Puppeteering AI - Interactive Control of an Artificial Dancer

ANON, anon, anon

ANON, anon, anon

Generative machine learning models for creative purposes play an increasingly prominent role in the field of dance and technology. A particularly popular approach is the use of such models for generating synthetic motions. Such motions can either serve as source of ideation for choreographers or control an artificial dancer that acts as improvisation partner for human dancers. Several examples employ autoencoder-based deep-learning architectures that have been trained on motion capture recordings of human dancers. Synthetic motions are then generated by navigating the autoencoder's latent space. This paper proposes an alternative approach of using an autoencoder for creating synthetic motions. This approach controls the generation of synthetic motions on the level of the motion itself rather than its encoding. Two different methods are presented that follow this principle. Both methods are based on the interactive control of a single joint of an artificial dancer while the other joints remain under the control of the autoencoder. The first method combines the control of the orientation of a joint with iterative autoencoding. The second method combines the control of the target position of a joint with forward kinematics and the application of latent difference vectors. As illustrative example of an artistic application, this latter method is used for an artificial dancer that plays a digital instrument. The paper presents the implementation of these two methods and provides some preliminary results.

CCS Concepts: • **Applied computing → Performing arts**; • **Human-centered computing** → *Interactive systems and tools*; • **Computing methodologies → Machine learning algorithms**.

Additional Key Words and Phrases: dance and technology, artificial dancer, motion synthesis, deep learning

ACM Reference Format:
anon and anon. 2022. Puppeteering AI - Interactive Control of an Artificial Dancer. In *CHI 2022: ACM CHI Conference on Human Factors in Computing Systems, April 30–May 05, 2022, New Orleans, LA.* ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Recent progress in machine learning (ML) has led to the adoption and proliferation of data driven techniques for artistic applications. In particular, the application of generative ML systems for the purpose of creating imagery and music have garnered much public visibility. Nevertheless, the field of dance has an enthusiastic community of its own that experiments with creative uses of generative ML. Within dance, two prominent lines of experimentation involve the use of generative ML systems to assist the ideation of a choreographer and the creation of artificial dancers with whom human dancers can improvise with during rehearsal or performance. In both cases, ML systems are typically trained on pose or motion material recorded from human dancers and then subsequently used to create synthetic poses or motions. Despite the technical similarity of these approaches, their integration into the creative workflow poses different challenges for choreographers and dancers. A choreographer can explore a system's capability to produce potentially interesting motions by navigating it's solution space in an undirected and open ended manner. Later on, the choreographer can then freely select, combine and modify the obtained synthetic motions. A dancer who is improvising in real-time with an ML-based artificial dancer cannot rely on rare chance occasions of obtaining interesting synthetic motions. Rather, he or she needs to acquire at least to some degree an understanding for how an ML-system can be guided towards promising regions of its solution space and how to anticipate the synthetic motions that can be obtained from there.

The project *Puppeteering AI* addresses some of these challenges by adapting a ML system that has originally been developed for choreographic purposes with methods for real-time interaction. The original system entitled *anon* employs an autoencoder architecture to synthesize short pose sequences in combination with a blending and sequencing mechanism for concatenating the generated sequences[anon]. For this system, a varitey of techniques have been proposed for creating synthetic pose sequences all of which are based on a navigation of the autoencoder's latent space. Unfortunately, the application of these techniques are hampered by the fact that it is difficult to gain an intuitive understanding for the organization of the latent space. The reason for this is that the relationship between motion data and its encoding is intractable and the spatial organization of the latent space is rarely related to perceptual aspects of the data.

To circumvent this issue, *Puppeteering AI* follows an alternative approach. This approach is based on controlling the generation of synthetic motions on the level of the motion itself rather than its encoding. The chosen approach is reminiscent of puppeteering, in that a human performer manipulates a small subset of the joints of an artificial character, while its remaining joints remain under the control of other processes. In traditional puppeteering, these other processes are based on physics, whereas in *Puppeteering AI* they are based on autoencoding.

In this paper, two different methods are presented that follow this principle. Both methods are based on the interactive control of a single joint of an artificial dancer while the other joints remain under the control of the autoencoder. The first method combines the control of the orientation of joint with iterative autoencoding. The second method combines the control of the target position of a joint with forward kinematics and the application of latent difference vectors.

