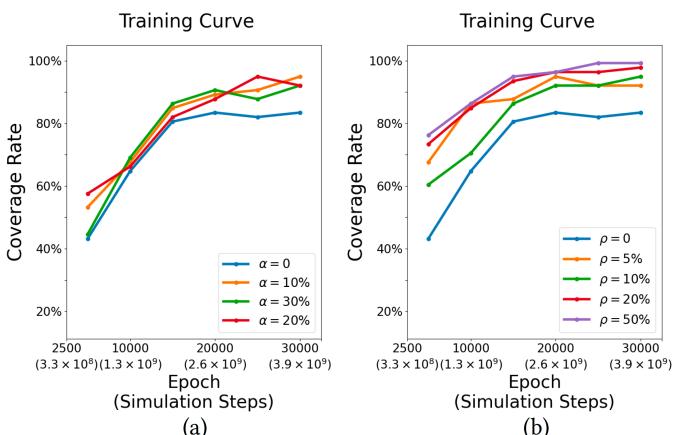
Figure 7: Transition coverage and probability between different skills. We show all 87 motion clips in the Sword&Shield dataset. Our model achieves a higher coverage rate and more even transition probability distribution compared with ASE and CALM.
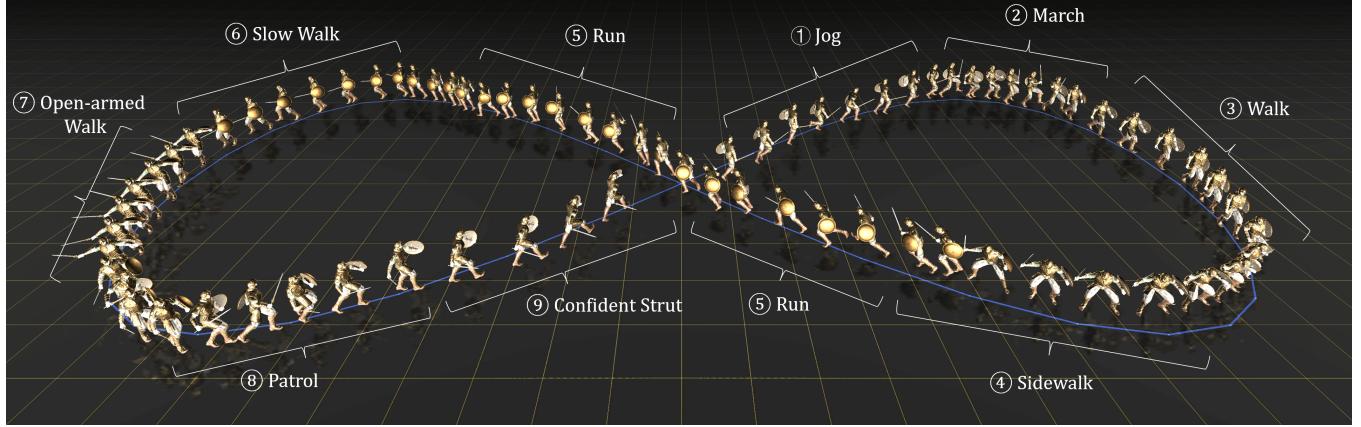

