The Wanderings of Odysseus in 3D Scenes

Yan Zhang Siyu Tang
ETH Zürich
{yan.zhang,siyu.tang}@inf.ethz.ch

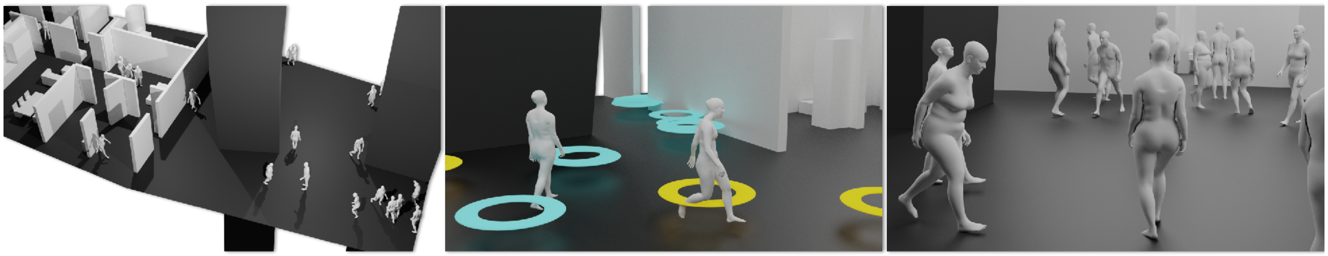

Figure 1. We propose **GAMMA**, an automatic and scalable solution to populate the 3D scene with digital humans, which have 1) varied body shapes, 2) plausible body-ground contact, and 3) realistic and perpetual motions to reach goals (e.g. rings in the middle figure), with the off-the-shelf path-finding algorithm.

Abstract

Our goal is to populate digital environments, in which digital humans have diverse body shapes, move perpetually, and have plausible body-scene contact. The core challenge is to generate realistic, controllable, and infinitely long motions for diverse 3D bodies. To this end, we propose generative motion primitives via body surface markers, or GAMMA in short. In our solution, we decompose the long-term motion into a time sequence of motion primitives. We exploit body surface markers and conditional variational autoencoder to model each motion primitive, and generate long-term motion by implementing the generative model recursively. To control the motion to reach a goal, we apply a policy network to explore the generative model's latent space and use a tree-based search to preserve the motion quality during testing. Experiments show that our method can produce more realistic and controllable motion than state-of-the-art data-driven methods. With conventional path-finding algorithms, the generated human bodies can realistically move long distances for a long period of time in the scene. Code is released for research purposes at: https://yz-cnsdqz.github.io/eigenmotion/GAMMA/

1. Introduction

In recent years, the rapid development of 3D technologies has accelerated the creation of a digital replica of the real world and initiated new ways that people interact with the world and communicate with each other. However, there is no existing solution to automatically populate the digital world with realistic virtual humans, which move and act like real ones. This work aims to enable virtual humans to cruise within a 3D digital environment, similar to Odysseus, who arrived home after wandering and hazards. The virtual humans follow randomized routes, pass individual waypoints, and reach the destination, while retaining realistic body shape, pose, and body-scene contact. Such technology can considerably enrich AR/VR user experiences and has many downstream applications. For example, having virtual humans strolling inside the digital model of a medieval city can make the experience more vivid, which allows real users to follow their guidance for better sightseeing. Beyond AR/VR, virtual humans can provide architects with a blueprint, enabling better foresight into design functionalities and defects.

This is particularly relevant to character animation in graphics, foremost in the gaming industry. Conventionally, a set of 3D characters are pre-designed, and a motion dataset is pre-recorded. To let characters respond to user inputs or background events, motions from the dataset are created via motion graph [34] or motion matching [13]. Although current AAA games demonstrate highly realistic character motion, this conventional technology cannot easily handle a massive number of characters with different behaviors [26]. Motion generalization across diverse characters usually requires extra motion re-targeting procedures [5]. Moreover, the generated motions are often deterministic, close to the pre-recorded clips, and hence have limited diversity.

The availability of large-scale motion capture datasets (e.g. AMASS [44]) facilitates the learning of generative motion models. They can effectively produce motions based on the motion in the past [80], action labels [58], scene context [25] and music [40], without limit on a specific character. Although the motion realism is improved by replacing 3D skeletons with expressive body meshes, e.g., SMPL-X [52], the generated motion is limited to a few seconds. It often has jittering, foot-skating, and other artifacts. To populate the digital environment, we need a fully automated way to generate long-term (potentially infinite) and realistic motions for a large variety of human shapes.

This is a considerably challenging task and far beyond the scope of existing solutions to our knowledge. The first obstacle we encounter is how to generate infinitely long, diverse, and stochastic human motion sequences. Existing methods regard motion as a standard time sequence of high-dimensional feature vectors and propose to model it with a single deep neural network. However, the uncertainty of human motion grows as time progresses. It is unclear whether a deep neural network has sufficient power to represent a perpetual motion.

To overcome this issue, we decompose a long-term motion into a time sequence of motion primitives, model each primitive, and compose them to obtain a long-term motion. Our insight comes from psychological studies. Human reaction time to visual stimuli is about 0.25 seconds [1, 68]. Namely, humans cannot control their body motion immediately after seeing a signal but have to wait for 0.25 seconds to give a response due to body inertia. Therefore, we let a motion primitive span 0.25 seconds. In this case, it mainly contains unconscious body dynamics, which are shorter, more deterministic, and easier to model. Specifically, we exploit the body surface markers to represent the body in motion [80] and use conditional variational autoencoder (CVAE) [33, 64] to model body dynamics. To efficiently recover the 3D body from markers, we design a body regressor with recursions. By blending the marker predictor and body regressor, our model can synthesize realistic long-term motion, which is perceptually similar to high-quality mocap sequences, e.g. from AMASS [44]. Of note, our marker-based motion primitive is generalizable to various body shapes, which initiates populating the 3D scene with a massive amount of virtual humans of different identities.

The second challenge is how to let the virtual humans move naturally within 3D scenes, towards a designated destination, while considering the geometric constraints of the environment. Inspired by Ling et al. [41], we propose a novel motion synthesis pipeline with control, which consists of a policy network and a tree-based search mechanism. We formulate long-term motion generation as a Markov decision process, and use a policy network to explore the CVAE latent space. By sampling from the policy, the body can gradually move to the goal, while keeping the foot-ground contact plausible. Simultaneously, we organize the motion generation process into a tree structure, which searches best motion primitives at each generation step and rejects unrealistic ones. We perform experiments to evaluate motion realism and controllability. Results show that our method can produce realistic long-term motions in 3D scenes and outperform state-of-the-art methods. Combing with conventional path-finding algorithms, e.g., navigation mesh baking and A* search [24, 63], we can populate large-scale 3D scenes with a massive number of virtual humans, which have diverse body shapes, cruise following paths, and finally reach their destinations.

