Raw Music from Free Movements: Early Experiments in Using Machine Learning to Create Raw Audio from Dance Movements

Daniel Bisig[1] and Kıvanç Tatar[2]

[1] Center for Dance Research, Coventry University, Coventry, United Kingdoms
ad5041@coventry.ac.uk
[2] Independent, Vancouver, B.C., Canada
contact@kivanctatar.com

Abstract. *Raw Music from Free Movements* is a deep learning architecture that translates pose sequences into audio waveforms. The architecture combines a *sequence-to-sequence* model generating audio encodings and an adversarial autoencoder that generates raw audio from audio encodings. Experiments have been conducted with two datasets: a dancer improvising freely to a given music, and music created through simple movement sonification. The paper presents preliminary results. These will hopefully lead closer towards a model which can learn from the creative decisions a dancer makes when translating music into movement and then follow these decisions reversely for the purpose of generating music from movement.

Keywords: movement computing, movement sonification, dance, deep learning, audio synthesis

1 Introduction

Music-Dance and *Dance-Music* practices explore artistic possibilities at the intersection of experimental music and contemporary dance. *Digital Music-Dance Instruments* (*DMDIs*) (Strauss, Tatar, & Nuro, n.d.) are a technology where a dancer controls an interactive music system (Tatar & Pasquier, 2019). As a multi-modal technology, *DMDIs* transform embodied gestures into sonic gestures. These technologies have been emerging in the literature, although they are not explicitly referred to as *DMDIs* (Erdem, Schia, & Jensenius, 2019; Tragtenberg, Calegario, Cabral, & Ramalho, 2019).

DMDIs draw from the rich expertise of dancers who explore improvisation within the practice of *Music-Dance*. Integrating the dancers' own intuitions, techniques, and knowledge about gestural expression into the design of *DMDI* is a challenging research task. The *Raw Music from Free Movements* (*RAMFEM*) system constitutes the authors' first attempt to design a *DMDI* system by starting from the creative decisions a dancer makes when translating music into movement and then reverse these decisions for the purpose of generating music from movement. An important aspect of *RAMFEM's* capability to learn from and recreate existing movement and music relationships is its operation in the raw audio domain. Because of this, *RAMFEM* can be applied to any recordings of movement and music, capture their correlations, and subsequently recreate the acoustic characteristics of the music through embodied gestures.

2 Background

Previously, Machine learning (ML) has been applied to multi-modal applications of dance and music. Although some examples exist for the creation of movement in response to a given music and for the control of music through movement, no research has been published on the use of ML for translating dance movement into raw audio, to the best of the authors' knowledge. However, previous research exists on the application of ML for the generation of raw audio. All these approaches form important backgrounds for this paper and are briefly surveyed.

2.1 ML-based Interactive Control of Music

Several examples of applying ML for controlling music through interaction are in the context of *Interactive Machine Learning.* This field proposes a new design paradigm for creating gestural interfaces (Gillies, 2019) that exploits tacit and embodied knowledge about movement. Many of these interfaces serve the creation of digital musical instruments (Fiebrink & Caramiaux, 2016). Several ML-based tools have been released for the artistic community such as *Wekinator* (Fiebrink & Cook, 2010) and *ml.lib* (Bullock & Momeni, 2015). These tools provide simple ML systems that learn from small training sets, operate in real-time, and integrate into creative workflows. Because of their simplicity, these ML systems can not directly generate raw audio. Rather, they are typically employed for mapping gestural input to control parameters for an external audio engine.

2.2 ML-based Generation of Raw Audio

Two ML architectures commonly used for the creation of raw audio are generative adversarial networks (GAN) and autoregressive (AR) systems. *WaveGAN* (Donahue, McAuley, & Puckette, 2018) is one of the first GANs for modeling audio waveforms. A more recent example is *GANSynth* (Engel et al., 2019) which operates directly on invertible spectra. A further example is *MelGAN* (Kumar et al., 2019) which is mainly used to invert mel-spectrograms for speech synthesis applications. *MelGAN* generates audio of higher fidelity than previous GANs.

