Reinforcement Learning Based Character Controlling

Wang Jingbo
Department of Information Engineering
The Chinese University of Hong Kong
Shatin, Hong Kong
wj020@ie.cuhk.edu.hk

Yin Zijing
Department of Information Engineering
The Chinese University of Hong Kong
Shatin, Hong Kong
yz020@ie.cuhk.edu.hk

Abstract

Character controlling is a longstanding problem in understanding the behavior of human. This task aims to generate various and high quality human motion in the simulated environment as in the real world controlled by human. Therefore, how to use the captured real human motions is the crucial component to solve this problem. Rather than regressing the human motion directly in previous motion prediction methods, in this project, a reinforcement learning method is adopted in to our framework to learn robust control policies capable of imitating a broad range of example motion clips. Besides, we also explore the new reward functions to encourage the motion similarity between the real human and the virtual character. With these new rewards, the algorithm will convergence faster than recent advances. The representation of our project can be found in this **link**.

1 Introduction

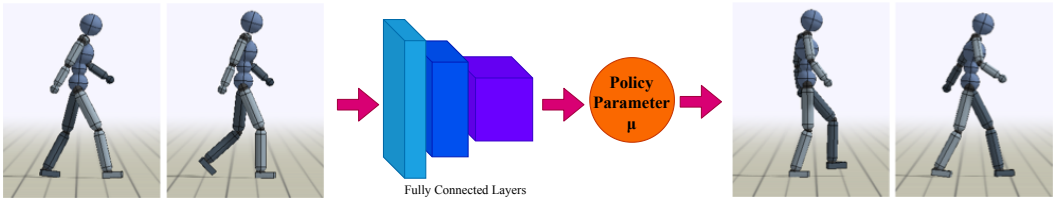

Figure 1: Framework for character controlling as motion mimicking.

Character controlling is widely utilized in many real-world applications, such as computer games, VR/AR, and motion capture. Recent methods always control what the human will do by predicting the future human motion directly, under the observation of previous human motion. However, under these methods, people cannot control the motion of virtual character in an explicit way. And thus, this problem is worth further exploration.

To solve this problem, recently, more researchers begin to model virtual character by mimicking human motions in the real world. Given human motions, the goal of this framework is to learn controllers, which have the ability to control the virtual character do same actions as the real human motions. Thus, directly, reinforcement learning, which been widely used in controlling and computer vision, can be introduced into this frame work (38; 37). Under this framework in Figure 1, the character controlling problem is changed to learn policy parameter of agents with specific predefined action, state and rewards, which can measure the differences between virtual character and real human.

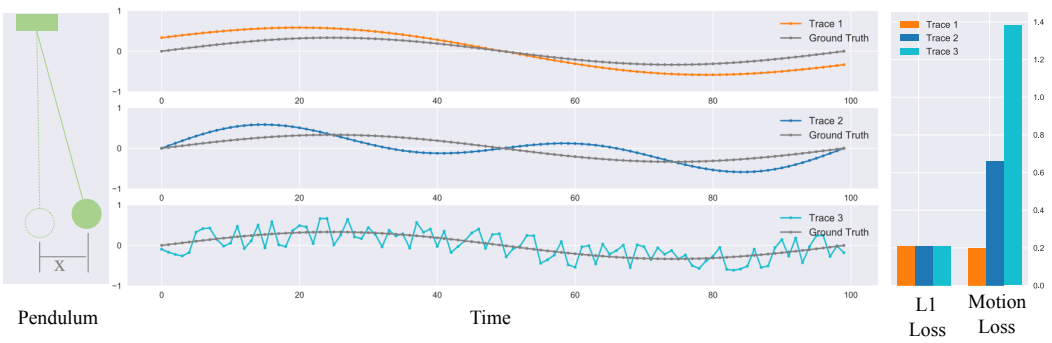

Figure 2: Motion loss is a better description of human motion.

