From Universal Humanoid Control to Automatic Physically Valid Character Creation

ZHENGYI LUO, YE YUAN, and KRIS M. KITANI, Carnegie Mellon University, USA

Automatically designing virtual humans and humanoids holds great potential in aiding the character creation process in games, movies, and robots. In some cases, a character creator may wish to design a humanoid body customized for certain motions such as karate kicks and parkour jumps. In this work, we propose a humanoid design framework to automatically generate physically valid humanoid bodies conditioned on sequence(s) of pre-specified human motions. First, we learn a generalized humanoid controller trained on a large-scale human motion dataset that features diverse human motion and body shapes. Second, we use a design-and-control framework to optimize a humanoid’s physical attributes to find body designs that can better imitate the pre-specified human motion sequence(s). Leveraging the pre-trained humanoid controller and physics simulation as guidance, our method is able to discover new humanoid designs that are customized to perform pre-specified human motions. Video demos can be found at the project page.

Fig. 1. Discovered designs for parkour, jump-rope, crawl, jumping jacks, and cartwheel.

1 INTRODUCTION

We consider the task of determining the body structure of a digital character that possesses the physical qualities necessary to perform a certain range of tasks. For example, a digital character that needs to perform cartwheels or back-flips might require a light body, or a character that needs great balance might have a heavier lower body and a lower center of mass. To generate such digital characters whose physique is representative of the type of actions it must perform, we proposed a method for automatically generating novel humanoid designs that are physically valid and conditioned to perform a set of predefined actions.

Concretely, given the sequence(s) of human motion captured by motion capture, our method will output a humanoid design that can successfully imitate that sequence in a physics simulator (Fig. 1). To find humanoid designs that are suited for performing a diverse set of motions, we utilize physics simulation and data-driven motion imitation techniques to enforce physical plausibility and human likeness. Grounded in physical laws, our method can recover character designs that can serve as blueprints to aid the ideation process for creating uniquely looking yet physically controllable characters.

The challenges are two-fold: (1) the design space to search for suitable characters is immense; (2) we desire our recovered character design to specialize in performing the input motion while staying useful for general daily motion tasks (e.g., walking and running). However, controlling a humanoid to imitate diverse reference motions inside the physics simulation itself is a challenging and time-consuming task [Mourot et al. 2021]. Start-of-the-art methods often take days or weeks to train suitable controllers for a single character design [Fussell et al. 2021; Luo et al. 2021; Wang et al. 2020; Won et al. 2020; Yuan and Kitani 2020]. This can be attributed to the difficulty of controlling a simulated character with such high degrees-of-freedom (e.g., 69 on the SMPL [Loper et al. 2015] human model) and the dynamics mismatch between simulated rigid bodies and real-world physics. As a result, while algorithmic discovery of agent design and morphology has been applied to simple walkers and hoppers [Luck et al. 2019; Wang et al. 2019; Yuan et al. 2021a] (such as those defined in the popular OpenAI Gym [Brockman et al. 2016]), learning specialized humanoid designs given a reference motion has not yet been attempted.

In this paper, we tackle automatic and efficient character discovery from two complementary angles: (1) learning a general character controller that can perform tens of thousands of human motions, given reference motions and pre-scanned human physique; (2) given a sequence(s) of reference motion, we form a dual design-and-control policy for intelligently altering the character’s design. Specifically, we first incorporate the human body shape into the state space of our controller and learn from a large-scale human motion dataset [Mahmood et al. 2019] accompanied by realistic human body shape annotation. Based on the key observation that foot geometry and compliance are key contributors to agent instability [Park et al.
2 RELATED WORKS