AR systems have been used to create longer audio segments. These systems typically predict audio waveforms one sample at a time. Pioneering examples include *WaveNet* (Oord et al., 2016) and *SampleRNN* (Mehri et al., 2016). Both examples show excellent performance on text to speech tasks but struggle to capture the long-time structure of music. *Hierarchical Wavenet* (Dieleman, Oord, & Simonyan, 2018) is an extension of *Wavenet* that is better suited for music generation. It combines a *Wavenet* with a vector quantified variational autencoder (VQ-VAE) (Oord, Vinyals, & Kavukcuoglu, 2017) for separately learning higher and lower level music structures. *JukeBox* (Dhariwal et al., 2020) is a recent AR system that can create several minutes long audio waveforms. It uses a hierarchical VQ-VAE in combination with a transformer architecture. All these architectures are computationally heavy in training and inference.

2.3 ML-based Translation of Music into Movement

Several studies have explored the application of ML for translating music into movement. Tang et al. describe an AR system trained on audio and motion capture recordings of professional dancers (Tang, Jia, & Mao, 2018). Training focuses on the identification and synchronisation of rhythms in music and dance. Lee et al. (2019) describe an AR system that is trained on existing music videos. Their approach also focuses on the synchronisation of rhythms in music and dance. Ren et al. (2019) employ a GAN that learns from dance videos and tries to match the rhythm of music and movement and their emotional characteristics. Sun et al. (2020) employ a GAN that is trained on existing dance videos, while using a similarity metrics between original and generated movement. Qi et al. (2019) trained different sequence to sequence transducer (Seq2Seq) architectures on music videos. They found a Seq2Seq architecture with self-attention to perform best. A sophisticated Seq2Seq architecture has been presented by Li et al. (2020). This architecture combines two transformer models, one for capturing movement and one for capturing musical context. These architecture have been trained on a wide range of objectives such as beat synchronisation, physical plausibility, and diversity of movements.

3 Implementation

The current architecture of *RAMFEM* consists of three components: an adversarial autoencoder (AAE), a sequence to sequence transducer (Seq2Seq), and an audio concatenation mechanism. The source code, trained models, and audio and motion capture data required for testing and training are available online [3] [4].

AAE improves Variational Autoencoders (VAEs) by replacing the Kullback–Leibler (KL) divergence term in the loss function with the introduction of discriminator networks. This eliminates the issues related to the KL-divergence multiplier in the loss function of VAEs (Kingma & Welling, 2019). The AAE in *RAMFEM* encodes and decodes short audio waveforms into and from latent vectors. In its current implementation, the waveforms are 256 samples long and the latent vectors have a dimension of 32. The AAE consists of four neural networks (Fig. 1, right side): audio encoder, audio decoder, audio discriminator, prior discriminator. The audio encoder consists of four 1D convolution (1D-Conv) layers followed

[3] `https://zenodo.org/record/4656086`

[4] `https://github.coventry.ac.uk/ad5041/RawMusicFromFreeMovements`

by one fully connected (FC) layer. The 1D-Conv layers employ a kernel size of 7 and a stride of 4, and their number of channels doubles with each layer starting at 32 and ending at 256. The FC layer contains 32 units. The decoder mirrors the encoder's layer arrangement and consists of one FC layer followed by four transposed 1D-Conv layers. The audio discriminator distinguishes between original and reconstructed waveforms. Its architecture is identical to that of the encoder with the introduction of an additional FC layer at the end that contains one unit. The prior discriminator distinguishes between latent vectors and random variables following a true Normal distribution.

The Seq2Seq takes a sequence of poses as input and translates them into a sequence of audio encodings. These encodings are passed to an audio decoder which transforms them into waveforms. The architecture consists of five neural networks (Fig. 1, left side): sequence encoder, sequence decoder, audio decoder, audio encoding discriminator, and audio discriminator. The audio decoder and audio discriminator are reused from the AAE. The sequence encoder and decoder are deterministic and don't possess an attention mechanism. The sequence encoder consists of three recurrent layers with 512 *Long short-term memory* (LSTM) (Hochreiter & Schmidhuber, 1997) units each. The sequence decoder consists of three recurrent layers with 512 LSTM units each and a last FC layer with 32 units. After a sequence encoding step, the hidden state of the sequence decoder is initialized with the hidden state of the sequence encoder, and the encoding of the first waveform is provided as first input to the sequence decoder. The audio encoding discriminator distinguishes between original audio encodings and predicted audio encodings. Its architecture consists of three FC layers with 32, 32, and 1 units.

The audio concatentation mechanism takes a sequence of waveforms, applies a *Hanning* window as amplitude envelope to each of them, and then concatenates them with a 50% overlap to create the final audio sequence.

