

Learning Physics-Based Full-Body Human Reaching and Grasping from Brief Walking References

Supplementary Material

1. Methods Details

1.1. Character and Degrees of Freedom

To assess the effectiveness of our approach to modeling a character’s walking and manipulation behaviors, we use our framework to train sophisticated 3D simulated characters to perform a range of tasks. This involves using a character based on AMP [11], where we have modified the spherical hand structure into a more dexterous configuration, such as the Shadow Hand [3]. The humanoid model consists of a body with 15 joints featuring 28 degrees of freedom (DoF) and a dexterous hand with 23 joints providing 25 DoF.

1.2. State and Observation

As our humanoid character is an adaptation of AMP [11] and ShadowHand [3], the state is primarily derived from AMP body states with modifications to the right hand using UniDexGrasp hand states [15]. This state comprises a collection of features detailing the configuration of the character’s body and its dexterous hand. These features encompass:

- Height of the root from the ground.
- Rotation of the root in local coordinates.
- Root’s velocity (linear and angular) in local coordinates.
- Local rotation of each joint.
- Local velocity of each joint.
- Local rotation of right hand joints.
- Local velocity of right hand joints.
- Local position of right hand joints.
- Right hand joint DoF.
- Velocity of right hand joint DoF.
- Positions of feet, left hand, and right fingertips in local coordinates.

In low-level policy training, observations consist of body states. For high-level downstream tasks, observations encompass full-body states along with task-specific data. For instance, in a grasping task, observations may include the object’s and table’s position and rotation, as well as the distance between the right fingertips and the target object.

1.3. Stage-based Task-specific Reward

As outlined in the primary document, the high-level policy is optimized using rewards tailored to specific tasks, along with motion-prior rewards. In reaching and grasping tasks, task-specific rewards r_G are divided into four distinct phases:

1. direction and walking, encouraging correct facing and

target velocity.

2. pre-grasping, focusing on moving the hand to approach the grasp pose.
3. grasping, employing a reward mechanism similar to UnidexGrasp[15] to promote effective grasping.
4. post-grasping, aiming at maintaining the grasped object, with an additional reward for preserving body equilibrium.

In the first stage, the location reward is designed to guide the agent toward the target by combining three components: the position reward, the velocity reward, and the facing reward. The combined reward is formulated as:

$$r_{\text{location}} = w_{\text{pos}} \cdot r_{\text{pos}} + w_{\text{vel}} \cdot r_{\text{vel}} + w_{\text{face}} \cdot r_{\text{face}}, \quad (1)$$

where:

$$r_{\text{pos}} = \exp(0.5 \times \|\tilde{p}_{\text{tar}} - \tilde{p}_{\text{root}}\|^2), \quad (2)$$

$$r_{\text{vel}} = \exp(-4.0 \times (v_{\text{tar}} - (n_{\text{tar}} \cdot \tilde{v}_{\text{root}}))^2), \quad (3)$$

$$r_{\text{face}} = 1 + (n_{\text{tar}} \cdot \tilde{f}). \quad (4)$$

Here, v_{tar} is the target velocity. \tilde{v}_{root} , \tilde{p}_{tar} and \tilde{p}_{root} represent the x-y plane velocity of the root, the x-y plane positions of the target and the root, respectively. \tilde{f} represents the projection of the facing direction onto the x-y plane. n_{tar} is the normalized direction to the target:

$$n_{\text{tar}} = \frac{\tilde{p}_{\text{tar}} - \tilde{p}_{\text{root}}}{\|\tilde{p}_{\text{tar}} - \tilde{p}_{\text{root}}\|}. \quad (5)$$

Additional masks are applied to adjust the rewards based on proximity or alignment: - If $\|\tilde{p}_{\text{tar}} - \tilde{p}_{\text{root}}\| < 0.5$, all rewards are boosted. - If the agent is moving in the wrong direction ($n_{\text{tar}} \cdot \tilde{v}_{\text{root}} \leq 0$), the velocity reward is set to zero.

In the second stage, the reward can be computed by $r_{\text{reach}} = \exp(-\|p_{\text{tar}} - p_{\text{reach}}\|^2)$. Here, p_{reach} represents the position of the reaching end-effector (the palm center), while p_{tar} represents the ideal reaching position. The ideal position is defined as 0.2 m above the object, at a distance of one-third of the table’s width from its center, and oriented towards the direction from which the person approached.

In the third stage, the reward is defined by grasping-related metrics, similar in UnidexGrasp [15]. This reward, r_{grasp} , evaluates the agent’s ability to successfully lift and stabilize an object while maintaining effective grasp control. It combines three components: the grasp quality reward, the object height reward, and the object velocity

penalty. These components are weighted and combined as follows:

$$r_{\text{grasp}} = w_{\text{grasp}} \cdot r_{\text{grasp_quality}} + w_{\text{height}} \cdot r_{\text{height}} + w_{\text{obj-vel}} \cdot r_{\text{obj-vel}}, \quad (6)$$

where:

$$r_{\text{grasp_quality}} = 2 - 0.5 \cdot d_{\text{finger}} - 1.0 \cdot d_{\text{hand}}, \quad (7)$$

$$r_{\text{height}} = 0.1 + 0.5 \cdot \frac{h_{\text{object}}}{h_{\text{lift_target}}}, \quad (8)$$

$$r_{\text{obj-vel}} = -0.2 \cdot \text{clamp}(v_{\text{object}} - v_{\text{threshold}}, 0, 5). \quad (9)$$

Here, d_{finger} denotes the distance between the agent’s fingers and the object, while d_{hand} represents the distance between the agent’s hand and the object. h_{object} is the height of the object above the table, normalized by the target lifting height $h_{\text{lift_target}}$. The object height reward, r_{height} , is applied only if the agent is in contact with the object. v_{object} indicates the velocity of the object and $v_{\text{threshold}}$ is a velocity threshold used to penalize excessive movement. This reward structure effectively balances the need for precise grasping, stable lifting, and minimal object movement, guiding the agent to complete the task efficiently and effectively.

