DOI: 10.1111/cgf.14636
ACM SIGGRAPH / Eurographics Symposium on Computer Animation 2022
D. L. Michels and S. Pirk
(Guest Editors)

Volume 41 (*2022*), *Number 8*

Synthesizing Get-Up Motions for Physics-based Characters

A. Frezzato[1] and A. Tangri[2] and S. Andrews[1]†

[1]École de technologie supérieure, Canada
[2]Manipal Institute of Technology, India

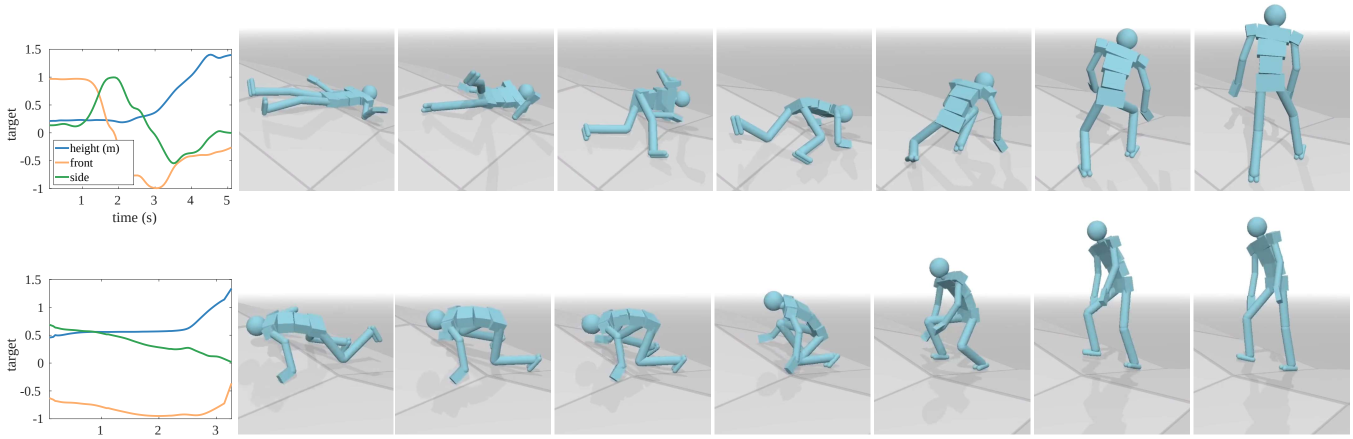

Figure 1: *Get-up motions synthesized using our DRL framework. Style is achieved by curves specifying the desired vertical height and torso orientation of the character. Distinct motion styles are possible, such as rolling on the right (top) and pushing straight up (bottom).*

Abstract

We propose a method for synthesizing get-up motions for physics-based humanoid characters. Beginning from a supine or prone state, our objective is not to imitate individual motion clips, but to produce motions that match input curves describing the style of get-up motion. Our framework uses deep reinforcement learning to learn control policies for the physics-based character. A latent embedding of natural human poses is computed from a motion capture database, and the embedding is furthermore conditioned on the input features. We demonstrate that our approach can synthesize motions that follow the style of user authored curves, as well as curves extracted from reference motions. In the latter case, motions of the physics-based character resemble the original motion clips. New motions can be synthesized easily by changing only a small number of controllable parameters. We also demonstrate the success of our controllers on rough and inclined terrain.

CCS Concepts

• ***Computing methodologies*** → *Physical simulation;* • ***Theory of computation*** → *Reinforcement learning;*

Keywords: physics-based character animation, deep reinforcement learning, get-up motions, standing, motion control

1. Introduction

Physics-based characters in recent years have demonstrated impressive capabilities to perform human tasks ranging from walking and running [MTT*17, PALvdP18, BCHF19], climbing [NBRH19], throwing [CL18, LXAK21], and even boxing [WGH21]. Typically, the character must perform these tasks while avoiding falling, and in the case that it does, a *get-up* motion is synthesized to return the character to a state where it can continue performing the task.

However, few approaches have focused specifically on the task of getting up. Get-up motions are complex and contact rich, often requiring many contacts between the humanoid and

† sheldon.andrews@etsmtl.ca

the environment. This makes synthesizing get-up motions from arbitrary prone or supine postures challenging since the valid configuration space is constrained by the surrounding environment and self-collisions, and dynamic control is further constrained by limitations of the physics-based character model. Therefore, current motion synthesis techniques often lack the variety, complexity, and naturalness that is exhibited by real humans get-up motions.

Many existing methods rely on motion capture data, and thus are only capable of producing get-up motions that imitate clips found in a motion database, whereas approaches that do not rely on reference motion often produce highly energetic motions and appear unnatural. Additionally, nearly all the work in this area demonstrates getting up from a flat terrain, and obviously humans can successfully get-up and stand in more complex environments. Finally, there are a variety of ways in which humans can go from lying down to standing up, and to our knowledge no previous work has explored authoring the style of get-up motions without having to explicitly pose or animate the character model, which can be tedious.

Motivated by these challenges, we focus on the problem of synthesizing realistic get-up motions for physics-based characters. Our approach is based on a deep reinforcement learning (DRL) framework, and provides a simple user interface to change the style of the motion in which curves determine the speed and form of the synthesized motions. The resulting animations exhibit more variety compared to imitation learning and reference motion tracking, yet with a controllable stylization. Furthermore, we demonstrate that our technique is suitable for a variety of rough and sloped terrains. A curriculum learning approach gradually increases the slope and roughness of the terrain to learn control policies that can successfully get-up for many different environments.

2. Related Work

The surveys by [GP12] and [MHLC*21] provide an overview of approaches that have been developed in recent years for controlling physics-based based characters. Deep learning (DL) neural network models have demonstrated an excellent ability for synthesizing complex and agile motions, and thus in this section we mainly focus on summarizing DRL and related machine learning techniques for animating physics-based characters.

Learning from motion. Learning human skills from a collection of animation clips is a popular approach to generate natural looking motions for physics-based characters [PALvdP18, LPY16, MTT*17]. Chentanez et al. [CMM*18] addressed fall recovery using a special policy to resume locomotion in case of failure. Complex tasks may be achieved by learning composite control policies that combine skills [PALvdP18], or by organizing low-level controllers into graph structure [LPY16]. Bergamin et al. [BCHF19] proposed a framework to learn responsive locomotion controllers based on exemplars produce by a motion matching technique. Other approaches leverage large databases of unorganized motion clips to learn a motion generator the produces a sequence of poses tacked by the controller [PRL*19]. These approaches produce natural looking and agile get-up motions, but without explicit style control. There has recently been interest on methods that handle challenging continuous control tasks using low-level controllers trained on task-agnostic motions [WGH20, WGH21, PCZ*19]. Peng et al. [PGH*22] recently addressed the problem of robust recovery from a fallen state by learning fast and agile motions that allow the agent to quickly recover from large perturbations. However, the character often exhibits superhuman-like abilities during recovery. The authors mention that having more get-up motions in their database would help to create natural recovery motions. Our approach offers more fine-grained control over the style and speed of get-up motions.

Latent motion models. Latent variable models learned from motion capture data have been shown to perform well for synthesizing natural and stylized motion. Yuan et al. [YK20] used a variational autoencoder (VAE) to robustly imitate motion capture trajectories using dynamical models. A related technique called generative adversarial imitation learning (GAIL) was employed by Wang et al. [WMR*17] to learn robust motion controllers from a small number of motion clips that also demonstrated diversity in the synthesized behaviors and an ability to transition between locomotion styles. More recently, Peng et al. [PMA*21] proposed a technique based on generative adversarial learning to synthesize natural motions from a large database of motions that produces compelling styles for a variety of different tasks. Other approaches encode the action space of an RL controller using a latent subspace representation [LZCVDP20, MHG*19, PCZ*19, MTA*20]. In our work, a pose VAE similar to the one used by Yin et al. [YYVDPY21] is trained and used to evaluate the naturalness of motions produced by the control policy. A discussion about this choice can be found in Section 3.5.

