Character Motion Synthesis Based on Deep Learning: A Survey

Anjian Chen

Faculty of Arts & Science, University of Toronto, Toronto, Ontario, Canada

Ansonchen927@outlook.com

Abstract. Character motion synthesis can be more cost-effective, flexible, and time-efficient compared to motion capture or traditional animation. As character motion synthesis regards multiple major industries, along with the development in deep learning techniques, character motion synthesis based on deep learning conspicuously receives substantial attention, resulting in numerous related studies that need to be analyzed and summarized. This paper presents an overview of character motion synthesis based on deep learning. Firstly, it epitomizes methods incorporating different types of neural networks, then encapsulates methods that did not utilize neural networks but simply deep learning, such as deep reinforcement learning, and lastly summarizes and evaluates the advantages and limitations of different deep learning methods on character motion synthesis.

Keywords: Character motion synthesis, deep learning, neural networks, character animation, recurrent neural networks, convolutional neural networks, generative adversarial networks, autoencoders, deep reinforcement learning.

1. Introduction

Character motion synthesis, the process of generating or simulating realistic character movements, can be applied to several fields, including animation and film industry, video games, robotics, biomechanics, augmented and virtual reality, etc. As the vast and rapid development in computer technology, along with the general public desiring better motion qualities, character motion synthesis quickly becomes one of the hottest topics in the field of computer graphics, resulting in the emergence, design, implementation, and testing for various methods in the recent years, with many of them achieving really high standards, oftentimes transcending the state-of-the-art. Figure 1 shows some examples of synthesized motions.

Figure 1. Various synthesized motions [1]

Lately, as more and more computer scientists become familiar with deep learning, deep learning and neural networks gradually advance into the center of attraction, and many attempt to incorporate this mature technique into character motion modeling, prediction, and synthesis [2]. As more rigorous standards and scrupulous demands, methods are mostly competing and trying to be prominent in features including computational efficiency [3-7], motion accuracy and realisticity [1, 4, 8-13], behavior complexity [14-21], database requirements [3, 22-25], training time [5, 18, 23, 26-27], model generalizing capability [10, 15, 27-32], and movement prediction stability [11, 33-40]. Apart from physics-based motion synthesis, which also have received great attention, using deep learning

and neural networks better allow models to learn and handle complicated non-linear patterns and relationships, and then extract, combine, and generate complex character motions directly from data, though with trade-offs on relative computational efficiency and the cost of losing partial physical accuracy and interpretability [17, 22, 37, 41-42].

The following section of this paper are organized as follows: Section 2 expatiates models which integrates different neural networks including recurrent neural networks (RNN), long short-term memory networks (LSTM), generative adversarial networks (GAN), convolutional neural networks (CNN), autoencoders, etc. Section 3 demonstrates non-neural network methods, mostly based on deep reinforcement learning and other deviates of deep learning. Section 4 discusses the advantages and disadvantages for different approaches elaborated above with tables and data. Finally, Section 5 presents the conclusion.

2. Methods with Neural Networks

Incorporating neural networks into motion synthesis provides a powerful and controllable framework, though the quality of the output depends more on whether there is a well-defined dataset for training, compared to approaches using deep learning techniques. Neural networks for motion synthesis do not require a reward signal unlike the DRL approach and are effective at learning complex patterns in the data, allowing explicit control on motion characteristics and interpolation and blending of motions [43]. There are multiple kinds of neural networks that plenty of research exploited, including CNN, RNN, LSTM, GAN, Autoencoders, and other variations ornamented in the following subsections.

2.1. Recurrent Neural Networks

Recurrent Neural Networks are well-suited for tasks involving sequential data, as the order and timing of character actions are essential, this ability to capture temporal dependencies is crucial for motion synthesis [2]. Besides, RNNs can also capture long-term dependencies in the training data, meaning they can remember information from previous time steps and use it to influence predictions at later time steps [28]. Moreover, RNNs allow refinement and adjustment of the synthesized motion since the output of the network is fed back as input for the next time step [18]. Ghorbani et al. utilized a deep hierarchical recurrent framework which can be tuned via weak control signals, aiming to preserve the stochastic nature of human motion simultaneously generate realistic spatiotemporal motion sequences, resulting in a higher inception score, which measures motion diversity, and a lower Fréchet Inception Distance Score, which measures the difference between the generated and the real motion samples [12]. Zhang and Panne proposed an autoregressive RNN, curriculum learning and a loss function, conditioned on target keyframes. Even when the height of the keyframe was adjusted, as shown in figure 2, the generated motion can follow the new keyframe constraints with a smooth trajectory. Though when the timing of sequences was modified, the framework did not track all keyframes precisely as the network is biased toward a more physically plausible result [44]. Harvey and Pal presented a Recurrent Transition Network, which only needed under 15MB of memory, using encoder-recurrent-decoder structure to perform movement transitions between the starting and ending frames. The training process required no labeling of gaits or frames but heavily relied on the quality of the data. It also required postprocessing to blend the synthesized motion with the ending target position due to unidirectionally using past frames only to infer transitions [45]. Harvey et al. later proposed a time-to-arrival embedding modifier, ensuring robustness to varying transition lengths, and a scheduled target noise modifier, allowing variations in generated transitions, with RNNs. This system can quickly generate quality motions between sparse keyframes with temporal and spatial variations but didn't allow control over those variations and struggled to generate transitions if the conditions were outside the training set [46]. Tang et al. proposed a multi-layer non-linear network based on RNN, a modified highway unit (MHU) to predict future skeleton using the last frame and summarized motion context, and a gram matrix loss for minimization for penalizing mean pose

convergence, which appeared normally in LSTM models, to better predict and generate motion in the long term [39]. Fragkiadaki et al. introduced Encoder-Recurrent-Decoder (ERD) networks, which combined representation learning with learning temporal dynamics for modeling human kinematics. By jointly training both the encoder and decoder recurrent networks, this model learned the representation for recurrent prediction, labeling, and dynamics, but required a large amount of training data [47]. This method used a time-series model with gating networks and a novel feature called local motion phase. Compared to using a single global phase, which requires a careful labeling process or explicit rules, local motion phase computed each bone independently and automatically, while ensuring asynchronous encoding for each contact. It has a larger training dataset than LSTM and can generate and combine motions that are not in the training set [24]. Pavllo et al. introduced a RNN called QuarterNet, which is based on quaternions for rotation parameterization. This model generated motion in real-time, allowing better control of time and space constraints [48].

Figure 2. Jump motions with edited apexes, where the green shows the original motions [44]

2.1.1 Long Short-Term Memory Networks

There is a special kind of RNN called Long Short-Term Memory Network, which is explicitly designed to address the vanishing and exploding gradient problems that can occur in traditional RNNs. With gating mechanisms that control the flow of information through the network, LSTMs can selectively remember or forget information from the previous steps and are better at retaining relevant information over extended periods, which traditional RNNs might be struggling with [4]. Jin et al. proposed a LSTM-based machine learning model to synthesize realistic head and eye animations in three-party conversations. However, this research ignores the semantic aspect and detailed finger motion for conversation, also it can't extract the prosody of an individual speaker if conversation overlapped [4]. Huang et al. demonstrated a deep neural network that can reconstruct human poses from 6 Inertial Measurement Units (IMUs), trained by a bidirectional RNN. By using the SMPL model and outputting it to LSTM architecture, it ensured to have sufficient data for training, which was the largest IMU dataset at that time. Due to a BIRNN, the model obtained temporal information from both the past and the future, resulting in smoother, qualitatively better predictions, where some sample motion frames are shown in figure 3 [6]. Lee et al. proposed a RNN using LSTM encoder and decoder units. This model didn't require structured inputs nor phase variables, can handle extreme control, avoiding undesirable visual artifacts, and generated animation faster. However, there exists a trade-off between learning speed and robustness and performance and accuracy were greatly dependent on the size and quality of the training data [2]. Zhou et al. proposed a special type of RNN called "acLSTM". This model resolved a general issue of LSTM networks, that is if the initial input has slight difference from the ground truth, the accumulated output will eventually cause the sequence to diverge or freeze. Thus, acLSTM does not fail even after 300 seconds whereas previous works at most maintain a couple seconds, achieving long term motion simulations using LSTM [18]. Martinez et al. proposed a sequence-to-sequence (seq2seq) architecture with sampling-based loss and residual connections, moving from a multi-layer LSTM to a single gated recurrent unit (GRU) without requiring a spatial encoding layer, to investigate a simpler model that outperforms other RNN method on short-term motion prediction. However, the performance during the long-run is worse [49]. Crnkovic-Friis proposed a LSTM model to generate choreographic sequences called CHOR-RNN, which was capable of producing choreography style, syntax, and to some extent semantics. However, it was limited to solo choreographies, and more hours of mocap data might improve the system's performance [50]. Henter et al. proposed a probabilistic model based on normalizing flow, using

autoregression and LSTMs to enable arbitrarily long time-dependencies. The model is task-agnostic and enable control schemes for the output motion with no algorithmic latency, but it is computationally heavy, requiring frames to be generated in sequence, and the output may switch between diverse locomotion modes and styles in an unstructured way if external guide was absent [51]. Graves demonstrated a LSTM network, which was able to generate both discrete and real-valued sequences with complex, long-range structure using next-step prediction, allowing recurrent network to condition its predictions on an auxiliary sequence, synthesizing realistic and diverse motions [8]. Wang et al. This research proposed a combination of a generator based on LSTM and a refiner network similar to GAN to add realism by using an adversarial loss, with further embedded contact information to improve quality. The network was compact, could handle both nonlinear dynamics and long-term temporal dependencies of human motion, and its size won't increase along with the amount of training data [7].