In this project, we follow the problem definition in the well-known framework (37). Firstly, we follow the PPO optimization method which is widely used in reinforcement learning and reproduced their results which are reported in previous projects. Rather than implementing their method directly, we also do the following exploration. We also add new reward functions, which encourage the motion similarity between virtual characters and the real human motion, to speed up the Convergence process of the proposed algorithm.

2 Related Work

Reinforcement Learning Reinforcement learning provides a feasible way for motion imitation, therefore agent can learn from trials and errors to practice human actions. Value iteration can be used in motion synthesis to develop kinematic controllers (26). However, some unnatural behaviors may occur in those manually designed controllers, such as lag and fold-over. Recently, deep neural network models have been introduced to RL for some challenging work (39) (19)(31). Some RL methods based on model-free imitation learning algorithm, such as Generative Adversarial Imitation Learning(17), can obtain significant performance in large and high-dimensional environments.

Motion Imitation Motion imitation has a long history in computer animation area. One early application was designed to walk stably by following the kinematic target trajectory (41). Reference motions can be simulated to produce realistic human locomotion (25).To generate more natural human motions, reference motions have also been used in reward function (38).

Motion Generation Recently, lots of work begin to focus on pose sequence generation. HP-GAN (2) combines the Seq2Seq model to the GAN framework for motion generation. Cai *et al.* (4) propose a Two-Stage GAN to generate the spatial and temporal information respectively for pose generation. PSGAN (47) takes the initial pose as input and action label as the condition to generate pose sequence for video generation. CSGN (45) formulates both generator and discriminator as graph convolution and generates pose sequence from noise sequence directly. Action2Motion (12) generates human pose sequences with a CVAE model for the given action.

Motion Prediction Pose prediction is also another important task to understand human behaviors. For given continuous pose sequences, these models can predict the future human motion at a few time steps. *Encoder-Recurrent-Decoder* (ERD) (9) incorporates encoder and decoder models before and after the recurrent units for motion prediction. Based on the Seq2Seq (42) model, Martinez *et al.* (33) predicts the velocities rather than the positions of joints for motion prediction. Ac-Lstm (29) enhances the capability of LSTM by training the mixture of synthesized frames and observed frames. Graph convolution network (GCN) is also widely used in motion prediction in recent advances (7; 27; 32).

3 Motion Modeling

We illustrate the motion modeling of virtual characters in this section before we define the reinforcement learning framework for this character controlling task. We first define the represent the human

motion sequence and then demonstrate the method to measure the differences between prediction and target human motion.

Human Motion Keypoint sequence (P_t) is the most direct representation of human, but it ignores the rotation of human bodies and can not describe the complex human motion exactly. Thus, in this project, we add the the joint rotation sequence (Q_t) into the human modeling, which models the relative position of different joints directly. This rotation is always represented by quaternion, which is easily to measure the differences of angles.

Motion differences The direct way to measure the differences between prediction and target motion is the point to point loss, such as the distance of keypoints and the joint rotation as following:

$$L_P = \sum_{k=1}^{K} ||\hat{p_t^k} - p_t^k|| \tag{1}$$

$$L_R = \sum_{k=1}^{K} ||\hat{q_t^k} - q_t^k|| \tag{2}$$

which $\hat{p_t^k}$ and $\hat{q_t^k}$ are the predicted keypoints and joint rotation of k joint at t time step.

Although these two function measures the point to point differences, we also argue that this two metrics can not reflect the motion of the human exactly. As shown in Figure 2, we simplify the human motion as the pendulum motion. Although the third motion is significant worse than the first and second motion, the point-to-point loss of them of them is same. We find out the long-term motion can help to judge these motions. As shown in figure 3, we propose our motion encoding as long-term joint movement, which is encoded by the pre-defined operator and the motion difference is defined as following:

$$e_t^k = p_t^k * p_{t-T}^k \tag{3}$$

$$L_M = \sum_{k=1}^{K} ||\hat{e_t^k} - e_t^k|| \tag{4}$$

As shown in figure 2, this motion difference can represent the differences of these motion. Thus, we propose the motion metric to measure this difference and we find out that this metric is beneficial for the convergence of motion mimicking.