Trajectory optimization and other learning methods. One of the objectives of our work is that the character is able to get-up on non-flat terrain. Ling et al. [XLKvdP20] proposed a curriculum learning approach for locomotion on non-flat terrain. Curriculum learning methods have also succeeded for learning complex tasks by increasing the difficulty over the training [YYVDPY21, LLLL21]. We are inspired by this type of approach and use curriculum learning to gradually train the agent to get-up on increasingly complex terrains. Heess et al. [HTS*17] trained physics-based humanoids to perform locomotion tasks without reference motion. Also, trajectory optimization approaches have been successful animating characters getting up in complex environments [MTP12, TET12, ABdLH13]. Lin and Huang [LH12] specifically addressed the problem of authoring get-up motions. Their approach uses a rapidly-exploring random tree (RRT) and a physics-based filter to generate intermediary poses from a motion capture database that animate between a lying down pose and key frame poses. Our approach requires only the initial state and the desired style of motion that is encoded in the input curves. The entire motion that we produce is physically plausible and generated within a natural pose space. Early work by Faloutsos et al. [FvdPT01] proposed a framework that combines several controllers using a state machine that is capable of returning to a balanced state. In our work, a single policy uses a latent space of poses to execute a get-up motion from a fallen state.

Robotics. Standing controllers have long been of interest for the robotics community as well. Morimoto and Doya [MD98] proposed a hierarchical RL framework for generating standing motions for simple humanoids. Kanehiro et al. [KFH*07] generated get-up motions by interpolating between the current pose of the robot and its closest pose in a predefined state graph. Other work by Fujiwara et al. [FKK*03] has similarly divided get-up motions into phases based on a contact configuration graph.

Human motion. Standing motions have been well studied in the biomechanical and kinematics literature. In particular, getting up from the floors has long been studied as a measure of the physical performance of elderly persons [AT00, KAS*16]. To our knowledge, no taxonomy exists to describe and classify the many different styles of get-up motions. However, Bohannon and Lusardi [BL04] identified typical strategies used by humans during supine-to-standing tasks: side-sit to half-kneel pivot, quadruped push-up, and sit-up and roll-over. In the results section, we demonstrate that our framework is capable of generating motions for each of these strategies. Recently, Tao et al. [TWGVdP22] proposed a framework for synthesizing get-up motions without using a database of reference clips. Natural motions are obtained by using a curriculum-based approach that modulates the torque of the character to learn successful controllers, and new behaviors are created by tracking slower versions of motion generated by fast policies. In contrast, we propose another approach by training a single policy that provides an extra user control for the style of the motion. Furthermore, our controllers are robust to terrain variations.

3. Methodology

An overview of our DRL framework is shown in Figure 2. A single control policy is used for both the get-up and standing phases of the motion, although we consider these phases separately in how the agent is trained. Further details about training are presented later in Section 5.

Control over the speed and style of get-up motions is achieved by curves that specify the desired height and torso orientation of the character during the course of the motion. These curves may be extracted from real motion data or authored (e.g., using a spline). Additionally, a C-VAE is trained using a motion capture database that includes getting up motions, as well as many other motions, and provides a latent space for synthesizing natural poses.

3.1. Character Model

The character model used in our experiments is shown in Figure 4. It is comprised of $n = 19$ joints with $m = 49$ controllable degrees of freedom (DOF). Boxes, capsules, and spheres are used to approximate the body shape. The height and shape follow an average human morphology with a mass of 65 kg and height of 1.7 m [PEA83]. Each joint is controlled by a proportional derivative (PD) servo. Torque limits, stiffness and damping values are the same as in DeepMimic [PALvdP18]. In addition, joint angle limits reflect realistic human kinematics to ensure naturalness of the motions. The front direction is computed using the cross product of the vector connecting the neck and hips and the side vector. The side vector is oriented with the hips and directed to the character's right side. The height tracking position is located at the base of the neck.

3.2. Environment

Several environments are used for training the agent that range from flat terrain, rough terrains, and sloped terrain. Rough terrain variations are created by perturbing the height of 1.0 m × 1.0 m tiles of a flat terrain. Examples are shown in Figure 3. A Coulomb coefficient of friction $\mu = 1.0$ is used for all character-terrain interactions.

The agent is provided with a height map of the surrounding terrain similar to the one used by DeepMimic [PALvdP18]. However, no convolution network is used to process the height map, and instead a simpler fully connected network architecture is used. The height map is a 1.0 m × 1.0 m grid with 25 cm accuracy, which is computed using the vertical distance between the hips and the ground. We found that this resolution is sufficient for the agent to learn good end effector placement, but without severely impacting simulation performance (i.e., the state vector size increases quadratically with the height map resolution, potential performance bottlenecks due to many distance queries against the terrain geometry).

3.3. Representation

The environment state $\mathbf{s} \in \mathbb{R}^{352}$ combines the character and task states:

$$\mathbf{s}_t = \begin{bmatrix} \mathbf{q}_t & \mathbf{p}_t & \omega_t & \mathbf{v}_t & \mathbf{w}_t & \mathbf{h}_t & \mathbf{t}_t & \mathbf{y}_{t-1} \end{bmatrix}$$

The subscript t refers to values from the current simulation step, and $t-1$ refers to values from the previous step. The components of the state vector are described in detail in the subsequent sections.

3.3.1. Character State

The pose of the character is given by $\mathbf{q} \in \mathbb{R}^{76}$, which contains the rotations of each link as quaternions. The character state also contains the position of each link relative to the root, $\mathbf{p} \in \mathbb{R}^{57}$, and the angular and linear velocities of each link, $\omega \in \mathbb{R}^{57}$ and $\mathbf{v} \in \mathbb{R}^{57}$ respectively. The global orientation and translation are not stored since our tasks are primarily concerned with standing up (i.e., global translation is unnecessary, and the orientation is sufficiently encoded by the torso angle).

The state also contains kinematic information related to the features determining the style of the get-up motion. Specifically, the vector $\mathbf{w} = \begin{bmatrix} \mathbf{w}_{\mathrm{h}} & \mathbf{w}_{\theta_x} & \mathbf{w}_{\theta_y} \end{bmatrix} \in \mathbb{R}^3$ contains the height of the torso, $\mathbf{w}_{\mathrm{h}}$, the angle between the torso side facing direction and the global up direction, $\mathbf{w}_{\theta_x}$, and similarly the angle between the torso front direction and global up direction, $\mathbf{w}_{\theta_y}$.

The height map $\mathbf{h} \in \mathbb{R}^{25}$ is an 5×5 grid uniformly sampled around the character that encodes the vertical distance of the hips to the ground, as described in Section 3.2. The height map is aligned with the local character frame. Finally, the previous filtered policy action $\mathbf{y}_{t-1} \in \mathbb{R}^{65}$ is also part of the character state.