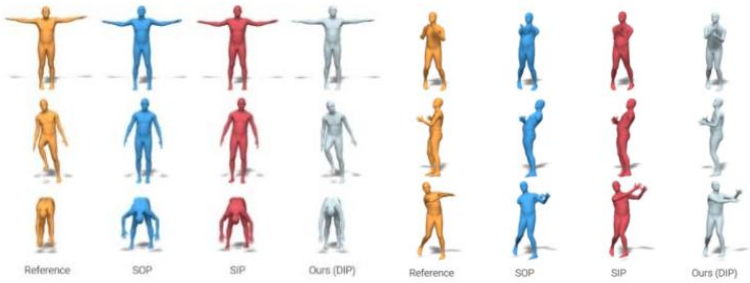

Figure 3. Sample frames of synthesized motions [6]

2.2. Convolutional Neural Networks

Convolutional Neural Networks provide spatial understanding and improve pose estimation, joint detection, and context awareness. By using 3D CNN content encoder and motion encoder, Fan and Kankanhalli proposed a method alleviated the problems of occlusion and distortion and was more robust to the noise from content information. Obtaining a unified framework to disentangle video into object, motion and background and train the framework in an unsupervised manner, the synthesized motion clip had a higher, nearly doubled motion cosine similarity [52]. Hou et al. proposed a novel causal convolutional neural network (CCNet) based on WaveNet with additional encoder and decoder and 1D convolution layers to allow the model taking skeleton configurations as an input and translate the motion data into features and back to the predicted motions, successfully synthesized motions with complex trajectories shown in figure 3. However, if the initial seed frames are set to be low, then jitters will occur in the generated motions [53]. Wang et al. proposed a novel scene-aware generative framework, introducing the scene context into convolutional sequence generation networks to respectively model the trajectories and fine-grained body movements. This framework could effectively synthesize human motions that are not only diverse in both trajectories and body movements, but also coherent with both structure and semantics of the given scene [54]. Tang et al. proposed the Motion Temporal Convolution Network (MTCN), replacing phase input with Temporal Convolution Network motion features. By augmenting a Style Neural State Machine (SNSM) to better transit between and interpolate styles, this modified MTCN-IN model could accurately reproduce the original motions in the training data, flexibly make online transition between styles according to the control for further diversity, and create new styles by merging the existing ones [20]. Aristidou et al. proposed a Bag-of-Motifs framework extracting motion motifs and signatures from motion data, which improve motion recognition, retrieval, segmentation, and synthesis. By embedding them into a feature-space using a triplet loss CNN, the model could synthesize contextually similar motions, although it was time-consuming to retrain the network when more human motion data are available [55]. Pavllo et al. further proposed a new version of QuaterNet that is based on CNNs rather than the original RNNs. By training with a position loss that performs forward kinematics on a parameterized skeleton, the model benefited from a constrained skeleton and from proper weighting across different joint prediction errors [34].

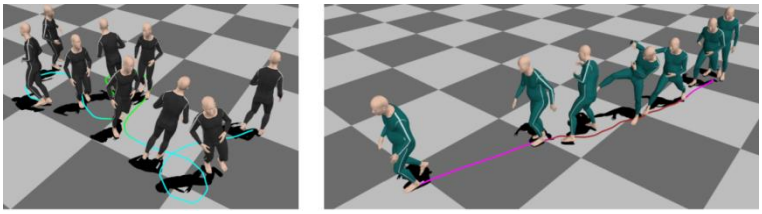

Figure 4. Synthesized motions complex trajectory and complex movements [53]

2.3. Generative Adversarial Networks

The adversarial training process in Generative Adversarial Networks encourages the generator to generate motions that are difficult for the discriminator to distinguish from real motions, which leads to robust models that can better handle variations and resemble high-quality real-world movements [10, 11, 13-14, 33, 56, 57]. Struye et al. proposed the TimeGAN, which is a novel approach that uses GAN to train on two sub-system, one generator and one discriminator. Compared to the only other approach that time, as the model converts data to a normal distribution, instead of a multi-modal Gaussian distribution in FFT, it generalizes better to outdoor virtual environments, but its runtime is substantially worse than FFT [10]. Shiobara and Murakami utilized GAN, by training both the generator and the discriminator, the generator can simulate various natural motions. The performance of both generator and discriminator improved during the beginning of the training, but halted and didn't alter in the last 125 epochs, which can be improved by enabling stable learning such as not using dropout [33]. Habibie et al. trained a GAN based model that simulates 3D human facial expression, head rotation, and upper body motion. The qualitative results showed this model slightly outperformed other methods on both naturalness and synchronization for facial and body-and-hand synthesis. However, the discriminator might produce inconsistent results since it was trained only on ground-truth motion sequences, and this model did not apply to lower body movements [57]. Merel et al. extended a generative adversarial imitation learning from limited state features. One benefit of using imitation is due to lack of good objective functions for complex motions. This model didn't require a hand-design metric and it presented a more than twice dimensional flexibility, partial results are shown in figure 5. However, the model only imitated limited behaviors with limited reuse, and the discriminator in the GAN didn't work well as mocaps have noise and are dynamically inconsistent for the imitator body [14]. Nikdel et al. proposed a novel method called DMMGAN, which combined a conditional GAN with a transformer-based encoder and a GRU to generate both the trajectory and the 3D pose of human motions [58]. Malek-Podjaski and Deligianni proposed Attention-based Wasserstein GAN with Gradient Penalty for learning both short and long-term motion synthesis. By applying self-attention, both generator and discriminator networks can learn how different output regions relate to each other, acquiring distant spatial relationships, resulting in more realistic motion synthesis. Blend loss and skeleton loss were also applied to the network to increase the accuracy and continuity of the produced motions [11]. Kutsuzawa et al. introduced conditional Wasserstein GANs with Gradient Penalty (cWGANs-GP) based on conditional GANs, which did not require designing characteristics of the latent space by hand. This method searched for the best motion from only valid motions represented in the latent space, avoiding possible convergence on an invalid motion, and did not require any relearning as it could generate motions suitable for specific situations [59]. Won et al. proposed a learning-based framework named Adversarial Correspondence Embedding (ACE), which leveraged adversarial learning and GANs to generate natural character motions while guiding the correspondence learning and preserving high-level motion semantics via a feature loss. Nevertheless, this model did not consider the difference in dynamic capabilities between morphologies, limited to a single character, and struggled to handle characters without clear notions of arms and legs [60].

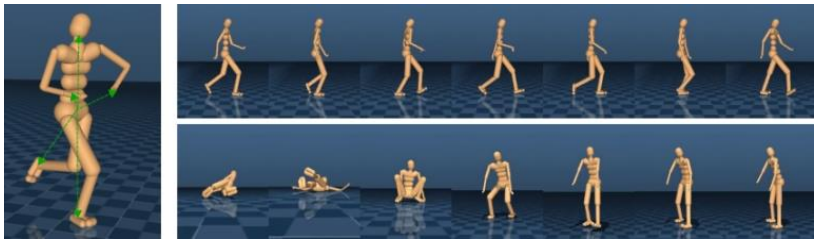

Figure 5. Equispaced frames by GAN imitations and a vector-corrected humanoid [14]

There were also successful attempts to combine generative adversarial training with other types of neural networks. Ferstl et al. proposed a RNN with a generative adversarial paradigm, producing gesture motion based on input speech. By using multiple discriminators, the entire speech-to-gesture generation was separated to a series of sub-tasks, allowing the automatic phase labelling to enforce a more realistic gesture phase structure, though the output motions still lacked realism [13]. Barsoum et al. proposed a modified version of Wasserstein GAN with a custom loss function, a generator derived from RNN, and a discriminator based on multilayer network (MLP). This HP-GAN took into consideration human anatomy, thus generating better human motions, but still couldn't tell if the training has converged even with the improvements [56]. Nishimura proposed a model named CNN-GAN, which was trained to generate a 4-second-long motion. The model could generate motions which seemed like two characters are interacting, though the behavior of a person depends on its context [40].