2.1 Simulated Character Control

Controlling a simulated character inside a physics simulation [Chentanez et al. 2018; Fussell et al. 2021; Peng et al. 2018a, 2021; Raibert and Hodgins 1991; Wang et al. 2020; Won et al. 2020; Yuan and Kitani 2020] has attracted increasing attention from the animation and graphics community. It has also been successfully applied in other fields, such as computer vision [Luo et al. 2021; Yuan et al. 2021b]. Originally developed by learning a single motion clip [Peng et al. 2018a], deep reinforcement learning (DRL) has shown impressive results in motion imitation that extend to controllable characters capable of imitating thousands of motion clips [Chentanez et al. 2018; Luo et al. 2021; Wang et al. 2020; Won et al. 2020]. More recent advances focus on improving the time of environmental interaction through model-based reinforcement learning [Fussell et al. 2021], as well as improving the use of motion clips through generative models [Peng et al. 2021] and motion warping [Lee et al. 2021]. Although these controllers can achieve impressive imitation results, they are often limited to a small subset of human motion defined by task and training data. To learn a more universal humanoid controller, residual force control [Yuan and Kitani 2020] is proposed to help balance the humanoid throughout the imitation process and has been successfully applied to learn controllers that can mimic up to tens of thousands of motion clips [Luo et al. 2021; Yuan et al. 2021b]. However, these methods have been focusing on obtaining better controllers [Mourot et al. 2021] for a single simulated character without considering body shape and weight variations.

2.2 Character Design Optimization

Automated character and agent design has been extensively explored in both the graphics and robotics community, since computer-aided designs have great potential to create elegant and effective characters or robots. Earlier work often employ evolutionary methods such as CMA-ES [Luck et al. 2019] to jointly optimize for better design and control policies. Similar to character control, recent advances in DRL based methods also achieved impressive results in optimizing not only the character’s physical attributes such as bone length and weight [Ha 2019; Schaff et al. 2019], but also its skeleton structure and morphology [Luck et al. 2019; Wang et al. 2019; Yuan et al. 2021a]. However, most of the prior arts have been focusing on a low-dimensional, single task setting, where the character is tasked to achieve simple tasks such as “moving forward” or “hopping around” inside a simulation. Due to the high dimensionality of possible design parameters and morphology, such frameworks have yet to be applied to humanoids, where controlling a humanoid for motion imitation is an unsolved task itself. Although some exploration has been made in training controllers that can adapt to different body shape parameters [Won and Lee 2019], the controllers learned in this setting can simple tasks such as moving forward or imitating a single motion clip. In this work, we tackle the challenging task where we aim to discover humanoid designs that not only specialize in a given motion clip but also retain the ability to imitate general motion sequences.

3 APPROACH

The problem of discovering motion-dependent design can be formulated as follows: given a sequence of human motion \( \mathbf{q}_{1:T} \), we aim to determine a suitable humanoid design \( \mathbf{D} \) that can better imitate the given sequence. We first train a universal body shape-dependent humanoid controller (Sect. 3.2) which will be fixed during the design optimization process. Then, we optimize the agent’s design parameters \( \mathbf{D} \) using a design-and-control framework (Sec. 3.3). As a notation convention, we use \( \hat{\cdot} \) to denote the target pose, and normal symbols without accents to denote poses from the physics simulator (Mujoco [Todorov et al. 2012]).
We first utilize an automatic simulated character creation process to vertices to bones based on the highest weight value. On the basis of \( q^3 \D\) pose State. The simulation state \( S \) consists of the character’s 3D position \( q^p \in \mathbb{R}^3 \) and orientation \( q^q \in \mathbb{R}^{3 \times 3} \) for each one of the human joints. Velocity \( \dot{q} \) is defined as a concatenation of linear and angular velocities \( \dot{q} = (\dot{q}^p, \dot{q}^q) \). These values encompass the full simulation state of the humanoid and are computed by the simulator at each time step. Policy. The policy computes the mean of a Gaussian distribution \( \mu \) with a fixed covariance matrix \( \Sigma \): \( \pi^B_\theta(s_t, \dot{q}_t, D) = \mathcal{N}(\mu, \Sigma) \). During training, we sample the action from the distribution to explore different actions, and during testing, we use the mean action. Given the simulation state \( s_t \) and reference motion \( \tilde{q}_t \), the policy will first use a feature extraction function \( T_E \) to extract the difference between the simulated character state and the reference motion and transform these features into an character-centric coordinate system for computation. Specifically:

\[
(T_{CC}(\tilde{q}^t, q^p - \tilde{q}^t, \dot{q}_t), (q^q \odot \tilde{q}^q), \tilde{q}^q) = T_E(s_t, \tilde{q}_t),
\]

\( T_{CC} \) transfers the input quantities from the global space into the character’s local coordinate frame based on the character root joint’s translation and position. Character design properties \( D \) will be specified in Sec. 3.3.