In the final stage, the reward is defined as r_{goal} , which combines the proximity of the agent to the target and its grasp quality. The reward is computed as:

$$r_{\text{goal}} = 3 \cdot r_{\text{location}} + 3 \cdot \text{clamp}(1.5 + r_{\text{grasp_quality}}, 0, 5). \quad (10)$$

Here, r_{location} represents the location reward as defined in the first stage, with the target position updated to match the goal in this stage.

Furthermore, we include an additional reward, r_{stage} , defined as a bonus granted when the transition condition is satisfied.

1.4. Transition Condition

To ensure smooth progression between stages, a set of transition conditions is defined. These conditions evaluate whether the agent has successfully completed the current stage’s objectives and is ready to move to the next stage. The conditions are:

- The distance between the root and the object is less than 1 meter.
- The palm is directly above the object, with a vertical distance of less than 0.1 meters.
- The object is lifted vertically by more than 0.1 meters relative to its initial position.

1.5. Feature Alignment Mechanism

We aim for the generated motions to accomplish diverse tasks at a macro-level, similar to the generated data, while preserving realistic patterns at a micro-level. These patterns represent common traits observed in real-world data. For instance, humans naturally synchronize their right hand and left foot to maintain balance, move their forearm driven by the upper arm, and exhibit joints that rarely bend or have limited bending angles. Intuitively, we hypothesize that such patterns arise from local observations. To incorporate this understanding, we modified the architecture of our critic network.

Previously, the network’s first layer directly processed the entire 223-dimensional full-body observation through three layers of MLPs. To better capture the hierarchical nature of motion patterns, we introduce five separate sub-networks, each dedicated to processing the local observations of specific body parts (torso and four limbs). Each sub-network performs an initial transformation on its respective input, producing part-specific features f_0 . These features are then passed into the subsequent shared layers of the network, enabling a more structured and progressive analysis.

This revised architecture allows the network to focus first on part-wise information, capturing localized patterns. Then it gradually integrates information across different parts, enabling the model to form increasingly global representations. This hierarchical progression ensures that both local patterns and global coherence are effectively captured.

As indicated in the pilot study, MoCap-Reach and MoCap-Walk exhibit evident clustering in the shallow layers of the network, but this clustering diminishes in the deeper layers. This observation suggests that, despite differences in orientation, real motions share common patterns, particularly at the part-wise level. Additionally, the first level of the network also demonstrates some degree of clustering, which provides insights into shared coordinate systems across different body parts. These patterns are referred to as "local features."

Inspired by this clustering behavior, we introduce a regularization term that encourages the generated motion to exhibit similar local feature locations. Specifically, we aim for the distance between two sets of features to remain within the variance of walking data. During space tuning, we derive the 10-step motion and input it into the original critic, which is trained on walking motions, to calculate the mean value of the features as the current feature $f_i(s, z)$.

We then introduce an additional reward to align the generated features with the pre-calculated walking feature distribution. Let the i -th feature have mean μ_i and variance σ_i respectively. The current feature is extracted, and the Mahalanobis distance is used to quantify its deviation from the feature distribution:

$$d_{f_i}^{ma} = \sqrt{(f_i(s, z) - \mu_i)(\sigma_i + \epsilon \mathbb{I})^{-1}(f_i(s, z) - \mu_i)} \quad (11)$$

We add a constant ϵ to avoid zero eigenvalues in σ . The reward is then:

$$r^{feats} = - \sum_{f_i} w_{f_i} d_{f_i}^{ma} \mathbb{1}(d_{f_i}^{ma} \geq \text{thres}_{f_i}) \quad (12)$$

where w_{f_i} and thres_{f_i} represent the weight for feature f_i and the corresponding threshold, which prevents excessive reduction in diversity. Notably, the feature extractor takes both motions and latent variable z as input. For a given motion, the discriminator (which only takes the state as input) evaluates whether it matches the dataset, while the critic also assesses whether the motion is aligned with the initial space manifold. Since we can pre-compute the inverse of the covariance matrix, this avoids the need to repeatedly compute the inverse at each step, significantly saving computation time. In our approach, w_{f_i} is adjustable during training. Our best-performing implementation aligns the features only in the first two layers.

It is worth noting that many studies improve motion flexibility by leveraging local motion information. For instance, Jang et al.’s Motion Puzzle [4] and Lee et al.’s physics-based controllers [5] focus on enhancing motion adaptability. PMP [2], on the other hand, provides greater flexibility for non-repetitive motions. However, these methods often overlook the interdependent patterns between body parts, leading to challenges in maintaining balance and producing natural motion.

1.6. Data Generation

Our method is similar to FLEX [14]. Utilizing pre-trained hand-grasping [10] and human pose priors [13], our approach employs a gradient-based optimization process across multiple objectives to minimize losses related to hand-object interaction, balance constraints, and task alignment to synthesis a grasping pose.

Once a grasping pose is synthesized, we interpolate it into continuous motions. Specifically, denote the target pose with root translation $p_{\text{root}}^{\text{tar}} = (x_{\text{root}}, y_{\text{root}}, z_{\text{root}})$ and joint local rotations q_i^{tar} . Using spherical linear interpolation (SLERP), we generate T frames of motion starting from an initial standing position with root translation $p_{\text{root}}^{\text{init}} = (0, 0, 0)$ and joint local rotations q_i^{init} . At frame t , the root translation $p_{\text{root}}^t = (x_{\text{root}}^t, y_{\text{root}}^t, z_{\text{root}}^t)$ and joint local rotations q_i^t are computed as follows:

$$p_{\text{root}}^t = \text{slerp}(p_{\text{root}}^{\text{init}}, p_{\text{root}}^{\text{tar}}, t/T) \quad (13)$$

$$q_i^t = \text{slerp}(q_i^{\text{init}}, q_i^{\text{tar}}, t/T) \quad (14)$$

where $\text{slerp}(\cdot, \cdot, \alpha)$ denotes the interpolation and t is the current frame in the interpolation from the initial position to the target pose over T frames.

