Table 2. The average movement in the body chain when deactivating each local motion phase in degrees/frame (Left Foot, Right Foot, Left Hand, Right Hand) and m/s (Ball).

Movement	LF	RF	LH	RH	Ball
All Phases	399	391	169	306	3.89
$\mathcal{P}$ w/o LF	323	320	150	273	3.52
$\mathcal{P}$ w/o RF	326	325	268	156	3.49
$\mathcal{P}$ w/o LH	387	378	171	304	3.54
$\mathcal{P}$ w/o RH	390	387	296	169	3.72
$\mathcal{P}$ w/o Ball	363	358	163	283	3.56
No Phases	21	22	15	21	0.86

9.3 Generative Variations and Generalization

We show examples where we can vary the movements of the characters by sampling noise (see Fig. 15). The variation in the control signals produced by the noise sampling results in different body and ball movements during holding, dribbling and interactions.

Our system can also synthesize novel combination of movements that are not observed in the training set. For example, the dataset contains the motion where the ball is dribbled behind the back while moving forward and at the same position, but not while walking backward or side-stepping (see Fig. 16, left). Our system can learn these behaviors and combine them, allowing the character to dribble behind the back while walking backward and side-stepping. Similarly, the dataset does not contain data where the player holds the ball and run, but our system can produce such movements by providing such a control signal (see Fig. 16, right).

9.4 Character Interaction

We simulate a one-on-one basketball play by letting two players freely control their own character using their gamepads (see Fig. 1). Our system can automatically trigger realistic behaviors: the dribbling character points its free arm towards the opponent character to block it from approaching. The defense character standing in the way automatically spreads its arms to stop the dribbling character from moving forward. These response motions are learned from the training data where the players are interacting with each other. Note that character movements during the interaction sometimes appear confused or unrealistic due to the uncommon trajectories produced by the user. The space of spatial relations between the characters is very large and the training data only covers a small subset of such space. The full character trajectory control (2D translation + 1D translation) is also rather complex and it is not always easy for the user to control the character to respond to the other character in a realistic manner. A higher-level controller that outputs a control signals or partially blend the user control signals to them could potentially improve the realism in path generation.

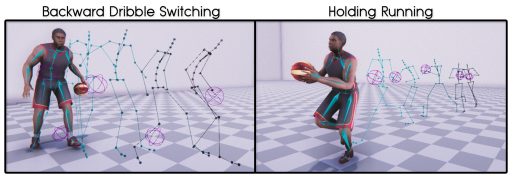

Fig. 16. Generalization to uncaptured movement combinations.

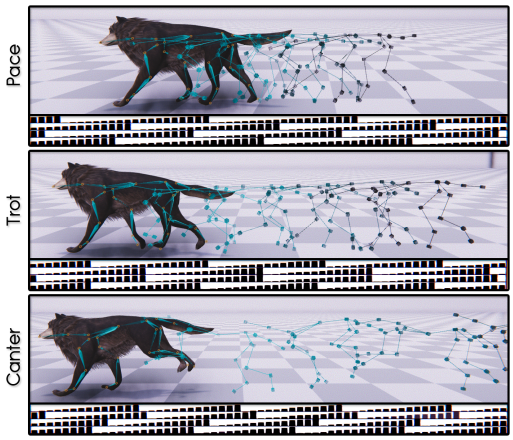

Fig. 17. The quadruped locomotion generated by our system: changing the target velocity will produce different local motion phase patterns, resulting in different locomotion modes.

9.5 Evaluating the Influence of Local Motion Phases

Using a forward dribbling motion as an example, we evaluate the influence of the local motion phases to the motion by deactivating each of them and observing the effect. The average updates are measured in degrees/frame for each selective bone chain, and in m/s for the ball movement. The results are shown in Table 2. If either left or right foot phase is deactivated, the other can slightly compensate for the inactive one. If the left or right hand is deactivated, the network automatically starts dribbling with the other hand that is active. The ball movement shows higher redundancy in connection with other phases. When all phases are turned off the character is completely stiff and slides.

9.6 Quadruped Motion Learning

We also train a quadruped controller using the local motion phase computed for every foot based on its contact with the ground. We use the same architecture and control system as in MANN [Zhang et al. 2018] but only switch the input to the gating network from the feet velocity to the local motion phases. This greatly helps to improve the response and generality of the control. As observed in Fig. 17, different locomotion modes can be synthesized by simply changing the target velocity of the character. For the sitting to walk, the MANN has rather slow response from switching from sitting to walking, where our system immediately starts to walk given the user instruction. In terms of generalization, our model can synthesize movements that are not observed during training, such as side-stepping and quick turning (see supplementary video) while MANN will simply slide the body without stepping for such unobserved control signals.