Action. The action \( a_t = (p^A_1, k_i, e_i) \) consists of the target joint angles \( p_i \), for the PD controllers attached to each joint, PD controller gains \( k_i \), and the external residual force \( e_i \) applied to the humanoid’s foot. Here we adopt the meta-PD controller \( [Yuan et al. 2021b] \) and the residual action representation \( [Park et al. 2019] \) popular in humanoid control tasks. For each joint \( i \), the torque applied can be computed as \( \tau^t = k^A_i \circ (p^A_i - p_i) - k_d^A \circ p_i \), where \( k^A_i \) and \( k_d^A \) are learned gains by the meta-PD controller. \( p^A_i \) is the target joint angle, \( p_i \) the current joint angle, and \( e_i \) the current joint velocity. \( \circ \) is the element-wise multiplication. Based on the observation that foot compliance and flexibility are crucial in helping the humanoid stay stable \( [Park et al. 2018] \), we resort to applying residual force on the foot geometry to best match the ground reaction force induced by foot muscle and shoes. For each foot geometry (ankle and toe as defined in SMPL), we apply five external forces \( e_i = (e_1^A, e_2^A, ..., e_3^A) \) where \( e_i^A \) is the first external force applied to the ankle/toe at timestep \( t \) for each external force \( e_i^A \), it consists of the contact point position on the geom’s local coordinate frame \( p^A_1 \in \mathbb{R}^3 \), its direction \( d^A_1 \in \mathbb{R}^3 \), and magnitude \( s^A_1 \in \mathbb{R}^1 \). To ensure that the policy does not abuse the external force, we constrain that the external force can only be applied when the foot is in contact with other geoms (e.g. the ground), as illustrated in Fig.3:

\[
e_i^A = \begin{cases} (p^A_1, d^A_1, s^A_1), & \text{if geometry A is in contact with the ground} \\ 0, & \text{otherwise} \end{cases}
\]

Reward. We use the standard motion tracking reward based on comparing the simulated character state \( s_t \) and the input reference motion \( \tilde{q}_t \):

![Fig. 2. Visualization of the character design space our algorithm can search through. From left to right: 1) the original humanoid shape, 2) & 3) by sampling the SMPL shape space \( \beta, \beta \) & 4) & 5) by sampling the SMPL shape space and geoms.](image)
Fig. 3. Residual force applied at the humanoid’s foot to compensate for the dynamics-mismatch between real foot/shoes and simulation.

\[ r_t = w_p \cdot r_p + w_o \cdot r_o + w_e \cdot r_e + w_{ef} \cdot r_{ef} \]

\[ r_p = \exp[-2.0\|\mathbf{q}_i^t \oplus \hat{\mathbf{q}}_i^t\|^2], \text{ character rotational pose reward} \]

\[ r_o = \exp[-0.005\|\mathbf{q}_i^t - \mathbf{q}_i^t\|^2], \text{ humanoid reward} \]

\[ r_e = \exp[-5\|\mathbf{q}_i^t \oplus \mathbf{q}_i^t\|^2], \text{ character positional pose reward} \]

\[ r_{ef} = \exp[-\|\mathbf{e}_i\|^2], \text{ residual force reward} \]

where \( w_p, w_o, w_e, w_{ef} \) are the respective reward weights.

**Training procedure.** We train on the training split of the AMASS dataset [Mahmood et al. 2019] and remove motion sequences that involve human-object interactions (such as stepping on stairs or leaning on tables). This results in 11402 high-quality motion sequences. At the beginning of each episode, we sample a fixed-length sequence (maximum length 300 frames) to train our motion controller. We use reference state initialization [Peng et al. 2018a] to randomly choose the starting point for our sampled sequence and terminate the simulation if the simulated character deviates too far from the reference motion \( \mathbf{q}_i^t \). To choose which motion sequence to learn from, we employ a hard-negative mining technique based on the success rate of the sequence during training. For each motion sequence \( \mathbf{q}_i^t = \mathbf{q}_0^t, \ldots, \mathbf{q}_{500}^t \), we maintain a history queue (of maximum length 50) for the most recent samples is the sequence: \( h_i^t \), where each \( h_i^t \) is a Boolean variable indicating whether the controller has successfully imitated the sequence during training. We calculate the expected success rate \( s^t \) of the sequence based on the exponentially weighted moving average using the history \( s^t = \text{ewma}(h_1^t, \ldots, h_{50}^t) \). The probability of sampling sequence \( i \) is then \( P(\mathbf{q}^i) = \frac{\exp(-s^t/r)}{\sum_j \exp(-s^j/r)} \), where \( r \) is a temperature parameter. Intuitively, the more we fail at imitating a sequence, the more likely we will sample it. Each \( s^t \) is initialized as 0.5.