Following this, we retarget the SMPL-X parameters to our humanoid model, similar to InterScene [8], ensuring that the generated motions align with the desired body structure and task requirements. To ensure physical plausibility, we enforce constraints on the generated motion: the target pose is set to rest on the left hand, and the minimum height of both feet remains consistent with the ground throughout the motion. By adhering to these constraints, the generated motion not only respects physical limitations but also aligns closely with task-specific objectives, such as precise hand-object interactions.

2. Implementation Details

2.1. Dataset

Table 1 presents the motion capture files used in our dataset. The dataset emphasize straightforward walk motion together with various but simple turning actions, constructing the brief walk reference.

Each motion sequence lasts approximately 2-5 seconds, capturing detailed and nuanced human locomotion dynamics. The inclusion of varied turning motions alongside straight walking ensures a well-rounded dataset suitable for walking to, reaching for, grasping, turning and walking back. The weights assigned to each motion type reflect their relative importance or frequency, with larger weights assigned to simple walking sequences and smaller weights to specific turning actions. This approach provides a balanced representation, aiding in effective training and evaluation.

2.2. Network Architecture

The network architecture builds upon the design used in ASE [12], with modifications tailored to meet the requirements of our system.

The **low-level policy** is implemented as an actor network that maps a state s and latent z to a Gaussian distribution over actions. This policy is realized using a fully connected network with three hidden layers of sizes [1024, 1024, 512] (the same configuration as the encoder and high-level policy), followed by linear output units. The **critic** for the value function divides its input into five components. Each component is processed independently through a small fully connected network. The resulting outputs are concatenated and passed through a fully connected layer with a single linear output unit, providing the value.

The **encoder** $q(z|s, s')$ and **discriminator** $D(s, s')$ are jointly modeled by a shared network. Separate outputs are used to compute the encoder’s mean $\mu_q(s, s')$, normalized to $\|\mu_q(s, s')\| = 1$, and the discriminator’s sigmoid output.

File Name	Weight
ACCAD_Female1Walking_c3d_B9_-_walk_turn_left_(90)	0.01463157
ACCAD_Female1Walking_c3d_B10_-_walk_turn_left_(45)	0.01463157
ACCAD_Female1Walking_c3d_B11_-_walk_turn_left_(135)	0.01463157
ACCAD_Female1Walking_c3d_B12_-_walk_turn_right_(90)	0.01463157
ACCAD_Female1Walking_c3d_B13_-_walk_turn_right_(45)	0.01463157
ACCAD_Female1Walking_c3d_B14_-_walk_turn_right_(135)	0.01463157
ACCAD_Female1Walking_c3d_B15_-_walk_turn_around_(same_direction)_s1	0.02663157
ACCAD_Female1Walking_c3d_B15_-_walk_turn_around_(same_direction)_s2	0.02663157
ACCAD_Female1Walking_c3d_B3_-_walk1	0.05263157
ACCAD.s007_QkWalk1	0.05263157
amp_humanoid_walk	0.05263157
CMU.07.01	0.10263157
CMU.07.02	0.10263157
CMU.07.07	0.10263157

Table 1. **Walking Dataset:** Weights for different walking and turning.

The **high-level policy** uses two hidden layers of sizes [1024, 512] to generate unnormalized latents \bar{z} . These are normalized to $z = \bar{z}/\|\bar{z}\|$ before being passed to the low-level policy.

2.3. Simulation Environment

The experiments utilize Isaac Gym [7], which is a highly efficient physics simulator that operates on a GPU. The training process incorporates 4096 simultaneous environments executed on one NVIDIA V100 GPU, achieving a simulation rate of 120Hz. Every neural network is developed using PyTorch [9].

2.4. Training Details and Hyper-Parameters

2.4.1. Initial Space Training

We trained the initial space (the initial low-level policy) using a walking dataset. The training process spanned 10,000 epochs and took approximately 12 hours. Details of the hyper-parameters are provided in Table 2.

2.4.2. Tuning with feature alignment

We tuned the initial space on the augmented dataset. This training uses 3000-6000 epochs(depends on the converge rate) and lasts for about 6-10 hours. We add a different weight of the r^{feats} for different configurations. The configuration $[w_{f_0^1}, w_{f_0^2}, w_{f_0^3}, w_{f_0^4}, w_{f_0^5}, w_{f_1}, w_{f_2}, w_{f_3}]$ represents the corresponding weight. The hyper-parameters can be found in Table 3.

2.4.3. Active Strategy in Data Generation

We employ active strategy in generating data as formulated below:

$$W_j = s_0 + w_{succ} \frac{\max_i sr_i - sr_j}{\max_i sr_i - \min_i sr_i} + w_{disc} \frac{\max_i \bar{p}_i - \bar{p}_j}{\max_i \bar{p}_i - \min_i \bar{p}_i} \quad (15)$$

Hyper-parameters	Value
Learning Rate	2e-5
Episode Length	300
Action Distribution Variance	0.055
Discount γ	0.99
TD(λ)	0.95
Disc/Enc Mini-batchsize	4096
Policy Mini-batchsize	16384
Disc Grad Penalty Weight	5
Latent Dimension	64
Diversity Objective Bonus	0.01
Disc Weight Decay	0.0001
Enc Weight Decay	0.000
Disc Reward Weight	0.5
Enc Reward Weight	0.5

Table 2. Hyper-parameters for Low-level Policy Training

Feature Configurations	Reward Weight	Threshold
[0, 1, 1, 1, 1, 0, 0, 0]	0.008	1
[1, 1, 1, 1, 1, 0, 0, 0]	0.008	1
[0, 1, 1, 1, 1, 0.5, 0, 0]	0.005	1
[1, 1, 1, 1, 1, 0.5, 0, 0]	0.005	1
[1, 1, 1, 1, 1, 0.5, 0.5, 0]	0.005	1

Table 3. Hyper-parameters for Features Alignment

where W_j denotes the overall score of the j -th task. In our implementation, we set s_0, w_{succ}, w_{disc} to 0.2, 0.4, 0.4, respectively.