2.4. Autoencoders

By reconstructing motion data and comparing it to the original, autoencoders can detect and remove anomalies or outliers in the training data, extract essential motion features, denoise and reduce the dimensionality of the training data, and generate intermediate or extended motion sequences for smooth transitions between known motions. Du et al. presented a novel generative model to combine statistical motion modeling and style transfer, with the encoder and decoder both modeled by a four-layer fee-forward network. The content and style motions are encoded and used to train a conditional variational autoencoder to model the conditional distribution, resulted in synthesized motions with different styles shown in figure 6 [61]. Starke et al. proposed an unsupervised novel neural network called the Periodic Autoencoder that transformed unstructured movements into a periodic manifold, learning features with different amplitudes and frequencies with their corresponding timing, results shown in figure 7. This model can synthesize motions with more movements per second and reduce foot sliding without foot contact labels, but it cannot resolve the ambiguity of which motion to generate next [62]. Wang and Neff presented a novel method for compressing, indexing, retrieval, and reconstruction of motion databases based on extracting high level and non-linear deep signatures using multichannel autoencoder. However, the down sampling during the training process might cause a loss of temporal information [25]. Habibie et al. proposed a novel approach by combining variational inference, control signal, and deep learning modules to construct a recurrent variational autoencoder that greatly reduced the predictive error, where motion sequence gradually converging into a mean static pose, for long sequences and allowing novel motion generation without initial frames from existing sequences [63]. Tevet et al. proposed a motion generation network based on training an autoencoder that leverages the knowledge encapsulated in the Contrastive Language-Image Pretraining model. This framework's latent structure successfully induced semantics and disentanglement, but struggled to understand directions, capture some motion styles, and be consistent for out-of-domain cultural reference examples [64]. Kundu et al. proposed a novel probabilistic generative approach named Bidirectional Human motion prediction GAN (BiHMP-GAN), which is a variation from GAN. With pose embedding, this model divided cases for unrealistic motion generation, but yet tried to generate complex motion sequences [65].

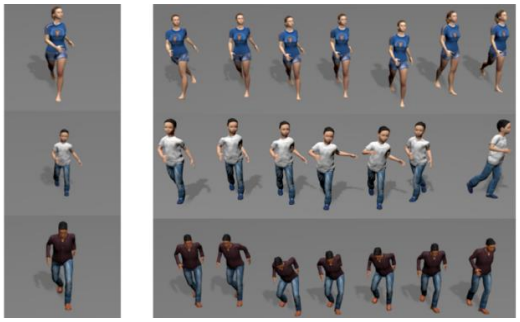

Figure 6. Stylistic motion primitives deformed from left-side neural walk using autoencoder [61]

There were also successful endeavors for combining autoencoders with RNNs. Murakami and Ikezawa proposed a human motion generative model using RNN and Variational AutoEncoders (VAE), where RNN could represent dependency between motion and latent variables during past time and the VAE model could extract human motion features and represent them as a probability distribution in low dimensional latent space. Although the generated posture of each frame was almost natural, the transition between them was not smooth [66]. Ghosh et al. combined a 3-layer LSTM and a dropout autoencoder and proposed the Dropout Autoencoder LSTM model, which performed better as the autoencoder was primarily used for representation learning and pose reconstruction and temporal modeling were separated into independent tasks. This model can generate natural actions in long time horizons, but since the model lacked physics-based feedback, it could not identify small errors in overall orientation [38].

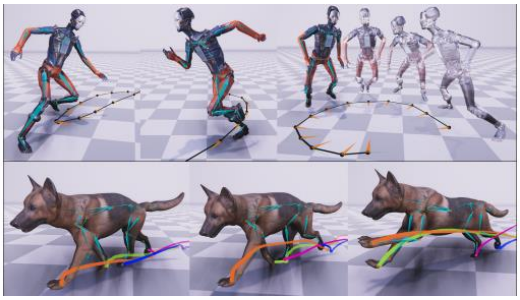

Figure 7. Diverse types of locomotion synthesized using the learned phase manifold [62]

2.5. Other Neural Networks

There are also other types of neural networks that cannot be classified as above categories but still produces high-quality synthesized motions. By using spatial differences and spatiotemporal patterns of a motion capture sequence, Mo et al. used a graph-based neural network agent to estimate and select keyframes from the CMU mocap dataset. It has similar capability to estimate keyframes but significantly reduces inference latency compared to greedy methods [67]. Yu et al. proposed a two-level neural network architecture based on feed-forward inference to solve full-body inverse kinematics problem in multi-contact situations with real-time performance, which could produce crowd motions in real-time, as shown in figure 8 [68]. Holden et al. proposed a novel model called Phase-Functioned Neural Network (PFNN), which uses a cyclic function to calculate weights and takes input user controls, character's previous state, and the geometry of the scene into account, thus can generate motions on different terrain settings, shown in figure 9. Once trained, the system can execute in 0.0008 seconds the fastest and require only 10 MB the smallest, regardless of the size of training datasets. However, it required 30 hours of training and manual labeling phases and gaits [69]. Gaisbauer et al. utilized deep neural networks to blend the start and end postures and generate in-between motion, where the synthesized motions are more natural than the generated motions through linear blending, and self-collisions also reduced. However, this approach didn't consider temporal relation of motion clips, thus the network may generate identical motions if the starting and ending poses are the same, regardless of the velocity and acceleration of the motions in between [43]. Zhang et al. proposed a time series model with a novel neural network structure called Mode-Adaptive

Neural Networks (MANN), whose weights are dynamically computed by gating network, that can produce smooth transitions when gradually changing the speed, meaning it can generalize to motion deviations from standard gaits. It also offers more flexibility to align different modes of locomotion and the simulated leg motions were more natural with bigger amplitude [32]. Starke et al. proposed a Neural State Machine with bi-directional controller, interaction sensor and environment sensor for better precision and to avoid unnatural movements, as shown in figure 10. The system runs in real-time and performs better overall in responsiveness and precision compared to other time-series models, like LSTM, PFNN, and MANN. However, when switching the goal action, the input into the network can change discontinuously, resulting in abrupt movements [70]. Bütepage proposed a feed-forward network with a bottleneck, trained symmetric, time-scale, and hierarchy temporal encoders to predict future motion frames. Compared to RNNs, these encoders are more generalized and robust as they were able to infer poses of the missing body parts from an unseen dataset [15]. Hou et al. proposed a novel two-part auto-regressive neural network, adopting attention mechanisms and with the help of consistency loss and feature fusion layer, can synthesize and generate high-quality motions and ensure well coordination between upper and lower body parts. However, TPTN performed poorly on motions in the air and motions that interact with the environment [71]. Qin et al. proposed a two-stage transformer encoder-based framework to synthesize motion in-betweening, where a context transformer first generates rough transitions and a detail transformer then refine the details of the motions. With keyframe positional encoding and learned relative positional encoding, the model is robust and can generalize to longer transitions with faster training speed, but undesired results might appear when the input context is ambiguous or distinct from the training set [28]. Holden et al. proposed a leaned alternative to the motion matching algorithm by breaking the algorithm down into three unique neural networks. The system can provide finer control when there is no clear match to select, but the networks were not trained to generalize thus it could not produce new motions unseen in the training data, and moreover there was a trade-off between quality improvements, by increasing the number of hidden layers, and runtime evaluation [26]. Smith et al. proposed a method utilized three neural networks that work together to output pose, foot contact, and timing of the generated motion, allowing the system transfer motion styles in real time with very little run-time memory footprints. However, this method was incapable of handling self-intersection artifacts, required a large dataset, and succeeded only for limited motion styles [3]. Merel et al. proposed a neural network architecture named neural probabilistic motor primitives, which was designed to perform one-shot imitation while learning a dense embedding space of individual motor skills. By utilizing linear feedback policy cloning, the model could transfer expert behavior using a single rollout, though currently it has been restricted to motor behaviors which do not involve interactions with objects [72]. Rajamäki and Hämäläinen presented a sampling-based model predictive control (SMPC), combining fast but imprecise nearest neighbor learning and slow but precise neural network training, acquiring the capability to act instantly and learn from experience, require no training data beyond what it generates through simulation. The model can produce stable locomotion while having a small sampling budget, thus running in real-time or near real-time [36]. Pan and Manocha proposed a method to automatically generate active animations of reduced deformable bodies using spacetime optimization with accounts on physics constraints, environmental forces, and DMP-based controller parametrization, which could be seen as a special kind of one-input-one-output neural network. However, this approach might not exhibit the same level of naturalness due to lack of keyframes [73].

Figure 8. Synthesized crowd motions in real-time [68]

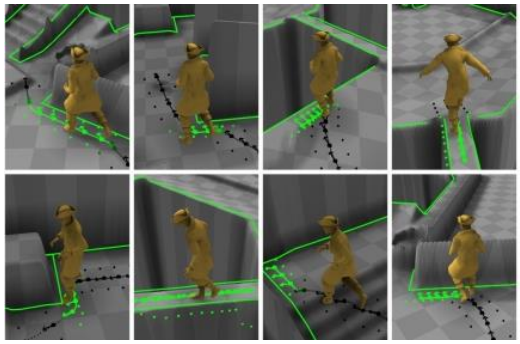

Figure 9. Results of the character where future trajectories collide with the environment [69]

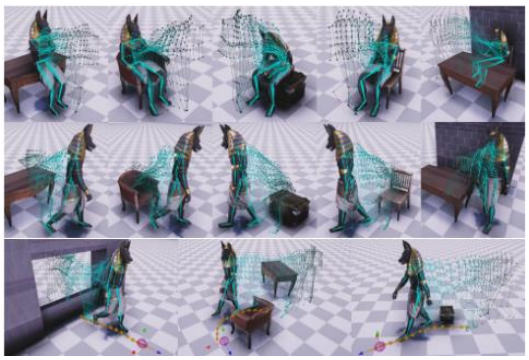

Figure 10. Result of Neural State Machine generating motions for various tasks [70]