### 3.3 Motion-dependent Character Discovery

To discover suitable character designs based on a sequence(s) of similar motions, we employ a two-stage design and control framework. While prior arts [Wang et al. 2019; Yuan et al. 2021a] apply similar techniques in finding simple characters for the task of “moving forward”, we focus on discovering designs that are capable of specializing in an input sequence while retaining its ability to perform diverse activities. We achieve this by pairing a humanoid controller with an additional design policy and optimizing the agent design through first-order optimization. Instead of training both the control and design policy jointly from scratch, which can be unstable and time-consuming, we utilize a pretrained humanoid controller and freeze its weights during the design optimization process. This way, the pretrained controller acts as a human dynamics prior that constrains the design policy to only find agents that are controllable yet more suitable for imitating the specified motions.

**Agent Design Parameters.** The design parameters are defined as:

\[ D \pm (\beta, w, b, f, m, g). \]

Here we modify the humanoid’s SMPL shape parameter \( \beta \), weight \( w \), height \( h \), joint friction loss \( f \) (joint dry friction) [Todorov et al. 2012], the size and density of each bone’s geometry \( m \), and motor gear for each joints’ actuator \( g \). These parameters control the simulation properties of the character such as mass, inertia, and motor strength. Fig. 2 shows possible humanoid designs by sampling the SMPL shape parameters and sizes for each bone’s geometry.

**Dual Policy for design and control.** Given a pretrained control policy \( \pi^C_D(\mathbf{a}_t|\mathbf{s}_t, \mathbf{q}_t, D) \), we augment it with a design policy \( \pi^D_\theta(\mathbf{a}_t|\mathbf{s}_t, \mathbf{q}_t, D) \) that aims to maximize the accumulated reward conditioned on the pretrained controller:

\[ D^* = \arg\max_D \mathbb{E}_{D,\pi^C_D}[\sum_{t=1}^T Y^{t-1} r_t], \quad (4) \]

Design policy \( \pi^D_\theta(D|s_0, q_0) \) is conditioned on the pose of the first frame of the input reference motion \( q_0 \) and computes the optimal design for the input sequence. Each episode is divided into two stages: design and control. At the beginning of each episode, we first use the design policy to change the agent’s physical characteristics \( D \), and then roll out the control policy for motion imitation. Combining the two policies, we form the overall decision-making framework \( \pi^\theta \):

\[ \pi^\theta(\mathbf{a}^D_t, D|s_t, \mathbf{q}_t, D_t) = \begin{cases} \pi^D_\theta(D|s_0, \mathbf{q}_0), & \text{design stage} \\ \pi^C_\theta(\mathbf{a}_t|s_t, \mathbf{q}_t, D), & \text{control stage}. \end{cases} \quad (5) \]

The design and control algorithm can be seen in Fig. 4 and Algorithm 1. We use the same reward and optimization algorithm (PPO) as in the motion imitation case, since the main objective remains the same: better motion tracking. Notice that the agent does not receive any reward \((r_0 = 0)\) at the design stage since interacting with the environment while changing the character’s design is undesirable. Rather, the reward signal for the design policy comes from the control policy’s interaction with the environment. We learn a new value function conditioned on the agent design to find the best design parameters:

\[ V(s, D) \pm \mathbb{E}_{D, \pi^C_D}[\sum_{t=1}^T Y^{t-1} r_t]. \quad (6) \]

### 4 EXPERIMENTS

We conduct our experiments on the motion imitation task and try to gauge the effectiveness of 1) the general humanoid controller for the general motion imitation task and 2) our humanoid design discovery process’s ability to find suitable character designs based on input motion sequence(s) of different categories and requirements. To test our motion imitation framework, we report motion imitation
We report motion tracking quality through a suite of pose-based and physics-based metrics:

### 4.1 Metrics

For agent design discovery, we report results on finding characters for 1) finding the best design for single sequences (parkour, karate, belly dancing, boxing), 2) for a category of sequences (dancing and running) and 3) for the whole AMASS training split.