We name our method **GAMMA**, for **G**ener**A**tive **M**otion primitive via body surface **MA**rkers. Code and model are released for research purposes.

2. Related Work

Character animation. A 3D character is normally rigged with a skeleton. To animate it, mocap data is used to actuate the skeleton, and the body mesh is deformed accordingly. To control the character, e.g. following a walking path, methods like motion graph [34] or motion matching [12, 13, 84] are to search for the most suitable motion clips from a dataset and blend them to remove discontinuities. Despite high motion realism, these methods are not well scalable to animate a massive number of characters with a vast mocap dataset [26]. Although recent methods like [27, 37] exploit neural networks to improve scalability and efficiency, the generated motions are often deterministic, close to the training samples, and hence lack diversity.

Generative models for 3D body motion synthesis. Based on large-scale datasets, learned generative models can synthesize motions based on certain conditions. For motion prediction, the model aims at generating motion that is expected to be close to the ground truth, provided on a motion from the past [6, 7, 14, 16, 17, 20–22, 38, 39, 43, 47, 48, 53, 67, 73, 83]. When considering motion uncertainty, stochastic motion prediction is proposed to generate diverse plausible motions based on the same motion seed [8, 9, 11, 18, 41, 46, 70, 75, 76]. Although these methods can generalize well across various body shapes and actions, motion realism is seldom considered. For example, the body is represented with stick figures. The generated motion has a limited time horizon. Scene constraints are not considered, global body configurations are not included.

More recent works exploit parametric body models e.g. SMPL-X [52], and directly produce motions of expressive body meshes. Compared to the skeleton, the motion of 3D bodies looks more realistic to our human eye and is directly compatible with existing rendering pipelines. Zhang et al. [80] extend stochastic motion prediction from stick figures to 3D bodies, which generate diverse global motions using a body surface marker-based representation. Petrovich et al. [58] proposes a transformer VAE to generate stochastic 3D body motions based on action labels. Zhang et al. [79] model dynamics of SMPL-X [52] body parameters with a recurrent network for generating perpetual motions of diverse 3D bodies. Wang et al. [71] generate scene-aware root motion trajectories and synthesize body poses along the trajectory. Test-time optimization procedures are then applied to improve the body-scene contact. Stark et al. [65] propose a neural state machine model to synthesize character motions interacting with objects, such as sitting on the chair and carrying the box. Hassan et al. [25] propose to generate stochastic and dynamic body-scene interactions by synthesizing the goal and the motion individually with neural networks. Ling et al. [41] design an autoregressive VAE to model body dynamics of two consecutive frames, and exploit reinforcement learning to train a policy for task-driven motion synthesis. Rempe et al. [59] propose an autoregressive model to learn 3D body motions, which is generic for body shapes and motion types. Although it focuses on motion estimation, HuMoR can generate random and long-term motions with plausible body-ground contact.

Body motion control. Motion can be controlled in different ways. A straightforward approach is to extend a generative motion model by introducing control signals as additional conditions, like in [23, 28, 32, 54]. Despite their effectiveness, the learned model is sensitive to the train-test domain gap, generating invalid motions when the control signal is much different from the training data. Another control approach is optimization-based, in which the control signal, e.g., fixed foot position when contacting with the ground, is regarded as the data term, and the generative model is used for regularization, e.g. [29, 72]. This shares similarity with body motion recovery from observations as in [30, 42, 59, 78]. They can produce high-quality results, but have high computational costs.

Many reinforcement learning (RL) methods are designed to control infinitely long motion to complete specific tasks. Most of them exploit physical simulation for body dynamics modeling, and propose policy networks to control, e.g. joint torques. To obtain human-like motions, additional data, e.g. videos or mocap sequences, is used for the character to imitate [10, 49, 55, 56]. To automatize which motion to imitate, Peng et al. [57] propose an adversarial imitation learning mechanism, designing an adversarial motion style reward during policy training. Yuan et al. [77] focus on physics-aware motion estimation. They create a humanoid according to the SMPL blend skinning, and thus their humanoid is more human-like than the skeleton. Ling et al. [41] first propose a CVAE-based generative motion model, and then train a policy to produce latent variables for motion control.

Ours vs. others. Our method is data-driven, so no need to create a humanoid in physical simulation as in [49, 77].

We are inspired by MotionVAE [41] and HuMoR [59] for synthesizing task-oriented behavior and motion estimation, respectively, in which the generative model's latent variables are manipulated to produce desired motions. Similar to MotionVAE [41], our method contains a generative motion model and a policy network, but is extended with novel motion models and control methods. Specifically, we first learn models of marker-based motion primitives, using the large-scale motion dataset AMASS [44], and then learn the policy based on the pre-trained motion model, so our method is generic for various body shapes and action types. The policy output distribution is regularized by a KLD term in addition to the PPO term (see Eq. (10)), to encourage the motion naturalness. Moreover, we exploit a tree-based search scheme to further improve the motion quality during testing. Therefore, our method is suitable for populating 3D scenes in an efficient and scalable manner.

3. Method

3.1. Preliminaries

SMPL-X [52]. SMPL-X is a differentiable parametric body model. Provided a compact set of body parameters, it yields an expressive body mesh of a fixed topology with 10,475 vertices, including face and hand details. In our work, the body parameter set is denoted as $\boldsymbol{B} = \{\boldsymbol{\beta}, \boldsymbol{\Theta}\}$. The body shape $\boldsymbol{\beta} \in \mathbb{R}^{10}$ is the lowest 10 components in the SMPL-X body shape PCA space. The body configuration has $\boldsymbol{\Theta} = \{\boldsymbol{t} \in \mathbb{R}^3, \boldsymbol{R} \in \mathbb{R}^3, \boldsymbol{\theta} \in \mathbb{R}^{63}, \boldsymbol{\theta_h} \in \mathbb{R}^{24}\}$, including root translation, root orientation in axis-angle, 21 joint rotations in axis-angle, and two hand poses in the MANO PCA space [60], respectively. We then denote a SMPL-X mesh as $\mathcal{M}(\boldsymbol{B})$. By sampling $\boldsymbol{\beta}$ from the standard normal distribution and sampling $\boldsymbol{\theta}$ from the Vposer [52], we can obtain 3D bodies of diverse shapes and poses quickly. Although our method is implemented with SMPL-X, it can be straightforwardly extended to other parametric body models, e.g. [51, 74].

MOJO [80]. MOJO is a solution to stochastic motion prediction of 3D body meshes, including a surface marker-based representation, a CVAE motion model, and a recursive scheme for 3D body recovery. Corresponding to the marker placement in mocap systems, the MOJO body markers are selected from the SMPL-X mesh template, converting the body motion into a time sequence of point clouds. Compared to the representations like joint locations, surface markers have richer body shape information and provide more constraints on the body degrees-of-freedom (DoF). The reprojection scheme exploits these benefits and effectively recovers the body mesh from predicted markers via optimization. At each time step during inference, markers are selected on the recovered body mesh, and then used as input for the next time step. Since the markers are reprojected to the valid body shape space repeatedly, the marker prediction error is hardly accumulated as time progresses, keeping the 3D body valid.

