

# EMP: Executable Motion Prior for Humanoid Robot Standing Upper-body Motion Imitation

**Abstract**—To support humanoid robots in performing manipulation tasks, it is essential to study stable standing while accommodating upper-body motions. However, the limited controllable range of humanoid robots in a standing position affects the stability of the entire body. Thus we introduce a reinforcement learning based framework for humanoid robots to imitate human upper-body motions while maintaining overall stability. Our approach begins with designing a retargeting network that generates a large-scale upper-body motion dataset for training the reinforcement learning (RL) policy, which enables the humanoid robot to track upper-body motion targets, employing domain randomization for enhanced robustness. To avoid exceeding the robot’s execution capability and ensure safety and stability, we propose an Executable Motion Prior (EMP) module, which adjusts the input target movements based on the robot’s current state. This adjustment improves standing stability while minimizing changes to motion amplitude. We evaluate our framework through simulation and real-world tests, demonstrating its practical applicability. [Project page](#).

## I. INTRODUCTION

The humanoid form allows humanoid robots to better adapt to human environments, tools, and human-machine interactions. We aim to enable humanoid robots to perform human-like movements, allowing for better mapping of human motions onto the robots. This enables them to quickly learn human motion skills, which lays the foundation for executing subsequent task operations.

However, many challenges remain in the practical implementation of humanoid robots mimicking human motions. The complex dynamic characteristics of humanoid robots, along with their high-dimensional state and action spaces, complicate motion control. While model-based controllers have shown remarkable results in whole-body motion imitation [1]–[3], the computational burden of complex dynamics models restricts these methods to simplified models, limiting their scalability for dynamic motions.

Recently, reinforcement learning methods have gained popularity in the field of humanoid robotics. Initially, RL was employed in the graphics community to generate humanoid motions from human motion data for animated characters [4], [5]. Additionally, RL controllers have been developed for bipedal robot walking [6], [7], whole-body control [8], and humanoid teleoperation [9].

Our work focuses on humanoid robots imitating human upper-body movements because humanoid robots perform tasks with their upper bodies while standing most of time. The mainstream frameworks are shown in Figure 1. In humanoid reinforcement learning for motion imitation, we observe a conflict between stability and similarity rewards. When joints are entirely controlled by the RL policy for whole-body control

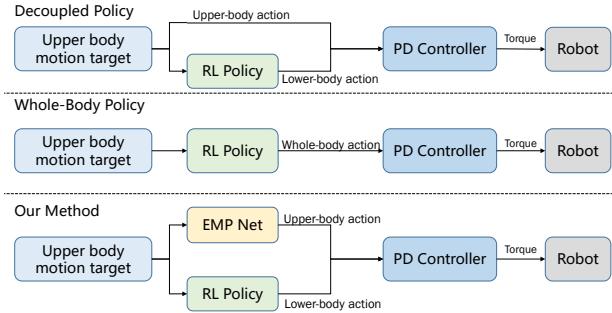


Fig. 1: Different Motion Imitation Framework. (a) Decoupled Policy, such as PMP [10], only generates lower-body actions and execute upper-body target directly. (b) Whole-Body Policy, like HumanPlus [11] and Exbody [8], controls whole body joint to imitate whole body motion target. (c) Our method introduce a executable motion prior to optimize upper-body motion target while RL policy provides lower-body actions.

like [9], [11], vibration and deviation on base and upper-body actions can occur. Conversely, directly executing upper-body actions may lead to the robot’s limited control capacity exceeding the RL policy’s capabilities, resulting in a loss of balance.

In this paper, we present a system for humanoid robots to imitate human upper-body motions while maintaining whole-body stability. Combined with imitation learning and reinforcement learning, Figure 2 shows our framework. First, we design a graph convolutional network to retarget human motions to humanoid movements, creating a motion dataset for training a robust RL imitation policy. Next, we train a RL policy for upper-body motion imitation using retargeted motions. This policy manages the lower-body joints to maintain balance, while upper-body targets are directly sent to robot to ensure alignment with targets.

When humans are performing upper-body actions, they can recognize potential dangers and make motion adjustments in a timely manner. Inspired by this, we propose an **Executable Motion Prior (EMP)** that modifies the input target upper-body motions based on the robot’s current state. This approach enhances standing stability while minimizing alterations to the motion amplitude. Utilizing the dataset obtained from motion retargeting and the trained RL controller, we train an EMP network. This network transforms unstable actions into stable ones by simultaneously encoding the robot’s current state and action objectives into a latent space and decoding them into new, more reasonable action objectives, functioning as an action optimization module before RL controller. Finally we deploy this framework in real-world humanoid robots.