2.4.4. High-level Policy Training

Using a task-specific reward $r = w_G r_G + w_{p_1} r_{p_1} + w_{p_2} r_{p_2}$, we train a high-level policy to execute reaching and grasp-

ing. The training hyper-parameters are provided in Table 4. Furthermore, the parameters related to the reward design are detailed in Table 5.

Hyper-parameters	Value
Learning Rate	2e-5
Episode Length	300
Action Distribution Variance	0.1
Discount γ	0.99
TD(λ)	0.95
Disc Mini-batchsize	4096
Policy Mini-batchsize	16384
LR Schedule	constant
Disc Grad Penalty Weight	5
Disc Weight Decay	0.0001
w_G	0.4
w_{p_1}	0.2
w_{p_2}	0.4

Table 4. Hyper-parameters for High-level Policy Training

Parameter	Value
w_{pos}	0.3
w_{vel}	0.6
w_{face}	0.1
w_{grasp}	1.0
w_{height}	2.0
$w_{\text{obj-vel}}$	1.0
$h_{\text{lift_target}}$	0.2 m
$v_{\text{threshold}}$	30m/s

Table 5. Reward Function Parameters

3. Experimental Details

3.1. Details about Baselines

In the implementation of AMP, we adopted a 7:3 ratio between the task loss and discriminator loss, which produced the highest success rate during our experiments. When the ratio was set to 1:0, AMP degraded to Fullbody PPO. The same ratio was used for AMP*, with the only difference being that the motion prior in AMP* was trained on a dataset that included interpolated data, similar to Ours.

For PMP and PSE, we modified the discriminator in AMP to a part-wise design and trained it on walking data. In PMP, we adopted the same 7:3 ratio between the task loss and discriminator loss as used in AMP. To further improve the success rate, we implemented a dedicated arm module for grasping, using rollout trajectories as references. During the second and third stages of the reaching task, the right-hand joints were encouraged to move closer to the refer-

ence. Similarly, in PSE, we introduced a part-wise discriminator as well. However, in the space training stage, the task and discriminator weight ratio was set to 1:1.

3.2. Details about User Study

In this study, human preference was used to evaluate the naturalness of motions. Specifically, we recruited **100** volunteers to compare the performance of different policies through video-based assessments. The evaluation process consisted of the following steps:

- **Random Motion Generation:** For each policy, we randomly selected four sets of scene parameters and rendered motion sequences for these parameters in the Isaac simulator.
- **Side-by-Side Video Comparison:** For each pair of policies to be compared, the motion sequence videos were presented side by side to the volunteers. Each video was recorded from three different viewpoints, allowing the volunteers to comprehensively observe the motion performance.
- **Volunteer Judgments:** After viewing the videos, the volunteers selected the policy that they perceived to exhibit more natural motion.

The users will rate the motions from N different aspects, selecting the best option for each aspect. The table in the main text shows the results of these comparisons. The baseline values represent the weighted sum of the proportion of volunteers who selected the baseline policy as the better option (in our case, the weights are all $\frac{1}{N}$).

3.2.1. Users(Q): Quality Evaluation

This evaluation focuses on the overall quality of the motions. For the grasping process, we further divided the assessment into four aspects: approaching the table, the grasping process, moving towards the target, and overall coherence.

- **Naturalness of Approaching the Table:** This evaluates how the integration of additional data influences the naturalness of the original walking motion.
- **Grasping Process:** As a crucial part of the entire sequence, this examines how naturally the agent extends its hand and successfully grasps the object.
- **Moving Towards the Target:** This assesses the agent’s ability to recover and move towards the target smoothly after grasping.
- **Overall Coherence:** This evaluates the naturalness of transitions between different stages of the motion and the overall intentionality of the entire sequence.

3.2.2. Users(I): Issues Judgment

In this evaluation criterion, we focus on the evaluation of detailed issues. Users are instructed to pay attention to specific details we identified (commonly observed unnatural patterns) and select the policy with fewer issues.

When comparing against the baseline, we highlighted four representative and classic issues: shuffling, sliding, near-loss of balance, and overly exaggerated or unnecessary movements.

For the ablation study, we first asked a subset of users to watch CIRCLE [1] interpolated data and real MoCap data. From their feedback, we identified the most frequently mentioned issue keywords, which were then used as evaluation criteria. We provided users with six motions corresponding to a specific ratio and asked them to rank these motions from worst to best based on the criteria mentioned above. A motion ranked in the i -th position received a score of $i - 1$. Finally, the overall scores are presented in the table.

3.3. Details about Pred Score

Considering that the visual differences introduced by the added features are relatively subtle, we adopted a more fine-grained evaluation approach for this section. Specifically, we used interpolated data generated by the CIRCLE strategy as negative examples and real motion capture data as positive examples to train a discriminator. This discriminator was then used to evaluate our generated motions. The scores presented in the table represent the average scores of 1,000 randomly sampled motions.