The paper also presents a combination of *Puppeteering AI* with a physical modeling based sonification system. The sonification system operates as a virtual musical instrument that the artificial dancer plays. This approach illustrates the possibility of integrating the *Puppeteering AI* system into an artistic performance settings than combines dance and music.

2 RELATED WORK

With recent progress in machine learning, the use of data-driven methods for creating synthetic motion has gained in popularity[16]. Data driven methods have proven effective in generating synthetic motions that look natural and expressive. Data-driven motion synthesis techniques have also been adopted in the context of dance[9] for capturing and imitating the idiosyncratic movement material of individual choreographers and dancers.

Several ML-systems based on neural networks have been proposed as co-creative tools for choreographers. A system entitled Chor-rnn implements a recurrent neural network[3]. The authors of this system suggest a collaborative workflow in which choreographer and tool take turns in creating motion material. In two publications, different deep-learning architectures are compared with regard to their usefulness for choreographic purposes. Based on a subjective evaluation of mixed density networks, autoencoders, and Long-Short Term Memory (LSTM) networks, the authors conclude that LSTMs perform best on criteria such as posture prediction, temporal coherence, motion consistency, and aesthetics[10]. Another comparison between autoencoders and LSTMs places a stronger focus on the flexibility of creating motion variations[17]. This comparison ends up given more attention to autoencoders than LSTMs.

Machine-learning systems have also been employed to realize interactive artificial dancers. Such an approach has been thoroughly explored by Berman and James[1, 2, 4] and McCormick and colleagues[12–14]. Both teams experimented with a variety of machine learning techniques to obtain a system capable of synthesizing motions that are similar to those of a human dancer. The *LuminAI* system employs a clustering mechanism to select motions for an artificial dancer that mirror with some deviations those of an interacting human dancer[11]. The *Viewpoints AI* system implements an interaction-based authoring approach for creation synthetic motions that combines ideas from case-based and imitative learning[8].

When working with autoencoders, navigating the latent space of encodings is a popular method for creating new motion material. This approach has been chosen both by researchers working with encodings of poses (e.g. [2, 10, 17]) and researchers working with encodings of pose sequences (e.g. [5–7]). One of the main drawbacks of navigation latent space pertains to the difficulty of obtaining an understanding for the typically obscure relationship between latent vectors and their decoding.

3 IMPLEMENTATION

Puppeteering AI employs an architecture that has been previously developed for a choreographic tool entitled —-[anon]. This architecture remains mostly unchanged and is briefly summarized. The physical modeling synthesis system has been previously developed for a music performance entitled *anon*[anon]. This system is also largely unchanged and therefore only briefly summarized. What has changed in the meantime is the implementation of *anon* in a form that is suitable for creating synthetic motions in real time and its combination with two different methods for interactive control. These changes are introduced in more detail.

3.1 *anon*

The ML part of *anon* consists of an adversarial autoencoder that has been trained on motion capture (mocap) recordings of a professional dancer who freely improvised to music. The model's architecture consists of three neural networks, an encoder, decoder, and discriminator (figure 1 left side). The encoder and decoder parts are autoregressive and operate on sequences of poses. The encoder takes as input a sequence of poses and generates a low dimensional representation of this sequence in the form of a latent vector. The decoder takes as input a latent vector and reconstructs a sequence of poses. The discriminator takes as input a latent vector and produces as output a binary value indicating its assessment of whether the vector follows a Gaussian distribution or not. This mechanism ensures that the space of latent vectors is free of gaps and that distances within it represent a measure of similarity.

The sequence blending mechanism concatenates the short pose sequences generated by the autoencoder into longer sequences (figure 1 right side). The mechanism draws inspiration from methods in computer music that combine short sound fragments to generate longer sounds: *Granular Synthesis*[18] and *Concatenative Synthesis*[19, 21]. Since poses are represented by joint orientations in the form of unit quaternions, their blending is based on spherical linear interpolation (SLERP)[20]. Source sequences are blended one after the other with a target sequence with which they are aligned at successively increasing positions. The amount of blending between source and target is controlled by a window function (*Hanning* in this case).