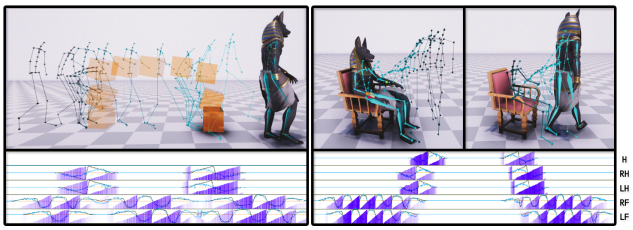

Fig. 18. Snapshots of carrying a box and sitting down/standing up from a chair and the corresponding local motion phases.

9.7 Object Interaction Learning

We apply the local motion phase to the object/environment interaction examples in [Starke et al. 2019]. The snapshots of picking up and putting down a box and sitting on and standing from a chair as well as the corresponding local motion phases are shown in Fig. 18. Our system can stably produce realistic movements automatically, while [Starke et al. 2019] requires the motion to be aligned via a global phase that is carefully handcrafted per action type.

10 DISCUSSION

Our system improves on the evolution of expert-based character controllers [Holden et al. 2017; Starke et al. 2019; Zhang et al. 2018]; a common question refers to what enables the previous networks to perform well on their given tasks, and why they cannot directly be used for each other domain, i.e. biped, quadruped, interaction. What all those models have in common is that they aim to segment and dynamically interpolate the movements of the same class and timing using jointly trained expert networks.

The PFNN [Holden et al. 2017] uses a global phase variable to compute the blending weights of the experts for animating biped locomotion. This is possible because the locomotion always follows the same left-right foot-fall pattern. On the other hand, defining a single phase variable is not possible for quadruped locomotion due to inconsistency of the foot-fall pattern at different speed.

In MANN [Zhang et al. 2018], the handcrafted phase function is replaced with a gating network that uses the foot velocities as features. The foot velocities naturally produce a well-separating feature space to segment the movements depending on the locomotion mode gaits - which unfortunately is not the case for regular biped locomotion.

The NSM [Starke et al. 2019] utilizes the gating structure developed in MANN to learn various types of interaction motions such as sitting on chairs, opening doors, carrying objects under a single structure that enables to define a single global phase separately for each task. The NSM generalizes the PFNN by expanding its initial scope from pure locomotion to different types of scene interaction motions. However, labeling and defining rules for each motion phase as well as requiring structured motion capture data of synchronized movements remains an issue.

With our enhancement of using the local motion phase, we can

- expand the scope of movements to fast-paced, asynchronous interactions with dynamic objects and other characters,
- overcome limitations of global phases in order to handle unstructured biped locomotion,
- improve quadruped motion in generating more detailed and responsive movements and
- produce variations within same actions using a generative control scheme.

Alternative Design of the Generative Control Scheme. An alternative structure for the generative control scheme could be cGAN [Mirza and Osindero 2014], for example as applied to add photo-realistic details to images [Ledig et al. 2017], or stylized details to terrains [Zhao et al. 2019]; we experimented with such a structure but found this is not straightforward. For such a design, we need to define a condition extracted from the ground truth control signal. If we use a filtered, smooth control signal as a condition (as a smoothed terrain is used as a condition in [Zhao et al. 2019]), during the training we are making an assumption that the control signals produced by the user control has similar properties to those produced by the smoothing filter. This is a very strong assumption and we find this is not necessarily the case. A better way to extract the condition is that we use the actual control signals from the user to train the cGAN on. But capturing those control signals from the user, and then mapping them to the corresponding motion is not straightforward and may require manual and labor intensive processes. Our design to construct a latent space of the control signal in an autoencoder fashion and adding noise at runtime can effectively produce a large variation of movements observed during training.

Limitations. The system will fail when the control signal given to the network is far from what have been observed during training. For example, when the user moves the body and the ball in random directions using the two joysticks, a very random control vector can be produced, and the generative control model will have difficulty in projecting it to a realistic control signal, resulting in penetrations etc. The unobserved control signal can mislead the output of the motion prediction network. Fixing this will require either preparing a generative control model that is trained with noise values at the level of such random user control, or preparing a logic that filters unrealistic control signals.