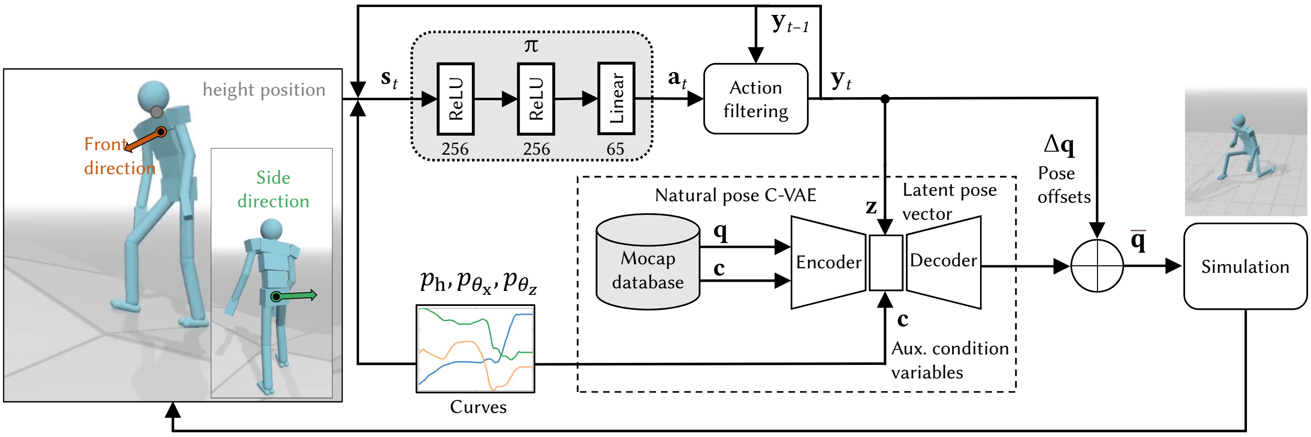

Figure 2: *An overview of our framework. The policy π produces a get-up motion that tracks curves defining the height and torso orientation trajectories. A pose C-VAE, trained using a motion capture database, helps to ensure that target poses for PD servos appear natural. The pose distribution is conditioned by the vector* **c** *that contain the three curves features. Offsets are applied to the natural pose to discover new styles, but also to react properly to the physics and adapt to the terrain variation. Temporal filtering of actions helps during the training to find better terrain adaptation and smoother exploration.*

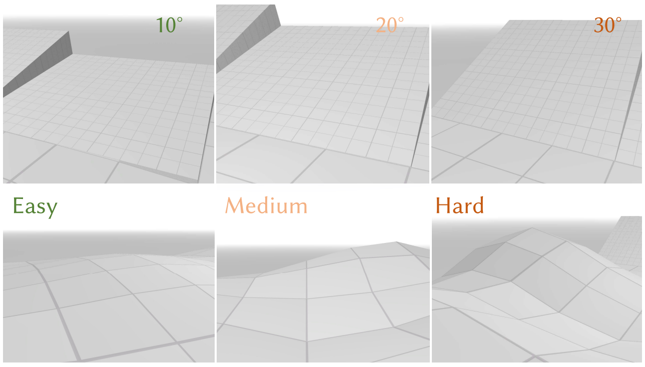

Figure 3: *The training environment contains terrain with four regions of difficulty, including three sloped terrains (top row) and easy, medium, and hard rough terrains (bottom row). A curriculum learning approach begins training on a flat region and more difficult regions are unlocked as training progresses.*

3.3.2. Task State

Vector $\mathbf{t} = \begin{bmatrix}\mathbf{t}_\mathrm{h} & \mathbf{t}_{\theta_x} & \mathbf{t}_{\theta_z}\end{bmatrix} \in \mathbb{R}^{12}$ contains the target height, side facing angle, and front facing angles, respectively, which are specified at 0 ms, 100 ms, 200 ms, 300 ms in the future. These target values give the agent important indications about the desired style of get-up motion. The values in **t** are sampled from three curve functions– p_h, p_{θ_x}, p_{θ_z}– that define the target height, side torso orientation, and front torso orientation, respectively. Curves may be manually authored, or extracted from a reference motion clip. Further details about the feature curves used for controlling the style of get-up motion are provided in Section 4.

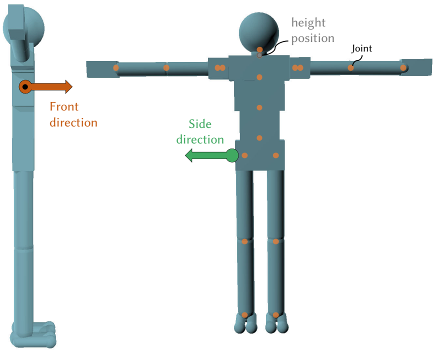

Figure 4: *Our character model, showing the height tracking position and the torso side and front facing vectors.*

3.4. Actions

Actions produced by the policy contain a latent parameter vector $\mathbf{z}' \in \mathbb{R}^{16}$ and joint angle offsets $\Delta\mathbf{q}' \in \mathbb{R}^{49}$, such that

$$\mathbf{a} = \begin{bmatrix}\mathbf{z}' & \Delta\mathbf{q}'\end{bmatrix} .$$

Action filtering [BCHF19] is further used to produce smoothly changing target angles for the PD servos, and we specifically observed that it helped to improve end effector placement on challenging terrains. Filtered actions are computed by

$$\mathbf{y}_t = \beta\mathbf{a}_t + (1-\beta)\,\mathbf{y}_{t-1} ,$$

where $\mathbf{a}_t$ is the action from the policy at time t, $\mathbf{y}_{t-1}$ is the filtered action from the previous time step, and $\beta = 0.2$ is the smoothing coefficient. The latent pose parameters **z** and offsets $\Delta\mathbf{q}$ are extracted from the filtered action, such that

$$\mathbf{y}_t = \begin{bmatrix}\mathbf{z} & \Delta\mathbf{q}\end{bmatrix}$$

The final joint angle targets for PD servos are computed as

$$\bar{\mathbf{q}} = \textsc{Decoder}\left(\mathbf{z}, \mathbf{c}\right) + \Delta\mathbf{q},$$

where the decoder function maps a latent pose using $\mathbf{z}$ and $\mathbf{c}$ to full character pose by a process that is explained in Section 3.5. However, since a conditional VAE is trained and used to decode the latent pose vector, in addition to $\mathbf{z}$, the auxiliary variables

$$\mathbf{c} = \begin{bmatrix} p_{\mathrm{h}} & p_{\theta_x} & p_{\theta_z} \end{bmatrix}$$

are also required to compute the full character pose. The vector $\mathbf{c} \in \mathbb{R}^3$ contains the target height and torso orientations sampled at current time t.

3.5. Natural Poses

Humans motion is synergistic, with joints moving in coordination to accomplish tasks. We therefore exploit this characteristic by learning a natural pose manifold for motions produced by our character controllers. We build on the β-VAE proposed by Yin et al. [YYVDPY21] for generating natural poses. However, rather than training an auto-regressive model on a motion capture dataset, we condition the model on the features tracked by our controller. Specifically, a conditional VAE (C-VAE) is used to regress poses that are also conditioned on the target height and torso angles, which are computed for each pose in the mocap dataset.

We found that conditioning the motion prior in this way helps the policy to generate poses that are more appropriate for the target height and torso angles. This results in faster learning and reduced training times compared to a standard pose VAE.

The natural pose C-VAE is shown in Figure 2. Only the decoder is used at run-time to compute full character poses from filtered latent parameters $\mathbf{z}$, which are generated by the policy, and the auxiliary variables, $\mathbf{c}$, which contains the target height and torso angles.

The LAFAN1 dataset [HYNP20] is used to train the C-VAE. Specifically, we use only motions sequences from the *Fall and get up* theme, which contains approximately 20 min / 36,000 frames of human motion. Our analysis indicates that about 30% of the frames are from get-up motions.

The C-VAE is trained with $\beta = 1 \times 10^{-6}$ and learning rate of 1×10^{-4}. The encoder and decoder are modeled as fully-connected neural networks with two layers of 256 units and *tanh* activation. For training the C-VAE, the pose height and angles are concatenated with the input pose for the encoder, and with the input latent vector for the decoder. Because the mo-cap data contain only joint angles, we compute the angles based on the character stance, this process is done once before training the C-VAE. The model is trained for 100 epochs and a batch size of 128 is used. The β-VAE loss is optimized using Adam. We performed a preliminary principal component analysis (PCA) on the training data to select the dimension of the latent vector and found that 16 components was sufficient to cover 85% of the variance across all poses in the LAFAN1 dataset, which is slightly higher than the 13 used by Yin et al.

4. Controlling the Get-up Style

Our approach for synthesizing various styles of get-up motions does not require imitating a full motion trajectory. Rather, a simple interface allows authoring three curves that define a few desired characteristics of the motion. One curve defines the desired vertical height of the character, and two other curves give the desired torso orientation.