### 4.2 Motion Imitation

Table 2 shows the quantitative results of our controller on motion imitation. Here we report the results of UHC with different configurations: UHC w/o RFC is our controller without using any external residual force; UHC-RFC-Root is similar to [Luo et al. 2021] where the external force is applied at the root of the character; UHC-RFC-Foot is our proposed method where the external force is applied on the foot and only when the foot is in contact with the ground; UHC-RFC-All is a configuration where we apply an external force to every joint in the human body. UHC-RFC-All can serve as an oracle method showcasing the best performance of a physics-based humanoid controller, where imitation quality rather than full physical realism of the simulated character is the priority. As can be shown in the result, our proposed method outperforms the previous state-of-the-art in terms of motion imitation quality on all metrics and is close to the physically implausible UHC-RFC-All case. Compared to UHC-No-RFC, we can see that our external force on the foot is effective compensating for the foot instability induced by the simple foot design (as indicated by the high acceleration error, the

Fig. 4. Given a reference motion sequence, our framework first uses a design policy \( \pi^D_\theta \) to sample potential candidates, and then uses a control policy \( \pi^C_\theta \) to imitate reference motion to gather feedback (reward) about the designs.

---

**Algorithm 1: Character Design Discovery via Design-and-Control Optimization**

**Input:** pretrained humanoid control policy \( \pi^C_\theta \), target motion sequence(s) \( \mathcal{Q} \)

while not converged do

\[ M \leftarrow \emptyset \] // initialize sampling memory

while \( M \) not full do

\[ D_0 \leftarrow \text{initial humanoid design} \]

\[ \mathcal{q}_{1:T} \leftarrow \text{random sequence of motion from dataset } \mathcal{Q} \]

for \( t = 1 \) do

\[ D \sim \pi^D_\theta(D|s_0, \mathcal{q}_0) \] sample design action

\[ r_t \leftarrow 0; \text{store } (r_t, D) \text{ into } M; \]

end

\[ s_0 \leftarrow \text{initialize simulation state based on design } D; \]

for \( t = 2, \ldots, T \) do

\[ a_t \sim \pi^C_\theta(a_t|s_t, \mathcal{q}_t, D) \] sample control action from pretrained controller

\[ s_{t+1} \leftarrow T(s_t|s_t, a_t) \] simulation dynamics

\[ r_t \leftarrow \text{imitation reward}; \]

store \( (r_t, a_t, s_{t+1}, \mathcal{q}_t, D) \) into \( M; \)

end

update \( \pi^D_\theta \) with PPO using collected samples from \( M \)

end

**Return** \( \pi^D_\theta \) and \( D \)

---

results on both the training and testing split of the AMASS dataset. For agent design discovery, we report results on finding characters for 1), finding the best design for single sequences (parkour, karate, belly dancing, boxing), 2) for a category of sequences (dancing and running) and 3) for the whole AMASS training split.
We evaluate our character design discovery framework by assigning target motion from the train and test split of the AMASS dataset and that only the design parameters can significantly improve the imitation quality, as shown in Table 2. We can see that for each individual sequence, optimizing the design parameters can improve the imitation quality. For example, the belly dance sequence improves from 84.1% to 98.3% when optimizing the design parameters.

4.3 Character Design Discovery

We evaluate our character design discovery framework by assigning target motion from the train and test split of the AMASS dataset. The results are shown in Table 1, where the suffix indicates the number of motion sequences fed into the algorithm.

Table 1. Evaluation of agent specializing in various challenging tasks. Here the suffix indicates the number of motion sequences fed into the algorithm.