Despite several advantages, MOJO produces motions with limited length, and cannot control the motion generation process. In addition, the body fitting optimization significantly increases the computation time.

Motion formulation. We decompose a long-term motion into a time sequence of motion primitives with overlaps. Each motion primitive contains a motion seed $\boldsymbol{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_M\}$ with $M = \{1, 2\}$, denoting the overlapped frames, and the future frames $\boldsymbol{Y} = \{\boldsymbol{y}_1, ..., \boldsymbol{y}_N\}$. Each frame is represented by the concatenation of the 3D marker locations. The motion primitive is defined in a canonical space. Following MOJO [80], the canonical coordinate is located at the pelvis in the frame $\boldsymbol{x}_1$. The X-axis is the horizontal component pointing from the left hip to the right hip, the Z-axis is pointing up, and the Y-axis is pointing forward.

3.2. Generative Motion Primitives

3.2.1 Model Architecture

The network architectures of GAMMA are illustrated in Fig. 2. The generative motion primitive model consists of a marker predictor and a body regressor.

The marker predictor. In our work, we exploit the 'SSM2 67' marker placement in MOJO [80] due to its better empirical performance. The marker predictor is cast by a CVAE, which has a condition branch, an encoder, and a decoder. The encoder is only used during training. During inference, we can randomly draw $\boldsymbol{z}$ from $\mathcal{N}(0, \boldsymbol{I})$ to obtain different $\boldsymbol{Y}$ based on the same motion seed $\boldsymbol{X}$. Compared to [76,80], our model has no additional sampling module and latent DCT, and hence is easier to train. Depending on whether the motion seed $\boldsymbol{X}$ contains body dynamics, we design a 1-frame predictor and a 2-frame predictor, denoting the number of frames in the motion seed $\boldsymbol{X}$. Their properties are investigated in the appendix.

The body regressor. This network learns to recover the global translations and the joint rotations simultaneously from the markers within a motion primitive. It takes the predicted markers, an initial body parameter Θ_0, and a provided body shape as input. Similar to HMR [31], the residual blocks are employed in a recursive manner. We set Θ_0, including the global translation, orientation, body pose, and hand pose, to zero in our implementation. To improve back-propagation, the global body orientation and joint rotations are converted into the 6D continuous representation [82] in the input and inside of the network, and changed back to axis-angle at the output. Since the body shape is dependent

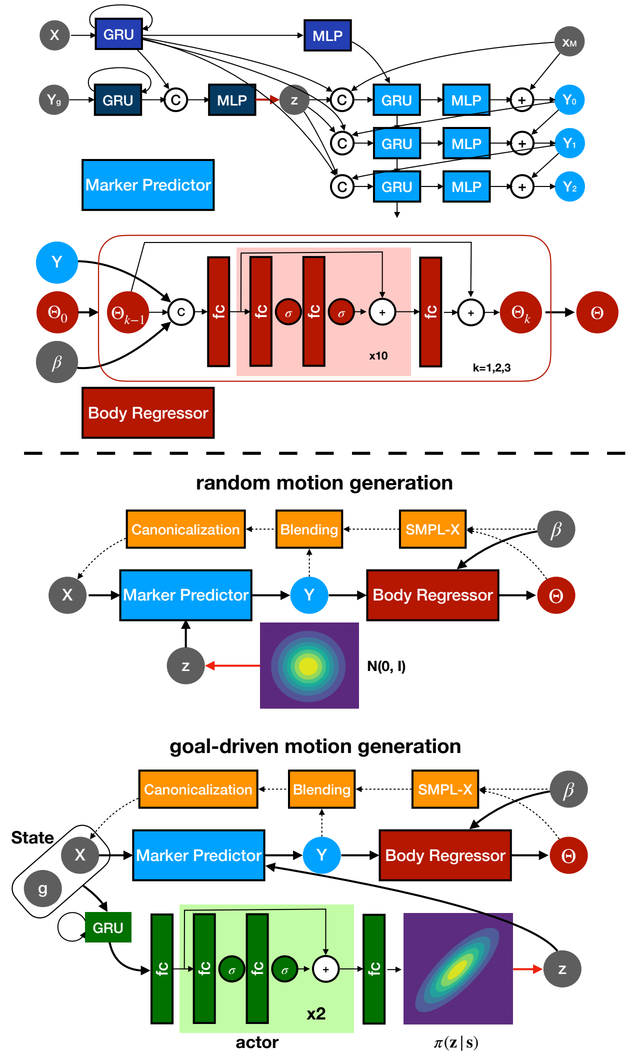

Figure 2. GAMMA architectures. The first two diagrams illustrate the marker predictor and the body regressor, respectively. The bottom two show their combinations. Random motion can be synthesized via sampling $\boldsymbol{z}$ from the standard normal distribution, and goal-driven motion can be synthesized via sampling $\boldsymbol{z}$ from the policy output. The red arrows denote the sampling operation, and the dash curves denote implementing motion primitive models recursively. See respective sections for detailed demonstrations.

on gender, we also propose two versions of body regressors for males and females, respectively, which are separately trained.

3.2.2 Training

Training the marker predictor. First, we train the predictor to learn each individual motion primitive. Similar to MOJO [80], the training loss contains a reconstruction term and a robust KL-divergence term to regularize the latent space, which is formulated to

$$\mathcal{L}_{predictor} = \mathbb{E}_{\boldsymbol{Y}}[|\boldsymbol{Y} - \boldsymbol{Y}^{rec}|] + \lambda \mathbb{E}_{\boldsymbol{Y}}[|\Delta \boldsymbol{Y} - \Delta \boldsymbol{Y}^{rec}|] \\ + \Psi\left(\text{KL-div}(q(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{Y})||\mathcal{N}(0, \boldsymbol{I}))\right), \tag{1}$$

in which ^{rec} means the reconstructed variable, Δ denotes the time difference, and $\lambda = 3$ in our implementation. The KLD term employs a robust function $\Psi(s) = \sqrt{1+s^2} - 1$ [15], which automatically reduces the gradient of the KLD term when its value is small, to alleviate posterior collapse.

Second, we fine-tune the predictor by rolling out longer sequences, like in [41, 48, 59]. Specifically, we use the last one or two frames of a generated motion primitive as the motion seed to generate the next motion primitive, and we minimize the same loss Eq. (1) as above. The transformation between motion primitives is based on the ground truth canonical coordinate. Since this rollout training process takes prediction errors into account, learned generative models can produce long motions stably during testing, and favors recovering body shape when the motion seed is not fully valid. In our experiments, the time horizon of the rollout is set to 8 motion primitives. The 1-frame model and the 2-frame model are trained separately.