11 CONCLUSION AND FUTURE WORK

We propose a novel feature called the local motion phase, which together with our mixture-of-experts architecture can learn fast and dynamic movements during basketball play. The features are extracted by a hybrid global and local optimization technique from unstructured motion capture data. We also propose a novel generative control model that can produce a wide variation of movements from abstract control signals. Fast and complex interactions between the character and the ball, or with another character can be synthesized in real-time after training the system with motion capture data of basketball plays.

One of the future work is that currently the local phase is defined based on contacts: extending this to non-contact movements is the next step. Another issue is the long turnaround time after making changes: current character control models need to be retrained from scratch when the control signal is readjusted. Learning the pure

motion manifold and the latent mapping from a decoupled control system will be interesting to significantly reduce the turnaround time. Finally, it will be interesting to extend our framework for physically-based animation by combining it with physically-based motion tracking methods [Bergamin et al. 2019; Hong et al. 2019; Park et al. 2019].

ACKNOWLEDGMENTS

We thank Fabio Zinno for his help acquiring the basketball dataset and the fruitful discussions on this research, as well as Matteo Loddo for skinning the character model. We further want to thank Mohsen Sardari, Harold Chaput, Daniel Guinn and Navid Aghdaie for the various support throughout this project. Finally, we also wish to thank the anonymous reviewers for their constructive comments.

REFERENCES

Okan Arikan and David A Forsyth. 2002. Interactive motion generation from examples. *ACM Trans on Graph* 21, 3 (2002), 483–490. https://doi.org/10.1145/566654.566606

Kevin Bergamin, Simon Clavet, Daniel Holden, and James Richard Forbes. 2019. DReCon: data-driven responsive control of physics-based characters. *ACM Trans on Graph* 38, 6 (2019), 206. https://doi.org/10.1145/3355089.3356536

Simon Clavet. 2016. Motion matching and the road to next-gen animation. In *Proc. of GDC*.

Stelian Coros, Philippe Beaudoin, and Michiel Van de Panne. 2010. Generalized biped walking control. *ACM Trans on Graph* 29, 4 (2010), 130. https://doi.org/10.1145/1778765.1781156

Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. 2015. Recurrent network models for human dynamics. In *Proc. ICCV*. 4346–4354. https://doi.org/10.1109/ICCV.2015.494

Félix G. Harvey and Christopher Pal. 2018. Recurrent Transition Networks for Character Locomotion. http://arxiv.org/abs/1810.02363

Rachel Heck and Michael Gleicher. 2007. Parametric motion graphs. In *Proc. I3D*. 129–136. https://doi.org/10.1145/1230100.1230123

Jessica K Hodgins, Wayne L Wooten, David C Brogan, and James F O'Brien. 1995. Animating human athletics. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*. 71–78. https://doi.org/10.1145/218380.218414

Daniel Holden, Taku Komura, and Jun Saito. 2017. Phase-functioned neural networks for character control. *ACM Trans on Graph* 36, 4 (2017), 42. https://doi.org/10.1145/3072959.3073663

Seokpyo Hong, Daseong Han, Kyungmin Cho, Joseph S Shin, and Junyong Noh. 2019. Physics-based full-body soccer motion control for dribbling and shooting. *ACM Trans on Graph* 38, 4 (2019), 1–12. https://doi.org/10.1145/3306346.3322963

Lucas Kovar and Michael Gleicher. 2004. Automated Extraction and Parameterization of Motions in Large Data Sets. *ACM Trans on Graph* 23, 3 (2004), 559–568. https://doi.org/10.1145/1186562.1015760

Lucas Kovar, Michael Gleicher, and Frédéric Pighin. 2002. Motion graphs. *ACM Trans on Graph* 21, 3 (2002), 473–482. https://dl.acm.org/doi/10.1145/566654.566605

Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *Proc. IEEE CVPR*. 4681–4690. https://ieeexplore.ieee.org/abstract/document/8099502

Jehee Lee, Jinxiang Chai, Paul SA Reitsma, Jessica K Hodgins, and Nancy S Pollard. 2002. Interactive control of avatars animated with human motion data. *ACM Trans on Graph* 21, 3 (2002), 491–500. https://doi.org/10.1145/566654.566607

Kyungho Lee, Seyoung Lee, and Jehee Lee. 2018. Interactive Character Animation by Learning Multi-objective Control. *ACM Trans on Graph* 37, 6 (2018). https://doi.org/10.1145/3272127.3275071