4.1. Spline Representation

A cardinal cubic B-spline is used to represent the trajectories of the three features tracked by our get-up controllers. Through experimentation, we found that nine control points are sufficient to synthesize a variety of get-up motions. The curves may be directly extracted from real motion clips (e.g., by evaluating the features for each animation and then sampling them at nine equally distributed instances in time). Alternatively, curves may be authored using an editing interface.

We define a feature curve $p(s)$ as a general function that returns the target trajectory based on the parameter s. The curve parameter lies in the interval $s \in [0, 1]$, and for motions with an arbitrary time interval, the normalized parameter value is computed as:

$$s = \min\left(\frac{t - t_0}{T}, 1\right) \tag{1}$$

Here, t_0 is the start time of the motion, T is the duration of the get-up motion, and t is the current simulation time.

4.2. Motion Features

Recall that the height of the character is measured as the vertical distance from the base of the neck to the average terrain height computed from the height map, and the torso orientations are computed as the angle between local directions of the upper and lower torso segments and the global vertical direction. These features of the character's posture are shown in Figure 4.

The location of the height position and torso vectors were determined by a trial-and-error process, with the goal being to simplify the authoring process while still providing sufficient control to create various natural-looking get-up motions. In preliminary experiments, we tested tracking the center of mass (COM) height. However, the character often cheats by raising its arms and legs in order to raise the COM. Also, the torso vectors are situated at two different locations: the lower and upper torso. This allows for some twisting of the torso during roll-over motions, and makes it easier to author curves for which the character can successfully go from lying down to standing.

4.3. Curve Editing

A simple interface for visualizing the get-up motions produced curve editing is shown in Figure 5. The editing tool may be used to make adjustments to an original reference motion, or alternatively to craft a get-up motion from scratch. Also, the duration of the motion can be adjusted simply by changing the duration T.

In the supplementary videos, we show an example of using this editing tool in which the control points of each feature curve are modified using sliders.

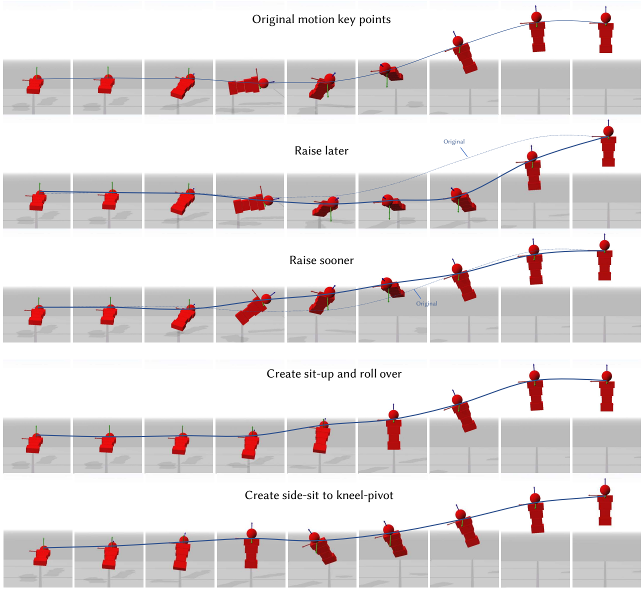
Original motion key points
Raise later
Original
Raise sooner
Original
Create sit-up and roll over
Create side-sit to kneel-pivot

Figure 5: *Adjustments are made to a reference motion curves (top). For instance, by having a delayed (second row) or earlier (third row) rise time. Curve editing may also be used to synthesize a different style of motion, such removing the turnover and converting the motion to a sit-up and roll over (fourth row) and create a side-sit to kneel-pivot (bottom row).*

5. Training

In this section, we present details related to training the get-up control policy for mapping the environment state $\mathbf{s}$ to an action $\mathbf{a}$ at each simulation step. The control policy is a distribution over actions $\pi(\mathbf{a}|\mathbf{s})$ for an arbitrary state $\mathbf{s}$. The distribution is modeled as a Gaussian distribution with state-dependent mean $\mu(\mathbf{s})$ and fixed covariance Σ:

$$\pi(\mathbf{a}|\mathbf{s}) = \mathcal{N}(\mu(s), \Sigma)$$

A deep neural network is used to implement π, where network inputs are processed by two fully-connected hidden layers, each with 256 units and a ReLU activation function. A final output layer with linear activation function produces the actions, as show in Figure 2.

Policy training is performed using reinforcement learning, where an agent learns from experience to maximize a reward over time. The policy generates an action $\mathbf{a}_t$ based on the current state $\mathbf{s}_t$, which causes the environment to transition to a new state $\mathbf{s}_{t+1}$. A reward r_t is used to convey if the action resulted in a good transition, or a bad one. Proximal policy optimization (PPO) [SWD*17] is used to update the policy. Specifically, the PPO-Clip algorithm with generalized advantage estimator GAE(λ). The general idea of this algorithm is to keep the new policy close to the old policy after an update using a clipping parameter ε. We refer the reader to the original paper for more details.

5.1. Rewards

Rewards for training the policy have a multiplicative form:

$$r_t = r_t^{\text{task}} r_t^{\text{natural}} \tag{2}$$

The term r_t^{task} is a task specific reward, and r_t^{natural} is a naturalness reward following Yin et al. [YYVDPY21].

5.1.1. Naturalness Reward

The natural reward is computed as:

$$r_t^{\text{natural}} = 1 - \text{CLIP}\left(\left(\frac{\|\Delta\mathbf{q}\|_1}{c}\right)^2, 0, 1\right) \tag{3}$$

Intuitively, Equation 3 penalizes large joint angle offsets $\Delta\mathbf{q}$ that generate target poses away from the natural pose manifold. During training, the value of c can be adjusted to control the amount of exploration at different stages. However, we found that a constant value of $c = 10$ was sufficient.

5.1.2. Task Reward

The get-up task reward also has a multiplicative form:

$$r_t^{\text{task}} = r_t^{\text{h}} r_t^{\theta} r_t^{\text{v}} r_t^{\text{p}} \tag{4}$$

Briefly, the term r_t^{h} encourages the agent to follow the height curve, r_t^{θ} encourages tracking the torso curves, r_t^{v} reduce the joint velocities toward the end of the motion, and r_t^{p} encourages the agent to track a neutral standing pose. Since the torso orientation and the height are dependent, a multiplicative reward helps the policy to produce motions where both the height and torso orientation reward are maximized. The height reward is computed as:

$$r_t^{\text{h}} = e^{-1.7\,|h_t - p_{\text{h}}(s)|} \tag{5}$$

The target height $p_{\text{h}}(s)$, and h_t is the current height of the character. A similar reward is used to track the torso orientation:

$$r_t^{\theta} = e^{-1.5\left(|\theta_{x,t} - p_{\theta_x}(s)| + |\theta_{z,t} - p_{\theta_z}(s)|\right)} \tag{6}$$

Target angles for the side and front torso vectors, $p_{\theta_x}(s)$ and $p_{\theta_z}(s)$ respectively, and $\theta_{x,t}$ and $\theta_{z,t}$ are the current torso angles.

In order to help the agent find a final stable standing pose, a velocity reduction reward penalizes angular motion of character links toward the end of the get-up motion:

$$r_t^{\text{v}} = (1 - \alpha_t) + \alpha_t e^{-0.05\|\omega_t\|} \tag{7}$$

The phase variable α_t transitions smoothly from $0 \rightarrow 1$ as the character stands up. The phase variable is computed as

$$\alpha_t = e^{-20\,(1-\phi)}\ , \tag{8}$$

where $\phi = p_{\text{h}}(s)/p_{\text{h}}(s=1)$ is the ratio between the current target height $p_{\text{h}}(s)$ and the final target height $p_{\text{h}}(s=1)$. The velocity reduction reward in Equation 7 allows the agent to move around unhindered during early stages of the get-up motion, but quickly begins to penalize fast motions as they transition to a standing pose.