Training the body regressor. The body regressor is trained with batches of canonicalized motion primitives. The training loss is based on forward kinematics, and is given by

$$\mathcal{L}_{regressor} = |M \circ \mathcal{M}(\boldsymbol{\Theta}, \boldsymbol{\beta}) - V_{gt}| + \alpha |\boldsymbol{\theta}_h|^2, \tag{2}$$

in which $M\circ$ denotes selecting marker vertices from the mesh template, V_{gt} denotes the ground truth body surface markers, and $\alpha = 0.01$ in our work. This hand regularization term is necessary, since there are only 3-4 markers for each hand, and none of them are on the fingers. The regressor is trained for male and female bodies separately. Although we only train the body regressors with ground truth markers, it performs stably when taking predicted markers as input in our trials. We did not observe the advantages of training the predictor and the regressor jointly in our experiments (see Sec. 4.1).

3.3. Motion Synthesis and Control

By implementing GAMMA recursively, we can either synthesize random motion by sampling $\boldsymbol{z}$ from $\mathcal{N}(0, \boldsymbol{I})$ at all steps, or control the motion to reach the goal by additionally applying a policy network. The recursion mechanism incorporates a SMPL-X body model, a motion blending module, and a canonicalization module, which are illustrated by dash curves and orange blocks in Fig. 2.

The SMPL-X module is to obtain the markers on the body mesh, which is regressed from predicted markers. Compared to the optimization-based reprojection scheme in MOJO [80], our regression is significantly faster without degrading the accuracy (see Tab. S2). However, due to lack of regularization, the regressor cannot guarantee the body is always valid. Invalid bodies can produce corrupted motion seeds, and hence destroy all future motion generation steps. Therefore, the blending module is essential to keep the recursion stable. Although one can use the predicted markers as the motion seed only, prediction error gradually accumulates over time in this case, causing high-frequent body movements and jitters.

3.3.1 Goal-driven Motion Policy

Inspired by [41], we formulate motion synthesis as a Markov decision process and control it via RL to reach a goal. Under this setting, we define a time sequence of tuples $\{(\boldsymbol{s}_t, \boldsymbol{a}_t, r_t)\}_{t=0}^{T}$, denoting the state, the action, and the reward at each primitive generation step.

The state. The state incorporates the motion seed and the normalized vectors from individual markers pointing to the goal in the canonical space. Given the goal location $\boldsymbol{g} \in \mathbb{R}^3$ on the ground plane, the state can be given by

$$\boldsymbol{s}_t = (\boldsymbol{X}, (\boldsymbol{g} - \boldsymbol{X})_{\boldsymbol{n}})^T \in \mathbb{R}^{l \times V \times 6}, \tag{3}$$

in which $l \in \{1, 2\}$ denotes the length of the motion seed, V denotes the number of markers, and $(\cdot)_{\boldsymbol{n}}$ denotes the normalized 3D vector with unit length. This state has balanced dimensions between the motion seed and the goal-based features. Normalizing the vector length is helpful to stabilize the policy training in our trials.

The action. Like in [41], we regard the latent variable $\boldsymbol{z}$ as the action to generate the current motion primitive.

The reward. The reward evaluates the quality of a generated motion primitive, which is given by

$$r = r_{path} + \beta_1 r_{ori} + \beta_2 r_{contact} + \beta_3 r_{vposer} + \beta_4 r_{goal}, \tag{4}$$

including the path following reward, the body orientation reward, the body-ground contact reward, the valid body pose reward, and the goal-reaching reward, respectively. Each of them ranges from 0 to 1. r_{path} encourages moving towards the goal while following the straight path, which can be given by

$$r_{path} = \frac{\langle (\boldsymbol{p}_T^{xy} - \boldsymbol{p}_0^{xy})_{\boldsymbol{n}}, (\boldsymbol{g}^{xy} - \boldsymbol{p}_0^{xy})_{\boldsymbol{n}} \rangle + 1}{2}, \tag{5}$$

in which $(\boldsymbol{p}_T^{xy} - \boldsymbol{p}_0^{xy})_{\boldsymbol{n}}$ denotes the normalized pelvis movement along the ground plane, and $(\boldsymbol{g}^{xy} - \boldsymbol{p}_0^{xy})_{\boldsymbol{n}}$ denotes the normalized direction to the goal along the ground plane. r_{ori} encourages the body to face the goal and can be formulated as

$$r_{ori} = \frac{\langle \boldsymbol{o}_T^{y}, (\boldsymbol{g}^{xy} - \boldsymbol{p}_T^{xy})_{\boldsymbol{n}} \rangle + 1}{2}, \tag{6}$$

in which $\boldsymbol{o}_T$ denotes the unit vector of the body facing direction at the last motion primitive frame. $r_{contact}$ encourages the body moving along the ground plane and discourage skating, which is formulated as

$$r_{contact} = e^{-|\boldsymbol{p}_T^z - \boldsymbol{p}_0^z|} \cdot e^{-|\boldsymbol{Y}_{vel}^*|_2}, \tag{7}$$

which discourages the pelvis Z-location shift and encourages the minimal marker speed close to zero within a motion primitive. r_{vposer} encourages the body pose to keep valid, and is formulated as

$$r_{vposer} = e^{-|\boldsymbol{\mu}(\boldsymbol{\theta})|_2}, \tag{8}$$

which gives a higher value if the encoded pose in the VPoser [52] latent space is closer to 0. The reward r_{goal} is added to all the motion primitives in the entire motion sequence according to the closest distance between any motion primitive to the goal. This reward can be formulated as

$$r_{goal} = \begin{cases} \left(\frac{L - |\boldsymbol{p}_*^{xy} - \boldsymbol{g}|_2}{L} \right)_+ & \text{if} |\boldsymbol{p}_*^{xy} - \boldsymbol{g}|_2 > \epsilon \\ 1 & \text{otherwise} \end{cases} \tag{9}$$

in which $\boldsymbol{p}_*^{xy}$ denotes the pelvis XY-location at the last frame of the closest motion primitive, L denotes the range of the activity area, $(\cdot)_+$ thresholds values to be non-negative, and ϵ is the tolerance defining the goal is reached.

Policy network and training. We employ the actor-critic mechanism [66] for training the policy. The actor produces a diagonal Gaussian distribution $\pi(\boldsymbol{z}_t|\boldsymbol{s}_t)$, and its architecture is illustrated in Fig. 2. The critic network shares the GRU with the actor and is only used in the training phase to estimate the expected return for the advantage function.

To train the policy, we set up a simulation environment. The area range L is a $20 \times 20m^2$ ground plane. At the beginning of the simulation, we create a character with a random gender, body shape, and pose, and place it in the center of the area with a random facing orientation. We then randomly set a goal on the ground following a uniform distribution. We first use the 1-frame model to produce 32 motion primitives from the same initial body, and then use the 2-frame model to generate motions for each body clone. The simulation terminates after generating 60 primitives, or all motions reach the goal. The goal is regarded as reached if its horizontal distance to the body pelvis is smaller than 0.75m in our trials.

