ModSkill: Physical Character Skill Modularization

Supplementary Material

This supplementary document provides additional implementation details (Sec. 1), an analysis of joint-level tracking errors comparing our method with PHC+ [37] (Sec. 2), and task specifications along with training curves for downstream tasks (Sec. 3). Extensive qualitative results are available in the supplementary video, where we demonstrate our method's capability to imitate various reference motion data and perform different downstream tasks.

1. Implementation Details

In ModSkill, we use a humanoid agent based on the SMPL kinematic model [30, 35], which consists of 24 rigid bodies, 23 of which are actuated. We divide the 24 rigid bodies of the simulated character into five body parts, $\mathcal{P} := \{$Left Leg (L-Leg), Right Leg (R-Leg), Left Arm (L-Arm), Right Arm (R-Arm), Torso$\}$. The corresponding grouping of rigid bodies is detailed in Tab. 6

Table 6. Body part grouping for skill modularization.

Body Part	Rigid Bodies
L-Leg	L-Hip, L-Knee, L-Ankle, L-Toe
R-Leg	R-Hip, R-Knee, R-Ankle, R-Toe
Torso	Pelvis, Torso, Spine, Chest, Neck, Head
L-Arm	L-Thorax, L-Shoulder, L-Elbow, L-Wrist, L-Hand
R-Arm	R-Thorax, R-Shoulder, R-Elbow, R-Wrist, R-Hand

Based on the body part grouping, we partition the input state accordingly. The Skill Modularization Attention Layer uses two-layer MLPs with dimensions [256, 64] to project the partitioned input states for each body part into keys, queries, and values, enabling attention across body parts. The resulting skill embeddings, with a dimensionality of 64, are then normalized to lie on the unit sphere. For each body part, we assign a low-level controller, implemented as a four-layer MLP with dimensions [2048, 1536, 1024, 512], which takes the corresponding skill embedding as input and produces PD targets for the actuated rigid bodies within that body part grouping. The policy network is trained for approximately 2×10^9 steps, with a learning rate of 2×10^{-5}, using reward function weights as specified in [37]. Generative adaptive sampling is applied every 2×10^8 steps. For each failed sequence, we utilize the corresponding text-label from HumanML3D [9] to generate $N = 3$ synthetic sequences using an off-the-shelf text-to-motion model [54]. Note that, in this work, we use a text-to-motion model to provide the generative prior for adaptive sampling, but alternative motion generation models, such as unconditional VAE-based models, can also be used. Our generative adaptive sampling strategy is not limited to text-to-motion models.

Note that ModSkill is a skill-learning framework designed to be flexible with arbitrary body-part configurations. We adopted a 5-part setting that allows for clear separation between limbs and is consistent with prior work on part-based modeling [16]. In Tab. 7, we show additional motion tracking results on AMASS-Test for two-part (Part 1: L-Arm, R-Arm, Torso; Part 2: L-Leg, R-Leg) and three-part (Part 1: L-Arm, R-Arm; Part 2: Torso; Part 3: L-Leg, R-Leg) configurations. We also detail a comparison of the corresponding model sizes for different body part configurations using ModSkill against PHC+ [37]. Despite the smaller model size compared to PHC+, our five-part modularization approach leads to more precise motion tracking and enables the formulation of reusable, modular skills without the need for progressive mining. Additionally, for the two-part and three-part configurations, our framework demonstrates smaller model sizes than the full-body baseline while achieving competitive performance, highlighting the advantages of skill modularization.

2. Joint-wise Tracking Error

Full-body tracking requires high precision. We present comprehensive joint-wise tracking error statistics across the five body parts defined in our skill modularization framework: right arm (Tab. 8), left arm (Tab. 9), right leg (Tab. 11), left leg (Tab. 10), and torso (Tab. 12). These detailed analyses highlight the effectiveness of skill modularization in achieving precise, high-fidelity physical character skill learning, demonstrating its potential for fine-grained motion control across distinct body parts.

Additionally, we observe higher errors at the end-effector joints (e.g., R-Toe, L-Toe, R-Hand, L-Hand) for both ModSkill and PHC+, suggesting that more complex motion samples or refined modeling of these joints may be needed. We also note increased errors in the right leg joints for our method compared to other body parts, which may indicate a bias or imbalance in the training data. However, the modular nature of our framework offers key advantages: we can not only isolate part-specific controllers during training but also leverage active learning to generate additional synthesized samples, helping to support skill learning for specific body parts.