Zimo Li, Yi Zhou, Shuangjiu Xiao, Chong He, Zeng Huang, and Hao Li. 2017. Auto-conditioned recurrent networks for extended complex human motion synthesis. *arXiv preprint arXiv:1707.05363* (2017). http://arxiv.org/abs/1707.05363

C Karen Liu and Zoran Popović. 2002. Synthesis of complex dynamic character motion from simple animations. *ACM Trans on Graph* 21, 3 (2002), 408–416. https://doi.org/10.1145/566654.566596

Libin Liu and Jessica Hodgins. 2018. Learning basketball dribbling skills using trajectory optimization and deep reinforcement learning. *ACM Trans on Graph* 37, 4 (2018), 142. https://dl.acm.org/doi/10.1145/3197517.3201315

Libin Liu, Michiel Van De Panne, and KangKang Yin. 2016. Guided learning of control graphs for physics-based characters. *ACM Trans on Graph* 35, 3 (2016), 29. https://doi.org/10.1145/2893476

Libin Liu, KangKang Yin, Michiel van de Panne, Tianjia Shao, and Weiwei Xu. 2010. Sampling-based contact-rich motion control. *ACM Trans on Graph* 29, 4 (2010), 128. https://doi.org/10.1145/1778765.1778865

Jianyuan Min and Jinxiang Chai. 2012. Motion graphs++: a compact generative model for semantic motion analysis and synthesis. *ACM Trans on Graph* 31, 6 (2012), 153. https://doi.org/10.1145/2366145.2366172

Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014). http://arxiv.org/abs/1411.1784

Tomohiko Mukai and Shigeru Kuriyama. 2005. Geostatistical motion interpolation. *ACM Trans on Graph* 24, 3 (2005). http://doi.acm.org/10.1145/1073204.1073313

Soohwan Park, Hoseok Ryu, Seyoung Lee, Sunmin Lee, and Jehee Lee. 2019. Learning predict-and-simulate policies from unorganized human motion data. *ACM Trans on Graph* 38, 6 (2019), 205. https://doi.org/10.1145/3355089.3356501

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. 2018. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Trans on Graph* 37, 4 (2018), 143. https://doi.org/10.1145/3197517.3201311

Marc H Raibert and Jessica K Hodgins. 1991. Animation of dynamic legged locomotion. In *ACM Siggraph Computer Graphics*, Vol. 25. ACM, 349–358. https://doi.org/10.1145/127719.122755

Charles Rose, Michael F. Cohen, and Bobby Bodenheimer. 1998. Verbs and Adverbs: Multidimensional Motion Interpolation. *IEEE Computer Graphics and Applications* 18, 5 (1998), 32–40. http://dx.doi.org/10.1109/38.708559

Charles F Rose III, Peter-Pike J Sloan, and Michael F Cohen. 2001. Artist-Directed Inverse-Kinematics Using Radial Basis Function Interpolation. *Computer Graphics Forum* 20, 3 (2001), 239–250. https://doi.org/10.1111/1467-8659.00516

Alla Safonova and Jessica K Hodgins. 2007. Construction and optimal search of interpolated motion graphs. *ACM Trans on Graph* 26, 3 (2007). https://doi.org/10.1145/1276377.1276510

Hyun Joon Shin and Hyun Seok Oh. 2006. Fat graphs: constructing an interactive character with continuous controls. In *Proc. SCA*. 291–298. http://dx.doi.org/10.2312/SCA/SCA06/291-298

Sebastian Starke, Norman Hendrich, and Jianwei Zhang. 2018. Memetic Evolution for Generic Full-Body Inverse Kinematics in Robotics and Animation. *IEEE Transactions on Evolutionary Computation* 23, 3 (2018), 406–420.

Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. 2019. Neural state machine for character-scene interactions. *ACM Trans on Graph* 38, 6 (2019), 209. https://doi.org/10.1145/3355089.3356505

Graham W Taylor, Geoffrey E Hinton, and Sam T Roweis. 2007. Modeling human motion using binary latent variables. In *Advances in neural information processing systems*. 1345–1352. https://papers.nips.cc/paper/3078-modeling-human-motion-using-binary-latent-variables

Emanuel Todorov and Weiwei Li. 2005. A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *Proceedings of the 2005, American Control Conference, 2005.* IEEE, 300–306.