Finally, the agent is encouraged to assume a neutral posture $\hat{\mathbf{q}}$ once they are standing. The reward

$$r_t^{\text{p}} = (1 - \alpha_t) + \alpha_t\, e^{-0.1\left(\sum_j |\log(\hat{q}_t^j, q_t^j)|\right)}\ , \tag{9}$$

where $\hat{q}^j$ and q_t^j are the target relative orientation and current relative orientation of the jth joint, respectively, and their difference is computed by a log map.

5.2. Initialization

The character is initialized at the start of each training episode either (i) lying on the floor or (ii) using a standing pose selected from one of the reference motion clips. In the case of the latter, reference poses are only used when training on flat terrain, since this matches to the original capture environment. We found that having several standing poses in the set of initial states is necessary for learning successful get-up motions since the agent explores states near the goal early in training. Whereas for irregular terrain, initial states are created by dropping a passive rag-doll model with a prone or supine posture onto the terrain at various locations. However, with this method for state initialization, we observed that sometimes the initial orientation of the torso was significantly different from the angles in the curves. This lead to problems with training, and so curves are filtered to have angles less than 30° from the initial orientation of the torso. In the case of initializing from a standing pose on irregular terrain, the feet of the character are re-oriented to match the local terrain slope.

5.3. Early Termination

The agent is expected to closely follow the height and torso orientation curves. Hence, a training episode is terminated when the character begins to deviate from the target trajectory. There are two conditions we consider: (i) the task reward is lower than 0.1, or (ii) the character's head touches the floor. If either of these conditions is sustained for more than 300 ms, the episode is terminated.

5.4. Terrain Curriculum

The character will struggle to make progress in early stages of training with rough or steeply sloped terrains. Therefore, a terrain curriculum learning approach gradually increases the difficulty of the terrain as training progresses. Effectively, the same terrain is used for all experiments, but it is divided into four regions (flat, easy, medium, and hard difficulties as shown in Figure 3. The agent is first trained on the flat region and must succeed in getting-up 200 times to unlock the next region. The agent revisits all unlocked regions during training in order to avoid catastrophic forgetting.

5.5. Training Curves

A collection of curves is used to train the agent for a diversity of get-up tasks. These curves are extracted from reference motion clips, as well as manually authored curves. Specifically, 13 curves are from reference motion clips, and 5 are authored curves designed to imitate various get-up styles. The agent is additionally presented each of the authored curves for three different durations (2.5 s, 5.1 s, 7.6 s). In total, 28 different curves are used to train the agent.

The initial posture of each reference motion is used to initialize the character model when the corresponding curve is being trained. However, for training on irregular terrains, curves are randomly selected from the entire collection and used to train the agent, given that the initial torso orientation condition explained in Section 5.2 is satisfied.

5.6. Implementation Details

Our framework is implemented using PyTorch and the PPO-Clip implementation is taken from Stable-Baselines3 [RHG*21]. We model the physics-based character and the environment using the Vortex dynamics engine [CM 19]. The simulation runs at 60 Hz and control policies are queried at 30 Hz. All experiments are performed on a Windows PC with NVIDIA GeForce GTX 1080 Ti GPU and 8 core Intel i9 CPU. No GPU is used to update the networks. Each policy is trained parallel using 16 processes.

Training requires 100M simulation steps and approximately 30 hours to complete. The learning rate decreases from 1×10^{-4} to 2×10^{-5} and standard deviation decreases from 0.6 to 0.3 uniformly every 20M steps. A mini-batch size of 256 is used. Other learning hyper-parameters are: a generalized advantage estimate 0.95, discount factor $\gamma = 0.99$, and clipping parameter $\varepsilon = 0.2$.

During training, the episode time includes both the get-up duration T plus a few second to train the standing phase. Thus, the total episode time is $T_{total} = T + T_{\text{stand}}$, where $T_{\text{stand}} = 3$ s is the additional time for standing.

6. Results

Here, we present some of the animations created using our framework. We first present examples of synthesizing get-up motion from curves extracted from reference motion clips, and show the robustness of our framework to synthesize the same motions on sloped and inclined terrains. We then demonstrate the authoring capabilities of our framework, and present some examples of manually authored get-up motions. Lastly, we perform an ablation study on the effectiveness of using a latent pose motion prior and the terrain curriculum during training.

6.1. Reference Motions

Figure 6 shows two different motions produced by feature curves extracted from clips in the LAFAN1 dataset. Additional get-up motions based on curves from the dataset can be found in the supplementary video. The character controllers are able to faithfully reproduce the style of the original motion in all the motions, and without explicitly tracking the pose.

The teaser and Figure 6 show examples of get-up motions taken from reference motions and retargeted to rough and sloped terrain. Despite the original motion being performed on flat terrain, the synthesized animations demonstrate that our get-up controller successfully tracks the height curve, and does reasonably well tracking the desired torso orientations. The character also effectively adapts its stance to the shape of the terrain at the end of the motion, and is able to achieve a stable stance despite the terrain slope. Additional motions on irregular terrain can be found in the supplementary video.

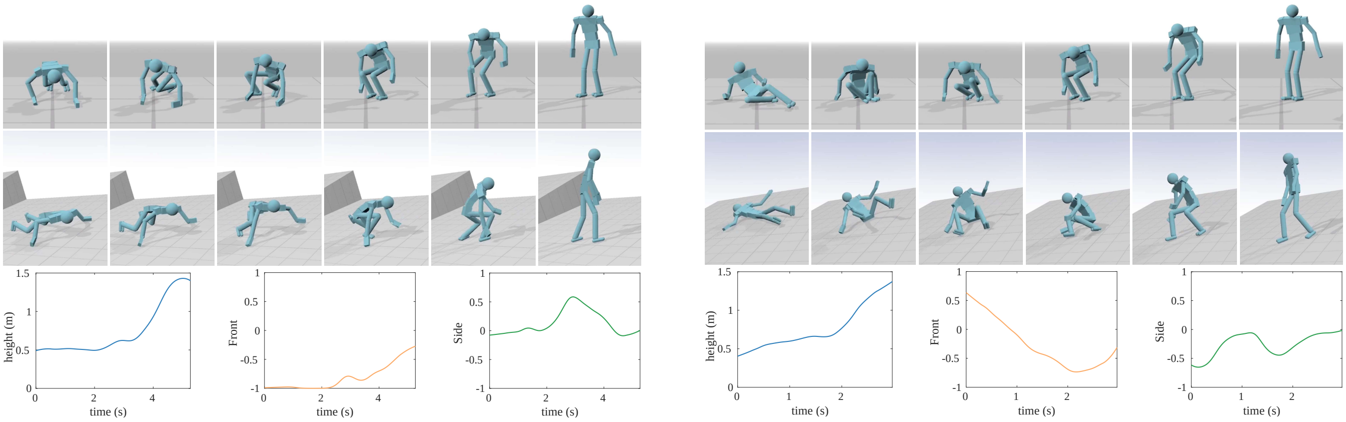

Figure 6: *Showing the character getting up on flat terrain (top) and sloped terrain (bottom) using feature curves that were extracted from reference motion clips. The agent is able to adapt to a more challenging environment and produce a similar get-up style.*

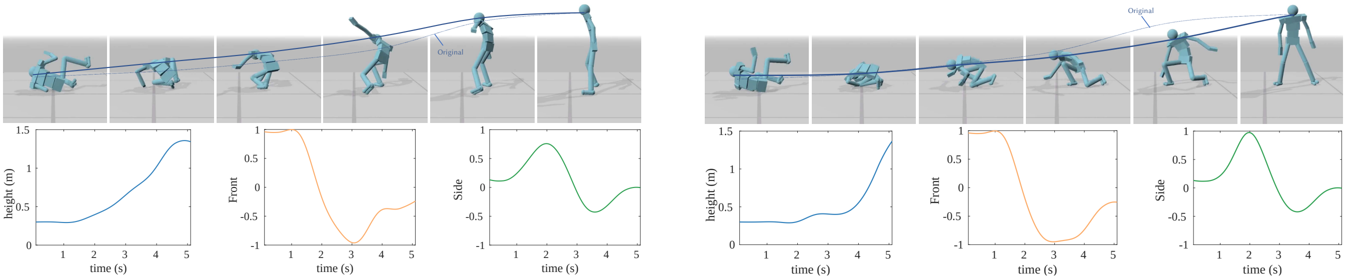

Figure 7: *Trajectories from a reference motion for sit-up and roll-over styles are edited to rise sooner (left) and later (right). Adjustments to the curves can be performed interactively, and the resulting motion is immediately shown to the user.*

6.2. Authoring Curves

Using our framework, it is simple to modify existing get-up motions in order to change the duration and form of the motion. For instance, Figure 7 shows the results of a sit-up and roll-over style of motion that has been changed to cause the character to rise sooner and later. This change requires small adjustments to the height and torso curves.