Figure 8: Motion variations (three colored in red, blue, and green are shown) are shown for each skill category. We test with the Sword&Shield dataset.
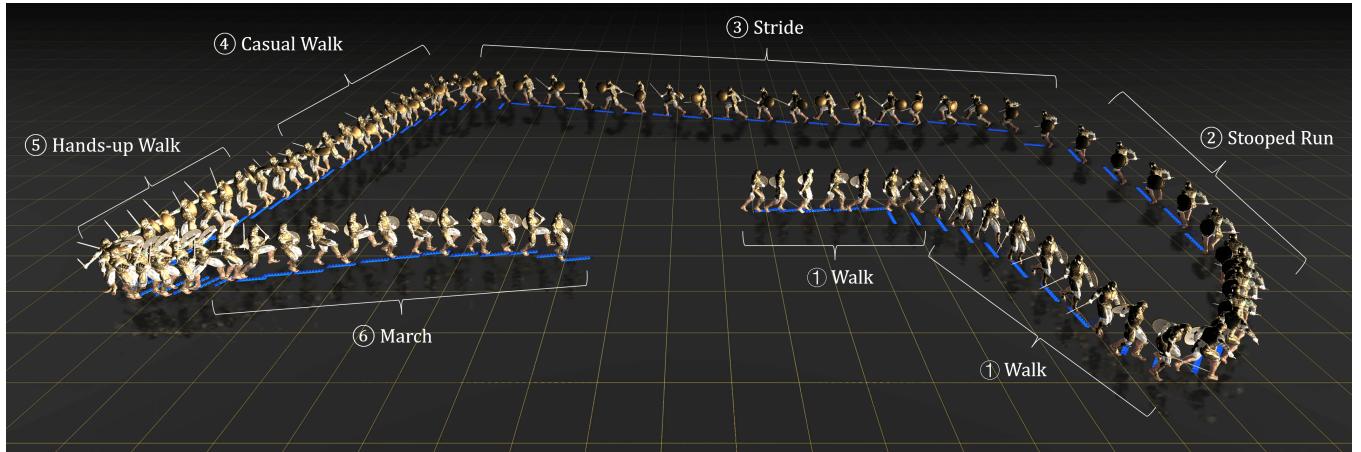

Figure 9: Visualization of trajectories of the character's root produced by random exploration with the low-level model. The trajectories are generated from random skill labels and latent codes. Starting from the same idle states, the character is able to move in different directions with realistic motions.
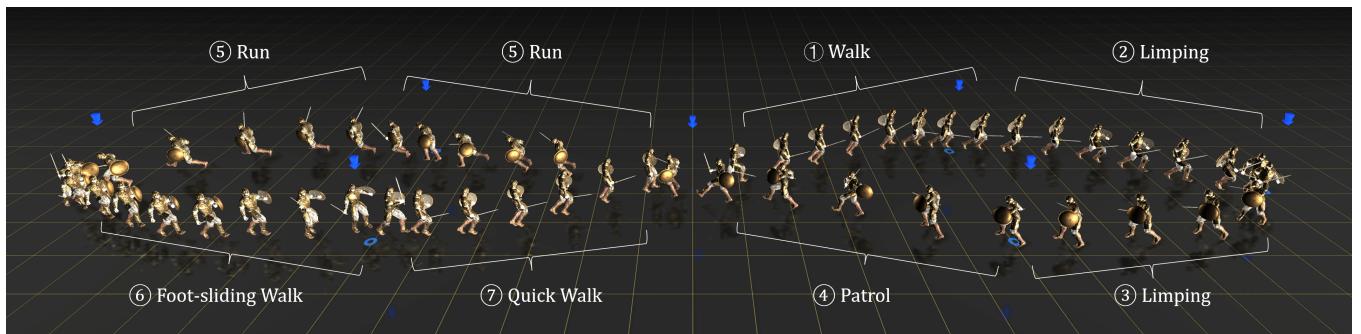

Figure 10: Influence of (a) the Focal Skill Sampling and (b) Element-wise Feature Masking to the training of the low-level model.

Figure 11: Path-follower: the character faithfully follows user-specified paths (blue path) under various specified skills, including jog, march, walk, sidewalk, run, slow walk, open-armed walk, patrol, and confident strut; More results are presented in the supplementary video.
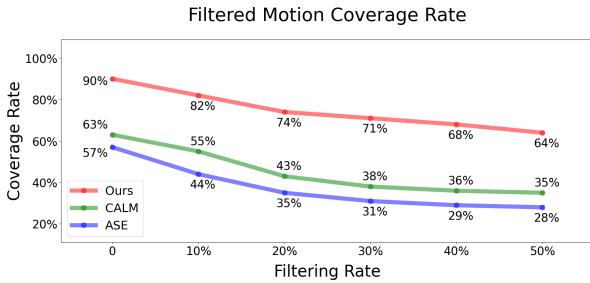

Figure 12: Directional Control: the character faithfully follows user-specified local moving direction (blue arrow) under various specified skills, including walk, stooped run, stride, casual walk, hands-up walk, and march; See more results in the supplementary video.