| Sequence | $S_{\text{succ}} \uparrow$ | $E_{\text{mpjpe}} \downarrow$ | $E_{\text{mpjpe-g}} \downarrow$ | $E_{\text{acc}} \downarrow$ |
|----------|-----------------|-----------------|-----------------|-----------------|
| Belly Dance-1 | 0% → 100% | 183.3 → 36.9 | 347.1 → 55.0 | 7.6 → 7.6 |
| Parkour-1 | 0% → 100% | 175.6 → 87.1 | 324.2 → 146.4 | 24.8 → 15.0 |
| Karate-1 | 100% → 100% | 35.9 → 30.1 | 45.8 → 39.9 | 7.1 → 7.1 |
| Crawl-1 | 100% → 100% | 66.4 → 40.6 | 307.6 → 67.2 | 3.8 → 4.3 |
| Cartwheel-1 | 0% → 100% | 160.9 → 37.4 | 284.8 → 66.2 | 8.1 → 4.9 |
| Dance-200 | 57.0% → 72.0% | 84.1 → 58.0 | 146.7 → 98.7 | 13.3 → 13.3 |
| Tennis-60 | 96.7% → 100% | 27.8 → 21.8 | 40.5 → 30.7 | 4.2 → 4.1 |
| Crawl-37 | 94.6% → 94.6% | 62.0 → 40.6 | 163.6 → 87.9 | 5.6 → 9.1 |
| Cartwheel-4 | 25% → 75% | 219.6 → 89.4 | 393.6 → 166.1 | 19.4 → 12.5 |
| Kick-302 | 98.3% → 98.3% | 45.5 → 38.8 | 75.4 → 62.4 | 7.5 → 9.1 |

Table 2. Evaluation of motion imitation for our humanoid controller using target motion from the train and test split of the AMASS dataset.

| Method | $S_{\text{succ}} \uparrow$ | $E_{\text{mpjpe}} \downarrow$ | $E_{\text{mpjpe-g}} \downarrow$ | $E_{\text{acc}} \downarrow$ |
|--------|-----------------|-----------------|-----------------|-----------------|
| AMASS dataset Training (11402 sequences) | | | | |
| No-RFC | 89.7% | 32.1 | 50.7 | 38.8 |
| RFC-RFOot | 94.7% | 34.5 | 50.9 | 4.5 |
| RFC-All | 94.6% | 39.3 | 38.3 | 1.2 |

The humanoid is jittering to try to stay upright. When UHC-RFC-Root and UHC-RFC-Foot are compared, it is clear that applying residual force only on the foot is more effective and has a more intuitive justification than using a supporting force on the character’s root.

5 LIMITATIONS AND FUTURE WORK

Currently, our agent discovery process is still relatively time-consuming: utilizing a pretrained controller, it takes around 3 hours to find suitable agents for a whole category of motion sequences, demonstrating its ability to generalize to a suite of diverse motions that shares similar traits. As motion is best seen in videos, we refer our reader to our https://youtu.be/uC0P2iB56kM for visual results.

One of our main goals for fixing the control policy is to discover agents that can perform specific actions in certain motion categories while preserving the agent’s ability to imitate general motions (such as walking and running). To this end, we evaluate these special agent’s performance on the AMASS test set. Results from Table 3 shows that the specialized agents, while imitating general motions (such as walking and running), still retain their ability to perform general motion and largely preserve their performance on the AMASS test split.
framework to generate diverse designs based on input motion at test time without additional training.

6 CONCLUSION

In this work, we present an automatic character design framework for generating controllable agents for various categories of highly dynamic human motion. Given a target motion sequence (either through MoCap, keyframe animation, or recovered from video [Peng et al. 2018b]), our pipeline can recover humanoid designs that can specialize on such sequences while retaining the ability to imitate diverse motion sequences. Notably, we also propose a more physically valid humanoid controller than prior art, and compensate for the lack of biologically sound foot design through residual force control. Recovered agents can serve as inspiration for creating physically valid controllable characters based on unique motion requirements.

REFERENCES

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym. ArXiv abs/1606.01540 (2016).
N. Chentanez, M. Müller, M. Macklin, Viktor Makoviychuk, and S. Jeschke. 2018. Physics-based motion capture imitation with deep reinforcement learning. Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games (2018).
Sammy Christen, Muhammed Kocabas, Emre Aksan, Jemin Hwangbo, Jie Song, and Otmar Hilliges. 2021. D-Grasp: Physically Plausible Dynamic Grasp Synthesis for Hand-Object Interactions. ArXiv abs/2112.03028 (2021).
Levi Fussell, Kevin Bergamin, and Daniel Holden. 2021. SuperTrack. ACM Transactions on Graphics (TOG) 40 (2021), 1 – 13.
David R Ha. 2019. Reinforcement Learning for Improving Agent Design. Artificial Life 25 (2019), 352–365.
Seyoung Lee, Sunmin Lee, Yongwoo Lee, and Jehee Lee. 2021. Learning a family of motor skills from a single motion clip. ACM Transactions on Graphics (TOG) 40 (2021), 1 – 13.
Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: a skinned multi-person linear model. ACM Trans. Graph. 34 (2015), 248:1–248:16.
Kevin Sebastian Luck, Heni Ben Amor, and Roberto Calandra. 2019. Data-efficient Co-Adaptation of Morphology and Behaviour with Deep Reinforcement Learning. In CoRL.
Zhengyi Luo, Ryo Hachiuma, Ye Yuan, and Kris Kitani. 2021. Dynamics-Regulated Kinematic Policy for Egocentric Pose Estimation. ArXiv abs/2106.05969 (2021).