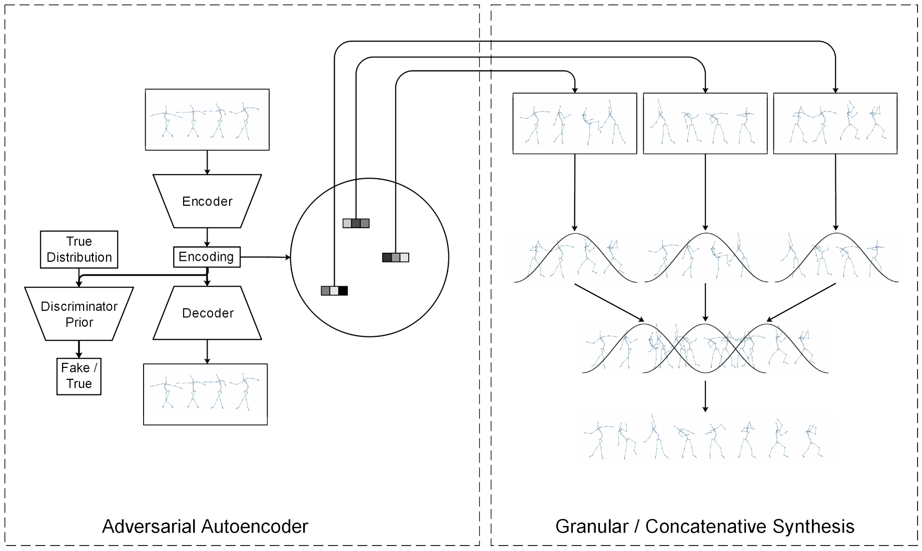

Fig. 1. *anon* System architecture. Shown on the left side is a schematic representation of the machine learning model depicting its three neural networks: an encoder, a decoder, and a discriminator. The trapezoid shapes of the encoder and decoder indicate the dimension reduction and expansion that is conducted by them. The circular region represents the latent space of pose sequence encodings. Shown on the right side is the pose sequence blending mechanism that mimics granular and concatenative synthesis techniques. Sequences are obtained by decoding sampled latent space vectors. Multiple decoded sequences are blended into longer sequences using overlapping windows which are depicted here as bell shaped curves.

3.2 *Puppeteering AI*

For creating synthetic motions for an artificial dancer, the original *anon* system has been modified for real-time application. As part of this adaptation, the system has been ported from *Tensorflow*[1] to the C++ API of *PyTorch*[2]. Furthermore, the length of the pose sequences the model operates on was reduced to 8 (as opposed to 128 used in some of the earlier experiments) which corresponds to a duration of 0.16 seconds.

Synthetic motions of the artificial dancer are created by autoencoding pose sequences obtained from a motion capture recording. Interaction is based on two different methods that interfere with a faithful reproduction of the original pose sequences. These methods provide interactive control of the orientation or position of a single joint, respectively.

3.2.1 Joint Orientation Control. This method controls the orientation of a selected joint relative to its parent joint. Since poses are represented as joint orientations, the input pose sequence used for autoencoding can be directly overwritten. By autoencoding a modified pose sequence, the interactively controlled joint orientations affect the orientations of the remaining joints. By iterating the autoencoding process several times while keeping the orientation of the interactively controlled joint fixed, the autoencoder convergences towards a pose sequence that is representative of both the original mocap material and the interactively controlled joint orientation. The number of iterations used for autoencoding and the number of iterations during which the interactively controlled joint is fixed do not necessarily have to be the same. If they are the same, then the joint orientation specified through interaction is fully visible in the final pose sequence. If they are not, then the interactively controlled joint orientation changes as result of the autoencoding process.

[1]Tensorflow: https://www.tensorflow.org/
[2]PyTorch: https://pytorch.org/

3.2.2 Joint Position Control. This method controls the position of a selected joint relative to the position of a root joint (the hip). Since poses are represented by orientations, the positions of joints need to be obtained through forward kinematics. The encodings of all those pose sequences are collected in which the selected joint is either close to its current position, or close to the intended target position. In order to maintain real-time performance, the encodings of all pose sequences in a given motion capture recording are precomputed and stored alongside the positions of the selected joint. These positions are organized in a spatial partitioning structure (a KD-Tree in this case) to quickly retrieve the encodings based on the Euclidean distance between joint positions. Following this principle, a fixed number of encodings is obtained for the current position and the target position of a selected joint, respectively. The encodings are then averaged and subtracted from each other to obtain the vector difference to be added to the encoding of the current pose sequence. The modified encoding is then decoded to obtain a pose sequence in which the selected joint is close to the intended target position.