Figure 13: Character Re-locating: the character re-locates to a user-specified target location (visualized by the blue disks on the ground and arrows in the air) with different specified skills, including walk, limping, patrol, run, foot-sliding walk, and quick walk; More results can be found in the supplementary video.
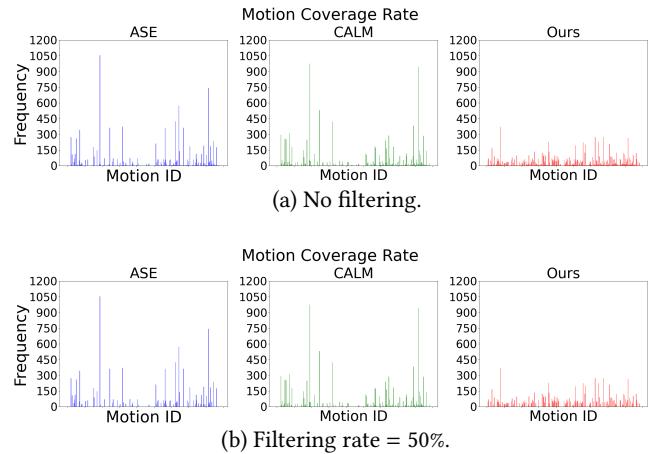
A   APPENDIX

A.1   Scalability

Figure A14: Comparison of the motion coverage on the Composite Skills dataset with 778 clips of 265 motion skills. The coverage rate of ASE and CALM falls dramatically with an increasing filtering rate, implying a serious unbalance of the coverage, whereas ours consistently produces high motion coverage rates.

Figure A14 presents the motion coverage rate calculated using different filtering rates. We can see ASE's coverage dramatically drops to 44%, 35%, and 28% at the filtering rate of 10%, 20%, and 50% filtering rate, respectively. The coverage rate of CALM drops from 55% to 43%, and 35% at the filtering rate of 10%, 20%, and 50% filtering rate. Meanwhile, we observed that CALM required more training steps to achieve a reasonable motion coverage rate. This implies a serious imbalance and instability of the motion coverage, i.e., many motion clips are matched only a few times, possibly due to stochastic factors existing in the randomly generated trajectories, as evidenced by Figure A15, that records the frequencies at which π produces trajectories that matched each motion clip in the dataset across $10,000$ trajectories.

A.2   Skill Label

A.2.1   Skill Label Acquisition. We investigate the performance of C·ASE on unorganized mocap data. The key to processing unorganized motion datasets is label acquisition. We demonstrate that an action recognition network [Duan et al. 2022] pretrained on NTURGB+D 120 dataset [Liu et al. 2020] can produce reliable skill labels. We test with raw unstructured motion clips from CMU Mocap dataset [CMU 2002] by finetuning the network on the aforementioned Composite Skills dataset, where we set the learning rate to be 0.1 with momentum being 0.9 and weight decay being $5{\times}10^{-4}$. The model is trained on a single A100 GPU. We set the batch size to be 128 and train for 80 epochs. We randomly divided the data set into the training set and test set by 9:1, and the top-1 accuracy on the test set was 76.9%. As illustrated in Figure A16, C·ASE effectively embeds the segmented skills. Note that the data in Figure A16 is taken from the test set. As demonstrated in Figure A17, our approach successfully captured the reference skills corresponding to each skill condition from the original unstructured motion clips.