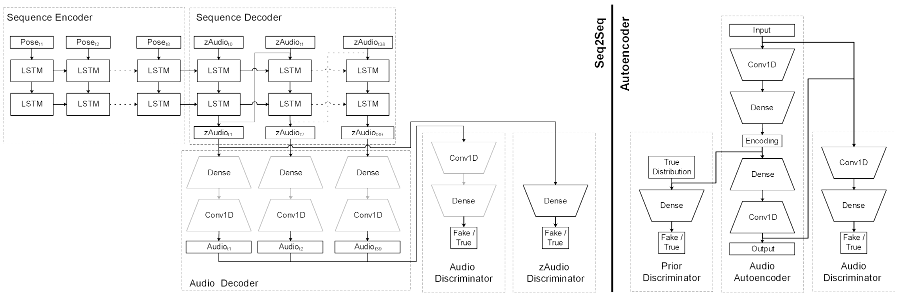

Fig. 1. The figure depicts the neural networks that form part of the sequence to sequence transducer (left side) and adversarial autoencoder (right side). Outlined shapes represent groups of layers with the same type of units. Shapes with dark outlines refer to layers whose weights are changed during Seq2Seq training. Shapes with light outlines refer to layers with fixed weights.

4 Dataset

Two different datasets were employed for training, named improvisation dataset and sonification dataset. The improvisation dataset consists of pose sequences and audio that have been recorded while a dancer was freely improvising to a given music. The dancer is an expert with a specialisation in contemporary dance and improvisation. The music consists of short excerpts of royalty free music including experimental electronic music, free jazz, and contemporary classic. The pose sequences have been acquired using the markerless motion capture system (*The Captury*) in the *iLab* at MotionBank, University for Applied Research, Mainz. The recording is 10 minutes in length which corresponds to a sequence of 30000 poses. Each pose consists of 29 joints whose relative orientations are represented by quaternions.

The sonification dataset contains the same pose sequences as the improvisation dataset. The audio of this dataset was created afterwards, through sonification, employing a very simple sound synthesis consisting of two sine oscillators controlled by the dancer's hands. The frequency and amplitude of each oscillator are proportional to the height and velocity of the corresponding hand, respectively. The authors created this dataset to verify the performance of *RAMFEM*.

5 Training

The training includes two stages where the AAE architecture and Seq2Seq architecture are trained in isolation.

For training the AAE, audio was mixed down from stereo to mono, equalized, re-sampled to 32000 Hz, and split into short overlapping excerpts of 256 samples in length. The data was split into an 80% training and 20% validation set. The autoencoder was trained in alternation with its two discriminators. Training progressed for 300 epochs using the Adam optimizer. The learning rate for the discriminators was kept constant at 4e-4. The learning rate for the autoencoder was kept constant at 1e-4 for the first 200 epochs and then reduced to 1e-5 for the second 100 epochs. The following loss functions were used: Categorical cross entropy for adversarial loss and mean square error for audio reconstruction loss.

For training the Seq2Seq, the pose sequence was split into overlapping excerpts with a length of 8 poses, each pose excerpt was paired with a sequence of 40 audio waveforms, each having 256 samples with 50% overlap to the next waveform, then the waveforms were encoded into latent vectors. The data was split into an 80% training and 20% validation set. During training, the weights of the pre-trained audio decoder and audio discriminator were kept fixed. The sequence encoder and decoder were trained in alternation with the audio encoding discriminator. Training progressed for 300 epochs using the Adam optimizer. The learning rate for the discriminators was kept constant at 4e-4. The learning rate for the sequence encoder and decoder was kept constant at 1e-4 for the first 200 epochs and then reduced to 1e-5 for the second 100 epochs. The following loss functions were used: Categorical cross entropy for adversarial loss and mean square error for audio reconstruction loss and audio encoding reconstruction loss.

A PC running Ubuntu 20.04 and equipped with an Intel i9-10900K CPU and a single Nvidia TITAN RTX GPU was used for training and inference. On this machine, training took about 15 hours for the AAE and 5 hours for the Seq2Seq. For inference with the Seq2Seq and the Decoder part of the AAE running in sequence, the translation of a pose sequence into 20 seconds of audio took about 130 seconds.

6 Results and Discussion

Examples illustrating the capabilities and shortcomings of the trained models are provided online[5]. The capability of the AAE to encode and reconstruct audio waveforms was evaluated qualitatively by comparing the original and reconstructed audio (Fig. 2, left and middle column). The comparisons show that the model reconstructs audio below 1000 Hz. Beyond this frequency, some spectral content is lost. The sonification dataset doesn't contain frequencies above 800 Hz, so the original and reconstructed audio are almost indistinguishable. For the improvisation dataset, the loss in high frequency content is clearly perceivable.