3.3 Digital Instrument

To experiment with the application of an artificial dancer in a performance situation that combines dance and music, a digital instrument has been developed. The instrument employs physical modeling synthesis that simulates a vibrating surface via a bank of resonating filters following a modal synthesis approach[15].

For *Puppeteering AI*, the filters are arranged in a cylindrical formation consisting of ten vertically stacked rings with ten filters each. Neighboring filters are acoustically connected both horizontally and vertically, with the former creating closed loops. Limiters change the amplitude of the audio signals traveling between filters and thereby control their amount of excitation and the level of feedback created in the closed loops. This formation surrounds the artificial dancer at a distance that can be reached by its extremities(see figure 6). Playing this instrument is based on the artificial dancer approaching the filters with some of its joints. Those filters that are sufficiently close to the dancer's joints are excited through an external audio signal. External audio signals employed here include impulses, impulse chains, and pink noise with a variable amplitude envelope.

4 EXPERIMENTS

The two interaction mechanisms have been tested by either setting the orientation or target position of a joint to a fixed value while simultaneously varying the parameters controlling the influence of interaction on the autoencoding process, or vice versa, by keeping the parameters fixed while changing the orientation or target position of a joint.

To evaluate the effect of the parameters for joint orientation control, the number of iterations for autoencoding and for overwriting the orientation of a selected joint were varied within range from 1 to 10 for each (see figure 2). The *right shoulder* was chosen as selected joint and its relative orientation was set to the quaternion equivalent of the Euler angles 0.0°, 0.0°, -120°.

To evaluate the effect of the joint orientation, the third Euler angle was varied between 180.0°and -120,0°while the first two Euler angles were kept constant at 0.0°. The control parameters were set to 5 iterations both for autoencoding and for overwriting the orientation of the selected joint. Again, the *right shoulder* was chosen as selected joint.

To evaluate the effect of the parameters for joint position control, the number of neighboring joint positions for calculating the latent vector difference between their corresponding pose sequence encodings, and the scaling of the latent vector difference were varied, with the former parameter ranging from 2 to 20 and the latter ranging from -1.0 to +1.0 (see figure 2). The *right hand* was chosen as selected joint and its target position relative to the hip joint was set to x -60.0, y 0.0, z 0.0.

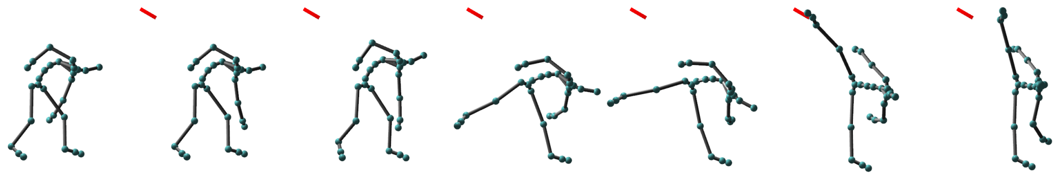
Fig. 2. Variation of Joint Orientation Control Parameters. A graphical representation of the artificial dancer is shown alongside an elongated red box that indicates the orientation of the selected joint (the right shoulder). The following abbreviations are used: *enc* stands for the number of autoencoding iterations, *int* stands for the number of iterations during which the joint orientation is overwritten with an interactively controlled value. From left to right: original mocap, enc 1 int 1, enc 5 int 0, enc 5 int 3, enc 5 int 5, enc 10 int 6, enc 10 int 10

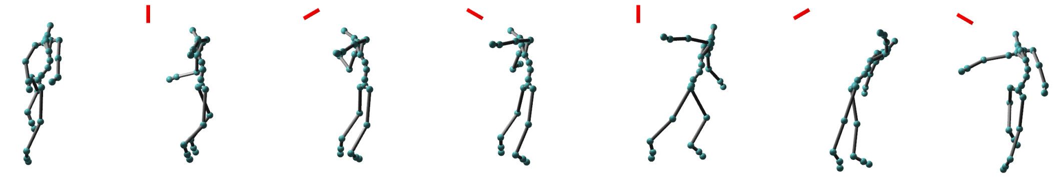
Fig. 3. Variation of Joint Orientation. A graphical representation of the artificial dancer is shown alongside an elongated red box that indicates the orientation of the target joint (the right shoulder). From left to right, the third Euler angle is set to: original mocap, 180.0°, 120.0°, 60.0°, 0.0°, -60.0°, -120.0°