4 Experiment Setups

4.1 Formulation

Here we take humanoid model as an example to define states, actions and reward function, which are three major parts in RL problems.

State State s represents the body structure of character model. Different parts of body are described as relative positions with respect to pelvis (which is the origin of coordinate system). Other configurations include its rotations and angular velocities. To be specific, x-axis is the facing direction of pelvis.

Action Action a represents the target orientations for each joint. After sampling, the orientations are sent to PD controllers(43). Then we can get information such as moment of force and input them into physical environment.

Reward The origin reward is divided into two parts, namely r_t^I and r_t^G. r_t^G is designed for specific tasks such as going towards target orientations and hitting the target. r_t^I encourages character to imitate reference motions and evaluates the disparity between reference data and imitations.Both of the two rewards have a weight, then we can get the total reward by adding them up.

$$r_t = w^I r_t^I + w^G r_t^G \tag{5}$$

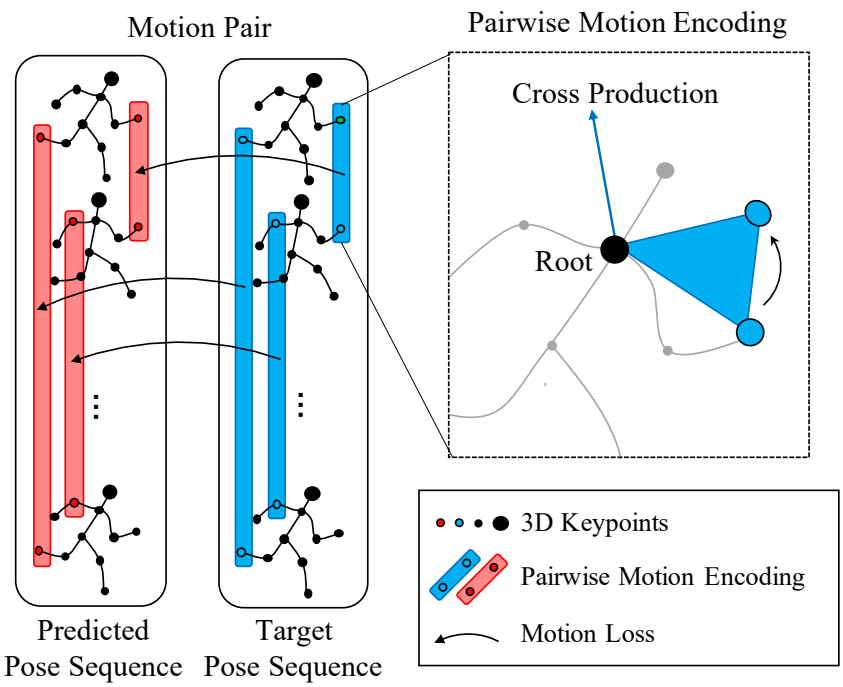

Figure 3: Illustrating of motion encoding to measure the difference between motion prediction and ground-truth.

Here, r_t^I can be further decomposed as four parts, which represent pose, velocity, end-effector and center-of-mass respectively. At each step t, r_t^I calculates imitation reward according to these parameters.

$$r_t^I = w^p r_t^p + w^v r_t^v + w^e r_t^e + w^c r_t^c \tag{6}$$

In these rewards, the formulation of r_t^p and r_t^e are same as L_R and L_P respectively.

Besides, we also adopt the motion metric as the new reward function when the time step is larger than the predefined motion range T. Thus, this reward function can be written as following:

$$r_t^I = w^p r_t^p + w^v r_t^v + w^e r_t^e + w^c r_t^c + \mathbf{1}(\mathbf{t} >= \mathbf{T}) w^m r_t^m \tag{7}$$

Although r_v is in this function, we argue that the effectiveness of the short-term motion is minor for speeding up convergence of this framework. We will evaluate the effectiveness of the new reward function can help the reinforcement in the following experiments.