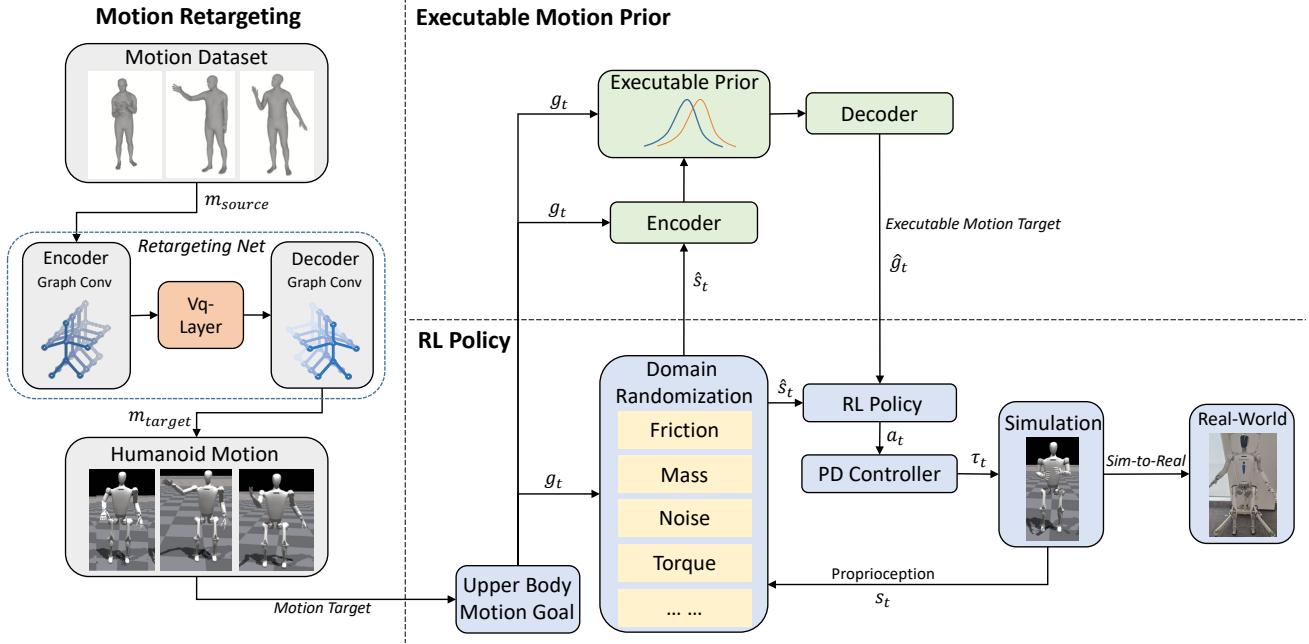


Fig. 2: Overview of our framework. **Motion Retargeting** (section III): We train a graph convolution retargeting network to convert human motions  $m_{source}$  to humanoid joint actions  $m_{target}$  as motion goal for imitation. **RL Policy** (section IV): We train an upper-body imitation policy for the humanoid to track the upper-body motion goal  $g_t$  while keeping balance. **Executable Motion Prior** (section V): We use a VAE-based network to adjust the goal motion based on the current state  $\hat{s}_t$ , improving stability.

Our contributions are as follows:

- 1) A RL based framework for humanoid imitating upper-body motion, which includes a motion retargeting network to transfer human motions to humanoid motions and a RL policy to control the robot while tracking upper-body motions;
- 2) An executable motion prior for the RL imitation policy system that adjusts target motions based on the humanoid's current state, enhancing stability while minimizing changes in motion amplitude;
- 3) A world model to simulate the state transition process of the environment for gradient backpropagation.
- 4) Sim-to-real transfer of our system that demonstrates its effectiveness in two humanoid robots.

## II. RELATED WORKS

### A. Motion Retargeting

Motion retargeting facilitates the transfer of motion data from a source character to a target character. In the context of animation, both optimization-based methods [12], [13] and learning-based methods [14], [15] are employed for motion transfer between animated characters. Learning-based methods tend to yield more efficient results and can facilitate motion transfer across different skeletal structures [16], [17].

In humanoid motion retargeting, Delhaisse et al. [18] use shared latent variable models to retarget motions between different humanoids. Ayusawa et al. [19] reconstruct human motion within the physical constraints imposed by humanoid dynamics and offer precise morphing function for different

human body dimensions. In our work we design a network to generate humanoid motions from human motions.

### B. Reinforcement Learning for Humanoid

Before RL-based controllers being used into real-world humanoids, they are often used in physics-based animation control [4], [5], [20]. Nevertheless, the humanoid avatars usually have a high degree of freedom, with minimal restrictions on joint positions, torques, and sometimes with additional auxiliary force [21].

On the other hand, realistic humanoids have complex dynamic models, and it is difficult to obtain privileged states, such as base velocity and height, from build-in sensors [8]. This makes it impossible to directly transfer RL models used for animated characters to physical humanoids. Li et al. [7] proposed an end-to-end RL approach and use task randomization to build a robust dynamic locomotion controller for bipedal robots. Radosavovic et al. [22] designed causal Transformer trained by autoregressive prediction of future actions from the history of observations and actions for real-world full-sized humanoid locomotion. Siekmann et al. [6] use stair-like terrain randomization to build a RL controller for humanoid traversing stair-like terrain. In this work we build a sim-to-real training process with domain randomization and deploy our policy to realistic humanoid.

### C. Humanoids Imitation from Human Motion

Traditional methods, such as model predictive control (MPC), use model-based optimization methods to minimize tracking errors under stability and contact constraints [23].

However, due to the high computational burden, the humanoid model is usually simplified [24], [25], which limits the accuracy of imitation.

RL controllers provides an alternative solution. Cheng et al. [8] train a whole-body humanoid controller with a large-scale motion dataset. He et al. [9] use a privileged policy to select a executable motion dataset, which helps training a robust RL policy for sim-to-real deployment. Fu et al. [11] train a task-agnostic low-level policy to track retargeted humanoid poses. Lu et al. [10] proposed a CVAE-based motion prior to enhance the robustness of controller. In this work we focus on tracking the upper-body motion targets and build a executable motion prior network to filter motions and a upper-body motion imitation RL policy to keep whole-body balance while tracking upper-body targets.

### III. RETARGETING HUMAN MOTION TO HUMANOIDS

#### A. Retargeting Network Architecture

Prior work [17] has achieved cross-skeleton motion retargeting between animated characters with graph network. We develop a network for motion retargeting from human to humanoid. Using networks for motion retargeting offers good real-time performance and generalization capabilities.

Figure 2 illustrates the structure of our retargeting net. We regard the upper-body skeleton of the humanoid and the human as a graph. Referring to the framework of VQ-VAE [26], our network consists of a motion encoder, a vq-codebook layer and a motion decoder.

The motion encoder  $f_e$  embeds the source motion from human. The source motion is represented as the positions of key nodes  $\mathbf{Q}_A \in \mathbb{R}^{N_A \times 3}$  and the features of edges  $\mathbf{E}_A$ . After passing through the graph convolutional layers, the source motion features are encoded into the latent space features :  $\mathbf{z}_A = f_e(\mathbf{Q}_A, \mathbf{E}_A)$ . A transformation net  $f_{tf}$  converts the latent features of input skeleton A into the latent features of output skeleton B:  $\mathbf{z}_B = f_{tf}(\mathbf{z}_A)$ . Then the codebook layer choose the nearest element of latent embedding vectors:

$$\mathbf{z}_e = \mathbf{e}_k \text{ where } k = \arg \min_j \|\mathbf{z}_B - \mathbf{e}_j\|_2 \quad (1)$$

The motion decoder  $f_d$  generates the target motion  $\mathbf{Q}_B \in \mathbb{R}^{N_B}$  (represented by joint angles) with latent embedding vector  $\mathbf{z}_e$  and edge features  $\mathbf{E}_B$ :  $\mathbf{Q}_B = f_d(\mathbf{z}_e, \mathbf{E}_B)$ .