In addition to synthesizing get-up motions that resemble reference motions, our framework can also synthesize unique get-up motions by authoring of the input curves. For the purpose of better understanding the feature curves and how they effect the style of get-up motion, we manually classified each of the 13 reference motions from the LAFAN1 dataset according to the strategies identified by Bohannon and Lusardi [BL04]. The mean and standard deviation of the curves for each class of get-up motion is shown in Figure 8. Using these plots as guidance, and the spline editing tool described in Section 4.3, we authored several different styles of get-up motion highlighted in the biomechanics literature. Figure 9 shows a sit-up and roll over, side sit to kneel pivot motion authored with our method.

The supplementary video also shows examples of editing the feature curves interactively, both on flat and irregular terrain. The motions are synthesized in real-time using our framework and the user is provided immediate feedback about the changes and how they affect key characteristics of the motion.

Overall	Flat	Easy	Medium	Hard
90%	100%	99%	95%	77%

Table 1: *The success rate for learned control policies on all terrain types. The agent can get-up on almost any flat, easy and medium type terrain. The agent is less successful with hard terrains, but is still able to get-up in most cases.*

6.3. Robustness Evaluation

We evaluate the robustness of the trained policy by starting from 100 different terrain locations and tracking the success of the agent. A success is counted when the agent is able to remain standing after getting-up from the given position. We observe good performance on flat, easy, and medium terrains. The agent can almost always get-up on these types of terrain. However, the agent sometimes fails to stand on hard terrains, which is often caused by the agent not finding suitable positioning of the feet and hands. Nevertheless, the agent is able to succeed in most cases. Table 1 presents statistics of the success-failure analysis for the different terrain types.

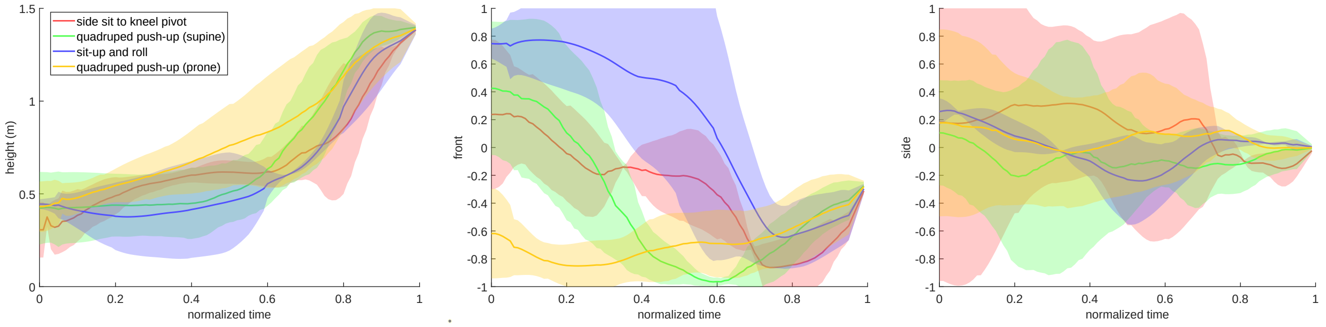

Figure 8: *The mean and standard deviation of feature curves extracted from reference motion clips and classified according to the strategies in Bohannon and Lusardi [BL04]. We distinguish between starting from a supine and prone posture for the quadruped push-up style. These plots provide a useful guide for authoring specific styles of get-up motion.*

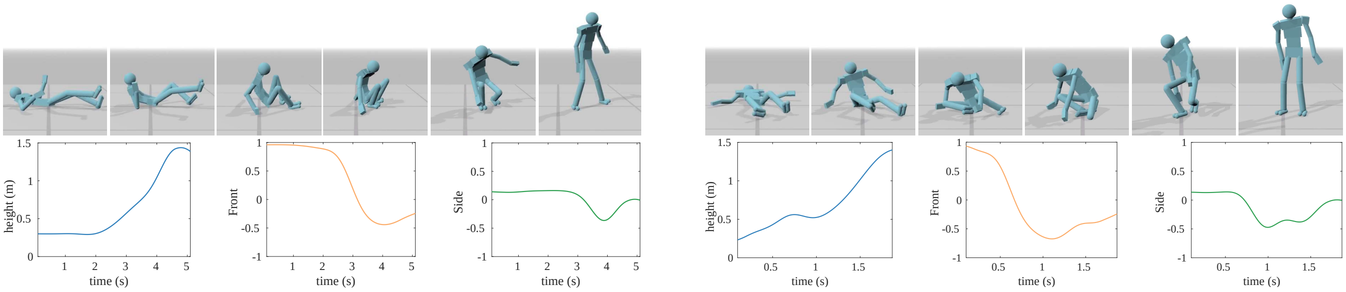

Figure 9: *Authored curves produce sit-up and roll (left) and side-sit to kneel-pivot (right) styles.*

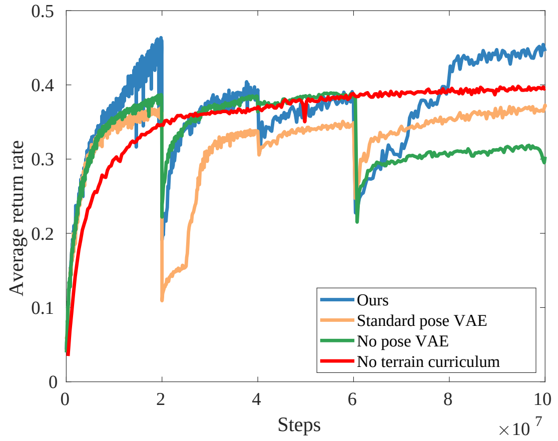

Figure 10: *An ablation comparison of our training methodology. Ours: the agent learns to get-up and stand on almost any terrain. Standard pose VAE: The agent is successful on flat and some irregular terrains. No terrain curriculum: The agent makes slow yet steady progress, but fails to get-up for some curve and terrain combinations. No pose VAE: Synthesized motions are unnatural.*

6.4. Ablation Study

Several ablation experiments were conducted to compare the effectiveness of our training methodology. The ablation experiments were conducted using 100M steps per experiment. The terrain difficulty was increased using a fixed schedule of every 20 million simulation steps, which roughly corresponds to the curriculum progression we observed during training.

Performance is measured as the average return rate, which is computed as the cumulative task reward divided by the total number of simulation steps required to perform the get-up and standing phases for all the curves and terrain positions in the training ensemble. This metric is used to compare the different training configurations since simply using the reward return is biased by the early termination, and we found that using the average return rate gives a clearer idea of how the agent performs. Figure 10 shows the performance using this metric for each training configuration we tested. Sudden drops in the average return rate indicate moments when the terrain difficulty is increased. A qualitative evaluation can also be found in the supplementary video, which is helpful for comparing the naturalness of motions. Each ablation experiment is further discussed below.

Standard pose VAE: We trained a pose VAE using the approach by Yin et al. [YYVDPY21] and use it in place of the C-VAE. The character is often able to follow the curve trajectories and produce several different motion styles, but often fails to reach a stable standing configuration. The agent shows good progress early-on, but that performance decreases after training advances to irregular and sloped terrains. The agent begins to succeed more frequently at standing toward the end of the training. However, we observed that some motions appear awkward and unnatural. Unlike the C-VAE, the postures chosen may be less appropriate in certain phases of the movement.