Specifically, we utilized the pre-trained feature extractor from MotionGPT to extract 512-dimensional features. Then these features were passed through three fully connected layers with a structure [512, 128, 128, 1] per unit. After activation, the final output was the score. We retained the checkpoint with the best validation accuracy, achieving a discrimination accuracy of 94.2%.

4. Additional Experiments and Visualization

4.1. Walking Phase Validation

Our method generates natural, high-quality walking motions, especially when far from the table (full video will be provided instead of clipping around grasping). We evaluated the 1st-phase motion using a discriminator trained on walking MoCap, achieving disc rewards of 0.487 which is close to those of directly reproducing motions using ASE/AMP (0.503/0.516) and the oracle (0.492), compared to the real data rewards of 0.892. During the pre-grasping(2nd-phase), while our method demonstrates significant improvements over existing approaches like Omni-grasp and Braun’s, some motions exhibit artifacts like sliding adjustments, primarily due to the need to generalize to **varying table widths** and the inherent complexity of interaction and **collision avoidance**. Even the oracle policy in mid-height shows noticeable adjustments in generalized scenarios. Noticing that MoCap data with high precision does not fully address these challenges and MoCap has its own limitations and biases, we turn to **explore more flexi-**



Figure 1. Diversity visualization

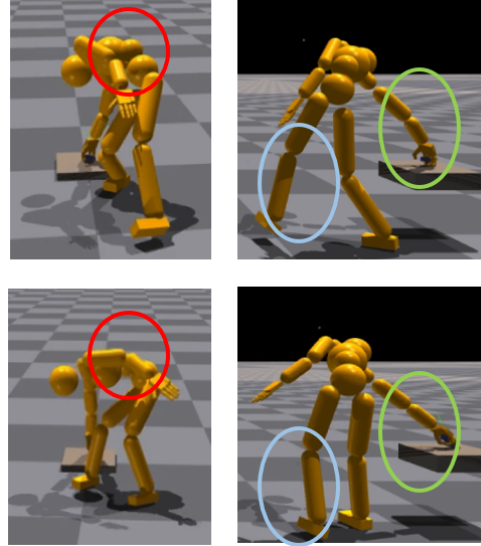


Figure 2. **Comparison of f_0 alignment:** The top image shows no alignment, while the bottom with f_0 alignment reveals key changes: the torso bends at the lower joint (red circle), a passive reach becomes a coordinated motion driven by the upper arm (green circle), and a suspended leg transitions to a standing leg (blue circle).

ble synthetic data.

4.2. Diversity of Motions

Diversity was not a primary focus in our approach, as we prioritized maximizing SR during RL exploration and pose generation. This led to optimized poses for different table heights converging. Enhancing pose diversity and increasing variance during exploration could yield different grasping patterns (shown in Figure 1).

4.3. More about Feature Alignment

In this section, we want to further analyze the effects of our feature alignment mechanism.

First, the effect of adding a shallow layer is particularly evident in the *partwise* patterns. For example, as shown in the red circles in Figure 2, when we add the torso feature f_0^1 , the bending pattern of the torso changes. Ini-

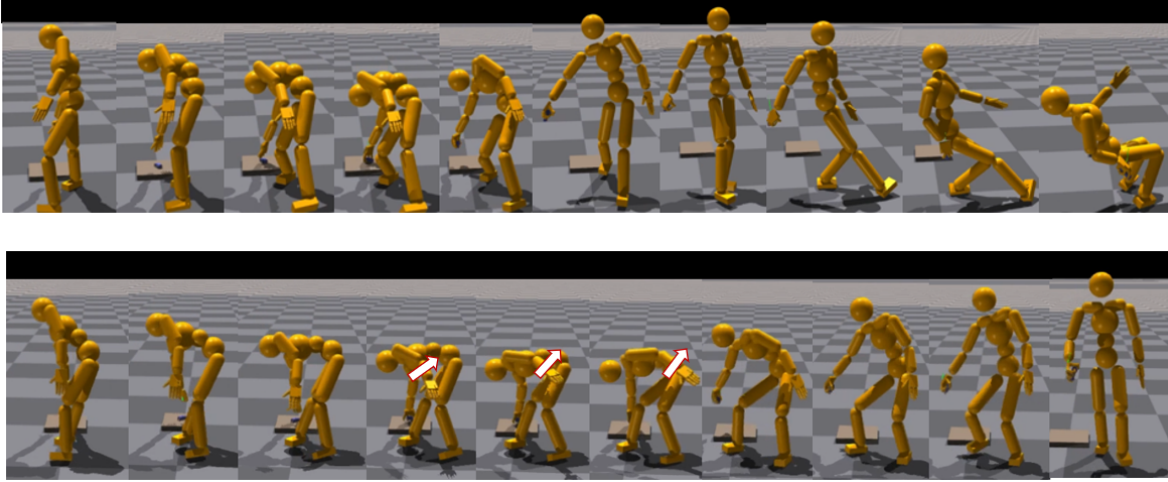


Figure 3. **Visualization of Stability Enhancement through Feature Alignment:** The lower figure, with f_0 and f_1 alignment, significantly improves stability during both the *grasp* phase and the *recover to walk* stage. The left hand raises swiftly (indicated by the arrow), and the left foot steps back quickly to maintain balance when grasping low objects.

tially, the bending originates from the closest joint to the head (torso1, corresponding to the chest and neck). After the addition, the upper body becomes mostly upright (borrowed from the *walk* pattern), with the bending localized to torso2 (similar to the waist). On the other hand, as indicated by the green circles, the hand posture without alignment appears relatively relaxed. This frequently results in body tilt dominating the motion, where the upper arm remains stiff while the forearm dangles loosely in a *reach* pattern. However, after alignment, a more natural pattern emerges: the upper arm drives the extension of the forearm, resulting in a more straight and coordinated reaching motion.