The key nodes are waist, torso, shoulder, elbow and wrist.

#### B. Training Loss

Combined with the method in [27], the training loss of our retargeting network is composed of five terms: end effector loss  $L_{ee}$ , orientation loss  $L_{ori}$ , elbow loss  $L_{elb}$ , embedding loss  $L_{emb}$  and commitment loss  $L_{com}$ . We list the losses in Tab I, where  $\mathbf{p}$  and  $\hat{\mathbf{p}}$  mean the node position of human and humanoid respectively,  $\mathbf{R}$  and  $\hat{\mathbf{R}}$  mean the end effector (namely wrist) rotation matrix,  $\text{sg}()$  means stop gradient.

### IV. RL CONTROL POLICY TRAINING FOR HUMANOID UPPER-BODY IMITATION

#### A. Overview

We decouple the whole-body control policy into  $\pi_{lower}$  and  $\pi_{upper}$ .  $\pi_{lower}$  is an RL-based policy which generates lower-

TABLE I  
TRAINING LOSS FOR RETARGETING NETWORK

Term	Expression	Weight
$L_{ee}$	$\left\  \frac{\mathbf{p}^{ee} - \mathbf{p}^{elb}}{\ \mathbf{p}^{ee} - \mathbf{p}^{elb}\ _2} - \frac{\hat{\mathbf{p}}^{ee} - \hat{\mathbf{p}}^{elb}}{\ \hat{\mathbf{p}}^{ee} - \hat{\mathbf{p}}^{elb}\ _2} \right\ _2^2$	100
$L_{ori}$	$\ \mathbf{R} - \hat{\mathbf{R}}\ _2^2$	100
$L_{elb}$	$\left\  \frac{\mathbf{p}^{elb} - \mathbf{p}^{sho}}{\ \mathbf{p}^{elb} - \mathbf{p}^{sho}\ _2} - \frac{\hat{\mathbf{p}}^{elb} - \hat{\mathbf{p}}^{sho}}{\ \hat{\mathbf{p}}^{elb} - \hat{\mathbf{p}}^{sho}\ _2} \right\ _2^2$	100
$L_{emb}$	$\ \text{sg}(\mathbf{z}_e) - \mathbf{e}\ _2^2$	10000
$L_{com}$	$0.25\ \mathbf{z}_e - \text{sg}(\mathbf{e})\ _2^2$	10000

TABLE II  
REWARDS EXPRESSIONS AND WEIGHTS

Term	Expression	Weight
Regularization		
Base orientation	$\exp(-10\ \mathbf{rpy}_t^{xy}\ _1)$	3.0
Projected gravity	$\exp(-20\ \mathbf{pg}_t^{xy}\ _2)$	3.0
Base height	$\exp(-100 h_t - h^{\text{ref}} )$	0.2
Base linear velocity	$\exp(-10\ \mathbf{v}_t\ _2)$	0.75
Base angular velocity	$\exp(-20\ \omega_t\ _2)$	0.75
Base acceleration	$\exp(-3\ \mathbf{v}_t - \mathbf{v}_{t-1}\ _2)$	0.2
Leg DoF position	$\exp(-100\ \mathbf{q}_t^{\text{leg}} - \mathbf{q}^{\text{leg,ref}}\ _2)$	1.0
Feet contact	$\mathbf{1}(F_{\text{feet}}^z \geq 5)$	0.5
Feet slip	$\mathbf{1}(F_{\text{feet}}^z \geq 5) \times \sqrt{\ \mathbf{v}_t^{\text{feet}}\ _2}$	0.2
Energy		
Action range	$\ \mathbf{a}_t\ _1$	-0.075
Action rate	$\ \mathbf{a}_t - \mathbf{a}_{t-1}\ _2^2$	-1.5
Action acceleration	$\ \mathbf{a}_t + \mathbf{a}_{t-2} - 2\mathbf{a}_{t-1}\ _2^2$	-1.5
Torques	$\ \tau_t\ _2^2$	-1e-5
Dof velocity	$\ \dot{\mathbf{q}}_t\ _2^2$	-1e-4
Dof acceleration	$\ \ddot{\mathbf{q}}_t\ _2^2$	-1e-7

body actions from proprioception state to keep the humanoid robot standing in balance while tracking upper-body motions. The upper-body policy  $\pi_{upper}$  is a open loop controller, namely our executable motion prior (EMP) network, which is detailed in Section V.

#### B. State Space

We consider our RL control policy as a goal-conditioned policy  $\pi : \mathbf{G} \times \mathbf{S} \rightarrow \mathbf{A}$ , where  $\mathbf{G}$  is goal space that indicates the upper-body motion target ,  $\mathbf{S}$  is the observation space and  $\mathbf{A}$  is the action space for lower-body joints.

We define goal state as  $\mathbf{g}_t \triangleq \mathbf{q}_{\text{target}} \in \mathbb{R}^{15}$ , where  $\mathbf{q}_{\text{target}}$  represents the target joint position of upper-body joints, including two 7-dof arms and one 1-dof waist. The action is denoted as  $\mathbf{a}_t \in \mathbb{R}^{12}$ . We define our observation state as  $\mathbf{s}_t \triangleq [\mathbf{q}_t, \mathbf{a}_{t-1}, \mathbf{rpy}_t, \mathbf{g}_t]$ , where  $\mathbf{q}_t \in \mathbb{R}^{27}$  indicates the joint position and  $\mathbf{rpy}_t \in \mathbb{R}^3$  is the euler angle of robot base. We combine states of last  $T$  frames together as  $\mathbf{S}_t = \{\mathbf{s}_{t-T:t}\} \in \mathbb{R}^{T \times 65}$  to utilize history message. We set  $T = 15$  in experiments. The action space consists of 12-dim joint

position targets (two 6-dof legs). The joint actions will be converted to joint torque by a PD controller.

### C. Reward Design

The rewards are detailed in Tab II, where  $h^{\text{ref}}$  is reference height of base,  $q^{\text{leg,ref}}$  is reference joint positions of legs.

Our policy focuses on upper-body motion imitation while standing, so we just set  $v_t^{\text{ref}} = 0$  and  $\omega_t^{\text{ref}} = 0$ .