3. Methods without Neural Networks

One of the most crucial issues for motion synthesis is to produce a large variety of motions while simultaneously making sure the results are being natural and realistic [42]. It is time-consuming and technically challenging to cover a vast majority of required motions in a scalable and controllable fashion [27]. Nevertheless, with the success of deep learning techniques, it provides researchers an opportunity to develop more efficient and flexible solutions [67]. Since methods assimilating neural networks oftentimes require large amounts of training data, are difficult to interpret its internal working and decisions, and generally are computationally intensive, which also require significant resources, several researchers considered approaches for motion synthesis without utilizing a neural network [3, 4, 51]. To synthesize complex movements such as martial arts, Starke et al. proposed a modular deep learning framework which alleviated problems of long training time, could respond to quickly changing inputs, and was robust as it can generate unseen motions, shown in figure 11. However, there were discrepancies for certain movements and when the trajectories of different motion layers were not aligned, the model might present unrealistic behaviors [27]. What's more, by applying semantics of the environment to a deep-learning model, Paduraru pushed motion simulations to become more realistic, increasing the triggered hard collision avoidance mechanism while remaining the same inference time. However, it had a tradeoff as the agents would be fitful when they were barely constrained by the mechanism [9]. Holden et al. proposed a deep learning framework to map high-level parameters to an output motion, which the unsupervised nonlinear manifold learning process is significantly easier. By learning from a large set of motion data without manual labeling or segmentation, this made the system more practical as users can easily add new motion data to the training process, and fast execution at runtime, meaning it could create crowd motion animations [74]. The fact that deep learning is limited in goal-directed behaviors, unable to adapt to changing environments, and lacking sequential decision-making, which potentially impair the performance of the framework, resulting in a heavier focus on deep reinforcement learning, instead of the simple deep learning strategy [17, 19, 29, 75-76].

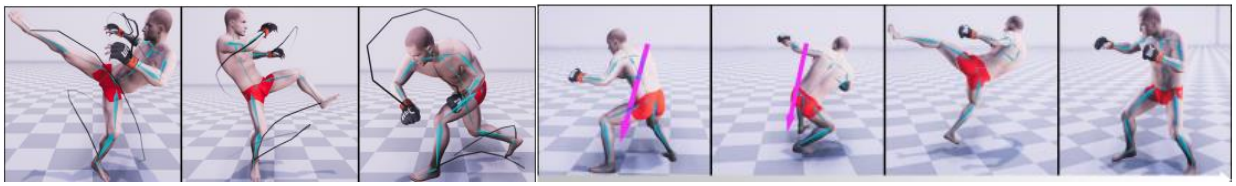

Figure 11. Synthesized martial arts movements with modular deep learning framework [27]

3.1. Deep Reinforcement Learning

Compared to deep learning, deep reinforcement learning enables the character to interact with its surroundings, learn from its previous actions, and adapt its behaviors based on feedback, which is critical for tasks where the characters need to respond to uncertain environments. In 2015, Xue Bin Peng et al. utilized DRL to simulate dynamic motions on challenging terrains. By combining tuple-based non-parametric approximations, value iteration based on positive temporal differences, and epsilon-greedy exploration, the model could control characters in high-dimensional and continuous spaces [77]. Then in 2016, they presented a novel DRL framework based on CACLA-style learning and a mixture of actor-critic experts, which could directly work with high-dimensional state and highly-dynamic terrain, avoiding the need for manual feature descriptors [78]. They further proposed a DRL framework with actor-critic learning algorithm and suggested that including basic local feedback in action parameterizations could improve policy performance and learning speed across different motions and character morphologies [79]. Further in 2017, they proposed a two-level hierarchy based on DRL. Training low-level controller (LLC) and high-level controller (HLC) separately, the model enables a degree of interchangeability as HLC can apply different intermediate motion objectives to multiple LLCs, exploring wide-ranged of behavior strategies and adapting to different terrains, as shown in figure 12 [5]. Later in 2018, they proposed a data-driven DRL framework with early termination in the training process to eliminate undesirable contacts and ensured the network produces the desired movements. Besides training on a humanoid, the model also trained a T-rex and a dragon, with much higher dimensional flexibility, to satisfy certain task objectives, thus proving the framework is capable of learning from artist-authored keyframes when there is no mocap data [17]. Moreover, Xie et al. presented a curriculum-based solution for character locomotion on complex terrains named ALLSTEPS. After training, three different characters can all perform stepping-stone tasks, as shown in figure 13, though they might occasionally miss a step in narrow situation, and the method ignores the influence of step transitions for less computation workload [21].

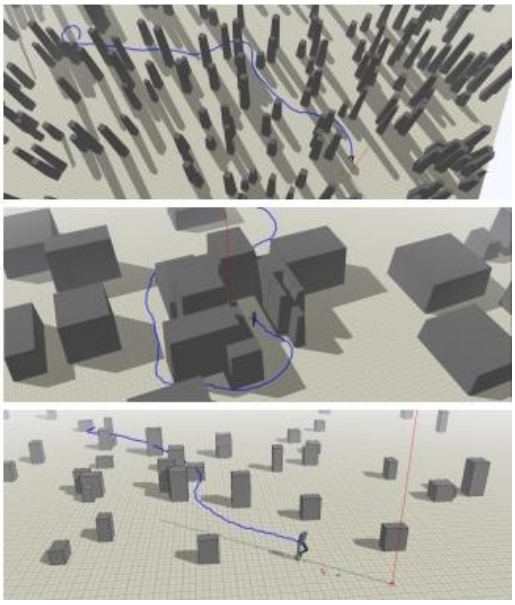

Figure 12. Character locomotion in multiple complex terrains [5]

Figure 13. Characters performing stepping-stone task on randomly generated terrain [21]

Other than being responsive to terrain uncertainties, DRL is also well-suited for tasks involving making a sequence of decisions, which requires dynamic character motions in evolving conditions. Sequential decision-making is closely linked to goal-oriented motions, which DRL also pertinent in. Clegg et al. presented a novel DRL framework aiming to simulate virtual character dressing. After separating an entire dressing process to several subtasks, the framework utilized a state machine and a policy sequencing algorithm to connect each subtask, producing a variety of successful dressing motions. Though given a 79% success rate, the simulation only runs in interactive rates rather than real-time, and the training process is computationally heavy [29]. Won et al. presented a novel Self-Regulated Learning (SRL), which combined DRL and an additional self-regulation control policy for aerobatics control. SRL was easy to implement and surprisingly effective for problems in which its sequential sub-goals have to be identified and addressed one-by-one to achieve the main goal [75]. Liu and Hodgins proposed a framework using a trajectory optimization approach with DRL to simulate basketball dribbling, a sequential and goal-oriented motion, with halved time length for control fragments, allowing tighter control of the ball's motion. The deep-learned non-linear arm control policies enable robust control of all cyclic basketball skills and ensure the learning process can finish in a reasonable time but wasn't capable of learning controllers for the skills that require accurate control of steps [16]. Chentanez et al. proposed a method utilized a tracking agent and a recovering agent, trained by DRL, to imitate motions in mocap clips from the CMU database, thus tolerated some number of perturbations and could synthesize unseen and continuous motions shown in figure 14 [42]. Arnold et. al proposed a new policy architecture based on DRL, operating efficiently with a large number of actions by leveraging prior information about the actions to embed them in a continuous space. This architecture design also allowed decoupled complexity [31].

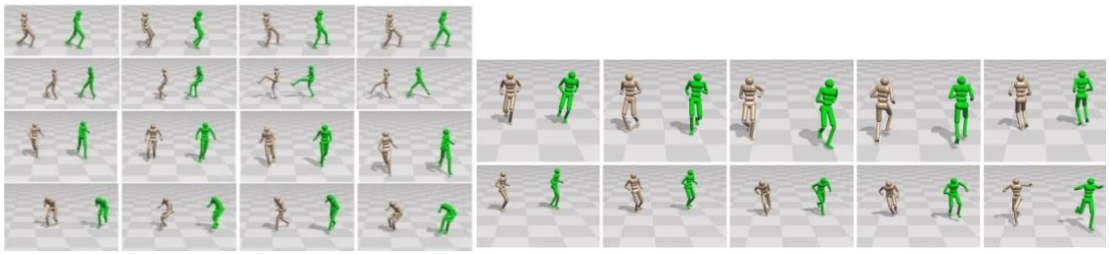

Figure 14. Unseen, continuous, synthesized motion [42]

With DRL, models are capable of handling where characters have access to partial information, and researchers can also update the database with more novel inputs for a better synthesized result. Yu et al. demonstrated a DRL approach with curriculum learning, virtual assistance to keep balance, and modification to the loss function to encourage symmetric behavior. Since it could produce locomotion controllers without prior knowledge about the motion, the model learned faster than all baseline methods [80]. Bergamin et al. proposed a new approach based on a DRL controller that learn from a large database of unstructured motion data, which had a higher degree of freedom, quicker response time, and lower runtime cost, due to the fast kinematic controller, the low simulation frequency used, and the low cost of policy evaluation [22]. Cho et al. proposed a data-driven character control method based on DRL, which could be seen as a motion matching deviant that used constructed databases and cluster recommendation components rather than greedy searching components. This method only required less than 24 hours of training time and could be trained on various motion data, but it was memory inefficient, relatively slow on updating speed, and limited on generating unseen motions [23]. Won and Lee proposed a novel algorithm based on DRL that learned

a physics-based controller equipped with body shape parameters, allowing it to deal with various body shapes, controlling characters immediately without any re-training process, and changing characters' shapes during the simulation, tending resolve the law of the jungle problem with biased training data generation and standardized reward values for different body shapes, as shown in figure 15 [41]. Research above illustrates how using DRL can help characters better counter complex environments, handle sequential decision-making and goal-oriented tasks, and manage to refine the training data that only has partial information.