No pose VAE: The get-up policy is trained to directly produce PD servo target angles, rather than outputting the latent vector and pose offsets. Additionally, the naturalness reward r_t^{natural} term is removed from Equation 2, and thus only the task reward is used. The agent struggles to get-up, and typically cannot attain a standing posture. As can be seen in the supplementary video, the best strategy it learns is to raise the neck and torso while lying on the ground. Furthermore, the animations appear unnatural.

No terrain curriculum: The character is sometimes able to get-up and stand without the terrain curriculum learning. However, the final average return rate is lower than our proposed method, as shown in Figure 10. The agent fails to perform certain types of motion because of a lack of early training only using the flat terrain. We interpret that this is due to the number of successful get-up styles being reduced by the omission of prior training on the flat ground.

6.5. Discussion

Here we discussion some additional observations about our framework, as well as its limitations.

Curve Authoring. Creating the authored curves involves first analyzing real motion data and gaining insight about typical curve trajectories (see Figure 8, and then reproducing similar shapes using the spline tool. During preliminary experiments, we tried creating motions with just the height and front torso orientation curve, but the agent often failed to produce the desired motion. An additional torso direction was added and this allowed the agent to reproduce the styles of the reference motions more reliably. Situating the torso directions at two different segments of the torso also gives the agent some flexibility to succeed, which renders the authoring task less difficult. Unfortunately, the curves provide little control over the character's global forward facing direction since torso orientations are measured relative to the global vertical direction only. This choice was made so that curves were not dependent on the initial configuration of the character.

Initial torso orientation. The controller is less robust to states where the orientation curves do not match the initial pose of the character. The agent sometimes succeeds in tracking the curves by quickly correcting its orientation, but the character often struggles to track the desired get-up style.

Steep Terrain. Getting up on steep and complex terrain is challenging. However, as shown in the supplementary video, the character is able to successfully get-up on a wide range of terrain types, including terrain with a 30° slope. We sometimes observe slipping between the feet and ground on the steepest terrains in the training environment.

Standard pose VAE vs C-VAE. In preliminary experiments, the PD servos used by our character model had high torque limits, and in this case the standard pose VAE seemed to perform well in synthesizing get-up motions. However, the torque limits were reduced in order to improve the naturalness of the motions, and this resulted in the agent struggling to learn. By learning a latent motion prior that is conditioned on the torso height and orientations, the overall learning and quality of the motions was improved.

Generative adversarial learning methods. Our framework uses a pose VAE that is conditioned on the height and orientation of the torso to produce natural poses. Alternatively, generative adversarial imitation learning has also been shown to synthesize a variety of natural and diverse behaviors for complex control tasks involving physics-based characters. Notably, Peng et al. [PMA*21] recently proposed AMP for synthesizing stylized motion, and a diversity of get-up motions is demonstrated in their results. The goal of their approach is to temporally composite a variety of skills to perform different behaviors, whereas our work focuses on synthesizing natural get-up motions using only a small number of controllable signals that guide the style of the get-up motion. We found that it is sufficient to simply restrict the agent to explore actions in a natural pose space without also embedding the pose transitions. Furthermore, since transitions are not part of the latent model, the policy is able to synthesize novel transitions that are not contained in the original motion database, which may be important for curve authoring applications.

7. Conclusion

We propose a method to learn controllers allowing physics-based characters to get-up on different terrains in a variety of different ways from arbitrary fallen states. Robustness to terrain variation is achieved by a curriculum training approach. The style of motion is controllable using only a few key features, which may be authored using spline curves or extracted from reference motions. The synthesized motions are natural, which is achieved using a pose VAE trained on a motion capture database and conditioned on the signals that guide the get-up motion. This not only improves the quality of the motion, but also improves training performance.

In the future, it would be interesting to further explore the use of feature curves to guide the synthesize of other types of motions, particularly for tasks where humans are capable of using many different modes or styles to accomplish the same task. We are especially interested by contact-rich motions, such as dexterous manipulation. Our interactive editing tool could also be applied for authoring novel motions in these cases. Extending our approach to non-humanoid characters may also be possible. However, our framework requires a pose prior constructed from a database of reference motions, and this introduces additional challenges if motion data is unavailable. Finally, while we primarily focus on creating rich get-up motions, it would be interesting to synthesize variations that consider additional objectives, such as holding an object or avoiding the use of an injured limb. However, this requires additional reward shaping and changes to the pose VAE training, since the space of natural and useful motions may change.

Acknowledgements

Special thanks to Michiel Van de Panne and Tianxin Tao for their stimulating discussions about get-up motions. We also thank Victor Zordan for his helpful feedback. This work was supported by an NSERC Discovery grant and a MITACS Globalink Research Internship.

References

[ABdLH13] Al Borno M., de Lasa M., Hertzmann A.: Trajectory optimization for full-body movements with complex contacts. *IEEE Trans. on Visualization and Computer Graphics 19*, 8 (2013), 1405–1414. doi:10.1109/TVCG.2012.325. 2

[AT00] Adams J. M., Tyson S.: The effectiveness of physiotherapy to enable an elderly person to get up from the floor: A single case study. *Physiotherapy 86*, 4 (2000), 185–189. doi:10.1016/S0031-9406(05)60962-5. 3

[BCHF19] Bergamin K., Clavet S., Holden D., Forbes J. R.: Drecon: Data-driven responsive control of physics-based characters. *ACM Trans. Graph. 38*, 6 (nov 2019). doi:10.1145/3355089.3356536. 1, 2, 4

[BL04] Bohannon R. W., Lusardi M. M.: Getting up from the floor. determinants and techniques among healthy older adults. *Physiotherapy Theory and Practice 20*, 4 (2004), 233–241. doi:10.1080/09593980490887993. 3, 8, 9

[CL18] Chemin J., Lee J.: A physics-based juggling simulation using reinforcement learning. In *Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games* (New York, NY, USA, 2018), MIG '18, Association for Computing Machinery. doi:10.1145/3274247.3274516. 1

[CM 19] CM Labs Simulations: Vortex Studio 2019c. http://www.cm-labs.com/vortex-studio/, 2019. [Online]. 7

[CMM*18] Chentanez N., Müller M., Macklin M., Makoviychuk V., Jeschke S.: Physics-based motion capture imitation with deep reinforcement learning. In *Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games* (New York, NY, USA, 2018), MIG '18, Association for Computing Machinery. doi:10.1145/3274247.3274506. 2

[FKK*03] Fujiwara K., Kanehiro F., Kajita S., Yokoi K., Saito H., Harada K., Kaneko K., Hirukawa H.: The first human-size humanoid that can fall over safely and stand-up again. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems* (2003), vol. 2, pp. 1920–1926 vol.2. doi:10.1109/IROS.2003.1248925. 3

[FvdPT01] Faloutsos P., van de Panne M., Terzopoulos D.: Composable controllers for physics-based character animation. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2001), SIGGRAPH '01, Association for Computing Machinery, p. 251–260. doi:10.1145/383259.383287. 2

[GP12] Geijtenbeek T., Pronost N.: Interactive character animation using simulated physics: A state-of-the-art review. *Computer Graphics Forum 31*, 8 (Dec. 2012), 2492–2515. doi:10.1111/j.1467-8659.2012.03189.x. 2

[HTS*17] Heess N., TB D., Sriram S., Lemmon J., Merel J., Wayne G., Tassa Y., Erez T., Wang Z., Eslami S. M. A., Riedmiller M. A., Silver D.: Emergence of locomotion behaviours in rich environments. *CoRR abs/1707.02286* (2017). arXiv:1707.02286, doi:10.48550/arXiv.1707.02286. 2

[HYNP20] Harvey F. G., Yurick M., Nowrouzezahrai D., Pal C.: Robust motion in-betweening. *ACM Trans. Graph. 39*, 4 (jul 2020). doi:10.1145/3386569.3392480. 5