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. 2019. AMASS: Archive of Motion Capture As Surface Shapes. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) (2019), 5441–5450.

Lucas Mourny, Ludovic Hoyet, François Le Clerc, François Schnitzler, and Pierre Hellier. 2021. A Survey on Deep Learning for Skeleton-Based Human Animation. ArXiv abs/2110.06901 (2021).

Hwangil Park, Ri Yu, and Jehee Lee. 2018. Multi-Segment Foot Modeling for Human Animation. EasyChair Preprints (2018).

Soohwan Park, Hoseok Ryu, Seyoung Lee, Summin Lee, and Jehee Lee. 2019. Learning predict-and-simulate policies from unorganized human motion data. ACM Transactions on Graphics (TOG) 38 (2019), 1–11.

Xue Bin Peng, P. Abbeel, Sergey Levine, and Michiel van de Panne. 2018a. DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills. ACM Trans. Graph. 37 (2018), 143:1–143:14.

Xue Bin Peng, Angjoo Kanazawa, Jitendra Malik, P. Abbeel, and Sergey Levine. 2018b. SFV: Reinforcement Learning of Physical Skills from Videos. ACM Trans. Graph. 37 (2018), 178:1–178:14.

Xue Bin Peng, Ze Ma, P. Abbeel, Sergey Levine, and Angjoo Kanazawa. 2021. AMP: Adversarial Motion Priors for Stylized Physics-Based Character Control. ACM Trans. Graph. 40 (2021), 144:1–144:20.

Marc H. Raibert and Jessica K. Hodgins. 1991. Animation of dynamic legged locomotion. Proceedings of the 18th annual conference on Computer graphics and interactive techniques (1991).

Charles B. Schaff, David Yunis, Ayan Chakrabarti, and Matthew R. Walter. 2019. Jointly Learning to Construct and Control Agents using Deep Reinforcement Learning. 2019 International Conference on Robotics and Automation (ICRA) (2019), 9798–9805.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. ArXiv abs/1707.06347 (2017).

Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. MujCoCo: A physics engine for model-based control. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (2012), 5026–5033.

Tingwu Wang, Yuntong Guo, Maria Shugrina, and Sanja Fidler. 2020. UniCon: Universal Neural Controller For Physics-based Character Motion. arXiv:2011.15119 [cs.GR]

Tingwu Wang, Yuhao Zhou, Sanja Fidler, and Jimmy Ba. 2019. Neural Graph Evolution: Towards Efficient Automatic Robot Design. ArXiv abs/1906.05370 (2019).

Jungdam Won, Deepak Gopinath, and Jessica Hodgins. 2020. A scalable approach to control diverse behaviors for physically simulated characters. ACM Trans. Graph. 39, 4 (2020).

Jungdam Won and Jehee Lee. 2019. Learning body shape variation in physics-based characters. ACM Trans. Graph. 38, 6 (Nov. 2019), 1–12.

Ye Yuan and Kris Kitani. 2020. Residual Force Control for Agile Human Behavior Imitation and Extended Motion Synthesis. In Advances in Neural Information Processing Systems.

Ye Yuan, Yuda Song, Zhengyi Luo, Wen Sun, and Kris M. Kitani. 2021a. Transform2Act: Learning a Transform-and-Control Policy for Efficient Agent Design. ArXiv abs/2110.03659 (2021).

Ye Yuan, Shih-En Wei, Tomas Simon, Kris Kitani, and Jason M. Saragih. 2021b. SimPoE: Simulated Character Control for 3D Human Pose Estimation. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021), 7155–7165.