TABLE III  
DOMAIN RANDOMIZATION

Term	Value
Friction	$\mathcal{U}(0.1, 2.0)$
Base Mass	$\mathcal{U}(-5.0, 5.0) + \text{default kg}$
Hand Mass	$\mathcal{U}(0, 2.5) + \text{default kg}$
Base Com	$\mathcal{U}(-0.05, 0.05) \text{ m}$
Link Inertia	$\mathcal{U}(0.8, 1.2) \times \text{default kg} \cdot \text{m}^2$
Link Mass	$\mathcal{U}(0.8, 1.2) \times \text{default kg}$
P Gain	$\mathcal{U}(0.8, 1.2) \times \text{default}$
D Gain	$\mathcal{U}(0.8, 1.2) \times \text{default}$
Motor Torque	$\mathcal{U}(0.8, 1.2) \times \text{default N} \cdot \text{m}$
Motor Damping	$\mathcal{U}(0.3, 4.0) \text{ N} \cdot \text{s}$
Motor Delay	$\mathcal{U}(0, 10) \text{ ms}$
Push Robots	interval = 5s, $v_{xy} = 0.5\text{m/s}$ , $\omega = 0.4\text{rad/s}$
Hang Robots	height = 0.1m, ratio = 20%
Init Joint Position	$\mathcal{U}(-0.1, 0.1) + \text{default rad}$
Action	$\mathcal{U}(0.98, 1.02) \times \text{default}$

### D. Domain Randomization

The domain randomization we use in our policy are listed in Tab III. we add random mass to the hands separately to enhance the terminal load capacity and We raise the robot by 0.1m with a probability of 20% during initialization.

### E. Termination Conditions

To improve training efficiency, we reset training process when the projected gravity on x or y axis exceeds 0.7.

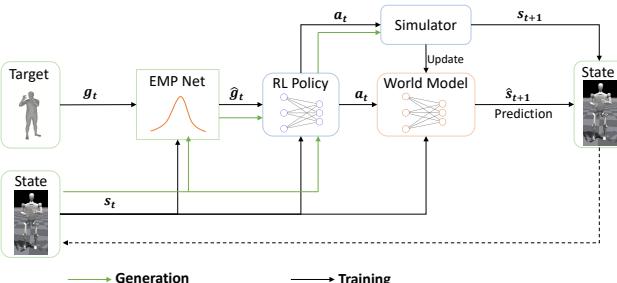


Fig. 3: Framework of our EMP System. EMP network generates optimized upper-body motion targets conditioned on the state of the robot. The world model learns the state transition model from the simulator for gradient backpropagation.

## V. EXECUTABLE MOTION PRIOR

### A. Overview

Our executable motion prior (EMP) network is designed based on the structure of Variational Autoencoder (VAE) [28].

Inspired by the framework of ControlVAE [29], we build our overall framework, shown in Figure 3. EMP adjusts the target upper-body motion based on the current state of the humanoid, improving the humanoid's standing stability while minimizing changes in the motion amplitude.

The EMP network consists of an encoder and a decoder, showed in Figure 2. The encoder is composed of 3 sub-networks: a state encoder  $f_s$ , a target encoder  $f_t$  and a fusion network  $f_\psi$ . The state encoder and target encoder encode the state and the target into the latent space variable  $z_1, z_2$ , respectively.

$$z_1, z_2 = f_s(s_t), f_t(g_t) \quad (2)$$

Then the fusion network encodes two variables into a single latent space vector  $z = f_\psi(z_1, z_2)$ , which follows a standard normal distribution  $z \sim \mathcal{N}(0, 1)$  and then the decoder generates new target for the humanoid. Then the EMP can be described as:

$$\hat{g}_t = f_\theta(s_t, g_t) \quad (3)$$

where  $\theta$  is the learnable variable of EMP net. The encoder and decoder net are both MLP networks.

### B. Training

The training process consists of two parts: world model  $f_w$  training and EMP training.

**World Model Training.** Due to the inability to obtain gradients from the robot state information in the simulation, we use a world model to simulate the state transition process of the humanoid robot environment. The world model predicts the next state of the humanoid robot depending on the current state and action:

$$\hat{s}_{t+1} = f_w(s_t, a_t) \quad (4)$$

where  $w$  is the learnable variable of world model. Then we have world model prediction loss:

$$L_{pre} = \|s_{t+1} - \hat{s}_{t+1}\|_2^2 \quad (5)$$

where  $s_{t+1}$  is the state given by the simulator, namely isaac-gym here and  $\hat{s}_{t+1}$  is the prediction of the world model. The state here is defined the same as section IV-B.

**EMP Training.** While the robot is losing its balance, the following situations usually occur: (1) The center of gravity is projected away from the support surface; (2) The robot's torso is no longer oriented vertically upwards. Therefore we train the network to avoid these situations. Meanwhile, the self-collision and smoothness of the motion can also influence the balance.

The training process of EMP is illustrated in Figure 3. We have the following losses:

i) *Reconstruction Loss.* The reconstruction loss  $L_{rec}$  encourages the generated motion  $\hat{g}_t$  to be as identical to the source target  $g_t$ . We define

$$L_{rec} = \|g_t - \hat{g}_t\|_2^2 \quad (6)$$

ii) *Orientation Loss.* The orientation loss  $L_{ori}$  promotes the humanoid's base to stay upright, which can improve the stability of the humanoid. Then  $L_{ori}$  is defined as

$$L_{ori} = \exp(-\|\widehat{pg}_{t+1}^{xy}\|_2^2) - 1 \quad (7)$$

where  $\widehat{\mathbf{pg}}_{t+1}^{xy}$  is the projected gravity vector, which is calculated from  $\widehat{\mathbf{rpy}}_{t+1}^{xy}$  predicted by the world model.

iii) *Collision Loss*. The collision loss  $L_{col}$  encourages the motion to reduce self-collision of the humanoid. We simplify the links that may collide into a spherical model, and calculate the distance between the links. We define

$$L_{col} = \sum_{i,j \in \mathbb{J}} \exp[-2(0.08 - \|\mathbf{p}_i - \mathbf{p}_j\|_2)] \quad (8)$$

where  $\mathbb{J}$  is the set of the links that may collide each other, we define  $\mathbb{J} = \{\text{torso, hand, sacrum, thigh}\}$  here.  $\mathbf{p}_i$  and  $\mathbf{p}_j$  mean the coordinate of the links centers, which can be calculated with forward kinematics (FK).