[KAS*16] Klima D. W., Anderson C., Samrah D., Patel D., Chui K., Newton R.: Standing from the floor in community-dwelling older adults. *Journal of aging and physical activity 24*, 2 (2016), 207–213. doi:10.1123/japa.2015-0081. 3

[KFH*07] Kanehiro F., Fujiwara K., Hirukawa H., Nakaoka S., Morisawa M.: Getting up motion planning using mahalanobis distance. In *Proceedings of IEEE International Conference on Robotics and Automation* (2007), pp. 2540–2545. doi:10.1109/ROBOT.2007.363847. 3

[LH12] Lin W.-C., Huang Y.-J.: Animating rising up from various lying postures and environments. *The Visual Computer 28*, 4 (2012), 413–424. doi:10.1007/s00371-011-0648-x. 2

[LLLL21] Lee S., Lee S., Lee Y., Lee J.: Learning a family of motor skills from a single motion clip. *ACM Trans. Graph. 40*, 4 (jul 2021). doi:10.1145/3450626.3459774. 2

[LPY16] Liu L., Panne M. V. D., Yin K.: Guided learning of control graphs for physics-based characters. *ACM Trans. Graph. 35*, 3 (may 2016). doi:10.1145/2893476. 2

[LXAK21] Luo Y., Xie K., Andrews S., Kry P.: Catching and throwing control of a physically simulated hand. In *Motion, Interaction and Games* (New York, NY, USA, 2021), MIG '21, Association for Computing Machinery. doi:10.1145/3487983.3488300. 1

[LZCVDP20] Ling H. Y., Zinno F., Cheng G., Van De Panne M.: Character controllers using motion VAEs. *ACM Trans. Graph. 39*, 4 (jul 2020). doi:10.1145/3386569.3392422. 2

[MD98] Morimoto J., Doya K.: Reinforcement learning of dynamic motor sequence: learning to stand up. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems.* (1998), vol. 3, pp. 1721–1726 vol.3. doi:10.1109/IROS.1998.724846. 3

[MHG*19] Merel J., Hasenclever L., Galashov A., Ahuja A., Pham V., Wayne G., Teh Y. W., Heess N.: Neural probabilistic motor primitives for humanoid control. In *International Conference on Learning Representations* (2019). doi:10.48550/arXiv.1811.11711. 2

[MHLC*21] Mourot L., Hoyet L., Le Clerc F., Schnitzler F., Hellier P.: A survey on deep learning for skeleton-based human animation. *Computer Graphics Forum* (2021). doi:https://doi.org/10.1111/cgf.14426. 2

[MTA*20] Merel J., Tunyasuvunakool S., Ahuja A., Tassa Y., Hasenclever L., Pham V., Erez T., Wayne G., Heess N.: Catch & carry: Reusable neural controllers for vision-guided whole-body tasks. *ACM Trans. Graph. 39*, 4 (jul 2020). doi:10.1145/3386569.3392474. 2

[MTP12] Mordatch I., Todorov E., Popović Z.: Discovery of complex behaviors through contact-invariant optimization. *ACM Trans. Graph. 31*, 4 (jul 2012). doi:10.1145/2185520.2185539. 2

[MTT*17] Merel J., Tassa Y., TB D., Srinivasan S., Lemmon J., Wang Z., Wayne G., Heess N.: Learning human behaviors from motion capture by adversarial imitation. *arXiv preprint arXiv:1707.02201* (2017). doi:10.48550/arXiv.1707.02201. 1, 2

[NBRH19] Naderi K., Babadi A., Roohi S., Hamalainen P.: A reinforcement learning approach to synthesizing climbing movements. In *IEEE Conference on Games 2019, CoG 2019* (London, United Kingdom, Aug. 2019), IEEE Conference on Computatonal Intelligence and Games, IEEE, pp. 1–7. IEEE Conference on Games, CoG ; Conference date: 20-08-2019 Through 23-08-2019. doi:10.1109/CIG.2019.8848127. 1

[PALvdP18] Peng X. B., Abbeel P., Levine S., van de Panne M.: DeepMimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Trans. Graph. 37*, 4 (jul 2018). doi:10.1145/3197517.3201311. 1, 2, 3

[PCZ*19] Peng X. B., Chang M., Zhang G., Abbeel P., Levine S.: MCP: Learning composable hierarchical control with multiplicative compositional policies. In *Advances in Neural Information Processing Systems 32*, Wallach H., Larochelle H., Beygelzimer A., d'Alché-Buc F., Fox E., Garnett R., (Eds.). Curran Associates, Inc., 2019, pp. 3681–3692. doi:10.48550/arXiv.1905.09808. 2

[PEA83] Plagenhoef S., Evans F. G., Abdelnour T.: Anatomical data for analyzing human motion. *Research Quarterly for Exercise and Sport 54*, 2 (1983), 169–178. doi:10.1080/02701367.1983.10605290. 3

[PGH*22] Peng X. B., Guo Y., Halper L., Levine S., Fidler S.: ASE: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Trans. Graph. 41*, 4 (jul 2022). doi:10.1145/3528223.3530110. 2

[PMA*21] Peng X. B., Ma Z., Abbeel P., Levine S., Kanazawa A.: AMP: Adversarial motion priors for stylized physics-based character control. *ACM Trans. Graph. 40*, 4 (jul 2021). doi:10.1145/3450626.3459670. 2, 10

[PRL*19] Park S., Ryu H., Lee S., Lee S., Lee J.: Learning predict-and-simulate policies from unorganized human motion data. *ACM Trans. Graph. 38*, 6 (nov 2019). doi:10.1145/3355089.3356501. 2

[RHG*21] Raffin A., Hill A., Gleave A., Kanervisto A., Ernestus M., Dormann N.: Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research 22*, 268 (2021), 1–8. URL: http://jmlr.org/papers/v22/20-1364.html. 7

[SWD*17] Schulman J., Wolski F., Dhariwal P., Radford A., Klimov O.: Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017). doi:10.48550/arXiv.1707.06347. 6

[TET12] Tassa Y., Erez T., Todorov E.: Synthesis and stabilization of complex behaviors through online trajectory optimization. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Vilamoura, Algarve,Portugal, 2012), IEEE, pp. 4906–4913. doi:10.1109/IROS.2012.6386025. 2

[TWGVdP22] Tao T., Wilson M., Gou R., Van de Panne M.: Learning to get up. *ACM Trans. Graph.* (jul 2022). doi:10.1145/3528233.3530697. 3

[WGH20] Won J., Gopinath D., Hodgins J.: A scalable approach to control diverse behaviors for physically simulated characters. *ACM Trans. Graph. 39*, 4 (jul 2020). doi:10.1145/3386569.3392381. 2

[WGH21] Won J., Gopinath D., Hodgins J.: Control strategies for physically simulated characters performing two-player competitive sports. *ACM Trans. Graph. 40*, 4 (jul 2021). doi:10.1145/3450626.3459761. 1, 2

[WMR*17] Wang Z., Merel J. S., Reed S. E., de Freitas N., Wayne G., Heess N.: Robust imitation of diverse behaviors. *Advances in Neural Information Processing Systems 30* (2017). doi:10.48550/arXiv.1707.02747. 2

[XLKvdP20] Xie Z., Ling H. Y., Kim N. H., van de Panne M.: ALLSTEPS: Curriculum-driven learning of stepping stone skills. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Goslar, DEU, 2020), SCA '20, Eurographics Association. doi:10.1111/cgf.14115. 2

[YK20] Yuan Y., Kitani K.: Residual force control for agile human behavior imitation and extended motion synthesis. *Advances in Neural Information Processing Systems 33* (2020), 21763–21774. doi:10.48550/arXiv.2006.07364. 2

[YYVDPY21] Yin Z., Yang Z., Van De Panne M., Yin K.: Discovering diverse athletic jumping strategies. *ACM Trans. Graph. 40*, 4 (jul 2021). doi:10.1145/3450626.3459817. 2, 5, 6, 9