Table 7. Motion Tracking Evaluation and Policy Network Model Size for Different Body-Part Configurations.

Method	Succ ↑	$E_{\text{g-mpjpe}}$ ↓	E_{mpjpe} ↓	E_{acc} ↓	E_{vel} ↓	Model Size (MB)
PHC+	99.2%	36.1	24.1	6.2	8.1	∼609
2-Part	97.1%	36.4	25.9	5.0	7.2	∼243
3-Part	98.6%	36.1	25.3	4.8	6.9	∼306
ModSkill (Ours)	**99.3%**	**32.2**	**22.7**	**4.4**	**6.3**	∼430

Table 8. **R-Arm:** Joint-wise Mean Position Errors (mm) on AMASS-Test.

Method	R-Thorax	R-Shoulder	R-Elbow	R-Wrist	R-Hand
PHC+	42.8	41.4	37.0	40.1	39.0
ModSkill	**20.6**	**21.2**	**21.8**	**29.8**	**30.5**

Table 9. **L-Arm:** Joint-wise Mean Position Errors (mm) on AMASS-Test.

Method	L-Thorax	L-Shoulder	L-Elbow	L-Wrist	L-Hand
PHC+	45.0	45.5	43.3	40.3	41.7
ModSkill	**20.6**	**21.4**	**27.0**	**29.3**	**31.5**

Table 10. **L-Leg:** Joint-wise Mean Position Errors (mm) on AMASS-Test.

Method	L-Hip	L-Knee	L-Ankle	L-Toe
PHC+	38.7	43.4	52.5	55.7
ModSkill	**20.1**	**26.3**	**31.7**	**33.9**

Table 11. **R-Leg:** Joint-wise Mean Position Errors (mm) on AMASS-Test.

Method	R-Hip	R-Knee	R-Ankle	R-Toe
PHC+	36.4	41.4	50.2	50.9
ModSkill	**21.7**	**31.6**	**38.9**	**47.9**

Table 12. **Torso:** Joint-wise Mean Position Errors (mm) on AMASS-Test.

Method	Pelvis	Torso	Spine	Chest	Neck	Head
PHC+	36.9	39.1	43.2	43.8	44.2	48.5
ModSkill	**19.7**	**21.0**	**22.0**	**21.6**	**20.0**	**20.5**

Following prior work [35], we use early termination during training and evaluation, where tracking terminated (i.e. considered a failure) if the average deviation of simulated joint positions from the reference exceeds 0.5 meters. However, this threshold is relatively lenient, as both the use of average deviation and the large 0.5-meter margin may overlook significant joint-wise errors that aren't fully captured by the current evaluation metrics. In Fig. 10, we report the full-body tracking success rate on AMASS-Test for termination distances ranging from 0.5 meters to 0.2 meters, with a step size of 0.05 meters. Notably, we observe an immediate decline in performance for PHC+ as the termination distance decreases, whereas our method maintains a higher success rate even under stricter termination conditions. This suggests that our approach is more robust, able to generalize effectively, and consistently preserves higher accuracy across a wider range of unseen sequences.

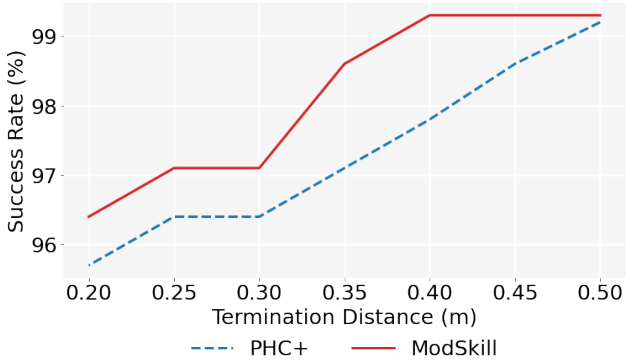

Figure 10. Effect of decreasing termination distance on full-body motion tracking success rate for AMASS-Test.