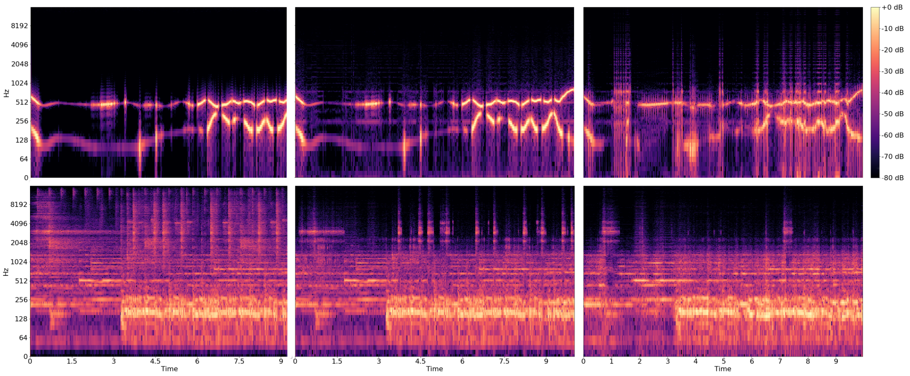

Fig. 2. The figure depicts log-frequency power spectrograms of the original audio (left) and audio reconstructed by the adversarial autoencoder (middle) and the sequence to sequence transducer (right). The audio is part of the sonification (top) and improvisation (bottom) datatsets.

The results obtained from training the Seq2Seq were more diverse. Two types of qualitative evaluations have been conducted: a comparison between the original and predicted audio using pose sequences from the training dataset (Fig. 2, left and right column), and a comparison between the predicted audio and a visual rendering of the pose sequences that are not from the training dataset.

The Seq2Seq trained on the sonification dataset generated audio that is acoustically similar to the original audio. The authors observed these results

[5] `https://zenodo.org/record/4656044`

with pose sequences from the training dataset, and also with pose sequences outside of the training dataset. Unfortunately, the quality of the predicted audio is low and includes noise, clicks, amplitude fluctuations, and dominant low frequencies. When the Seq2Seq was provided with pose sequences that were not in the dataset, the model generated audio that maintains the original acoustic characteristics. Interestingly, the model also preserved the original correlation between movement and music generating frequency sweeps as the hands change their height and changing amplitude in proportion to the velocity of the hands.

The results are similar for the Seq2Seq trained on the improvisation dataset when using pose sequences from the training dataset as input. The predicted audio is of low quality but acoustically similar to the original and the correlation between movement and audio is preserved. When the Seq2Seq was provided with pose sequences that were not in the dataset, the results were more difficult to interpret. Some of the acoustic elements from the original audio are still present but appear to lack a clear temporal structure. Some correlations between movement and audio are occasionally perceivable but only during moments when the limbs travel fast and far.

There are at least two takeaways from these findings. The lower quality of audio generated by the combination of Seq2Seq and AAE compared to the AAE alone indicates that it is inadequate to use the same audio reconstruction loss in both cases. The weak and likely inconsistent correlation between movement and music during free improvisation is likely one reason for the lackluster performance of the Seq2Seq trained on the improvisation dataset when provided with novel pose sequences as input. Both these issues are addressed in section (7).

7 Conclusions and Future Directions

It is encouraging that *RAMFEM* grasps some of the acoustic properties and correlations between movement and music with its current simple architecture. The authors plan to continue this research along three trajectories: experimentation with additional datasets, extended evaluation including interviews with dancers, and improvements to loss functions and model architectures.

Currently, the datasets used for training cover two extremes: very simple music with an equally simple movement correlation versus complex music with a possibly inconsistent movement correlation. It's worthwhile to create new datasets that lie in between these two extremes. A first group of datasets will be based on more sophisticated sonifications but maintain a simple correlation with movement. These datasets will be useful for verifying further iterations in the ML architecture design. A second group of datasets will be based on recordings of dancers improvising in a more controlled manner to music. This involves for each recording working with a subset of music and improvisation principles that are proposed by the dancers and subsequently maintained.

So far, evaluation has been conducted by the authors themselves. It is planned to incorporate the dancers' feedback into the evaluation. In particular, the evaluation of the generated music will take into account the dancer's reported principles of relating movement to music. It will be interesting to see which of these principles are learned by the models, and if the models expose some principles that the dancer wasn't initially aware of following.