We train the policy network for 500 epochs and run the simulation 8 times for each epoch. Therefore, we collect motion data from 4000 random body-goal pairs. We update the policy network at each epoch by minimizing the following loss

$$\begin{aligned}\mathcal{L}_{policy} = \mathcal{L}_{PPO} + \mathbb{E}\left[(R_t - V(\boldsymbol{s}_t))^2\right] \\ + \alpha\Psi\left(\text{KL-div}(\pi(\boldsymbol{z}|\boldsymbol{s})||\mathcal{N}(0, \boldsymbol{I}))\right),\end{aligned} \tag{10}$$

in which the first term updates the policy network with PPO [61], the second term is to update the critic network for better value estimation, and the third term is to regularize motion in the latent space. R_t denotes the expected return from rewards with a discount factor. More details are referred to the appendix.

3.3.2 Tree-based Search

Given the probabilistic nature of our generative model, there is no hard constraint on the body-scene interaction, when sampling motions from the latent space. Therefore, we exploit a tree-based search to discard motion primitives with inferior body-ground contact during test time.

Inspired by random tree-based motion planning [35, 36], we organize the motion generation process into a tree structure, with the root being the initial body pose. Each node denotes a motion primitive, and hence has one parent and multiple children. In addition, each node has a quality or cost value, and all nodes at the same level are ranked. During generation, we can keep the best K nodes to yield new motions, and discard the rest. Specifically, at the first level, we generate N primitive nodes from the root, and only keep the best K while discarding others. At the second level, we will have KN nodes after generation, and we keep the best K. And so on. This tree-based search works both for $\boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I})$ and $\boldsymbol{z} \sim \pi(\boldsymbol{z}|\boldsymbol{s})$. With the learned policy, the search space is significantly reduced.

The design of the node quality/cost depends on the application scenario and can be different from the reward. For example, one can set a high weight on the foot-ground contact to discard floating bodies, or put a high weight on the distance to the goal to encourage fast movement.

4. Experiment

In this section, we perform empirical evaluations on motion realism and motion controllability. Additional experiments, including the performance of the marker predictor and the body regressor, comparison with other body representations, the influence of bodies on motion generation, and runtime test, are presented in the appendix.

Dataset. We exploit the large-scale mocap data **AMASS** [44]. Specifically, we train the marker predictors on **CMU** [3], **MPI HDM05** [50], **BMLmovi** [19], **KIT** [45] and **Eyes Japan** [2]. When training the body regressors, we additionally use **BMLrub**, **Transitions** and **TotalCapture** [69], but exclude **KIT** for testing (see Sec. A.1).

For each mocap sequence, we downsample it to 40fps, and trim it into motion primitives. Each motion primitive clip contains 10 frames or 0.25 seconds, and is canonicalized before model training. In addition, we also prepare longer canonicalized clips with 10 motion primitives, which are used for training with rollout.

4.1. Evaluation on Motion Realism

We randomly select 100 static poses from **HumanEva** and **ACCAD**, respectively, and generate a 10-second motion based on each static pose.

Baseline and our methods. To our knowledge, HuMoR [59] is the state-of-the-art generative model for generating long-term motions of diverse 3D bodies. To produce motions with static poses like ours, we set zero velocities to the initial bodies and generate 10-second motions.

Our method has several versions. 'ours-e2e' denotes training predictor and regressor end-to-end (see appendix). 'ours-ro' exploits a marker predictor with rollout training, and blends the predicted markers and the reprojected markers by averaging. 'ours-reproj' exploits a marker predictor without rollout training, and only uses the reprojected markers on the 3D body as the next motion seed. The suffix '-xf' denotes that the number of frames in the motion seed is x. We also use 'ours-ro-policy' to test the influence of the policy network. In this case, we set a random goal for each initial static body, and generate a 10-second motion. Moreover, we randomly choose 200 sequences from **AMASS** to measure how far we are from generating real motion. For a fair comparison, neither test-time optimization nor tree-based search is performed in this experiment.

Evaluation metrics. The metrics are about *body-ground contact* and *perceptual study*. For the *body-ground contact*, we set a threshold height of 0.05m from the ground plane and a speed threshold of 0.075m/s for skating. Then the contact score is defined as

$$s = e^{-(\min |\boldsymbol{Y}_z| - 0.05)_+} \cdot e^{-(\min |\boldsymbol{Y}_{vel}| - 0.075)_+}, \tag{11}$$

in which $|\boldsymbol{Y}_z|$ and $|\boldsymbol{Y}_{vel}|$ denote the height absolute value and the velocity magnitude, respectively. $(\cdot)_+$ denotes the zero-threshold function equivalent to ReLU. Therefore, this score ranges from 0 to 1, and is the higher, the better. For the *perceptual study*, we render the 100 generated motions from **HumanEva** and let Amazon Mechanical Turk users evaluate on a six-point Likert scale from 'strongly disagree' (1) to 'strongly agree' (6), similar to [80, 81].

Results and discussion. The results are shown in Fig. 3. First, we can see that our methods consistently outperform HuMoR w.r.t. the body-ground contact. In particular, the motion generated by the policy network outperforms all others. This shows the contact reward for the policy training is effective. We can also observe that the end-to-end training is not favorable in this case. Second, user study results show that our methods produce perceptually more realistic motions than HuMoR. By comparing 'ours-ro-policy' and 'ours-ro-2f', the perceptual score increases by a large margin, indicating the policy's effectiveness. Since goal-driven behavior is common in our daily lives, our generated motions with goals are more perceptually realistic than long-term random motions.

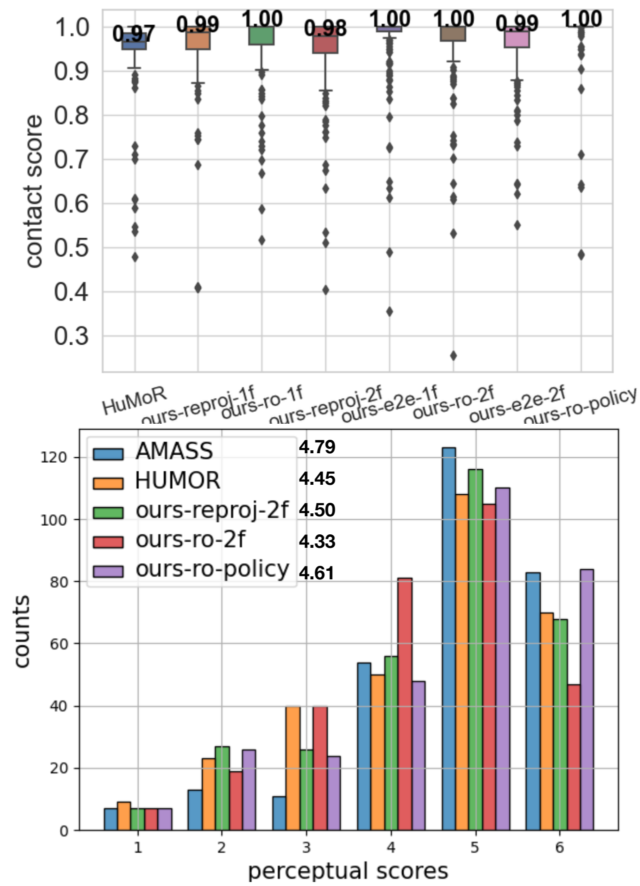

Figure 3. Motion realism analysis. From top to bottom, show the body-ground contact and the perceptual score, respectively. At the top, the box plot denotes the lower and the upper quartiles, and the numbers denote the median. At the bottom, the X and Y-axis denote the perceptual scores and the total counts, respectively. The mean perceptual scores are shown beside the legend.