In fact, using the skill label as a condition brings a lot of flexibility, which makes our method compatible with various input sources. For instance, methods that directly regress the skeleton rotation [Shi
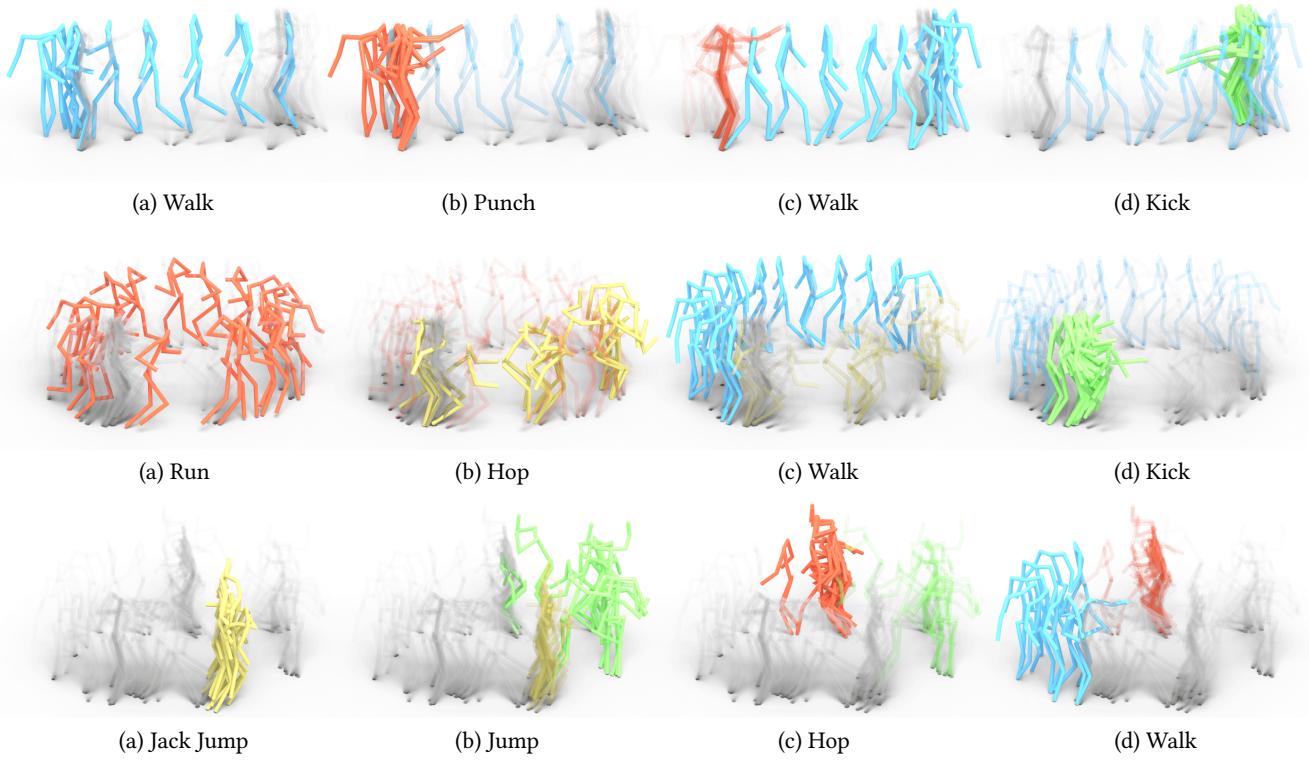
(a) No filtering.

(b) Filtering rate = 50%.

Figure A15: Frequencies at which the low-level policy produces motions that match all 265 motion skills in the Composite Skills dataset. We show distributions produced by the filtering rate of 0% and 50%. Compared to ASE and CALM, our method produces diverse motions that much more evenly cover all reference clips.