The current model architectures and loss functions have been chosen for their simplicity. Changing the audio reconstruction loss will likely improve the quality of the audio generated by the Seq2Seq. For comparing waveforms, loss functions based on *Differentiable Time Warping* (Cuturi & Blondel, 2017) or *Earth Mover's Distance* (Arjovsky, Chintala, & Bottou, 2017) have been suggested (Purwins et al., 2019). Alternatively, audio reconstruction loss could be based on criteria that are better aligned with human auditory perception (Ananthabhotla, Ewert, & Paradiso, 2019)(Manocha et al., 2020)(Steinmetz & Reiss, 2020)(Wright & Välimäki, 2020).

The 1D-Convolutions that form part of the AAE could be replaced by differentiable digital signal processing (DDSP) components (Engel, Hantrakul, Gu, & Roberts, 2020). By integrating DDSP components directly into the AAE, it is endowed with a stronger inductive bias for audio and will therefore likely require fewer trainable parameters, generate audio of better quality, and exhibit better computational efficiency. This last aspects is crucial for making the application of the architectures suitable for real-time interactive music creation scenarios.

Another modification concerns the length of the time window of Seq2Seq inputs. Sonification and improvisation situations have different requirements with respect to timing. Movement sonification should typically respond instantaneously to movement whereas for improvisation, a longer time window is required to conduct a perceptual integration of events (Wittmann & Pöppel, 1999) (Malloch et al., 2005). For this reason, the Seq2Seq architecture likely needs to be modified to predict longer sequences. Candidates are Seq2Seq architectures with attention (Bahdanau, Cho, & Bengio, 2014)(Luong, Pham, & Manning, 2015) or transformer architectures (Vaswani et al., 2017).

To summarize, there is still a long way to go towards the goal of creating a tool that captures some of the creative decisions a dancer makes when translating music into movement and then imitate these decisions for the purpose of generating music from movement. While the results obtained so far are rudimentary, they indicate that sequence to sequence transducers combined with raw audio generation techniques are promising candidate architectures for this task.

8 Acknowledgement

The authors' thanks go to the dancers who have contributed countless hours of their spare time to the motion capture recordings. Further thanks go to MotionBank for providing their infrastructure and assisting in the recordings. This research is conducted in the context of a Marie Curie Fellowship and is funded by the European Union. The collaboration of the second author has been supported by the Canada Council for the Arts.

References

Ananthabhotla, I., Ewert, S., & Paradiso, J. A. (2019). Towards a perceptual loss: Using a neural network codec approximation as a loss for generative audio models. In *Proceedings of the 27th acm international conference on multimedia* (pp. 1518–1525).

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein generative adversarial networks. In *International conference on machine learning* (pp. 214–223).

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

Bullock, J., & Momeni, A. (2015). Ml. lib: robust, cross-platform, open-source machine learning for max and pure data. In *Nime* (pp. 265–270).

Cuturi, M., & Blondel, M. (2017). Soft-dtw: a differentiable loss function for time-series. In *International conference on machine learning* (pp. 894–903).

Dhariwal, P., Jun, H., Payne, C., Kim, J. W., Radford, A., & Sutskever, I. (2020). Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*.

Dieleman, S., Oord, A. v. d., & Simonyan, K. (2018). The challenge of realistic music generation: modelling raw audio at scale. *arXiv preprint arXiv:1806.10474*.

Donahue, C., McAuley, J., & Puckette, M. (2018). Adversarial audio synthesis. *arXiv preprint arXiv:1802.04208*.

Engel, J., Agrawal, K. K., Chen, S., Gulrajani, I., Donahue, C., & Roberts, A. (2019). Gansynth: Adversarial neural audio synthesis. *arXiv preprint arXiv:1902.08710*.

Engel, J., Hantrakul, L., Gu, C., & Roberts, A. (2020). Ddsp: Differentiable digital signal processing. *arXiv preprint arXiv:2001.04643*.

Erdem, C., Schia, K. H., & Jensenius, A. R. (2019). Vrengt: A Shared Body-Machine Instrument for Music-Dance Performance. In *Proceedings of the international conference on new interfaces for musical expression (nime).* doi: 10.5281/zenodo.3672918

Fiebrink, R., & Caramiaux, B. (2016). The machine learning algorithm as creative musical tool. *arXiv preprint arXiv:1611.00379*.