According to the visualizations, we can see that the HuMoR can produce observable body jitters and other high-frequency non-plausible movements. In contrast, our methods produce smoother motions with more plausible body-ground contact. However, we observe some bodies moving stiffly in our results, particularly in the 1-frame-based models. Since these methods generate motions only based on a single frame, the continuity of higher-order dynamics is not guaranteed. The 2-frame-based models can alleviate this issue, especially when the body motion is slow and low-frequent.

4.2. Evaluation on Motion Control

Datasets and evaluation metrics. For testing, we randomly select 100 character-goal pairs from the simulation area. Like policy training, each body yields 32 motions. The simulation terminates when one body clone reaches the

	steps↓	*success rate*↑	*avg. dist.*↓	*contact score*↑
g-CVAE	24.1	0.91	3.46	0.8
policy(10)	33.9	0.95	3.32	0.94
policy(1)	**21.2**	**1.0**	3.60	0.93
policy(10) & top1	26.5	0.99	**0.04**	**0.97**
policy(10) & top4	23.4	0.96	1.98	0.96

Table 1. Comparison between motion control methods. The up/down arrows denote the score is the higher/lower the better. Best results are in bold, second best in blue.

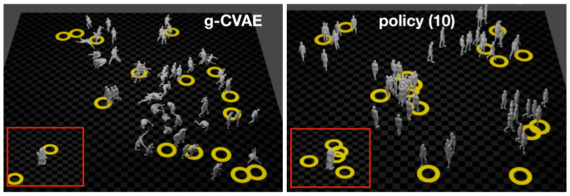

Figure 4. Perceptual comparison between control methods. The subfigures and the main figures denote the starting and the ending frame, respectively. The yellow ring denotes the goal.

goal or after generating 60 primitives.

The evaluation metrics include: 1) *steps*, i.e. the minimum number of motion primitives to achieve the goal. 2) *Success rate*. We regard a test as successful if anybody's clone reaches the goal within 60 motion primitives. 3) *Average distance*. We average the distances of all bodies to the target at the end of each simulation, which also measures how bodies are scattered when the goal is reached. 4) *Contact score*, which is defined in Eq. (11).

Baseline and our methods. A conventional baseline is to incorporate the goal into the generative motion model as an additional condition, like in [25, 65]. For a fair comparison, we add the goal-based features in Eq. (3) to our models and denote this baseline as 'g-CVAE'.

About our methods, we use 'policy (α)' to denote the version with the KLD term weight in Eq. (10). In addition, we use 'top k' to denote selecting the best k primitives at each tree level, according to the contact score and the distance to the goal.

Results and discussions. As shown in Tab. 1, our policy-based methods are superior to 'g-CVAE' w.r.t. all metrics, in particular the contact score. Between our policy-based methods, we find a lower KLD weight can lead to faster but less plausible motions. When exploiting search, the performance consistently improves, and is not sensitive to the value of k. Fig. 4 visualizes their results. We find 'g-CVAE' cannot keep the body valid, probably because the goals for training can only be from the data, but goals for testing are random. On the other hand, 'policy (10)' can largely preserve the motion realism while driving the bodies to the goal. Note that 'top 4' includes sub-optimal motions when calculating the average, so its number is higher than 'top 1'.

Moreover, we visualize how the policy is trained in Fig. 5. The initial policy network (at epoch 0) can produce valid goal-agnostic and random motions. In the early stage, the policy drives the body to the goal as quickly as possible, ignoring motion realism. With more training epochs, the KLD term in Eq. (10) provides more motion regularization. At 500 epochs, the goal-reaching behavior and the motion realism are well balanced. With more training epochs, the KLD term decreases, causing the motions to become more regularized while ignoring the goals.

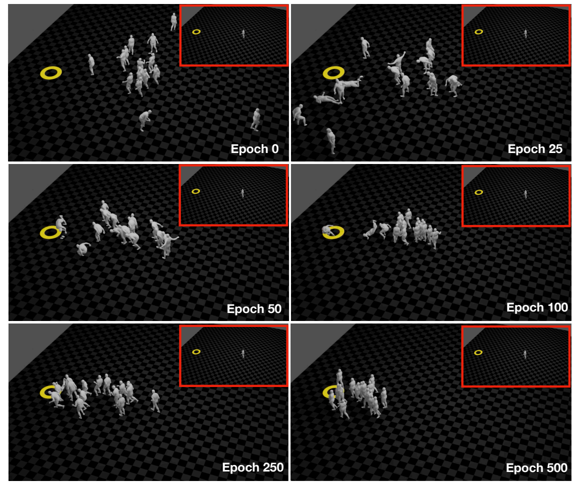

Figure 5. Visualization of policy training process. The notations are same to Fig 4, and the text indicates the training epochs.

5. Conclusion

In this paper, we propose an automatic solution to populate 3D scenes with diverse moving bodies. We learn generative models of body surface marker-based motion primitives and synthesize long-term motion with a policy network and tree-based search. Experiments show the effectiveness and advantages over baselines. Together with conventional path-finding algorithms, we can generate diverse people wandering in the digital environment.

Limitations. The generated motion is not fully physically plausible since our method is purely data-driven. For example, the body can tilt, ignoring gravity. Compared to AMASS sequences, motion realism still has room to improve. Motion generated by the policy may take a long time to reach a goal in the near distance, which is different from real human behavior. In the future, we will extend our work to synthesize more complex body-scene interactions.

Social impact. Although training male and female models separately can improve motion realism, our method could be potentially biased, if the male and female motion sequences are not well balanced in the dataset.

Acknowledgement. We appreciate Gramazio Kohler Research for providing architecture CAD models. Body visualizations are based on the SMPL-X blender add-on. This work is supported by the SNF grant 200021_204840.