3. Downstream Tasks

For generative downstream tasks, steer, reach, strike, and VR-tracking, we utilize a three-layer MLP with dimensions [2048, 1024, 512] for the high-level policy and train the policy using PPO [49] for approximately 2×10^9 steps with a learning rate of 2×10^{-5}. VR-tracking follows the reward function for the full-body tracking tasks. Please refer to our supplementary video for a comprehensive evaluation of our model on these tasks. Following [37, 47], the goal state and reward formulations for steer, reach, and strike are detailed below:

Steering. The goal state is defined as $s^t_{\text{steer}} := (d_t, v_t)$, where d_t and v_t represent the target direction and the desired linear velocity at timestep t, respectively. The objec-

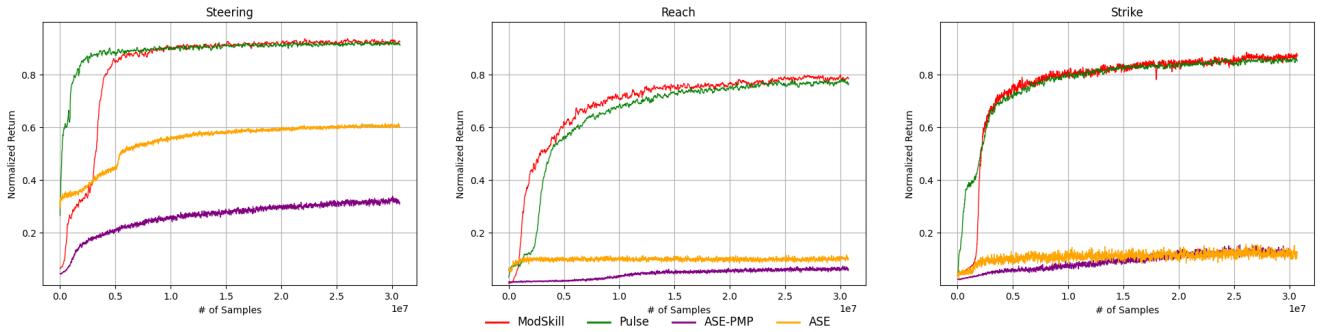

Figure 11. Normalized returns during training for downstream tasks.

tive of the policy is to control the character to travel along the specified direction at the desired velocity. The reward is defined as $r_{\text{speed}} = |v_t - v_{t0}|$, where v_t^0 is the root velocity of the humanoid.

Reach. In the reach task, we aim to minimize the distance between the simulated character's right hand and a desired 3D target point, c_t, randomly sampled from a 2-meter box centered at $(0, 0, 1)$. The goal state is $s_{\text{reach}}^t \equiv (c_t)$. Let $p_{\text{R-Hand}}$ be the position of the simulated character's right hand. The reward for reaching is calculated as the exponential of the negative squared distance between the right-hand position and the desired target point:

$$r_{\text{reach}} = \exp\left(-5\|p_{\text{R-Hand}} - c_t\|_2^2\right),$$

Strike. The objective of this task is to knock over a target object. We select the rigid bodies, R-Hand, R-Wrist, R-Elbow as the target body parts for contact, where the task terminates (e.g. considered a failure) if any body part other than the target body parts makes contact with the target object. The goal state $s_{\text{strike}}^t \equiv (x_t, \dot{x}_t)$ consists of the position and orientation x_t, linear and angular velocities $\dot{x}_t$ of the target object in the simulated character's frame of reference. The reward is defined as $r_{\text{strike}} = 1 - \mathbf{u}_{\text{up}} \cdot \mathbf{u}_t$, where $\mathbf{u}_{\text{up}}$ is the global up vector, and $\mathbf{u}_t$ is the up vector of the target object.

As shown in Fig. 11, we provide the training curves for the steering, reach, and strike tasks. We observe that our method initially exhibits slower progress than SOTA models, especially for tasks that utilize full-body template movements, such as speed and strike, due to the added complexity of modular skill spaces. However, the learning curve accelerates quickly, ultimately catching up to SOTA models with a similar convergence speed and scale of normalized returns. For more precise tasks, like reaching, that require targeted control of specific body parts, our method demonstrates faster learning, indicating better flexibility for character control. ASE-PMP shows similar trends, where part-level reward signals lead to a more complex skill learning process. For more precise tasks, like reaching, which targets specific body parts, our method demonstrates faster learning, indicating better flexibility. Additionally, when PMP is applied to a single large-scale dataset rather than multiple small-scale specialized datasets of task-specific scenarios, this can lead to less effective skill learning and transfer.