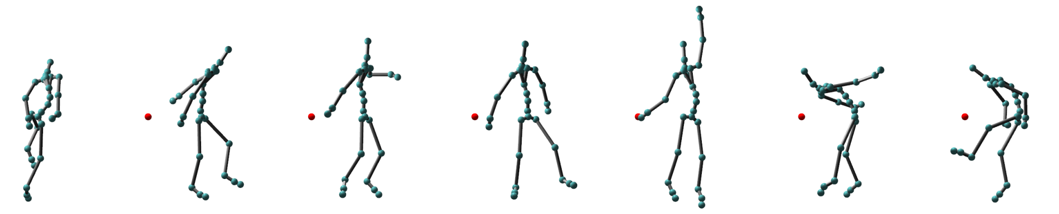
Fig. 4. Variation of Joint Target Position Control Parameters. A graphical representation of the artificial dancer is shown alongside a red ball that indicates the target position of the selected joint (the right hand). The following abbreviations are used: *png* stands for the number of position neighbors, *sca* stands for scaling of the latent vector difference. From left to right: original mocap, png 20 sca 0.5, png 2 sca 0.5, png 20 sca 1.0, png 2 sca 1.0, png 20 sca -1.0, png 2 sca -1.0

The evaluate the effect of the relative target joint position, the x and y coordinates where varied between -60.0 and +60.0, and -120.0 and +120.0, respectively while the z-coordinte was kept constant at 0.0. The control parameters were set to 20 for the number of neighboring joint positions and 1.0 for the scaling of the latent vector difference. Again, the *right hand* was chosen as selected joint.

The digital instrument was evaluated less systematically with an interaction setup that could potentially be used in a live performance. This setup uses a wearable sensor for controlling the target position of a single joint and a Midi-controller for changing parameters of the physical modeling synthesis system. The wearable sensor [3] measures absolute orientation. This orientation is converted into cylindrical coordinates and discretized so that the joint target position is superimposed on the location of the instrument's filters(see figure 6).

[3]Bosch BNO055: https://www.bosch-sensortec.com/products/smart-sensors/bno055/

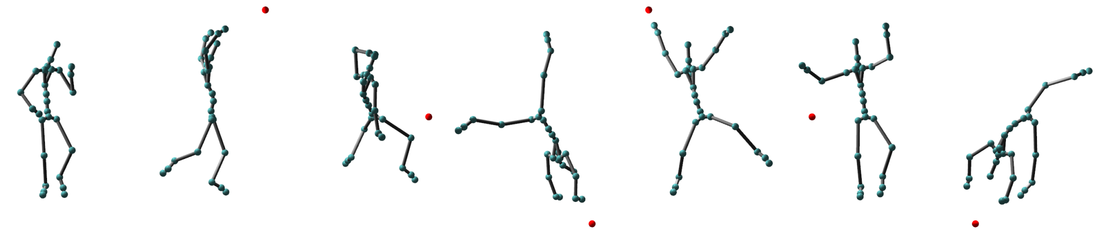

Fig. 5. Variation of Target Joint Position. A graphical representation of the artificial dancer is shown alongside a red ball that indicates the target position of the selected joint (the right hand). From left to right, the x and y coordinates of the target position are set to: original mocap, x60.0 y120.0, x60.0 y0.0, x60.0 y -120.0, x-60.0 y120.0, x-60 y0.0, x-60.0 y-120.0

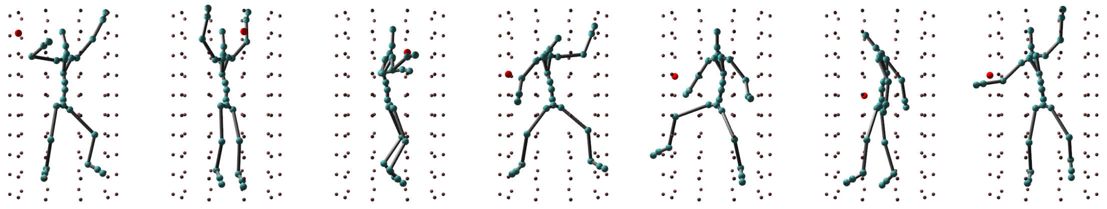

Fig. 6. Artificial Dancer with Digital Instrument. A graphical representation of the artificial dancer surrounded by small balls representing the filters that form part of a digital instrument. The images are successive stills of a screen recording that is available online as animation.