References

[1] Experiment: How fast your brain reacts to stimuli. 2

[2] Eyes Japan motion capture dataset. http://mocapdata.com/index.html. 6

[3] CMU graphics lab. CMU graphics lab motion capture database. http://mocap.cs.cmu.edu/, 2000. 6

[4] ACCAD MoCap System and Data. https://accad.osu.edu/research/motion-lab/systemdata, 2018. 1

[5] Kfir Aberman, Peizhuo Li, Dani Lischinski, Olga Sorkine-Hornung, Daniel Cohen-Or, and Baoquan Chen. Skeleton-aware networks for deep motion retargeting. *ACM Transactions on Graphics (TOG)*, 39(4):62–1, 2020. 1

[6] Emre Aksan, Peng Cao, Manuel Kaufmann, and Otmar Hilliges. Attention, please: A spatio-temporal transformer for 3D human motion prediction. *arXiv preprint arXiv:2004.08692*, 2020. 2

[7] Emre Aksan, Manuel Kaufmann, and Otmar Hilliges. Structured prediction helps 3D human motion modelling. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7144–7153, 2019. 2

[8] Sadegh Aliakbarian, Fatemeh Saleh, Lars Petersson, Stephen Gould, and Mathieu Salzmann. Contextually plausible and diverse 3d human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11333–11342, 2021. 2

[9] Emad Barsoum, John Kender, and Zicheng Liu. HP-GAN: Probabilistic 3D human motion prediction via GAN. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, pages 1418–1427, 2018. 2

[10] Kevin Bergamin, Simon Clavet, Daniel Holden, and James Richard Forbes. Drecon: data-driven responsive control of physics-based characters. *ACM Transactions On Graphics (TOG)*, 38(6):1–11, 2019. 3

[11] Apratim Bhattacharyya, Bernt Schiele, and Mario Fritz. Accurate and diverse sampling of sequences based on a "best of many" sample objective. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8485–8493, 2018. 2

[12] Michael Buttner. Machine learning for motion synthesis and character control in games. *Proc. of I3D 2019*, 2019. 2

[13] Michael Büttner and Simon Clavet. Motion matching-the road to next gen animation. *Proc. of Nucl. ai*, 2015, 2015. 1, 2

[14] Yujun Cai, Lin Huang, Yiwei Wang, Tat-Jen Cham, Jianfei Cai, Junsong Yuan, Jun Liu, Xu Yang, Yiheng Zhu, Xiaohui Shen, et al. Learning progressive joint propagation for human motion prediction. In *European Conference on Computer Vision*, 2020. 2

[15] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proceedings of the International Conference on Image Processing*, pages 168–172, 1994. 5

[16] Qiongjie Cui, Huaijiang Sun, and Fei Yang. Learning dynamic relationships for 3D human motion prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6519–6527, 2020. 2

[17] Lingwei Dang, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. Msr-gcn: Multi-scale residual graph convolution networks for human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11467–11476, 2021. 2

[18] Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016. 2

[19] Saeed Ghorbani, Kimia Mahdaviani, Anne Thaler, Konrad Kording, Douglas James Cook, Gunnar Blohm, and Nikolaus F Troje. Movi: A large multi-purpose human motion and video dataset. *Plos one*, 16(6):e0253157, 2021. 6

[20] Partha Ghosh, Jie Song, Emre Aksan, and Otmar Hilliges. Learning human motion models for long-term predictions. In *International Conference on 3D Vision*, pages 458–466. IEEE, 2017. 2

[21] Anand Gopalakrishnan, Ankur Mali, Dan Kifer, Lee Giles, and Alexander G Ororbia. A neural temporal model for human motion prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12116–12125, 2019. 2

[22] Liang-Yan Gui, Yu-Xiong Wang, Xiaodan Liang, and José MF Moura. Adversarial geometry-aware human motion prediction. In *European Conference on Computer Vision*, pages 786–803, 2018. 2

[23] Ikhsanul Habibie, Daniel Holden, Jonathan Schwarz, Joe Yearsley, and Taku Komura. A recurrent variational autoencoder for human motion synthesis. In *28th British Machine Vision Conference*, 2017. 3

[24] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968. 2

[25] Mohamed Hassan, Duygu Ceylan, Ruben Villegas, Jun Saito, Jimei Yang, Yi Zhou, and Michael Black. Stochastic scene-aware motion prediction. In *Proc. International Conference on Computer Vision (ICCV)*, pages 11374–11384, Oct. 2021. 2, 3, 8

[26] Daniel Holden. Character control with neural networks and machine learning. *Proc. of GDC 2018*, 2018. 1, 2

[27] Daniel Holden, Oussama Kanoun, Maksym Perepichka, and Tiberiu Popa. Learned motion matching. *ACM Transactions on Graphics (TOG)*, 39(4):53–1, 2020. 2

[28] Daniel Holden, Taku Komura, and Jun Saito. Phase-functioned neural networks for character control. *ACM Transactions on Graphics (TOG)*, 36(4):1–13, 2017. 3

[29] Daniel Holden, Jun Saito, and Taku Komura. A deep learning framework for character motion synthesis and editing. *ACM Transactions on Graphics (TOG)*, 35(4):1–11, 2016. 3

[30] Daniel Holden, Jun Saito, Taku Komura, and Thomas Joyce. Learning motion manifolds with convolutional autoencoders. In *SIGGRAPH Asia 2015 Technical Briefs*, pages 1–4. 2015. 3

[31] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and

pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7122–7131, 2018. 4

[32] Kacper Kania, Marek Kowalski, and Tomasz Trzciński. Trajevae–controllable human motion generation from trajectories. *arXiv preprint arXiv:2104.00351*, 2021. 3

[33] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Represention*s, 2014. 2

[34] Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion graphs. In *ACM SIGGRAPH 2008 classes*, pages 1–10. 2008. 1, 2

[35] Steven M LaValle et al. Rapidly-exploring random trees: A new tool for path planning. 1998. 6

[36] Steven M LaValle, James J Kuffner, BR Donald, et al. Rapidly-exploring random trees: Progress and prospects. *Algorithmic and computational robotics: new directions*, 5:293–308, 2001. 6

[37] Kyungho Lee, Sehee Min, Sunmin Lee, and Jehee Lee. Learning time-critical responses for interactive character control. *ACM Transactions on Graphics (TOG)*, 40(4):1–11, 2021. 2

[38] Chen Li, Zhen Zhang, Wee Sun Lee, and Gim Hee Lee. Convolutional sequence to sequence model for human dynamics. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5226–5234, 2018. 2

[39] Maosen Li, Siheng Chen, Yangheng Zhao, Ya Zhang, Yanfeng Wang, and Qi Tian. Dynamic multiscale graph neural networks for 3D skeleton based human motion prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 214–223, 2020. 2

[40] Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *ICCV*, 2021. 2

[41] Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel Van De Panne. Character controllers using motion VAEs. *ACM Transactions on Graphics*, 39(4):40–1, 2020. 2, 3, 5

[42] Miao Liu, Dexin Yang, Yan Zhang, Zhaopeng Cui, James M Rehg, and Siyu Tang. 4d human body capture from egocentric video via 3d scene grounding. In *2020 International Conference on 3D Vision (3DV)*. IEEE, 2021. 3