Table A2: Filtered motion coverage of C·ASE with randomly assigned labels. We report the performance at different filtering rates.
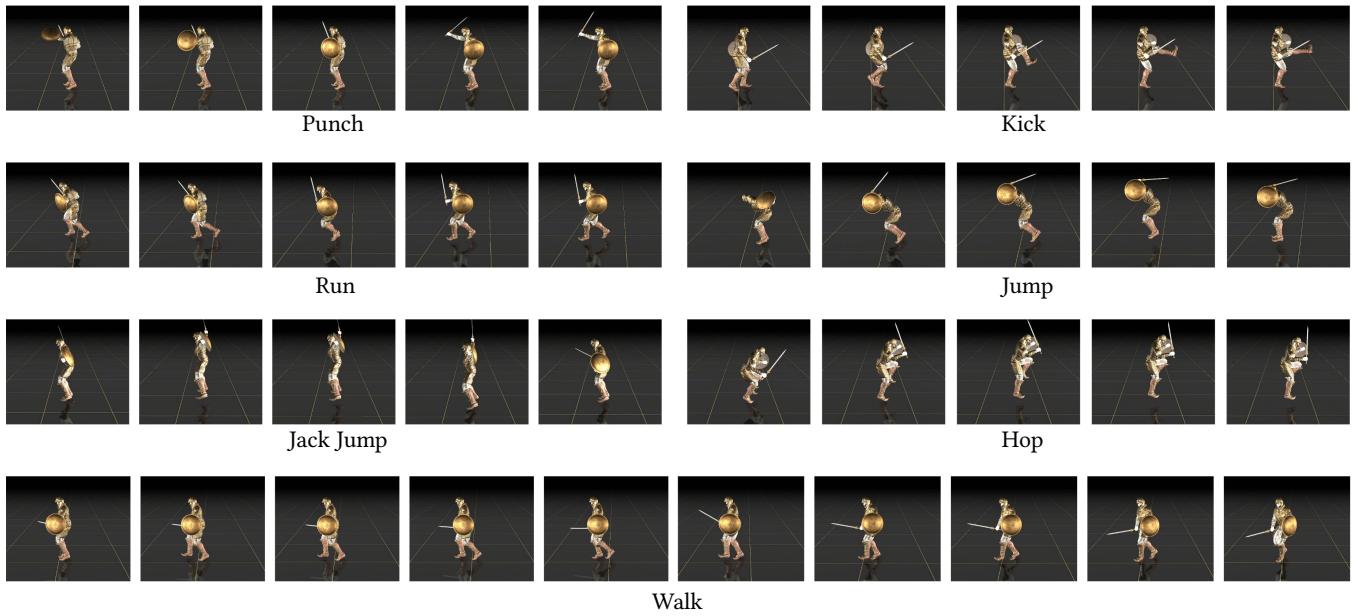
# Random	Filtering Rate					
Label	0	10%	20%	30%	40%	50%
16	78	57	45	43	41	40
32	87	62	55	52	51	49
64	90	83	79	77	77	75

et al. 2020] or joint rotation [Cai et al. 2022; Dou et al. 2022; Li et al. 2023, 2021b; Wan et al. 2021; Wang et al. 2023] of the parametric model, e.g., SMPL [Loper et al. 2015], from the video, can be directly applied to our labeling. Moreover, skill labels are typically aligned with the language of skill descriptions, which could open the door for language-guide motion control.

A.2.2   Random Labeling. Regarding the influence of labeling, we investigate the performance of C·ASE at different levels of randomness where we train C·ASE on Sword&Shield dataset containing 87 motions with fewer labels: that are *randomly* assigned to each clip. Due to the randomness of labeling, samples under each skill label are more heterogeneous than those in our default setting. The result can be found in Table A2. From Table A2, we could see that by randomly grouping the skills with assigned labels for skill embedding, the motion coverage may improve to a certain extent, but not significantly. Specifically, when the number of random labels is small (fewer groups), motion clips under each group are more heterogeneous. Thus, the improvement in the efficiency of skill embedding becomes limited. In fact, an efficient conditional skill embedding calls for not only skill partitions but also to ensure that the motions within each skill group are as homogeneous as possible.

(a) Walk   (b) Punch   (c) Walk   (d) Kick

(a) Run   (b) Hop   (c) Walk   (d) Kick

(a) Jack Jump   (b) Jump   (c) Hop   (d) Walk

Figure A16: Motion Segmentation results on unstructured motion clips. Each row represents one unstructured motion clip from the CMU Mocap dataset [CMU 2002]. We highlight each segmented skill in a specific color.

Punch   Kick

Run   Jump

Jack Jump   Hop

Walk

Figure A17: Various skills performed by C·ASE under different skill conditions based on the motion segmentation results.

A.3　More High-Level Tasks

In the following, we evaluate C·ASE in various representative high-level tasks, such as *Reach*, *Steering*, *Location*, and *Strike*, following [Peng et al. 2022]. For detailed task configurations and goal reward designs, we refer readers to [Peng et al. 2022]. Notably, in this paper, we automatically filter out motion clips whose variation of projected root trajectory on XY-plane is less than 0.3m within one episode for various downstream tasks with an emphasis on locomotion skills.