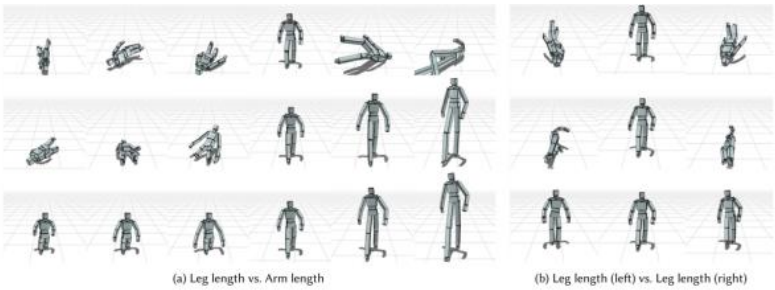

Figure 15. Comparison to methods without shape variation or adaptive sampling [41]

3.2. Other Deviations of Deep Learning

Apart from traditional deep learning and DRL, some researchers variated the deep learning algorithm, adding other policies, aiming to generate better results. Won et al. proposed a deep Q-learning (DQL) model with trajectory optimization and evolutionary strategy of Covariance Matrix Adaptation Evolution Strategy (CMA-ES). Compared to the standard DQL, this model not only converged faster but also discovered better control policy at the convergence, resulting in the trained character had various motor skills that generate agile maneuvers. Switching between motor skills is also immediate and effortless with the help of the neural network [81]. Lillicrap et al. proposed a model-free reinforcement method called deep deterministic policy gradient (DDPG) algorithm, applying DRL and Q-learning, using an actor-critic approach, a replay buffer, and batch normalization. With batch normalization, the model could learn effectively across many different tasks with differing types of units, without needing to manually ensure the set range for the units [82]. Liu and Hodgins attempted to combine DQL with neural networks, developed a control system with schedulers trained by a medium-sized neural Q-network, whose Q-value function was approximated by a Feedforward Neural Network (FNN). The learned scheduler could ensure the character to maintain balance during oscillation on non-flatten surfaces without external perturbations, as shown in figure 16. The character will frequently use out-of-sequence actions to regain balance and slowly turn to the target direction when encountering exterior forces [37].

Figure 16. Real-time simulation of the learned schedulers on various motion tasks [37]

4. Discussion

Methods that use neural networks can learn complex relationships in motion data, are generally more adaptable as networks can be fine-tuned to account different scenarios or motion styles and are more effective on modeling temporal dependencies. On the other hand, methods without neural networks generally provide explicit modeling due to manual-designed rules and heuristics and are less time-consuming, resource-intensive, and data-dependent. Nonetheless, both are capable of synthesizing high-quality and robust motions, transit between one motion to another, and respond to new inputs. To compare approaches presented above, table 1 puts together some representative methods in each category and lists the advantages and limitations of each.

Table 1. Advantages and limitations of different types of motion synthesis method

Types of Methods			Advantages	Limitation
Methods with Neural Networks	RNN	Autoregressive RNN [44]	Smooth trajectories when changing, style-and-physics aware, global consistency, simple and usable in online situations	Physically biased, takes 130 hours for training, lack of consideration of more than one keyframe in advance, discontinuities in motion, doesn't provide partial keyframe control
		Recurrent Transition Network [45]	Picks contexts directly and solely from data, models uneven ground locomotion, 15MB of memory	Heavily dependent on data quality, require target blending to match the ending frame due to unidirectional synthesis
	LSTM	LSTM for head and eye animation [4]	Uses gates to control information received from the previous state, better statistic and user-voted results	Major computation bottleneck, ignoring affective states of the interlocutors and semantics aspects, unclear extraction for overlapping speech
		Deep Inertial Poser [6]	Considers bidirectional temporal information, capable of reconstructing full-body motion from a sparse set of sensors in real time, rarely produces inter-penetrations despite noise in data	Difficult to effectively model accelerations, struggles when character is parallel to the floor
		LSTM encoder and decoder units [2]	Takes 5 milliseconds per frame, avoids undesirable visual artifacts, can learn many structurally different actions together without explicit representation of phases or nodes	Required manual labeling, performance and accuracy mainly depend on the size and quality of the training data, trade-off between learning speed and robustness
	CNN	Motion Temporal Convolution Network [20]	Reproduces original motions, flexible online transitions between styles, creates new styles by merging the existing ones	Trade-off between performance degradation and elimination of style information in the past motion sequence
		Casual CNN [53]	Takes skeleton configurations for better motion predictions, captures personalized style variation	Struggles to handle arbitrary skeleton variation, severe foot-ground penetration, jitters occur when initial seed frames were low
	GAN	TimeGAN [10]	Insensitive to the distributions within the dataset, more generalized to outdoor virtual environment, outperform earlier GAN-based approaches in multiple datasets	Challenging to generate very-slow motion time series, computationally inefficient, taking over 10 hours on a workstation computer
		Generative Adversarial imitation learning [14]	Doesn't require explicit, hand-design metric for imitation and demonstration data, more action dimensions	Inconsistent discriminator, difficult to develop and assess algorithms as well as when monitoring convergence, demonstrated on restricted behaviors and limited reuse, builds less in structure in controllers and functions
	Auto-encoder	Periodic Autoencoder [62]	Reduces foot-skating even when missing contact labels as inputs, synthesize more movements per seconds	Doesn't resolve ambiguity about which motion skill to generate, still requires user control or probabilistic techniques to sample from a learned distribution
		3-layer LSTM with a dropout autoencoder [38]	Fast to train since doesn't require any hyperparameter tuning, easier to model velocity representations, best performance when trained on multiple actions	Best numerical results do not correspond to the best qualitative long-term motion, inherently hard to produce both accurate short-term predictions and long-term forecasting
	Other	Phase-Functioned Neural Network [69]	Produces higher quality results than LSTMs, extremely fast and compact, requiring milliseconds of execution time and a few megabytes of memory, enables characters to adapt to different geometric environment	Requires manual checks, corrections, and gait labeling, slow computation for phase function, requires longer training time, trade-off between responsiveness and quality, struggles with complex interactions with the environment
		Mode-Adaptive Neural Network [32]	Produces gradually smooth transitions, quickly responds to direction and velocity inputs, more flexibility due to more degrees of freedom of the gating network	Limited training dataset, doesn't conduct any terrain fitting and adaptation

Method without Neural Networks	DL	Modular DL on Martial Arts motion [27]	Robust and can generalize to unseen movements, responsive to quickly changing inputs	Conceptually infeasible inputs will generate unrealistic motion, limited accuracy when reproducing specific motions, doesn't automatically extract control signals for higher-level actions
		Unsupervised non-linear manifold DL [74]	Does not require any manual motion segmentation or alignment, fast execution at runtime, suitable for motions of large crowds, allows users to easily add new motion to training set	There is an ambiguity between high-level parameters and the output motion
	DRL	DeepMimic: Data-driven DRL [17]	Selects the most appropriate clip for given situation and switch between clips whenever appropriate, can be readily applied to non-bipedal characters, significantly higher-dimensional action space, robust to external perturbation and generate plausible recovery results	Limits the ability to adjust the timing of the motion, learning process is time consuming, often requiring several days per skill, is based on a manually defined state-similarity metric, relative weighting of the imitation reward and task reward.
		Physics-based motion capture imitation with DRL [42]	Can imitate unseen clips, has a novel action space consisting joint torque and the gain of PD controllers, can recover from large disturbances, reset dead agents immediately to avoid stalling	Can fail to track certain motions such as flips, produces jittery motion in some cases as the agent was trying to optimize for more reward by performing micro movements
		DRL with Curriculum learning [80]	Superior data-efficiency, faster learning with curriculum learning and mirror symmetry loss, learn speed-appropriate gaits	The quality of the motion is still not on a par with previous method that exploits real-world data
	DQL	Covariance Matrix Adaptation Evolution Strategy [81]	Immediate and effortless switches between motor skills, converges faster and discovers better control policy, generates agile maneuvers can be extended to other types of locomotion	The quality of the initial reference trajectory substantially influences the quality of the simulated motor skills, insufficient keyframes to describe natural looking motions

There are also attempts to combine neural networks with DRL. Lee et al. proposed a teacher-student framework, where teacher learning was based on DRL and student learning adopted a RNN model consisting of stacked LSTM layers. By separating teacher and student policy, the model simplifies the learning challenge of spatial and temporal conditions, but there exists blurring at the beginning and ending of the generated action [83]. Lee et al. proposed a hierarchical network architecture based on DRL to address both long-term planning of trajectory mimicking and short-term muscle coordination in a unified framework, resulting in a scalable algorithm to simulate and control realistic human movements with highly-detailed musculoskeletal models [1]. Luo et al. proposed a data-driven, physics-based, controllable quadruped agent, combining the merits of GAN and DRL. The trained agent can effectively model interactions in complex navigation scenarios and dynamically changing environments [19]. Ling et al. introduced a kinematic generative model based on an autoregressive conditional variational autoencoder (MVAE), using DRL to train controllers to achieve movements. This model can produce robust, high-quality, long-term motion predictions and allow multiple control policies to be learned using the same motion model as learning of the task is separated from the learning of the dynamics, as shown in figure 17 [35]. Park et al. presented a predict-and-simulate framework that incorporates a RNN-based motion generator and DRL-based controller. By taking dynamic states from the physical simulation rather than the direct output of the network as input, this setup provided a lot of flexibility to the network, allowing the motion generator to deviate from the training data, be resilient to external perturbation, and physically interact with the environment. Some examples of synthesized motions are shown in figure 18 [76]. Approaches combining neural networks with DRL succeeded the advantages of both methods, but also simultaneously inherited the potential limitations of both, thus it really depends on how well the combinations were.