Beyond the visual differences, it is even more critical to emphasize the contribution of this feature to overall stability. Specifically, we observe that this feature significantly aids both the final stage of *grasp* and the *recover to walk* phase. For example(as shown in Figure 3), with the addition of features(especially f_1), the motion exhibits a clear pattern where the left hand is swiftly raised and the left foot quickly steps back to maintain balance when grasping very low objects. This coordinated movement of the limb is particularly beneficial for dynamic balance during the recovery phase in *walk*.

We observe that without the alignment of the features, the agent adopts a more 'aggressive' posture characterized by a bent right leg and a lifted left foot. While this posture is inspired by real grasping motions (e.g. actions similar to picking up a small ball in the ground), these movements rely on fine muscle coordination and balance that are challenging even for humans. Such extreme low-grasp postures make it difficult to recover quickly, let alone maintain balance for the agent.

To achieve the same task, a more robust strategy can be adopted to enhance the success rates. For example, keep both feet grounded and squat sideways instead of tilting. This movement resembles natural upright postures and walking, which inherently favor balanced grasping. **This strategy of "learning more stable task-completion motions" from *walk* can be further incorporated into humanoid robotics to help execute tasks.**

We also observed that **incorporating deeper features**, such as f_2 and f_3 , during feature alignment can **negatively impact the overall learning of reaching skills**. Specifically, when f_2 is included, the success rate decreases, even falling below the performance achieved without additional data. Furthermore, when f_3 is added, the policy fails to successfully walk to the table. This decline in performance arises because deeper features capture more information about global movement(as shown in the pilot study below). For new datasets, the objective of reaching the target and the goal of maintaining local features close to those from the walking diverge significantly. This divergence introduces inconsistency into the training process, leading to a disorganized feature space that hinders effective learning of the task.

4.4. Ablation Study of Data Ratio

In our active strategy, we emphasize **maximizing the utility of generated data to achieve high success rates with a minimal data ratio**. In this section, we provide a detailed analysis of the impact of the data ratio on policy performance and the role of "maximizing the utility of generated data."

As previously defined, the data ratio refers to the percent-

age of generated data relative to the original data used during sampling. As shown in Figure 5, we observe that when the data ratio is low, the success rate increases rapidly as the ratio grows, peaking at around 20-30%. In this phase, the active strategy plays a crucial role by specifically addressing tasks that walk data cannot handle, effectively avoiding the addition of irrelevant data.

However, adding more data does not necessarily compensate for the shortcomings of the random strategy. As we can observe, the success rate drops sharply when the ratio continuously increase, reaching nearly zero when the ratio exceeds 1:1. This is because the excessive addition of data makes learning the skill space more challenging. As new data increases, natural walking and turning skills are forgotten. As shown in the Figure 4, when the ratio exceeds 1:1, the character struggles to perform natural turns. When the ratio exceeds 1:2, the character focuses on switching between various skills to maintain balance, making it difficult to walk effectively.

Even with a ratio below 1:1 (data ratio ≤ 100), we observe that larger ratios result in significantly longer training times. As seen in ASE [12], even well-balanced weight designs require several days of training for relatively small differences, such as a 30-minute skill. This further highlights the value of first training a fundamental walk space then selectively expanding it. By doing so, we efficiently learn the truly reusable "walk" skill and focus on the task-relevant "reach" skill at minimal cost.

5. More about Pilot Study

5.1. Pilot study with larger dataset

In our pilot study, features from shallow layers show distinct clustering in real data and lower FID values. In contrast, deeper layers prioritize semantic information, reducing this clustering. However, even in deep layers, the FID values between MoCap Reach and MoCap Walk are lower than those between MoCap Reach and Generated Reach.

We hypothesize that this discrepancy arises due to the limited training data: although MoCap Reach and Generated Reach share similar global movements (e.g. reaching), the critic primarily identifies that they are distinct from "walk-like" motions but lacks a comprehensive understanding of the specific characteristics of reaching motions.

We believe that with a critic trained on a larger and more diverse dataset, MoCap Reach and Generated Reach—both representing similar semantic motions—would cluster more closely in the deep layers. This would further support the notion of a universal phenomenon: transferable local patterns are captured in shallow layers, whereas deeper layers reflect global, task-specific movements.

5.1.1. Dataset

To enhance the critic's ability to understand motions, we use more data to train the critic. The added data can be found below in Table 6:

File Name	Weight
RL_Avatar_Atk_Spin_Motion.npy	0.00724638
RL_Avatar_Standoff_Feint_Motion.npy	0.03105590
RL_Avatar_Dodge_Backward_Motion.npy	0.01552795
RL_Avatar_RunBackward_Motion.npy	0.01552795
RL_Avatar_WalkBackward01_Motion.npy	0.01552795
RL_Avatar_WalkBackward02_Motion.npy	0.01552795
RL_Avatar_Dodge_Left_Motion.npy	0.01552795
RL_Avatar_RunLeft_Motion.npy	0.01552795
RL_Avatar_WalkLeft01_Motion.npy	0.01552795
RL_Avatar_WalkLeft02_Motion.npy	0.01552795
RL_Avatar_Dodge_Right_Motion.npy	0.01552795
RL_Avatar_RunRight_Motion.npy	0.01552795
RL_Avatar_WalkRight01_Motion.npy	0.01552795
RL_Avatar_WalkRight02_Motion.npy	0.01552795
RL_Avatar_RunForward_Motion.npy	0.02070393
RL_Avatar_WalkForward01_Motion.npy	0.02070393
RL_Avatar_WalkForward02_Motion.npy	0.02070393
RL_Avatar_Standoff_Circle_Motion.npy	0.06211180
RL_Avatar_TurnLeft90_Motion.npy	0.03105590
RL_Avatar_TurnLeft180_Motion.npy	0.03105590
RL_Avatar_TurnRight90_Motion.npy	0.03105590
RL_Avatar_TurnRight180_Motion.npy	0.03105590
RL_Avatar_Fall_Backward_Motion.npy	0.00869565
RL_Avatar_Fall_Left_Motion.npy	0.00869565
RL_Avatar_Fall_Right_Motion.npy	0.00869565
RL_Avatar_Fall_SpinLeft_Motion.npy	0.00869565
RL_Avatar_Fall_SpinRight_Motion.npy	0.00869565
RL_Avatar_Idle_Alert(0)_Motion.npy	0.00434783
RL_Avatar_Idle_Alert_Motion.npy	0.00434783
RL_Avatar_Idle_Battle(0)_Motion.npy	0.00434783
RL_Avatar_Idle_Battle_Motion.npy	0.00434783
RL_Avatar_Idle_Ready(0)_Motion.npy	0.00434783
RL_Avatar_Idle_Ready_Motion.npy	0.00434783
CMU_07_02.npy	0.04070393
CMU_07_01.npy	0.04070393
CMU_07_07.npy	0.04070393
amp_humanoid_jog.npy	0.08316768
amp_humanoid_walk.npy	0.09316768
amp_humanoid_run.npy	0.08316768