Hybrid Action Space. To achieve the task-training objective, the high-level policy learns to output an optimal configuration of both the skill label c and the latent transition code z, sampled from the *discrete* skill set C and the *continuous* latent space $\mathcal{Z}$. Unlike previous work [Peng et al. 2022; Yao et al. 2022] with purely continuous action spaces, our high-level policy training falls into discrete-continuous action reinforcement learning [Fan et al. 2019; Li et al. 2021a]. Directly regressing the discrete skill label and latent transition code hinders learning and results in poor performance. Instead, we treat skill label prediction as a classification problem, with the high-level policy outputting a continuous probability distribution over skill labels. The skill label c is predicted as a continuous probability distribution and converted to the skill label index using Gumbel-Softmax [Jang et al. 2016] for the low-level controller, while latent z is drawn from a spherical Gaussian distribution.

Following ASE [Peng et al. 2022], we also reuse the pre-trained low-level model and train a separate high-level policy for each task with no human intervention, including *Location*, *Strike*, *Reach*, and *Steering*. As depicted in Figure A18, our model successfully accomplishes the given tasks while maintaining realistic motions, thanks to the pre-trained conditional adversarial embeddings. Specifically, when trained to strike a target object, the character seamlessly transitions from the locomotion state (e.g., running) to the sword attack skill, effectively and naturally completing the task. The character is also capable of resisting external perturbations; see more results in the supplementary video.

A.4　Implementation Details

A.4.1　Definition of State and Action. The state $\mathbf{s}$ for the discriminator D and encoder q is defined by the set of the height of the root from the ground, rotation of the root in the local coordinate frame, linear and angular velocity in the local coordinate frame, local rotation of each joint, local velocity of each joint, positions of hands, feet in the local coordinate frame. For low-level policy network π and high-level policy network ω, the state is additionally with the position of the shield and the tip of the sword in the local coordinate frame. The action $\mathbf{a}$ is in 31-D, specifying the target rotations for the PD controller at each joint.

A.4.2　Condition Randomization. At the beginning of each episode, we randomize the $(\mathbf{s}, c)$ pairs independently, selecting a random reference state from the dataset and a random skill label from the set of skill labels. This approach forces the character to perform the desired skill from an arbitrary state while producing realistic transitions between various skills, as we shall demonstrate. Similar strategies have been used in [Nair et al. 2018; Peng et al. 2018; Rajeswaran et al. 2017; Sharon and van de Panne 2005].

In the context of GAN, Randomly Initialized State (RIS) and Random Condition Signals (RCS) are applied to the policy that serves as a generator. Since the discriminator is taking paired reference motions and their condition signals, the generator (policy) will learn to match each condition signal to the corresponding motion. Combinatorial variations brought by RIS and RCS enable the policy to be robust and insensitive to the exact boundary of labels and allow for natural transitions between various skills, as shown in the video. We found that a bigger and more diverse dataset, i.e., the Composite Skills dataset, will not reduce our performance.