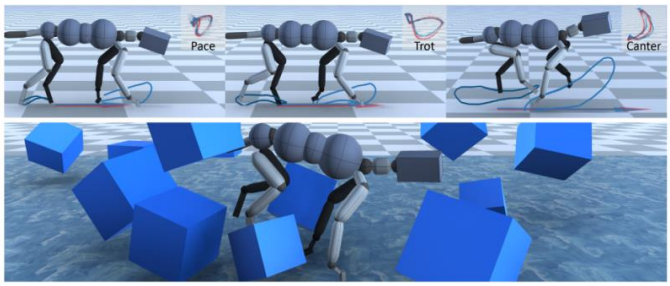

Figure 17. Non-bipedal agent produces movements and adapt to external perturbations [35]

One major issue for motion synthesis based on deep learning is its significant dependence on the quality and variety of the dataset. A potentially perfect dataset would accurately cover various actions with diverse speeds, styles, and environmental conditions, with each class being represented sufficiently to prevent any bias. The data should also be free of noise, artifacts, or errors as they can mislead the models to synthesize undesired results [10, 12, 14, 36, 46, 52, 76, 78]. There also exist a lot of trade-offs, such as between short-term and long-term motion simulation, transferability and overfitting, computational efficiency and robustness, motion quality and responsiveness, etc. Hence, navigating these trade-offs requires careful consideration of the particular requirements and constraints of the application for the synthesized motions, and researchers should design the models according to the specific tasks that the model needs to accomplish.

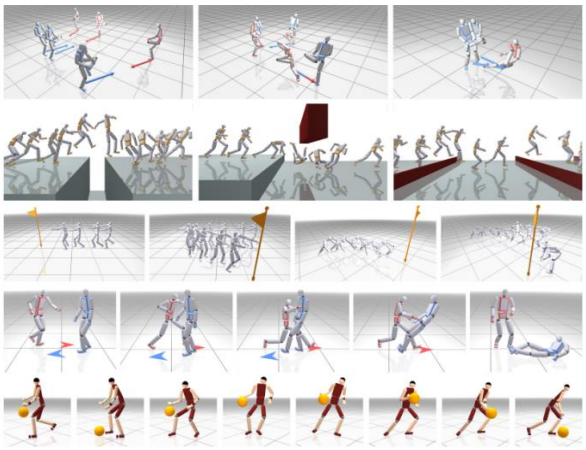

Figure 18. Various examples of synthesized motions using RNN generator and DRL controller [76]

5. Conclusion

This paper presents a survey of several existing deep learning methods on character motion synthesis. With the development of deep learning techniques, higher quality datasets, and computational resources, the synthesized motions become more realistic and robust than before and believed to be more and more prevailing in industries such as video games, movies, and animations. One future approach for motion synthesis based on deep learning is to try combining different networks and learning strategies, preserving the advantages of each method and simultaneously resolving the limitations of each to the utmost extent or according to the specific requirements, as trade-offs will always exist due to the nature of deep learning. Moreover, instead of merely based on learning schemes or neural networks, researchers can try to combine deep learning with physics-based controllers for more realistic and better-quality results.

References

[1] Seunghwan Lee, Moonseok Park, Kyoungmin Lee, and Jehee Lee. 2019. Scalable Muscle-Actuated Human Simulation and Control. ACM Trans. Graph. 38, 4, Article 73 (July 2019), 13 pages. https://doi.org/10.1145/3306346.3322972.

[2] Kyungho Lee, Seyoung Lee, and Jehee Lee. 2018. Interactive Character Animation by Learning Multi-Objective Control. ACM Trans. Graph. 37, 6, Article 180 (November 2018), 10 pages. https://doi.org/10.1145/3272127.3275071.

[3] Harrison Jesse Smith, Chen Cao, Michael Neff, and Yingying Wang. 2019. Efficient Neural Networks for Real-time Motion Style Transfer. Proc. ACM Comput. Graph. Interact. Tech. 2, 2, Article 13 (July 2019), 17 pages. https://doi.org/10.1145/3340254.

[4] Aobo Jin, Qixin Deng, Yuting Zhang, and Zhigang Deng. 2019. A Deep Learning-Based Model for Head and Eye Motion Generation in Three-party Conversations. Proc. ACM Comput. Graph. Interact. Tech. 2, 2, Article 9 (July 2019), 19 pages. https://doi.org/10.1145/3340250.

[5] Xue Bin Peng, Glen Berseth, KangKang Yin, and Michiel van de Panne. 2017. DeepLoco: Dynamic Locomotion Skills Using Hierarchical Deep Reinforcement Learning. ACM Trans. Graph. 36, 4, Article 41 (July 2017), 16 pages. DOI: http://dx.doi.org/10.1145/3072959.3073602.

[6] Yinghao Huang, Manuel Kaufmann, Emre Aksan, Michael J. Black, Otmar Hilliges, and Gerard Pons-Moll. 2018. Deep Inertial Poser: Learning to Reconstruct Human Pose from Sparse Inertial Measurements in Real Time. ACM Trans. Graph. 37, 6, Article 185 (November 2018), 15 pages. https://doi.org/10.1145/3272127.3275108.

[7] Zhiyong Wang, Jinxiang Chai, and Shihong Xia. 2018. Combining Recurrent Neural Networks and Adversarial Training for Human Motion Modelling, Synthesis and Control. arXiv: 1806.08666 (2018).

[8] Alex Graves. 2013. Generating sequences with recurrent neural networks. arXiv: 1308.0850.

[9] Ciprian Paduraru and Miruna Paduraru. 2022. Pedestrian motion in simulation applications using deep learning. In Proceedings of the 6th International ICSE Workshop on Games and Software Engineering: Engineering Fun, Inspiration, and Motivation (GAS '22). Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/3524494.3527624.

[10] Jakob Struye, Filip Lemic, and Jeroen Famaey. 2022. Generating Realistic Synthetic Head Rotation Data for Extended Reality using Deep Learning. In Proceedings of the 1st Workshop on Interactive eXtended Reality (IXR '22). Association for Computing Machinery, New York, NY, USA, 19–28. https://doi.org/10.1145/3552483.3556458.

[11] Matthew Malek-Podjaski, Fani Deligianni. 2017. Adversarial Attention For Human Motion Synthesis. arXiv: 2204.11751.

[12] S. Ghorbani, C. Wloka, A. Etemad, M. A. Brubaker, and N. F. Troje. 2020. Probabilistic character motion synthesis using a hierarchical deep latent variable model. In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '20). Eurographics Association, Goslar, DEU, Article 21, 1–15. https://doi.org/10.1111/cgf.14116.

[13] Ylva Ferstl, Michael Neff, and Rachel McDonnell. 2019. Multi-objective adversarial gesture generation. In Proceedings of the 12th ACM SIGGRAPH Conference on Motion, Interaction and Games (MIG '19). Association for Computing Machinery, New York, NY, USA, Article 3, 1–10. https://doi.org/10.1145/3359566.3360053.

[14] Josh Merel, Yuval Tassa, Dhruva TB, Sriram Srinivasan, Jay Lemmon, Ziyu Wang, Greg Wayne, and Nicolas Heess. 2017. Learning human behaviors from motion capture by adversarial imitation. CoRR abs/1707.02201 (2017). arXiv: 1707.02201.

[15] Judith Bütepage, Michael Black, Danica Kragic, and Hedvig Kjellström. Deep representation learning for human motion prediction and classification. arXiv preprint arXiv: 1702.07486, 2017.

[16] Libin Liu and Jessica Hodgins. 2018. Learning Basketball Dribbling Skills Using Trajectory Optimization and Deep Reinforcement Learning. ACM Trans. Graph. 37, 4, Article 142 (August 2018), 14 pages. https://doi.org/10. 1145/3197517.3201315.

[17] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. 2018. DeepMimic: Example-Guided Deep Reinforcement Learning of PhysicsBased Character Skills. ACM Trans. Graph. 37, 4, Article 143 (August 2018), 18 pages. https://doi.org/10.1145/3197517.3201311.

[18] Yi Zhou, Zimo Li, Shuangjiu Xiao, Chong He, Zeng Huang, and Hao Li. 2018. Auto-conditioned recurrent networks for extended complex human motion synthesis. arXiv preprint arXiv: 1707.05363 (2018).