Table 6. **Critic Dataset:** Weights for different motions

5.1.2. Result

When using a critic trained with a larger dataset, we observed deeper-level clustering of two macroscopic motion types, particularly in the final two layers of the network. At the same time, shallow-level clustering remained similar to that observed with our previous critic (shown in Figure 7). Although our dataset does not contain specific reaching mo-

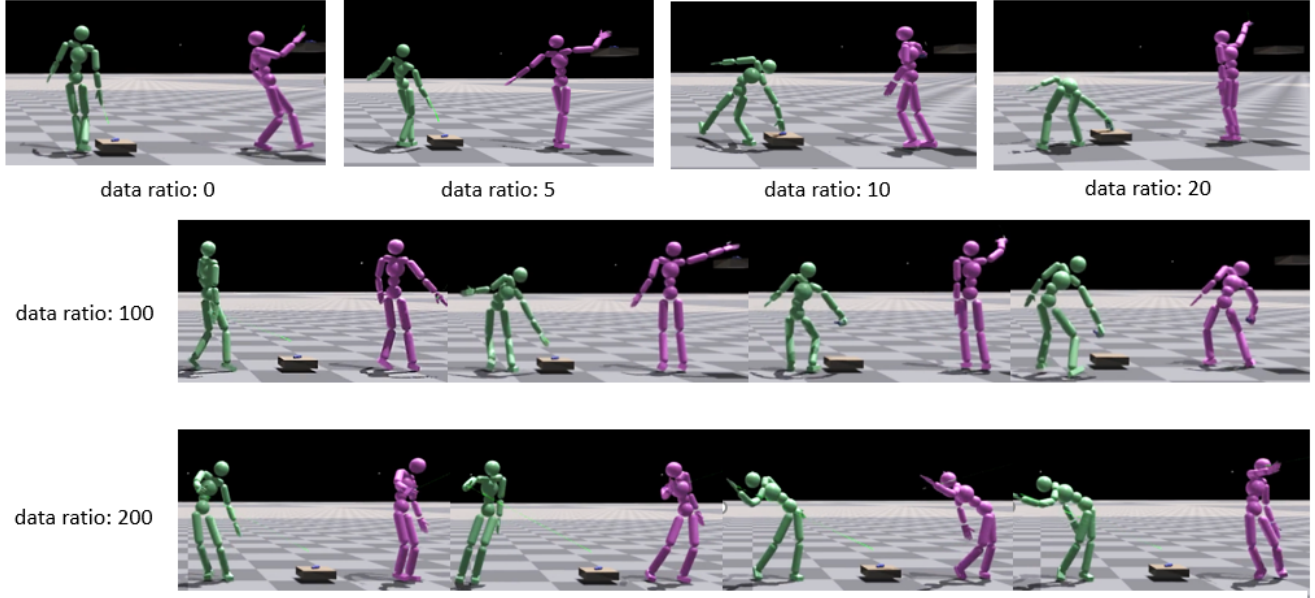


Figure 4. **Visualization of Reaching and Grasping with Varying Data Ratios(%)**: At low data ratios, task completion improves rapidly as the ratio increases. However, when the ratio exceeds 100%, the character struggles with natural turning, and beyond 200%, the character shifts focus to balancing between skills, hindering effective walking.

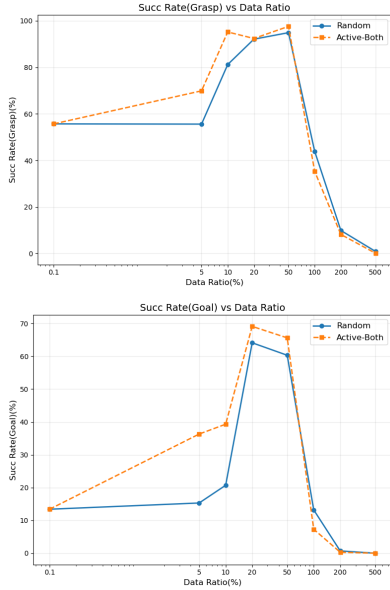


Figure 5. **Success Rate Curve with Varying Data Ratio(%)**: Using a small data ratio allows for an active strategy that specifically targets challenging tasks, resulting in better success rate. However, an excessively large data ratio can lead to a rapid decline in the success rate.

tions, the increased data volume enhanced the critic’s understanding. As a result, the value predictions were no longer solely based on “walk-like” patterns but also incorporated

the classification of macroscopic motion types.