Ruben Villegas, Jimei Yang, Duygu Ceylan, and Honglak Lee. 2018. Neural Kinematic Networks for Unsupervised Motion Retargetting. In *Proc. IEEE CVPR*. https://doi.org/10.1109/CVPR.2018.00901

J.M. Wang, D.J. Fleet, and A. Hertzmann. 2008. Gaussian Process Dynamical Models for Human Motion. *IEEE PAMI* 30, 2 (Feb 2008), 283–298. https://doi.org/10.1109/TPAMI.2007.1167

Douglas J Wiley and James K Hahn. 1997. Interpolation synthesis of articulated figure motion. *IEEE Computer Graphics and Applications* 17, 6 (1997), 39–45. https://doi.org/110.1109/38.626968

Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation* 1, 2 (1989), 270–280. https://doi.org/10.1162/neco.1989.1.2.270

Andrew Witkin and Michael Kass. 1988. Spacetime constraints. *ACM Siggraph Computer Graphics* 22, 4 (1988), 159–168. https://doi.org/10.1145/54852.378507

Shihong Xia, Congyi Wang, Jinxiang Chai, and Jessica Hodgins. 2015. Realtime style transfer for unlabeled heterogeneous human motion. *ACM Trans on Graph* 34, 4 (2015), 119. https://doi.org/10.1145/2766999

Yuting Ye and C Karen Liu. 2012. Synthesis of detailed hand manipulations using contact sampling. *ACM Trans on Graph* 31, 4 (2012). https://doi.org/10.1145/2185520.2185537

KangKang Yin, Kevin Loken, and Michiel Van de Panne. 2007. Simbicon: Simple biped locomotion control. *ACM Trans on Graph* 26, 3 (2007), 105. https://doi.org/10.1145/1276377.1276509

He Zhang, Sebastian Starke, Taku Komura, and Jun Saito. 2018. Mode-adaptive neural networks for quadruped motion control. *ACM Trans on Graph* 37, 4 (2018). https://doi.org/10.1145/3197517.3201366

Wenping Zhao, Jianjie Zhang, Jianyuan Min, and Jinxiang Chai. 2013. Robust realtime physics-based motion control for human grasping. *ACM Trans on Graph* 32, 6 (2013). https://doi.org/10.1145/2508363.2508412

Yiwei Zhao, Han Liu, Igor Borovikov, Ahmad Beirami, Maziar Sanjabi, and Kazi Zaman. 2019. Multi-theme generative adversarial terrain amplification. *ACM Trans on Graph* 38, 6 (2019), 200. https://doi.org/10.1145/3355089.3356553

A AUTOMATED DATA ANNOTATION

In this section, we describe the process of extracting all state variables from the data that is required for training. To reduce manual labor and avoid data inconsistency due to labeling errors, we design an automated workflow for feature extraction. All motion data processing is performed in the Unity engine. All features live in a time series window $\mathcal{T}_{-1s}^{1s}$ that can cover information of up to 13 uniformly-sampled points within 1s (6 samples) in the past and future around 1 centered root sample at current frame i.

A.1 Extracting Control Variables

To synthesize a wide variation of locomotion and dribbling movements of different speeds, facing directions and interactive maneuvers, we extract three main control channels from the data: trajectory path $\mathbf{T}_i$, action labels $\mathbf{A}_i$ and interaction vectors $\mathbf{I}_i$. All control variables are extracted for each time series sample within $\mathcal{T}_{-1s}^{1s}$.

The trajectory $\mathbf{T}$ is computed by projecting the hip bone and applying a Gaussian kernel $\mathcal{G}$ on the root rotation to prevent overfitting to the data, similar to [Holden et al. 2017]. In addition, since basketball plays cover fast movements such as spiral turns and quick ball switching maneuvers that can happen in a very short time window, we develop an adaptive filter that can take into account such quick movements:

$$r = \mathcal{G}(w_R \mathbf{R}_{\mathcal{T}}, \sigma) \tag{14}$$

where r is the smoothed root orientation, $\mathbf{R}_{\mathcal{T}}$ is the global orientation of the root around the vertical axis, σ is the standard deviation set to $\frac{1}{4}\mathcal{T}_{-1s}^{1s}$ and w_R is a weighting computed by applying another Gaussian filter to $\dot{\mathbf{R}}_{\mathcal{T}}$:

$$w_r = \mathcal{G}(\dot{\mathbf{R}}_{\mathcal{T}}, \sigma). \tag{15}$$

This has a nice effect where movements of the same gradient will produce a more narrow filter width and keep their important content, and movements of oscillating or noisy gradient will sum up to zero and produce a wider filter width to remove such artefacts.