[19] Ying-Sheng Luo, Jonathan Hans Soeseno, Trista Pei-Chun Chen, and WeiChao Chen. 2020. CARL: Controllable Agent with Reinforcement Learning for Quadruped Locomotion. ACM Trans. Graph. 39, 4, Article 38 (July 2020), 10 pages. https://doi.org/10.1145/3386569.3392433.

[20] Yingtian Tang, Jiangtao Liu, Cheng Zhou, and Tingguang Li. 2022. Online Motion Style Transfer for Interactive Character Control. arXiv preprint arXiv: 2203.16393.

[21] Zhaoming Xie, Hung Yu Ling, Nam Hee Kim, and Michiel van de Panne. 2020. ALLSTEPS: curriculum-driven learning of stepping stone skills. In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '20). Eurographics Association, Goslar, DEU, Article 20, 1–12. https://doi.org/10.1111/cgf.14115.

[22] Kevin Bergamin, Simon Clavet, Daniel Holden, and James Richard Forbes. 2019. DReCon: Data-Driven Responsive Control of Physics-Based Characters. ACM Trans. Graph. 38, 6, Article 206 (November 2019), 11 pages. https: //doi.org/10.1145/3355089.3356536.

[23] Kyungmin Cho, Chaelin Kim, Jungjin Park, Joonkyu Park, and Junyong Noh. 2021. Motion Recommendation for Online Character Control. ACM Trans. Graph. 40, 6, Article 196 (December 2021), 16 pages. https://doi.org/10.1145/ 3478513.3480512.

[24] Sebastian Starke, Yiwei Zhao, Taku Komura, and Kazi Zaman. 2020. Local Motion Phases for Learning Multi-Contact Character Movements. ACM Trans. Graph. 39, 4, Article 54 (July 2020), 14 pages. https://doi.org/10.1145/ 3386569.3392450.

[25] Yingying Wang and Michael Neff. 2015. Deep signatures for indexing and retrieval in large motion databases. In Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games (MIG '15). Association for Computing Machinery, New York, NY, USA, 37–45. https://doi.org/10.1145/2822013.2822024.

[26] Yingying Wang and Michael Neff. 2015. Deep signatures for indexing and retrieval in large motion databases. In Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games (MIG '15). Association for Computing Machinery, New York, NY, USA, 37–45. https://doi.org/10.1145/2822013.2822024.

[27] Sebastian Starke, Yiwei Zhao, Fabio Zinno, and Taku Komura. 2021. Neural Animation Layering for Synthesizing Martial Arts Movements. ACM Trans. Graph. 40, 4, Article 92 (August 2021), 16 pages. https://doi.org/10.1145/ 3450626.3459881.

[28] Jia Qin, Youyi Zheng, and Kun Zhou. 2022. Motion In-betweening via Twostage Transformers. ACM Trans. Graph. 41, 6, Article 184 (December 2022), 16 pages. https://doi.org/10.1145/3550454.3555454.

[29] Alexander Clegg, Wenhao Yu, Jie Tan, C. Karen Liu, and Greg Turk. 2018. Learning to Dress: Synthesizing Human Dressing Motion via Deep Reinforcement Learning. ACM Trans. Graph. 37, 6, Article 179 (November 2018), 10 pages. https://doi.org/10.1145/3272127.3275048.

[30] Jingwei Xu, Huazhe Xu, Bingbing Ni, Xiaokang Yang, Xiaolong Wang, and Trever Darrell. 2020. Hierarchical Style-based Networks for Motion Synthesis. arXiv preprint arXiv: 2008.10162.

[31] Gabriel Dulac-Arnold, Richard Evans, Hado van Hasselt, Peter Sunehag, Timothy Lillicrap, Jonathan Hunt, Timothy Mann, Theophane Weber, Thomas Degris, and Ben Coppin. 2015. Deep reinforcement learning in large discrete action spaces. arXiv preprint arXiv: 1512.07679 (2015).

[32] He Zhang, Sebastian Starke, Taku Komura, and Jun Saito. 2018. Mode-Adaptive Neural Networks for Quadruped Motion Control. ACM Trans. Graph. 37, 4, Article 145 (August 2018), 11 pages. https://doi.org/10.1145/ 3197517.3201366.

[33] Ayumi Shiobara and Makoto Murakami. 2021. Human Motion Generation using Wasserstein GAN. In Proceedings of the 2021 5th International Conference on Digital Signal Processing (ICDSP '21). Association for Computing Machinery, New York, NY, USA, 278–282. https://doi.org/10.1145/3458380.3458428.

[34] Dario Pavllo, Christoph Feichtenhofer, Michael Auli, and David Grangier. 2019. Modeling Human Motion with Quaternion-based Neural Networks. CoRR abs/1901.07677 (2019). arXiv: 1901.07677 http://arxiv.org/abs/1901.07677.

[35] Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel van de Panne. 2020. Character Controllers using Motion VAEs. ACM Trans. Graph. 39, 4, Article 40 (July 2020), 12 pages. https://doi.org/10.1145/3386569.3392422.

[36] Joose Rajamäki and Perttu Hämäläinen. 2017. Augmenting sampling based controllers with machine learning. In Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA '17). Association for Computing Machinery, New York, NY, USA, Article 11, 1–9. https://doi.org/10.1145/3099564.3099579.

[37] Libin Liu and Jessica Hodgins. 2017. Learning to schedule control fragments for physics-based characters using deep Q-learning. ACM Trans. Graph. 36, 3, Article 29 (June 2017), 14 pages. DOI: http://dx.doi.org/10.1145/3083723.

[38] Partha Ghosh, Jie Song, Emre Aksan, and Otmar Hilliges. 2017. Learning Human Motion Models for Long-term Predictions. arXiv preprint arXiv: 1704.02827 (2017).

[39] Yongyi Tang, Lin Ma, Wei Liu, and Weishi Zheng. 2018. Long-term human motion prediction by modeling motion context and enhancing motion dynamic. arXiv preprint arXiv: 1805.02513 (2018).

[40] Yusuke Nishimura, Yutaka Nakamura, and Hiroshi Ishiguro. 2020. Long-term Motion Generation for Interactive Humanoid Robots using GAN with Convolutional Network. In Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction (HRI '20 Companion), March 23– 26, 2020, Cambridge, United Kingdom. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3371382.3378386.

[41] Jungdam Won and Jehee Lee. 2019. Learning Body Shape Variation in Physicsbased Characters. ACM Trans. Graph. 38, 6, Article 207 (November 2019), 12 pages. https://doi.org/10.1145/3355089.3356499.

[42] Nuttapong Chentanez, Matthias Müller, Miles Macklin, Viktor Makoviychuk, and Stefan Jeschke. 2018. Physics-based motion capture imitation with deep reinforcement learning. In Proceedings of the 11th ACM SIGGRAPH Conference on Motion, Interaction and Games (MIG '18). Association for Computing Machinery, New York, NY, USA, Article 1, 1–10. https://doi.org/10.1145/3274247.3274506.

[43] Felix Gaisbauer, Jannes Lehwald, Janis Sprenger, and Enrico Rukzio. 2019. Natural Posture Blending Using Deep Neural Networks. In Proceedings of the 12th ACM SIGGRAPH Conference on Motion, Interaction and Games (MIG '19). Association for Computing Machinery, New York, NY, USA, Article 2, 1–6. https://doi.org/10.1145/3359566.3360052.

[44] Xinyi Zhang and Michiel van de Panne. 2018. Data-driven Autocompletion for Keyframe Animation. In MIG '18: Motion, Interaction and Games (MIG '18), November 8–10, 2018, Limassol, Cyprus. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3274247.3274502.

[45] Félix G. Harvey and Christopher Pal. 2018. Recurrent Transition Networks for Character Locomotion. In Proceedings of SIGGRAPH Asia 2018 Technical Briefs, Tokyo, Japan, December 4–7, 2018 (SA '18 Technical Briefs), 4 pages. https://doi.org/10.1145/3283254.3283277.

[46] Félix G. Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. 2020. Robust Motion In-betweening. ACM Trans. Graph. 39, 4, Article 60 (July 2020), 12 pages. https://doi.org/10.1145/3386569.3392480.

[47] Katerina Fragkiadaki, Sergey Levine, and Jitendra Malik. 2015. Recurrent Network Models for Human Dynamics. CoRR abs/1508.00271 (2015). arXiv: 1508.00271 http://arxiv.org/abs/1508.00271.

[48] Dario Pavllo, David Grangier, and Michael Auli. 2018. QuaterNet: A Quaternion-based Recurrent Model for Human Motion. CoRR abs/1805.06485 (2018). arXiv: 1805.06485 http://arxiv.org/abs/1805.06485.

[49] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. arXiv preprint arXiv: 1705.02445, 2017.

[50] Luka Crnkovic-Friis and Louise Crnkovic-Friis. 2016. Generative Choreography using Deep Learning. CoRR abs/1605.06921 (2016). http://arxiv.org/abs/1605.06921.

[51] Gustav Eje Henter, Simon Alexanderson, and Jonas Beskow. 2020. MoGlow: Probabilistic and Controllable Motion Synthesis Using Normalising Flows. ACM Trans. Graph. 39, 4, Article 236 (July 2020), 14 pages. https://doi.org/10. 1145/3414685.3417836.