iv) *Centroid Loss*. The centroid loss  $L_{cen}$  prompts the centroid of humanoid to stay in the range of support surface under foot.  $L_{cen}$  is defined as

$$L_{cen} = \min\{\exp(-7(0.03 - d)), 10\} - 1 \quad (9)$$

where  $d$  is the distance between the center of the foot support surface and the projection of the centroid onto the ground.

v) *Smoothness Loss*. The smoothness loss  $L_{smo}$  promotes the motion to be smooth and reduce the occurrence of motion mutations.  $L_{smo}$  is defined as

$$L_{smo} = \|\hat{\mathbf{g}}_t - \hat{\mathbf{g}}_{t-1}\|_2^2 + 0.2\|\hat{\mathbf{g}}_t + \hat{\mathbf{g}}_{t-2} - 2\hat{\mathbf{g}}_{t-1}\|_2^2 \quad (10)$$

vi) *Regularization Loss*. The regularization loss  $L_{reg}$  encourages the latent variable to conform to standard Gaussian distribution.  $L_{reg}$  is defined as

$$L_{reg} = \|\mathbf{z}\|_2^2 \quad (11)$$

where  $\mathbf{z}$  is the latent variable.

Finally we get overall loss for EMP training:

$$\begin{aligned} L = & \lambda_{rec} L_{rec} + \lambda_{ori} L_{ori} + \lambda_{col} L_{col} \\ & + \lambda_{cen} L_{cen} + \lambda_{smo} L_{smo} + \lambda_{reg} L_{reg} \end{aligned} \quad (12)$$

We set  $\lambda_{rec} = 20, \lambda_{ori} = 10, \lambda_{col} = 1, \lambda_{cen} = 10, \lambda_{smo} = 100, \lambda_{reg} = 1$  here. The overall training process is shown in Algorithm 1.

### C. Generation

The Generation process is illustrated in Figure 3. With the trained prior distribution, the EMP net generates the executable target for humanoid from source target and state.

## VI. EXPERIMENTS

### A. Simulation Experiments

**Hardware Platform.** The main humanoid platform we use is a full-sized robot (1.65m, 60kg) which feature 27 degrees of freedom, including two 7-dof arms (about 6kg for one arm, which brings higher load capacity and control difficulty), two 6-dof legs and one 1-dof in waist.

**Implementation Details.** The encoder and decoder of retargeting network are both graph convolutional neural network with three graph convolutional layers, and the hidden sizes are [16,32,64] and [66,32,16], respectively. The codebook of retargeting network has 2048 latent space vectors, each with a dimensionality of 64. The world model is implemented as

---

### Algorithm 1 Training process of EMP

---

```

1: for number of training epochs do:
2:   for batch of motions in training set do:
3:     Reset simulation environment;
4:     for  $t \leftarrow 0$  to  $T - 1$  do:
5:       Sample  $\mathbf{a}_t = \pi(\mathbf{s}_t, \mathbf{g}_t)$ ;
6:       Sample  $\mathbf{s}_{t+1}$  and  $\hat{\mathbf{s}}_{t+1} = f_w(\mathbf{s}_t, \mathbf{a}_t)$ ;
7:       Update world model  $f_w$  with  $\nabla_w L_{pre}$ ;
8:     end for
9:     Reset simulation environment;
10:    for  $t \leftarrow 0$  to  $T - 1$  do:
11:      Sample  $\hat{\mathbf{g}}_t = f_\theta(\mathbf{s}_t, \mathbf{g}_t)$ ;
12:      Sample  $\mathbf{a}_t = \pi(\mathbf{s}_t, \hat{\mathbf{g}}_t)$ ;
13:      Sample  $\hat{\mathbf{s}}_{t+1} = f_w(\mathbf{s}_t, \mathbf{a}_t)$ ;
14:      Update EMP  $f_\theta$  with  $\nabla_\theta L$ ;
15:    end for
16:  end for
17: end for

```

---

multi-layer perceptrons (MLP) with hidden size of [1024,512]. The state encoder and target encoder of EMP network are MLPs with hidden sizes of [1024,1024], and the fusion network and decoder are MLPs with hidden sizes of [2048,2048]. The RL training is conducted on an NVIDIA A800 (80GB) GPU and takes about 6 hours with a learning rate of 1e-3 in Isaac Gym [30]. The EMP network is trained on an NVIDIA RTX4060 GPU for 5 hours.

**Motion Dataset.** We use our retargeting network to build our humanoid motion dataset. We choose GRAB dataset [31] in AMASS dataset [32] as our source motion dataset. We train our network on the dataset and use the retargeting results for RL policy training. We divide these motions into smaller motions of the same length, with each motion being 60 frames long, and then reconnect them to eliminate the inconvenience caused by varying motion lengths. For EMP training, we divide these motions into smaller motions with 200 frames long, facilitating our batch collecting process.

**EMP Training.** We train our Executable motion prior (EMP) on retargeted GRAB dataset, which we randomly divided into a training set (1,070 motions) and a test set (270 motions). We train the world model and EMP with Adam [33] optimizer with an initial learning rate of 1e-3.

**Baselines.** We consider the following baselines:

- i) **Privileged Policy.** Referring to the settings in [9], the observation space for the privileged policy input includes all first-hand robot state, and no noise or domain randomization is added during training. The privileged policy demonstrates the upper limit of the robot's mobility.
- ii) **Whole-Body Policy.** Instead of only control lower-body joints, the whole-body policy controls all 27 joints. We train this policy based on the rewards and methods in Exbody [8] and HumanPlus [11]. We use this policy to track upper-body motions and keep lower-body joints in default angles.
- iii) **Decoupled Imitation Policy.** Our main RL policy, which controls lower-body joints to keep balance while tracking