4.2 Training algorithm

Policy gradient methods are fundamental in deep neural networks and reinforcement learning, but they are sensitive to step size and sometimes have poor efficiency. To solve these problems, Proximal Policy Optimization (PPO) was proposed as a simple but useful off-policy algorithm to train an agent. It has a new objective function to achieve small batch updates in multiple training steps, which can minimize the cost function and reduce the deviation from previous policy. In this algorithm, $r_t(\theta)$ denotes the probability ratio under the new and old policies, which is represented as follows:

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \tag{8}$$

In order to make the training more stable, the advantage function will be clipped if the probability ratio between the new and old policies. Then the final objective function can be represented as follows:

$$L_t(\theta) = min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t) \tag{9}$$

Where $\hat{A}_t$ is the estimated advantage at time t, and ϵ is usually set to 0.2.

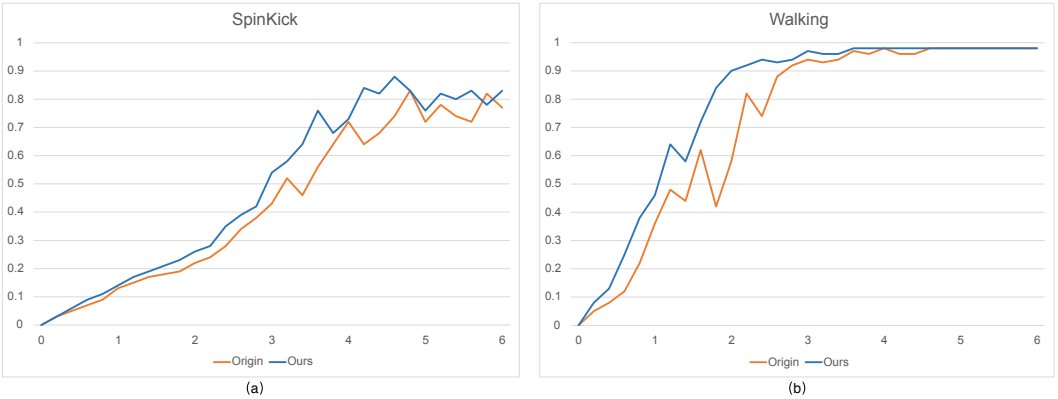

Figure 4: **Visualization result of learning curve** . The orange curves represent the learning process with origin reward function and the blue curves are for the reward with motion metric. Our method speeds up the convergence of the previous proposed framework.

5 Experiments

This section, we first illustrate the experiment settings of our project. Then, we evaluate our method can speed up the convergence of the previous reinforcement learning framework. At last, we will demonstrate several visualization results of the motion mimicking framework.

Experiment Settings We conduct all experiments based on *Tensorflow Toolbox* and the visualization results are based on *OpenGL*. All the reference human motions are provided by the DeepMimic Project (37) and we only focus on the mimicking the single motion of human character. Especially, in our project, we train our model on the hard action "SpinKick"and the easy action"Walking" for convenience to evaluate that our new reward function can help convergence of this framework. Besides, as (37), we add the negative exponential function to all motion metrics and follows the weights of these rewards.

Experiments Results We demonstrate the learning curves of "SpinKick" and "Walking" actions under the origin and proposed reward functions in Figure 4. Although both rewards can make the framework converge to a stable situation, our reward function can help this system converge faster than the system ignored long-term motion metric. Especially for the easy action " Walking", which just contains several simple motion patterns, the framework achieves the stable solution faster than the origin one significantly.

Quantitative Results At last, we will show how this model works by visualizing the action of controlled virtual character. For convenience, we just visualized the hard action "SpinKick" in the following figure 5. This figure show the "SpinKick" action controlled by the trained model and we think this action is very similar to the human action in the real world.

6 Conclusion

Character controlling is a difficult and long standing problem for computer vision and reinforcement learning. And the reward function is crucial for this system to learn accurate human motion efficiently. In this project, we first analyze the metric to measure the differences between the motion prediction and ground-truth. Then, based on the reproduced motion mimicking framework under the reinforcement learning. We find out that the motion reward can speed up the convergence of the complex system. At last, we visualize the final results of the virtual character who can do different actions as the reference humans. We believe that with more accurate motion description and efficient reinforcement learning framework, we model the virtual do more feasible, complex, and challenging actions as human.