A.4.3　Network Structure. We implement the policy networks π and ω, value function V, encoder q and discriminator D using MLP. Specifically, the low-level policy π is modeled by a neural network that maps a state $\mathbf{s}$ and latent $\mathbf{z}$ and skill label c to a Gaussian distribution over actions with a fixed diagonal covariance matrix. The network is implemented by the MLP with ReLU units and hidden layers containing $[1024, 1024, 512]$. Note that the discrete skill label c is embedded into a 64D space and concatenated to the input before being passed to the first layer of the network. A similar structure is used for the value function V but with a single linear output unit.
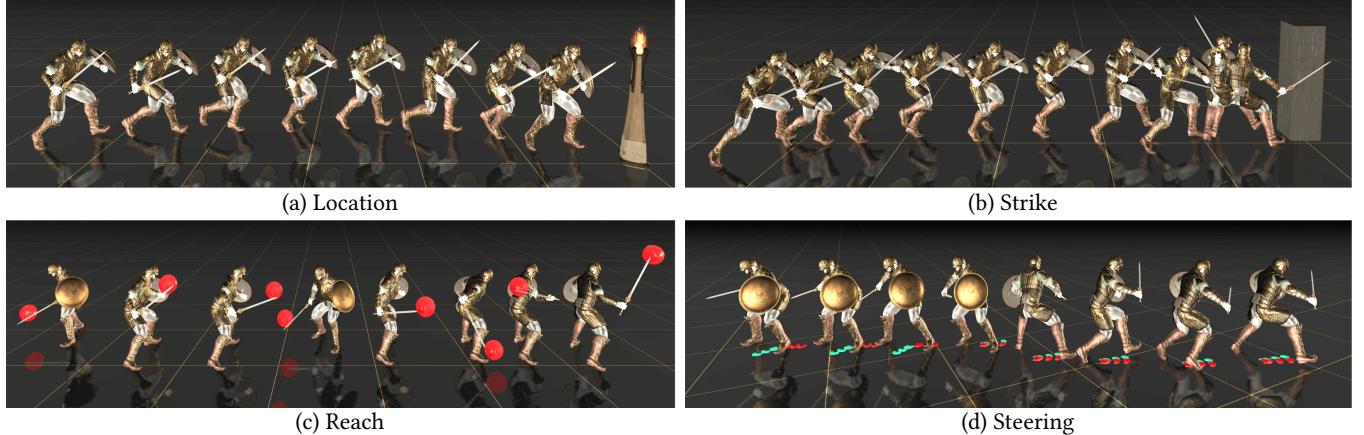
Different from ASE [Peng et al. 2022] that models the encoder and discriminator as one single network with different output heads, in our model, the encoder is modeled as one network using MLP of hidden layers containing $[1024, 1024, 512]$ with ReLU units while the discriminator is implemented in another network using MLP of hidden layers containing $[1024, 512]$ with ReLU units for discrimination. To condition the discriminator D, we increase the dimension of the output layer to be the number of skills and then the feature specified by c is used for discrimination. The conditional encoder is implemented similarly to π, where we embed the skill label and concatenate it to the input.

The high-level policy ω is implemented with 2 hidden layers with $[1024, 512]$ units, followed by a linear layer to predict the mean of latent code and skill label. Then the discrete skill label is obtained by taking the class with the highest probability.

A.5　Training Details

For each skill label c, we embed it into a 64D latent space. And the latent space for $\mathbf{z}$ is 16D. The random probability for Element-wise Feature Masking is set to be $\rho = 20\%$ while the update rate α is set to be 20% in Focal Skill Sampling. The other hyperparameter settings for low-level policy and high-level policy, as well as corresponding value functions, are the same as ASE [Peng et al. 2022]. Following ASE, we also randomize the sequence of latent codes during low-level policy training. Each latent is fixed for between 1 and 150 time steps before the update. The buffer length for the discriminator and encoder is set to be 20. During task training, the frequency of high-level policy and low-level policy are 6Hz and 30Hz, respectively. The skill label and latent produced by the high-level policy are repeated for 5 steps for the low-level policy.

We integrate reference motion from the CMU dataset into the conditional discriminator using adaptive masking for sword and shield components. Specifically, as the skill label serves as a condition when obtaining skill labels from the Sword&Shield dataset, we consider key points, including those for the sword and shield,

(a) Location

(b) Strike

(c) Reach

(d) Steering

Figure A18: A physically Simulated character can be trained to complete tasks using skills from the pre-trained low-level policy. The character successfully achieves the goals of high-level tasks while producing naturalistic behaviors. (a) Location: the character moves towards the target location, i.e., torch. (b) the character moves to the target object and knocks it over with its sword. (c) Reach: the character positions the tip of the sword at a target location. (d) Steering: the character moves along a target direction (red arrow) while facing a target heading direction (green arrow).

during the discriminator process. For skills derived from the CMU Mocap dataset, we apply masking to the sword and shield key points for both reference motion (real samples) and physically simulated motion (fake samples). This approach eliminates the need for the simulated character's sword and shield positions to align with the reference motion, as these positions may be unreliable during the retargeting process from the CMU Mocap dataset.

Furthermore, we employ curriculum learning for efficient training. At the beginning of the training, we disallow body-ground contact for the first 15000 epochs to accelerate the training process,

then permit it in subsequent epochs to learn those hard-to-learned movements.

A.6 Failure Cases

In this section, we show failure cases, e.g., characters only launch idling motion when conditioned on a skilled label that is not embedded well. Details can be found in our supplementary video. We also observed that there is a trend of coverage improving these skills in C·ASE as training proceeds.