5 RESULTS AND DISCUSSION

The experiments conducted so far provide some preliminary insights into the capability of the two interaction methods to influence the motion of an artificial dancer by means of a single interactively controlled joint.

The following insights have been gained concerning the interactive control of joint orientation in combination with iterative autoencoding. Iterative autoencoding alone without interaction alters the original pose sequence only marginally, especially when the number of iterations is kept low (up to five) and the chosen pose sequence doesn't contain any material that seldom appears in the motion capture material used for training. Iterative autoencoding in conjunction with interaction clearly affects the overall motion of an artificial dancer through the control of a single joint. This effect is small for a low number of autoencoding and control iterations but becomes very prominent as the number of these iterations increases. Accordingly, by altering the number of iterations, both the sensitivity of the artificial dancer to interaction and the balance between original and novel motion can be shifted. The orientation of the interactively controlled joint is only visible in the final synthetic motion if it is prevented from being modified by the autoencoder. Otherwise, the autoencoder overwrites this orientation with one that better matches the currently synthesized motion. This effect is particularly pronounced if the interactively controlled joint exhibits an unrealistic orientation.

The following insights have been gained concerning the interactive control of joint position in combination with forward kinematics and latent vector difference calculation. Changing the scale of the latent vector difference that is added to the encoding of the current pose clearly affects the influence of interaction on the resulting synthetic motions, with larger absolute scale values leading to stronger effects. As long as the value of the scale doesn't exceed a range from -1.0 to +1.0, the resulting synthetic motions remain realistic, even when the target position lies beyond reach for the artificial dancer. A positive scale causes the artificial dancer to approach the target position, whereas a negative scale makes it move away from the target position. Whether a target position can be reached depends not only on the distance of the position but also on the currently selected pose sequence for autoencoding. If the initial pose sequence places the interactively controlled joint close to the target position, then the synthetically created motion is very similar to the original pose sequence with only the limb containing the selected joint altered. If the interactively controlled joint lies far from the target position, the entire orientation and pose of the artificial dancer is strongly modified, leading to a synthetic motion that is more dissimilar from the original pose sequence. Changing the number of encodings that are used to calculate the latent difference vector has a more nuanced effect. For settings in which a target position can easily be reached, a low number of encodings (e.g. 2) tends to bring the the interactively controlled joint closer (or further away for negative scale) than a high number of encodings (e.g. 20). At the same time, a low number of encodings causes the overall synthetic motion to deviate more strongly from the original pose sequence, whereas a high number of encodings focuses deviation on the limb containing the interactively controlled joint.

The combination of an artificial dancer and a digital instrument has been evaluated by experimenting with different parameters for controlling joint target position and sound synthesis. The results of these experiments are available online as audiovisual screen recordings[4]. In all these experiments, the input pose sequences are chosen from a short excerpt of the original mocap recording. This causes the artificial dancer to exhibit partially periodic motions which in turn manifest as recurring melodic patterns in the acoustic output. The characteristics of these melodic patterns is dependent on the chosen joint target position. Frequently, the interactively controlled joint moves close enough to its target location to cause an excitation of a filter at this position. But occasionally, it is other joints of the artificial dancer that come into contact with filters that lie far away from the target position. Accordingly, interacting with an instrument playing artificial dancer leads to a musical result that is more varied than initially anticipated.

6 CONCLUSION

The results obtained so far indicate that both interaction methods are suitable for controlling an ML-based artificial dancer. Compared to approaches that are based on latent space navigation, the proposed methods are easier to comprehend and the resulting synthetic motions are more predictable. Furthermore, both methods work well in combination with interfaces such as wearable sensors whose output is of significantly lower dimension that the degrees of freedom of an artificial dancer. The use of a wearable sensor for controlling an artificial dancer playing a digital instrument illustrates an example artistic application scenario for such a setup.

There is still a lot of room for further research. In a next step, the authors plan to explore in more depth how the characteristics of a chosen initial pose sequence affects the type and range of synthetic motions than can be generated through interaction. The authors also plan to study how other (and possible multiple) selections of interaction joints affect synthetic motions. So far, each interaction method has been used in isolation. It would be interesting to experiment with running both interaction methods concurrently and in a complementary manner. And last but not least, the authors intend to test *Puppeteering AI* with a dancer. Letting a professional dancer experiment with this system and develop small choreographies for it will generate additional insights and thereby help with the system's future development.