Fiebrink, R., & Cook, P. R. (2010). The wekinator: a system for real-time, interactive machine learning in music. In *Proceedings of the eleventh international society for music information retrieval conference (ismir 2010)(utrecht)* (Vol. 3).

Gillies, M. (2019). Understanding the role of interactive machine learning in movement interaction design. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *26*(1), 1–34.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Kingma, D. P., & Welling, M. (2019). An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning*, *12*(4), 307–392.

Retrieved 2020-05-05, from `http://arxiv.org/abs/1906.02691` (arXiv: 1906.02691) doi: 10.1561/2200000056

Kumar, K., Kumar, R., de Boissiere, T., Gestin, L., Teoh, W. Z., Sotelo, J., ... Courville, A. (2019). Melgan: Generative adversarial networks for conditional waveform synthesis. *arXiv preprint arXiv:1910.06711*.

Lee, H.-Y., Yang, X., Liu, M.-Y., Wang, T.-C., Lu, Y.-D., Yang, M.-H., & Kautz, J. (2019). Dancing to music. *arXiv preprint arXiv:1911.02001*.

Li, J., Yin, Y., Chu, H., Zhou, Y., Wang, T., Fidler, S., & Li, H. (2020). Learning to generate diverse dance motions with transformer. *arXiv preprint arXiv:2008.08171*.

Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Malloch, S., et al. (2005). Why do we like to dance and sing. *Thinking in four dimensions: Creativity and cognition in contemporary dance*, 14–28.

Manocha, P., Finkelstein, A., Zhang, R., Bryan, N. J., Mysore, G. J., & Jin, Z. (2020). A differentiable perceptual audio metric learned from just noticeable differences. *arXiv preprint arXiv:2001.04460*.

Mehri, S., Kumar, K., Gulrajani, I., Kumar, R., Jain, S., Sotelo, J., ... Bengio, Y. (2016). Samplernn: An unconditional end-to-end neural audio generation model. *arXiv preprint arXiv:1612.07837*.

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., ... Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

Oord, A. v. d., Vinyals, O., & Kavukcuoglu, K. (2017). Neural discrete representation learning. *arXiv preprint arXiv:1711.00937*.

Purwins, H., Li, B., Virtanen, T., Schlüter, J., Chang, S.-Y., & Sainath, T. (2019). Deep learning for audio signal processing. *IEEE Journal of Selected Topics in Signal Processing*, *13*(2), 206–219.

Qi, Y., Liu, Y., & Sun, Q. (2019). Music-driven dance generation. *IEEE Access*, *7*, 166540–166550.

Ren, X., Li, H., Huang, Z., & Chen, Q. (2019). Music-oriented dance video synthesis with pose perceptual loss. *arXiv preprint arXiv:1912.06606*.

Steinmetz, C. J., & Reiss, J. D. (2020). auraloss: Audio-focused loss functions in pytorch. In *Digital music research network one-day workshop (dmrn+ 15)*.

Strauss, L., Tatar, K., & Nuro, S. (n.d.). *Iterative design processes and soma-based practices in instance.* (in review at *Organised Sound*)

Sun, G., Wong, Y., Cheng, Z., Kankanhalli, M. S., Geng, W., & Li, X. (2020). Deepdance: music-to-dance motion choreography with adversarial learning. *IEEE Transactions on Multimedia*, *23*, 497–509.

Tang, T., Jia, J., & Mao, H. (2018). Dance with melody: An lstm-autoencoder approach to music-oriented dance synthesis. In *Proceedings of the 26th acm international conference on multimedia* (pp. 1598–1606).

Tatar, K., & Pasquier, P. (2019). Musical agents: A typology and state of the art towards musical metacreation. *Journal of New Music Research*, *48*(1), 56–105. Retrieved 2018-09-11, from `https://www.tandfonline.com/doi/full/10.1080/09298215.2018.1511736` doi: 10.1080/09298215.2018.1511736

Tragtenberg, J., Calegario, F., Cabral, G., & Ramalho, G. (2019). Towards the Concept of "Digital Dance and Music Instrument". In *Proceedings of the international conference on new interfaces for musical expression (nime)* (p. 6).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Wittmann, M., & Pöppel, E. (1999). Temporal mechanisms of the brain as fundamentals of communication—with special reference to music perception and performance. *Musicae Scientiae*, *3*(1_suppl), 13–28.

Wright, A., & Välimäki, V. (2020). Perceptual loss function for neural modeling of audio systems. In *Icassp 2020-2020 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 251–255).