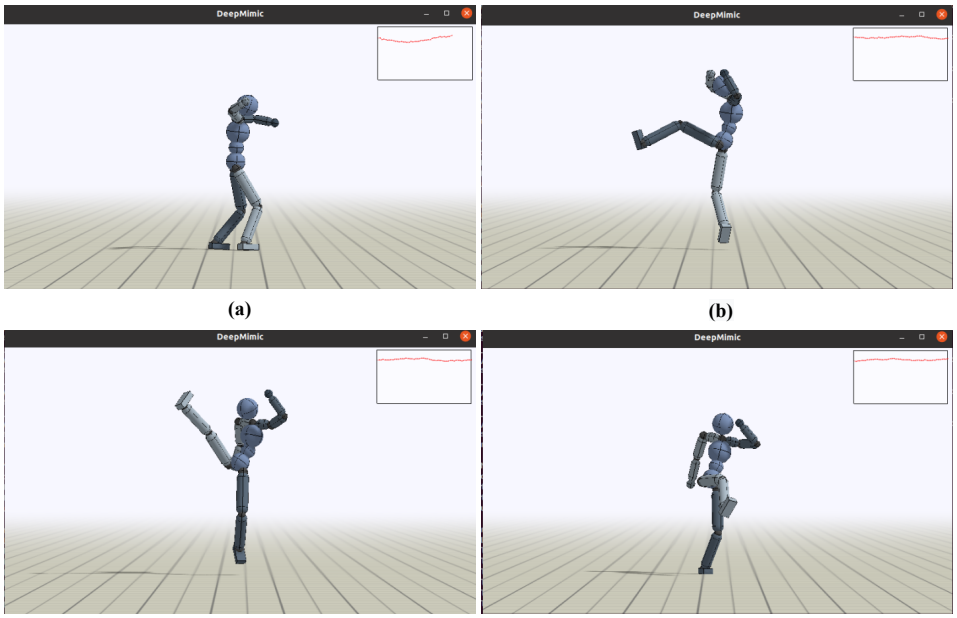

Figure 5: **Visualization result**. This reinforcement learning framework can control the virtual character do hard action similar with the human in real worlds.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] E. Barsoum, J. Kender, and Z. Liu. Hp-gan: Probabilistic 3d human motion prediction via gan. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 1418–1427, 2018.

[3] J. Butepage, M. J. Black, D. Kragic, and H. Kjellstrom. Deep representation learning for human motion prediction and classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6158–6166, 2017.

[4] H. Cai, C. Bai, Y.-W. Tai, and C.-K. Tang. Deep video generation, prediction and completion of human action sequences. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 366–382, 2018.

[5] Z. Cao, H. Gao, K. Mangalam, Q.-Z. Cai, M. Vo, and J. Malik. Long-term human motion prediction with scene context. In *ECCV*, 2020.

[6] N. Chentanez, M. Müller, M. Macklin, V. Makoviychuk, and S. Jeschke. Physics-based motion capture imitation with deep reinforcement learning. In *Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games*, pages 1–10, 2018.

[7] Q. Cui, H. Sun, and F. Yang. Learning dynamic relationships for 3d human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[9] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4346–4354, 2015.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[11] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

[12] C. Guo, X. Zuo, S. Wang, S. Zou, Q. Sun, A. Deng, M. Gong, and L. Cheng. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20)*, 2020.

[13] M. Hassan, V. Choutas, D. Tzionas, and M. J. Black. Resolving 3D human pose ambiguities with 3D scene constraints. In *International Conference on Computer Vision*, pages 2282–2292, Oct. 2019.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[16] N. Heess, D. TB, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, S. Eslami, et al. Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*, 2017.

[17] J. Ho and S. Ermon. Generative adversarial imitation learning. In *Advances in neural information processing systems*, pages 4565–4573, 2016.