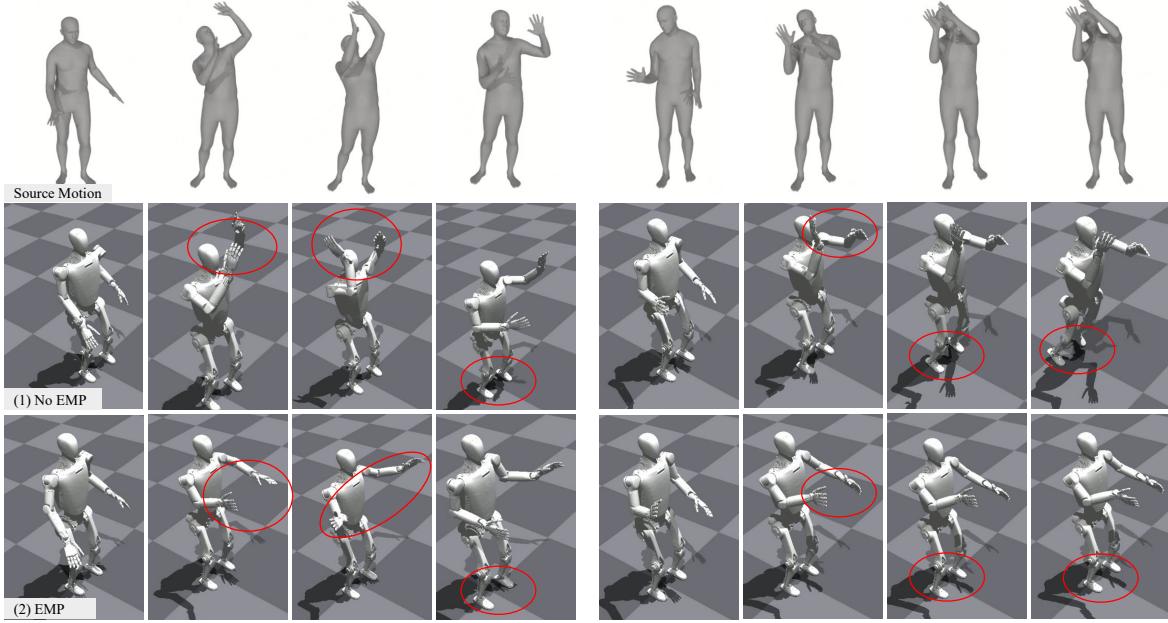


Fig. 4: Simulation experiments (left motion: hammer use, right motion: lightbulb screw). The results show that while executing dangerous motions, EMP network will optimize the unexecutable motion and keep the robot stand stably.

upper-body motions.

- iv) **Decoupled Imitation Policy with Predictive Motion Prior (PMP) [10].** We add PMP features into the observation state of decoupled policy.
- v) **Decoupled Imitation Policy with EMP.** Our full system, RL policy with executable motion prior.
- vi) **EMP when Danger.** Enable EMP only when regloss of the latent space exceeds 0.04. The regloss reflects the degree of the motion deviation from prior distribution.

**Metrics.** The metrics are as follows:

- **Success Rate (SUC).** We define imitation failed when termination conditions in section IV-E are triggered.
- **Mean upper-joint position reward (MJP).** We define upper-body joint position reward as  $r_{jp} = \exp(-\|q_t - g_t\|_2)$ .
- **Mean self-collision reward (MSC).** Self-collision often happens while tracking motions, which will seriously disturb the balance control of the robots. We use link contact force to evaluate this metric:  $r_{col} = -\|f_t\|_2$ . We only consider the contact between these links: torso, thigh, hand and sacrum.
- **Mean Base Velocity reward (MBV).**
- **Mean Base Acceleration reward (MBA).**
- **Mean Base Orientation reward (MBO).**
- **Mean upper-body action Smoothness (MUS).** We calculate the velocity rate of upper-body joints to evaluate smoothness:  $r_{smo} = \|\dot{q}_t - \dot{q}_{t-1}\|^2$

**Results.** We deployed our system on the humanoid robot. The results of simulation experiments are summarized in Tab IV. Note that we randomly add 0.8 ~ 1.2kg to both hands as hand load. The results reveal that our EMP methods outperform other baselines. Compared with Decoupled Policy, the Whole-Body Policy has a higher success rate and completely avoids collisions (the main reason is that we introduced upper-body

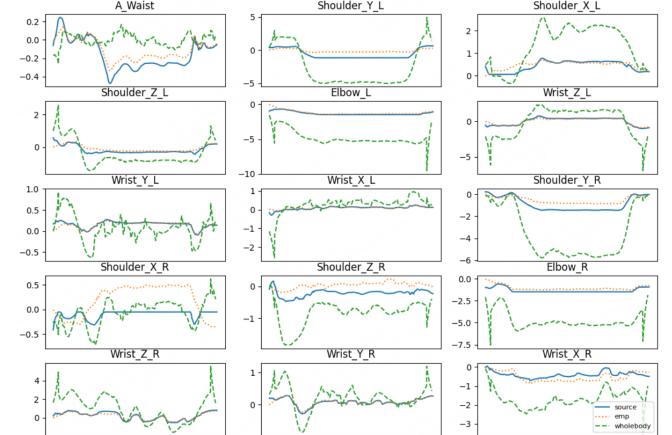


Fig. 5: The upper motion (lightbulb screw) of whole-body policy and EMP. The whole-body policy brings significant vibration to upper-body motions.

collision penalties during training). However, the whole-body policy performs poor in other metrics, especially upper-body motion smoothness. The PMP baseline has achieved a certain improvement in base decoupled policy, but its effectiveness is weaker than that of EMP.

The EMP network optimizes upper-body motion while minimizing deviations as much as possible, thereby improving control stability. The acceleration, velocity and orientation stability of the base are improved remarkably and the collision is reduced while the joint position error is lightly increased.

Figure 4 shows some simulation results. While the upper-body motions are executable, our framework maintains consistency with the initial motions. Once the amplitude of the motion exceeds the control capability of the controller, the EMP will optimize the motion to keep the overall robot stable and avoid falling situations. We analyze the upper-body motion variation curve in Figure 5. We can see that while acting

TABLE IV  
EXPERIMENT RESULTS

Baselines	Metrics						
	SUC ↑	MJP ↑	MSC ↓	MBV ↑	MBA ↑	MBO ↑	MUS ↓
Privileged Policy	100%	0.8121	0.3856	0.8186	0.7158	0.7702	2.4420
Whole-Body Policy	<b>100%</b>	0.7915	<b>0.0</b>	0.7153	0.6801	0.5204	7.6708
Decoupled Policy	97.0%	<b>0.8295</b>	0.3668	0.7973	0.7533	0.6699	<b>2.2842</b>
PMP	97.4%	0.8289	0.3741	0.7790	0.7295	0.6800	2.3022
EMP (Ours)	98.1%	0.8221	0.1494	<b>0.8036</b>	0.7588	<b>0.6892</b>	2.3678
EMP when Danger	98.1%	0.8221	0.1476	0.8029	<b>0.7602</b>	0.6868	2.3527

upper-body motions, whole-body policy exhibits noticeable oscillations, especially in wrist joints.