The action labels $\mathbf{A}$ describes whether the character is in the state of *Stand* or *Move*, and in the *Dribble*, *Hold* or *Shoot* state with respect to the ball. To compute the labels for *Stand* or *Move*, we compute the root velocity magnitude and transform it into a continuous value between 0 and 1 using a smooth-step function with threshold 0.5m/s. Thus, the sum of both of them always equals 1. For the *Dribble* label, we detect contact labels within the time series window to extract whether the character is interacting with the ball or not. Similarly, the *Hold* label is set active when both hands are in contact with the ball. To extract Shoot labels, we determine whether the ball is leaving contact with the hands and is above a certain height and velocity threshold.

The interaction vector features $\mathbf{I}$ are used to direct the ball control, such as the ball dribbling property, i.e. location, speed and height, controlling switching and special turning maneuvers, and also the ball position while holding. At every time series sample, the 3D vector pair $\mathbf{I}_i^p$ and $\mathbf{I}_i^m$, are computed by:

$$\mathbf{I}_i^p = (\hat{x}_i, h_i, \hat{z}_i) \qquad \mathbf{I}_i^m = (\frac{d\hat{x}_i}{dt}, \left\| \frac{dh_i}{dt} \right\|, \frac{d\hat{z}_i}{dt}) \tag{16}$$

where $(\hat{x}_i, \hat{z}_i)$ is the horizontally-normalized vector around the character root, and h_i the desired vertical height, and $\mathbf{I}_i^m$ is computed by their temporal finite difference. To process the horizontal control vectors, we also apply the same gradient-based Gaussian filtering as for the computation of the root trajectory in Eq. (14):

$$h = \mathcal{G}(w_H \mathbf{H}_{\mathcal{T}}, \sigma), \tag{17}$$

where h is the output height, $\mathbf{H}_{\mathcal{T}}$ is the original height curve, σ is the standard deviation in Eq. (14), and w_H is a weighting given by

$$w_H = \mathcal{G}(\mathbf{C}_{\mathcal{T}}, \sigma) \tag{18}$$

where $\mathbf{C}_{\mathcal{T}}$ is a block function representing the contact between the hand and the ball, which is obtained by the method described in Section A.2. This setting allows continuous control of the character between different actions as well as switching between offense and defense during runtime.

A.2 Extracting Ball Movements

The position and velocity of the ball is extracted from the motion capture data. However, there are numerous clips where the ball movement leaves the view of the optical motion capture, or when performing regular locomotion and not interacting with the ball, in which case the ball data is either missing or invalid. In such case, we clamp or project the ball to a position on a capsule surface around the character. More specifically, the ball coordinates within a radius of $r = 2.5$ around the character root are learned as:

$$\hat{\mathbf{B}} = w_b \mathbf{B} \tag{19}$$

where $\mathbf{B}$ are the parameters of the ball, including its position, velocity, upward vector and forward vector of the ball in the root space, and w_b is a weighting computed by $w_b = 1.0 - \|\mathbf{b}_p\|/r$ where $\mathbf{b}_p$ is the horizontal position of the ball in the root space. During runtime, computing the actual ball parameters can be done by dividing $\hat{\mathbf{B}}$ by the predicted w_b. This transforms the features within a reasonable distribution around the character and helps the network to better learn the ball movement. We wish the network to only learn movements of the ball in which range the character can actually interact with it.

A.3 Detecting Contact Labels

The required contact labels are automatically obtained by evaluating a primary condition for distance and secondary condition for velocity as denoted in Eq. (20). First, we perform collision checks within a specified distance threshold around the character bone. If this holds true, we check another threshold for the maximum velocity difference of the two contacting colliders to filter wrongly extracted and sliding contacts. This can be denoted as

$$c_i^k = \begin{cases} 1 & \text{if} \quad ||\mathbf{p}_i^k - \mathbf{p}^{\cdot}|| \leq d_{max} \wedge ||\mathbf{v}_i^k - \mathbf{v}^{\cdot}|| \leq v_{max} \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

where $\mathbf{p}_i^k, \mathbf{p}^{\cdot}$ and $\mathbf{v}_i^k, \mathbf{v}^{\cdot}$ are the position and velocity of bone k at frame i and those of another collider object $(\cdot)$ and d_{max}, v_{max} are threshold parameters (d_{max} adjusted according to the collider objects and $v_{max} = 1$).