[18] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[19] D. Holden, T. Komura, and J. Saito. Phase-functioned neural networks for character control. *ACM Transactions on Graphics (TOG)*, 36(4):1–13, 2017.

[20] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[21] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[22] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. 2014.

[23] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[24] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth international conference on 3D vision (3DV)*, pages 239–248. IEEE, 2016.

[25] Y. Lee, S. Kim, and J. Lee. Data-driven biped control. In *ACM SIGGRAPH 2010 papers*, pages 1–8. 2010.

[26] Y. Lee, K. Wampler, G. Bernstein, J. Popović, and Z. Popović. Motion fields for interactive character locomotion. In *ACM SIGGRAPH Asia 2010 papers*, pages 1–8. 2010.

[27] M. Li, S. Chen, Y. Zhao, Y. Zhang, Y. Wang, and Q. Tian. Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[28] X. Li, S. Liu, K. Kim, X. Wang, M.-H. Yang, and J. Kautz. Putting humans in a scene: Learning affordance in 3d indoor environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12368–12376, 2019.

[29] Z. Li, Y. Zhou, S. Xiao, C. He, Z. Huang, and H. Li. Auto-conditioned recurrent networks for extended complex human motion synthesis. *arXiv preprint arXiv:1707.05363*, 2017.

[30] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[31] L. Liu and J. Hodgins. Learning to schedule control fragments for physics-based characters using deep q-learning. *ACM Transactions on Graphics (TOG)*, 36(3):1–14, 2017.

[32] W. Mao, M. Liu, M. Salzmann, and H. Li. Learning trajectory dependencies for human motion prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9489–9497, 2019.

[33] J. Martinez, M. J. Black, and J. Romero. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2891–2900, 2017.

[34] J. Merel, Y. Tassa, D. TB, S. Srinivasan, J. Lemmon, Z. Wang, G. Wayne, and N. Heess. Learning human behaviors from motion capture by adversarial imitation. *arXiv preprint arXiv:1707.02201*, 2017.

[35] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019.

[36] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. A. Osman, D. Tzionas, and M. J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[37] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018.

[38] X. B. Peng, G. Berseth, K. Yin, and M. Van De Panne. Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Transactions on Graphics (TOG)*, 36(4):1–13, 2017.

[39] A. Rajeswaran, S. Ghotra, B. Ravindran, and S. Levine. Epopt: Learning robust neural network policies using model ensembles. *arXiv preprint arXiv:1610.01283*, 2016.

[40] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[41] D. Sharon and M. van de Panne. Synthesis of controllers for stylized planar bipedal walking. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2387–2392. IEEE, 2005.

[42] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[43] J. Tan, K. Liu, and G. Turk. Stable proportional-derivative controllers. *IEEE Computer Graphics and Applications*, 31(4):34–44, 2011.

[44] J. Wang, S. Yan, Y. Xiong, and D. Lin. Motion guided 3d pose estimation from videos. *arXiv preprint arXiv:2004.13985*, 2020.

[45] S. Yan, Z. Li, Y. Xiong, H. Yan, and D. Lin. Convolutional sequence generation for skeleton-based action synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4394–4402, 2019.

[46] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018.
[47] C. Yang, Z. Wang, X. Zhu, C. Huang, J. Shi, and D. Lin. Pose guided human video generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 201–216, 2018.
[48] Y. Ye and C. K. Liu. Synthesis of responsive motion using a dynamic model. In *Computer Graphics Forum*, volume 29, pages 555–562. Wiley Online Library, 2010.
[49] S. Zhang, Y. Zhang, Q. Ma, M. J. Black, and S. Tang. PLACE: Proximity learning of articulation and contact in 3D environments. In *International Conference on 3D Vision (3DV)*, Nov. 2020.
[50] Y. Zhang, M. Hassan, H. Neumann, M. J. Black, and S. Tang. Generating 3d people in scenes without people. In *Computer Vision and Pattern Recognition (CVPR)*, pages 6194–6204, June 2020.