Further pilot studies revealed that, under our critic network architecture, shallow-level features effectively capture the patterns of real motion and can transfer across different motion types and tasks. In contrast, deeper-level features encode more information about macroscopic motion forms and semantic details.

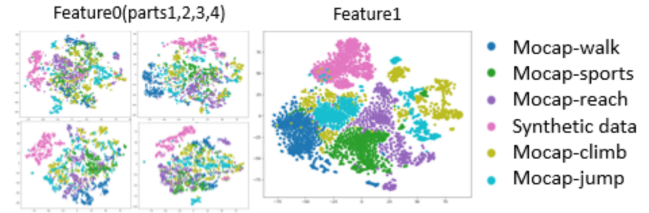


Figure 6. pilot study for various task

5.2. Pilot Study with various motions

Reaching and grasping are fundamental yet highly challenging tasks that require balancing high DOF while enabling precise manipulation, making them a rigorous testbed for motion generation. We also conducted pilot studies across various tasks, obtaining consistent results. All these motions are from AMASS [6]. For tasks beyond reaching, like shown in the Figure 6, MoCap data cluster together in shallow layers while synthetic data remain separated. These suggest our mechanism can capture transfer-

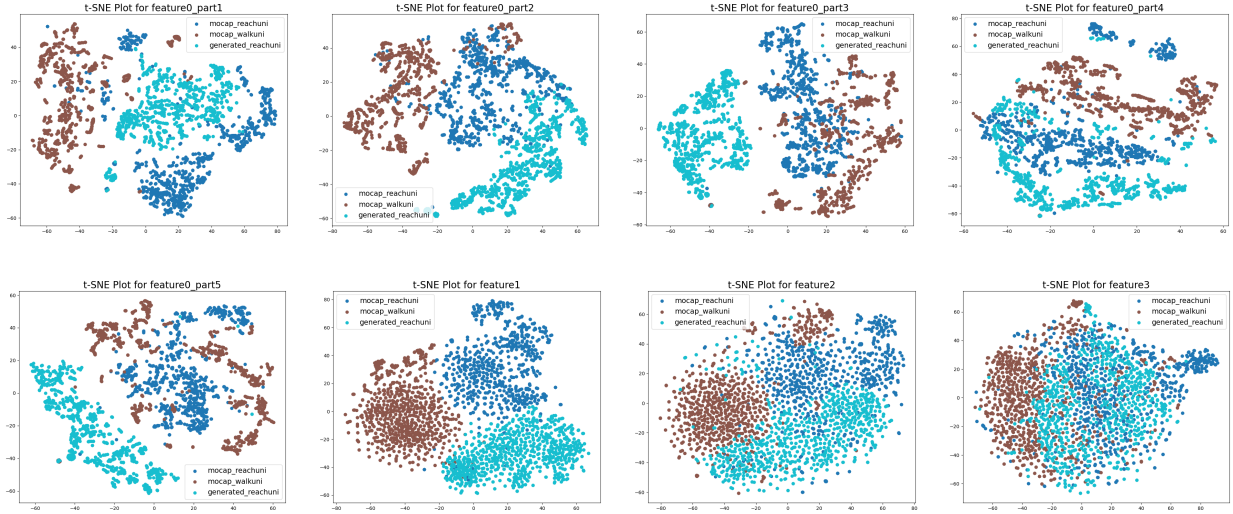


Figure 7. **t-SNE plots of features extracted at different levels of the more comprehensive critic network:** There is clear clustering within the MoCap data in shallow layers and a clear clustering within Reach data in deeper layers.

able patterns from walking beyond specific tasks.

References

- [1] Joao Pedro Araujo, Jiaman Li, Karthik Vetrivel, Rishi Agarwal, Deepak Gopinath, Jiajun Wu, Alexander Clegg, and C. Karen Liu. Circle: Capture in rich contextual environments, 2023. 6
- [2] Jinseok Bae, Jungdam Won, Donggeun Lim, Cheol-Hui Min, and Young Min Kim. Pmp: Learning to physically interact with environments using part-wise motion priors. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–10, 2023. 3
- [3] Shadow Robot Company. Dexterous hand series, 2005. Accessed: 2024-11-15. 1
- [4] Deok-Kyeong Jang, Soomin Park, and Sung-Hee Lee. Motion puzzle: Arbitrary motion style transfer by body part. *ACM Transactions on Graphics (TOG)*, 41(3):1–16, 2022. 3
- [5] Seyoung Lee, Jiye Lee, and Jehee Lee. Learning virtual chimeras by dynamic motion reassembly. *ACM Transactions on Graphics (TOG)*, 41(6):1–13, 2022. 3
- [6] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019. 9
- [7] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021. 4
- [8] Liang Pan, Jingbo Wang, Buzhen Huang, Junyu Zhang, Hao-fan Wang, Xu Tang, and Yangang Wang. Synthesizing physically plausible human motions in 3d scenes. In *International Conference on 3D Vision (3DV)*, 2024. 3
- [9] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 4
- [10] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image, 2019. 3
- [11] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (ToG)*, 40(4):1–20, 2021. 1
- [12] Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Transactions On Graphics (TOG)*, 41(4):1–17, 2022. 3, 8
- [13] Omid Taheri, Nima Ghorbani, Michael J Black, and Dimitrios Tzionas. Grab: A dataset of whole-body human grasping of objects. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 581–600. Springer, 2020. 3
- [14] Purva Tendulkar, Dídac Surís, and Carl Vondrick. Flex: Full-body grasping without full-body grasps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21179–21189, 2023. 3
- [15] Yinzheng Xu, Weikang Wan, Jialiang Zhang, Haoran Liu, Zikang Shan, Hao Shen, Ruicheng Wang, Haoran Geng, Yijia Weng, Jiayi Chen, et al. Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4737–4746, 2023. 1