[43] Zhenguang Liu, Pengxiang Su, Shuang Wu, Xuanjing Shen, Haipeng Chen, Yanbin Hao, and Meng Wang. Motion prediction using trajectory cues. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13299–13308, 2021. 2

[44] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pages 5442–5451, Oct. 2019. 2, 3, 6, 1

[45] Christian Mandery, Ömer Terlemez, Martin Do, Nikolaus Vahrenkamp, and Tamim Asfour. Unifying representations and large-scale whole-body motion databases for studying human motion. *IEEE Transactions on Robotics*, 32(4):796–809, 2016. 6

[46] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. Generating smooth pose sequences for diverse human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13309–13318, 2021. 2

[47] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Learning trajectory dependencies for human motion prediction. In *International Conference on Computer Vision*, pages 9489–9497, 2019. 2

[48] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2891–2900, 2017. 2, 5

[49] Josh Merel, Leonard Hasenclever, Alexandre Galashov, Arun Ahuja, Vu Pham, Greg Wayne, Yee Whye Teh, and Nicolas Heess. Neural probabilistic motor primitives for humanoid control. In *International Conference on Learning Representations*, 2019. 3

[50] Meinard Müller, Tido Röder, Michael Clausen, Bernhard Eberhardt, Björn Krüger, and Andreas Weber. Documentation mocap database HDM05. 2007. 6

[51] Ahmed A. A. Osman, Timo Bolkart, and Michael J. Black. STAR: Sparse trained articulated human body regressor. In *European Conference on Computer Vision*, volume LNCS 12355, pages 598–613, Aug. 2020. 3

[52] 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. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10975–10985, June 2019. 2, 3, 6

[53] Dario Pavllo, Christoph Feichtenhofer, Michael Auli, and David Grangier. Modeling human motion with quaternion-based neural networks. *International Journal of Computer Vision*, pages 1–18, 2019. 2

[54] Dario Pavllo, David Grangier, and Michael Auli. Quaternet: A quaternion-based recurrent model for human motion. In *British Machine Vision Conference*, 2018. 3

[55] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Trans. Graph.*, 37(4):143:1–143:14, July 2018. 3

[56] Xue Bin Peng, Angjoo Kanazawa, Jitendra Malik, Pieter Abbeel, and Sergey Levine. Sfv: Reinforcement learning of physical skills from videos. *ACM Transactions On Graphics (TOG)*, 37(6):1–14, 2018. 3

[57] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *arXiv preprint arXiv:2104.02180*, 2021. 3

[58] Mathis Petrovich, Michael J. Black, and Gül Varol. Action-conditioned 3D human motion synthesis with transformer VAE. In *International Conference on Computer Vision (ICCV)*, pages 10985–10995, October 2021. 2, 3

[59] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J. Guibas. Humor: 3d human motion model for robust pose estimation. In *International Conference on Computer Vision (ICCV)*, 2021. 3, 5, 7, 2

[60] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied Hands: Modeling and capturing hands and bod-

ies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), Nov. 2017. 3

[61] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 6

[62] Leonid Sigal, Alexandru O Balan, and Michael J Black. HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, 87(1-2):4, 2010. 1

[63] Greg Snook. Simplified 3d movement and pathfinding using navigation meshes. *Game programming gems*, 1(1):288–304, 2000. 2

[64] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28:3483–3491, 2015. 2

[65] Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. Neural state machine for character-scene interactions. *ACM Trans. Graph.*, 38(6):209–1, 2019. 3, 8

[66] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998. 6

[67] Yongyi Tang, Lin Ma, Wei Liu, and Wei-Shi Zheng. Long-term human motion prediction by modeling motion context and enhancing motion dynamic. In *International Joint Conference on Artificial Intelligence*, page 935–941, 2018. 2

[68] Warren H Teichner. Recent studies of simple reaction time. *Psychological Bulletin*, 51(2):128, 1954. 2

[69] Matthew Trumble, Andrew Gilbert, Charles Malleson, Adrian Hilton, and John P Collomosse. Total capture: 3d human pose estimation fusing video and inertial sensors. In *BMVC*, volume 2, pages 1–13, 2017. 6

[70] Jacob Walker, Kenneth Marino, Abhinav Gupta, and Martial Hebert. The pose knows: Video forecasting by generating pose futures. In *International Conference on Computer Vision*, pages 3332–3341, 2017. 2

[71] Jiashun Wang, Huazhe Xu, Jingwei Xu, Sifei Liu, and Xiaolong Wang. Synthesizing long-term 3d human motion and interaction in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9401–9411, 2021. 3

[72] Zhiyong Wang, Jinxiang Chai, and Shihong Xia. Combining recurrent neural networks and adversarial training for human motion synthesis and control. *IEEE transactions on visualization and computer graphics*, 27(1):14–28, 2019. 3

[73] Mao Wei, Liu Miaomiao, and Salzemann Mathieu. History repeats itself: Human motion prediction via motion attention. In *European Conference on Computer Vision*, 2020. 2

[74] Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. GHUM & GHUML: Generative 3D human shape and articulated pose models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6184–6193, 2020. 3

[75] Xinchen Yan, Akash Rastogi, Ruben Villegas, Kalyan Sunkavalli, Eli Shechtman, Sunil Hadap, Ersin Yumer, and Honglak Lee. MT-VAE: Learning motion transformations to generate multimodal human dynamics. In *European Conference on Computer Vision*, pages 265–281, 2018. 2

[76] Ye Yuan and Kris Kitani. DLow: Diversifying latent flows for diverse human motion prediction. *European Conference on Computer Vision*, 2020. 2, 4, 1

[77] Ye Yuan, Shih-En Wei, Tomas Simon, Kris Kitani, and Jason Saragih. Simpoe: Simulated character control for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7159–7169, 2021. 3

[78] Siwei Zhang, Yan Zhang, Federica Bogo, Marc Pollefeys, and Siyu Tang. Learning motion priors for 4d human body capture in 3d scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11343–11353, 2021. 3

[79] Yan Zhang, Michael J Black, and Siyu Tang. Perpetual motion: Generating unbounded human motion. *arXiv preprint arXiv:2007.13886*, 2020. 3

[80] Yan Zhang, Michael J Black, and Siyu Tang. We are more than our joints: Predicting how 3d bodies move. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3372–3382, 2021. 2, 3, 4, 5, 7, 1

[81] Yan Zhang, Mohamed Hassan, Heiko Neumann, Michael J Black, and Siyu Tang. Generating 3d people in scenes without people. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6194–6204, 2020. 7

[82] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019. 4

[83] Yi Zhou, Zimo Li, Shuangjiu Xiao, Chong He, Zeng Huang, and Hao Li. Auto-conditioned recurrent networks for extended complex human motion synthesis. In *International Conference on Learning Representions*, 2018. 2

[84] Fabio Zinno. Ml tutorial day: From motion matching to motion synthesis, and all the hurdles in between. *Proc. of GDC 2019*, 2019. 2