TABLE V  
ABLATION STUDY

Methods	SUC ↑	MJP ↑	MSC ↓	MBV ↑	MBA ↑	MBO ↑	MUS ↓
Full EMP	<b>98.1%</b>	<b>0.822</b>	<b>0.149</b>	<b>0.804</b>	<b>0.759</b>	<b>0.689</b>	<b>2.368</b>
EMP w/o smoothness	27.0%	0.637	0.211	0.702	0.591	0.555	5.434
EMP w/o orientation	2.6%	0.327	3.982	0.470	0.283	0.375	12.82
EMP w/o centroid	10.7%	0.396	2.850	0.531	0.232	0.422	11.00

### B. Ablation Study

To validate the impact of different losses on the effectiveness of EMP, we conducted ablation experiments on smoothness loss, orientation loss and centroid loss. As illustrated in Tab V, the results of the ablation experiments indicate that all three loss functions play an important role in the training of the EMP network. The absence of these loss functions not only affects the directly related metrics but also impacts the overall stability of the system. In contrast, the impact of the smoothness loss on the system is smaller than that of the other two losses.

### C. Real-world Experiments

**Deployment Settings.** We test our system on real-world humanoid robot platform. All the proprioception of the robot comes from build-in sensors. The algorithm we deployed on the real-world system is baseline iv). Our RL policy and EMP runs at 50Hz. The PD controller is running at 1kHz.

**Motions Imitation.** We test several human motions from AMASS dataset in Figure 6. Note that the safety rope connected to the head of the humanoid is just for protection.

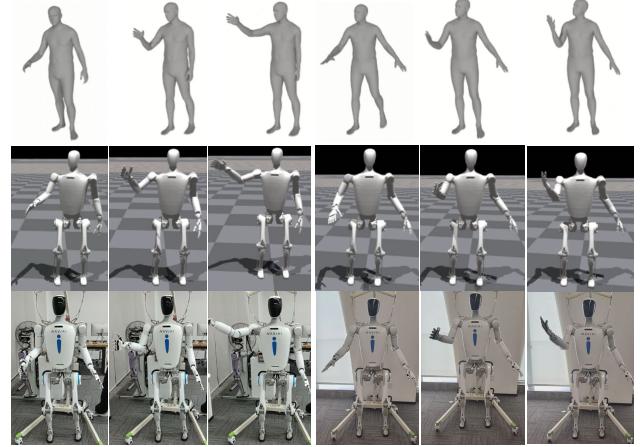


Fig. 6: Humanoid robot imitating dataset motion.

### D. Experiments on Another Platform

We have also deployed our system on another humanoid platform, which also features two 7-dof arms, two 6-dof legs and one 1-dof in waist.

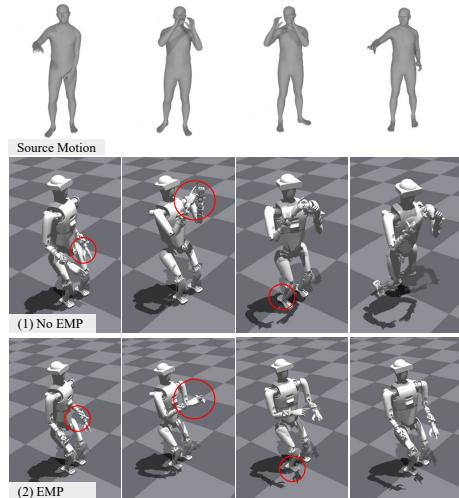


Fig. 7: Simulation experiment on another platform

The results are shown in Figure 7, and the metrics of partial baselines are shown in Table VI. The results show that our framework also performs well in older platforms.

TABLE VI  
SIMULATION RESULTS ON ANOTHER ROBOT

Baselines	SUC ↑	MJP ↑	MSC ↓	MBV ↑	MBA ↑	MBO ↑	MUS ↓
Privileged Policy	99.3%	0.840	1.317	0.787	0.284	0.702	3.642
Whole-body Policy	64.8%	0.318	2.893	0.525	0.176	0.480	8.148
Decoupled Policy	90.0%	0.841	1.748	0.792	0.381	0.727	1.604
EMP (Ours)	<b>97.8%</b>	<b>0.861</b>	<b>0.129</b>	<b>0.807</b>	<b>0.394</b>	<b>0.754</b>	<b>1.435</b>
EMP when Danger	95.9%	0.835	0.799	0.797	0.371	0.742	1.691

## VII. CONCLUSIONS AND FUTURE WORK

In this work, we introduce a framework that enables the humanoid to imitate upper-body motions retargeted from human motions. We train a retargeting network from a humanoid motion dataset and a upper-body imitation RL policy to control the humanoid to keep balance while tracking motions. Then our approach utilize executable motion prior before RL controller to transform difficult motions into executable targets that fit the humanoid control ability. Through simulations and real-world tests, we validated the effectiveness of our framework. However, we have not realized whole-body motion imitation due to high DoF and complex dynamics of the full-sized humanoid robot. Meanwhile, joint limitations of the robot result in a significant disparity between the retargeted motions and the source movements. We hope to address these limitations in future to build a whole-body motion imitation system.