REFERENCES

[1] Alexander Berman and Valencia James. 2014. Towards a live dance improvisation between an avatar and a human dancer. In *Proceedings of the 2014 International Workshop on Movement and Computing*. 162–165.

[4] *Puppeteering AI* and Digital Instrument: Animation 1, Animation 2, Animation 3, Animation 4

[2] Alexander Berman and Valencia James. 2018. Learning as Performance: Autoencoding and Generating Dance Movements in Real Time. In *International Conference on Computational Intelligence in Music, Sound, Art and Design*. Springer, 256–266.

[3] Luka Crnkovic-Friis and Louise Crnkovic-Friis. 2016. Generative choreography using deep learning. *arXiv preprint arXiv:1605.06921* (2016).

[4] Marco Gillies, Rebecca Fiebrink, Atau Tanaka, Jérémie Garcia, Frédéric Bevilacqua, Alexis Heloir, Fabrizio Nunnari, Wendy Mackay, Saleema Amershi, Bongshin Lee, et al. 2016. Human-centred machine learning. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. 3558–3565.

[5] Ikhsanul Habibie, Daniel Holden, Jonathan Schwarz, Joe Yearsley, and Taku Komura. 2017. A recurrent variational autoencoder for human motion synthesis. In *28th British Machine Vision Conference.*

[6] Daniel Holden, Jun Saito, and Taku Komura. 2016. A deep learning framework for character motion synthesis and editing. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–11.

[7] Daniel Holden, Jun Saito, Taku Komura, and Thomas Joyce. 2015. Learning motion manifolds with convolutional autoencoders. In *SIGGRAPH Asia 2015 Technical Briefs*. 1–4.

[8] Mikhail Jacob and Brian Magerko. 2015. Interaction-based Authoring for Scalable Co-creative Agents.. In *ICCC*. 236–243.

[9] Manish Joshi and Sangeeta Chakrabarty. 2021. An extensive review of computational dance automation techniques and applications. *Proceedings of the Royal Society A* 477, 2251 (2021), 20210071.

[10] Esbern Torgard Kaspersen, Dawid Górny, Cumhur Erkut, and George Palamas. 2020. Generative Choreographies: The Performance Dramaturgy of the Machine. In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications-Volume 1: GRAPP*. SCITEPRESS Digital Library, 319–326.

[11] Lucas Liu, Duri Long, Swar Gujrania, and Brian Magerko. 2019. Learning movement through human-computer co-creative improvisation. In *Proceedings of the 6th International Conference on Movement and Computing*. 1–8.

[12] John McCormick, Steph Hutchinson, Kim Vincs, and Jordan Beth Vincent. 2015. Emergent behaviour: learning from an artificially intelligent performing software agent. In *ISEA 2015: Proceedings of the 21st International Symposium on Electronic Art*. ISEA 2015, 1–4.

[13] John McCormick, Kim Vincs, Saeid Nahavandi, and Douglas Creighton. 2013-01-01. Learning to dance with a human. ISEA International; Australian Network for Art & Technology; University of Sydney.

[14] John McCormick, Kim Vincs, Saeid Nahavandi, Douglas Creighton, and Steph Hutchison. 2014. Teaching a digital performing agent: Artificial neural network and hidden markov model for recognising and performing dance movement. In *Proceedings of the 2014 International Workshop on Movement and Computing*. 70–75.

[15] Joseph Derek Morrison and Jean-Marie Adrien. 1993. Mosaic: A framework for modal synthesis. *Computer Music Journal* 17, 1 (1993), 45–56.

[16] Lucas Mourot, Ludovic Hoyet, François Le Clerc, François Schnitzler, and Pierre Hellier. 2021. A Survey on Deep Learning for Skeleton-Based Human Animation. In *Computer Graphics Forum*. Wiley Online Library.

[17] Mariel Pettee, Chase Shimmin, Douglas Duhaime, and Ilya Vidrin. 2019. Beyond imitation: Generative and variational choreography via machine learning. *arXiv preprint arXiv:1907.05297* (2019).

[18] Curtis Roads. 2004. *Microsound*. MIT press.

[19] Diemo Schwarz et al. 2004. Data-driven concatenative sound synthesis. (2004).

[20] Ken Shoemake. 1985. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*. 245–254.

[21] Aymeric Zils and François Pachet. 2001. Musical mosaicing. In *Digital Audio Effects (DAFx)*, Vol. 2. Citeseer, 135.