[52] Hehe Fan and Mohan Kankanhalli. 2022. Motion = Video - Content: Towards Unsupervised Learning of Motion Representation from Videos. In ACM Multimedia Asia (MMAsia '21). Association for Computing Machinery, New York, NY, USA, Article 2, 1–7. https://doi.org/10.1145/3469877.3490582.

[53] Shuaiying Hou, Weiwei Xu, Jinxiang Chai, Congyi Wang, Wenlin Zhuang, Yu chen, Hujun Bao, and Yangang Wang. 2021. A Casual Convolutional Neural Network for Motion Modeling and Synthesis. arXiv preprint arXiv: 2101.12276.

[54] Jingbo Wang, Sijie Yan, Bo Dai, and Dahua Lin. 2021. Scene-aware Generative Network for Human Motion Synthesis. arXiv preprint arXiv: 2105.14804.

[55] Andreas Aristidou, Daniel Cohen-Or, Jessica K. Hodgins, Yiorgos Chrysanthou, and Ariel Shamir. 2018. Deep Motifs and Motion Signatures. ACM Trans. Graph. 37, 06, Article 187 (November 2018), 13 pages. https://doi.org/10.1145/3272127.3275038

[56] Emad Barsoum, John Kender, and Zicheng Liu. 2017. HP-GAN: Probabilistic 3D human motion prediction via GAN. arXiv preprint arXiv: 1711.09561.

[57] Ikhsanul Habibie, Weipeng Xu, Dushyant Mehta, Lingjie Liu, Hans-Peter Seidel, Gerard Pons-Moll, Mohamed Elgharib, and Christian Theobalt. 2021. Learning Speech-driven 3D Conversational Gestures from Video. In 21th ACM International Conference on Intelligent Virtual Agents (IVA ’21), September 14–17, 2021, Virtual Event, Japan. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3472306.3478335.

[58] Payam Nikdel, Mohammad Mahdavian, and Mo Chen. 2022. DMMGAN: Diverse Multi Motion Prediction of 3D Human Joints using Attention-Based Generative Adversarial Network. arXiv preprint arXiv: 2209.09124.

[59] Kyo Kutsuzawa, Hitoshi Kusano, Ayaka Kume, and Shoichiro Yamaguchi. 2019. Motion Generation Considering Situation with Conditional Generative Adversarial Networks for Throwing Robots. arXiv preprint arXiv: 1910.03253.

[60] Tianyu Li, Jungdam Won, Alexander Clegg, Jeonghwan Kim, Akshara Rai, and Sehoon Ha. 2023. ACE: Adversarial Correspondence Embedding for Cross Morphology Motion Retargeting from Human to Nonhuman Characters. arXiv preprint arXiv: 2305.14792.

[61] Han Du, Erik Herrmann, Janis Sprenger, Noshaba Cheema, Somayeh Hosseini, Klaus Fischer, and Philipp Slusallek. 2019. Stylistic Locomotion Modeling with Conditional Variational Autoencoder. In Eurographics 2019 - Short Papers, Paolo Cignoni and Eder Miguel (Eds.). The Eurographics Association. https://doi.org/10.2312/egs.20191002.

[62] Sebastian Starke, Ian Mason, and Taku Komura. 2022. DeepPhase: Periodic Autoencoders for Learning Motion Phase Manifolds. ACM Trans. Graph. 41, 4, Article 136 (July 2022), 13 pages. https://doi.org/10.1145/3528223.3530178.

[63] Ikhansul Habibie, Daniel Holden, Jonathan Schwarz, Joe Yearsley, and Taku Komura. 2017. A recurrent variational autoencoder for human motion synthesis. In Proceedings of the British Machine Vision Conference (BMVC’17). BMVA Press, Durham, UK, Article 119, 12 pages. https://doi.org/10.5244/C.31.119.

[64] Guy Tevet, Brian Gordon, Amir Hertz, Amit H. Bermano, and Daniel Cohen-or. 2022. MotionCLIP: Exposing Human Motion Generation to CLIP Space. arXiv preprint arXiv: 2203.08063.

[65] Jogendra Nath Kundu, Maharshi Gor, and R Venkatesh Babu. 2018. BiHMP-GAN: Bidirectional 3D Human Motion Prediction GAN. arXiv preprint arXiv:1812.02591 (2018).

[66] Makoto Murakami and Takahiro Ikezawa. 2022. Human Motion Generation Using Variational long Network. In 2022 6th International Conference on Digital Signal Processing (ICDSP) (ICDSP 2022), February 25–27, 2022, Chengdu, China. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3529570.3529588.

[67] Clinton Mo, Kun Hu, Shaohui Mei, Zebin Chen, and Zhiyong Wang. 2021. Keyframe Extraction from Motion Capture Sequences with Graph based Deep Reinforcement Learning. In Proceedings of the 29th ACM International Conference on Multimedia (MM '21). Association for Computing Machinery, New York, NY, USA, 5194–5202. https://doi.org/10.1145/3474085.3475635.

[68] Moonwon Yu, Byungjun Kwon, Jongmin Kim, Shinjin Kang, and Hanyoung Jang. 2019. Fast Terrain-Adaptive Motion Generation using Deep Neural Networks. In SIGGRAPH Asia 2019 Technical Briefs (SA ’19 Technical Briefs), November 17–20, 2019, Brisbane, QLD, Australia. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3355088.3365157.

[69] Xue Bin Peng, Glen Berseth, and Michiel van de Panne. 2015. Dynamic terrain traversal skills using reinforcement learning. ACM Trans. Graph. 34, 4, Article 80 (August 2015), 11 pages. https://doi.org/10.1145/2766910.

[70] Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. 2019. Neural State Machine for Character-Scene Interactions. ACM Trans. Graph. 38, 6, Article 178 (November 2019), 14 pages. https://doi.org/10.1145/3355089.

[71] Shuaiying Hou, Hongyu Tao, Hujun Bao, and Weiwei Xu. 2023. A Two-part Transformer Network for Controllable Motion Synthesis. arXiv: 2304.12571.

[72] Josh Merel, Leonard Hasenclever, Alexandre Galashov, Arun Ahuja, Vu Pham, Greg Wayne, Yee Whye Teh, and Nicolas Heess. 2018. Neural probabilistic motor primitives for humanoid control. arXiv preprint arXiv: 1811.11711 (2018).

[73] Zherong Pan and Dinesh Manocha. 2018. Active Animations of Reduced Deformable Models with Environment Interactions. ACM Trans. Graph. 37, 3, Article 36 (August 2018), 17 pages. https://doi.org/10.1145/3197565.

[74] Daniel Holden, Jun Saito, and Taku Komura. 2016. A deep learning framework for character motion synthesis and editing. ACM Trans. Graph. 35, 4, Article 138 (July 2016), 11 pages. https://doi.org/10.1145/2897824.2925975.

[75] Jungdam Won, Jungnam Park, and Jehee Lee. 2018. Aerobatics Control of Flying Creatures via Self-Regulated Learning. ACM Trans. Graph. 37, 6, Article 181 (November 2018), 10 pages. https://doi.org/10.1145/3272127.3275023.

[76] Soohwan Park, Hoseok Ryu, Seyoung Lee, Sunmin Lee, and Jehee Lee. 2019. Learning predict-and-simulate policies from unorganized human motion data. ACM Trans. Graph. 38, 6, Article 205 (November 2019), 11 pages. https: //doi.org/10.1145/3355089.3356501.

[77] Xue Bin Peng, Glen Berseth, and Michiel van de Panne. 2015. Dynamic terrain traversal skills using reinforcement learning. ACM Trans. Graph. 34, 4, Article 80 (August 2015), 11 pages. https://doi.org/10.1145/2766910.

[78] Xue Bin Peng, Glen Berseth, and Michiel van de Panne. 2016. Terrain-adaptive locomotion skills using deep reinforcement learning. ACM Trans. Graph. 35, 4, Article 81 (July 2016), 12 pages. https://doi.org/10.1145/2897824.2925881.

[79] Xue Bin Peng and Michiel van de Panne. 2016. Learning Locomotion Skills Using DeepRL: Does the Choice of Action Space Matter? arXiv preprint arXiv: 1611.01055 (2016). http://arxiv.org/abs/1611.01055.

[80] Wenhao Yu, Greg Turk, and C.Karen Liu. 2018. Learning Symmetric and Low-Energy Locomotion. ACM Trans. Graph. 37, 4, Article 144 (August 2018), 12 pages. https://doi.org/10.1145/3197517.3201397.

[81] Jungdam Won, Jongho Park, Kwanyu Kim, and Jehee Lee. 2017. How to Train Your Dragon: Example-Guided Control of Flapping Flight. ACM Trans. Graph. 36, 4, Article 198 (July 2017), 13 pages. DOI: 10.1145/3130800.3130833.

[82] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. arXiv preprint arXiv: 1509.02971 (2015).

[83] Kyungho Lee, Sehee Min, Sunmin Lee, and Jehee Lee. 2021. Learning TimeCritical Responses for Interactive Character Control. ACM Trans. Graph. 40, 4, Article 147 (August 2021), 11 pages. https://doi.org/10.1145/3450626.3459826.