## REFERENCES

- [1] Y. Ishiguro, T. Makabe, Y. Nagamatsu, Y. Kojio, K. Kojima, F. Sugai, Y. Kakiuchi, K. Okada, and M. Inaba, "Bilateral humanoid teleoperation system using whole-body exoskeleton cockpit tablis," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6419–6426, 2020.
- [2] Y. Ishiguro, K. Kojima, F. Sugai, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, "High speed whole body dynamic motion experiment with real time master-slave humanoid robot system," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 5835–5841.
- [3] J. Ramos and S. Kim, "Humanoid dynamic synchronization through whole-body bilateral feedback teleoperation," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 953–965, 2018.
- [4] X. B. Peng, Y. Guo, L. Halper, S. Levine, and S. Fidler, "Ase: large-scale reusable adversarial skill embeddings for physically simulated characters," *ACM Transactions on Graphics*, vol. 41, no. 4, p. 1–17, July 2022. [Online]. Available: <http://dx.doi.org/10.1145/3528223.3530110>
- [5] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, "Amp: adversarial motion priors for stylized physics-based character control," *ACM Transactions on Graphics*, vol. 40, no. 4, p. 1–20, July 2021. [Online]. Available: <http://dx.doi.org/10.1145/3450626.3459670>
- [6] J. Siekmann, K. Green, J. Warila, A. Fern, and J. Hurst, "Blind bipedal stair traversal via sim-to-real reinforcement learning," 2021. [Online]. Available: <https://arxiv.org/abs/2105.08328>
- [7] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control," 2024. [Online]. Available: <https://arxiv.org/abs/2401.16889>
- [8] X. Cheng, Y. Ji, J. Chen, R. Yang, G. Yang, and X. Wang, "Expressive whole-body control for humanoid robots," 2024. [Online]. Available: <https://arxiv.org/abs/2402.16796>
- [9] T. He, Z. Luo, W. Xiao, C. Zhang, K. Kitani, C. Liu, and G. Shi, "Learning human-to-humanoid real-time whole-body teleoperation," 2024. [Online]. Available: <https://arxiv.org/abs/2403.04436>
- [10] C. Lu, X. Cheng, J. Li, S. Yang, M. Ji, C. Yuan, G. Yang, S. Yi, and X. Wang, "Mobile-television: Predictive motion priors for humanoid whole-body control," 2025. [Online]. Available: <https://arxiv.org/abs/2412.07773>
- [11] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn, "Humanplus: Humanoid shadowing and imitation from humans," in *arXiv*, 2024.
- [12] Z. Popović and A. Witkin, "Physically based motion transformation," in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '99. USA: ACM Press/Addison-Wesley Publishing Co., 1999, pp. 11–20. [Online]. Available: <https://doi.org/10.1145/311535.311536>
- [13] R. Rekik, M. Marsot, A.-H. Olivier, J.-S. Franco, and S. Wuhrer, "Correspondence-free online human motion retargeting," 2024. [Online]. Available: <https://arxiv.org/abs/2302.00556>
- [14] D. Holden, J. Saito, and T. Komura, "A deep learning framework for character motion synthesis and editing," *ACM Trans. Graph.*, vol. 35, no. 4, jul 2016. [Online]. Available: <https://doi.org/10.1145/2897824.2925975>
- [15] H. Jang, B. Kwon, M. Yu, S. U. Kim, and J. Kim, "A variational u-net for motion retargeting," in *SIGGRAPH Asia 2018 Posters*, ser. SA '18. New York, NY, USA: Association for Computing Machinery, 2018. [Online]. Available: <https://doi.org/10.1145/3283289.3283316>
- [16] K. Aberman, P. Li, D. Lischinski, O. Sorkine-Hornung, D. Cohen-Or, and B. Chen, "Skeleton-aware networks for deep motion retargeting," *ACM Transactions on Graphics*, vol. 39, no. 4, Aug. 2020. [Online]. Available: <http://dx.doi.org/10.1145/3386569.3392462>
- [17] H. Zhang, Z. Chen, H. Xu, L. Hao, X. Wu, S. Xu, R. Xiong, and Y. Wang, "Unified cross-structural motion retargeting for humanoid characters," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–14, 2024.
- [18] B. Delhaisse, D. Esteban, L. Rozo, and D. Caldwell, "Transfer learning of shared latent spaces between robots with similar kinematic structure," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 4142–4149.
- [19] K. Ayusawa and E. Yoshida, "Motion retargeting for humanoid robots based on simultaneous morphing parameter identification and motion optimization," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1343–1357, 2017.
- [20] Z. Luo, J. Cao, A. Winkler, K. Kitani, and W. Xu, "Perpetual humanoid control for real-time simulated avatars," 2023. [Online]. Available: <https://arxiv.org/abs/2305.06456>
- [21] Y. Yuan and K. M. Kitani, "Residual force control for agile human behavior imitation and extended motion synthesis," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 2020.
- [22] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath, "Learning humanoid locomotion with transformers," *arXiv:2303.03381*, 2023.
- [23] K. Darvish, L. Penco, J. Ramos, R. Cisneros, J. Pratt, E. Yoshida, S. Ivaldi, and D. Pucci, "Teleoperation of humanoid robots: A survey," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1706–1727, 2023.
- [24] J. Z. Zhang, S. Yang, G. Yang, A. L. Bishop, D. Ramanan, and Z. Manchester, "Slomo: A general system for legged robot motion imitation from casual videos," 2023. [Online]. Available: <https://arxiv.org/abs/2304.14389>
- [25] J. Ramos and S. Kim, "Dynamic locomotion synchronization of bipedal robot and human operator via bilateral feedback teleoperation," *Science Robotics*, vol. 4, no. 35, p. eaav4282, 2019. [Online]. Available: <https://www.science.org/doi/abs/10.1126/scirobotics.aav4282>
- [26] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," 2018. [Online]. Available: <https://arxiv.org/abs/1711.00937>
- [27] H. Zhang, W. Li, J. Liu, Z. Chen, Y. Cui, Y. Wang, and R. Xiong, "Kinematic motion retargeting via neural latent optimization for learning sign language," 2022. [Online]. Available: <https://arxiv.org/abs/2103.08882>
- [28] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2022. [Online]. Available: <https://arxiv.org/abs/1312.6114>
- [29] H. Shao, S. Yao, D. Sun, A. Zhang, S. Liu, D. Liu, J. Wang, and T. Abdelzaher, "Controlvae: Controllable variational autoencoder," 2020. [Online]. Available: <https://arxiv.org/abs/2004.05988>
- [30] V. Makovychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac gym: High performance gpu-based physics simulation for robot learning," 2021. [Online]. Available: <https://arxiv.org/abs/2108.10470>
- [31] O. Taheri, N. Ghorbani, M. J. Black, and D. Tzionas, *GRAB: A Dataset of Whole-Body Human Grasping of Objects*. Springer International Publishing, 2020, pp. 581–600. [Online]. Available: [https://doi.org/10.1007/978-3-030-58548-8\\_34](https://doi.org/10.1007/978-3-030-58548-8_34)
- [32] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "AMASS: Archive of motion capture as surface shapes," in *International Conference on Computer Vision*, Oct. 2019, pp. 5442–5451.
- [33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017. [Online]. Available: <https://arxiv.org/abs/1412.6980>