Next Level Choreography: Applying a Transformer Network to Generate Improvised Dance Motions

Zahra Asadi, Jonas Moons(✉), and Stefan Leijnen

Artificial Intelligence Research Group, University of Applied Science Utrecht, Heidelberglaan 15, 3584 CS Utrecht, The Netherlands
jonas.moons@hu.nl

Abstract. With recent developments in artificial intelligence, it is possible to generate human motion using deep learning. In this paper, a transformer deep learning algorithm is investigated to generate improvisation dance motions for the Another Kind of Blue (AKOB) data set. AKOB is an innovative dance group, located in The Hague, Netherlands, with a specialization in combining modern dance and technology. For this study, AKOB recorded various dance movements with different pieces of music using a motion capture system. This data is used to train a transformer network and generate sequences of improvisational dance using seed motions and musical input. The produced movements are visualized and compared to the ground truth of human motions to examine their quality. The results show possible human positions, but the speed of motions is a lot compared to the music. Also, sometimes the transition from one position to another is not feasible.

Keywords: Transformer Network · Improvisation Dance · Human Motion · Music

1 Introduction

In vision-based human motion recognition, algorithms are used to analyse the movement of people using camera-captured images. These algorithms have various applications in domains like sports, art, entertainment, surveillance, man-machine interfaces, and robotics [1]. Similarly, motion recognition can be used to analyse the choreography of dance. In the art discipline of modern dance, movements are combined with music to express artistic ideas [2]. These choreographies are then performed by professional dancers. With rapid advances in artificial intelligence, it is now possible to train neural networks models that are able to generate new choreographies [3].

Another Kind of Blue (AKOB) is a dance group that uses controlled drones and artificial intelligence (AI) in its performances [4]. Currently, AKOB uses drones that are pre-programmed to perform certain movements, based on recorded human movements. AKOB wishes to develop an algorithm that enables the drones to improvise dance

A. L. Brooks (Ed.): ArtsIT 2022, LNICST 479, pp. 515–525, 2023.
https://doi.org/10.1007/978-3-031-28993-4_36

motions without an external control. The first step towards this is to create a deep learning algorithm that generates human movements based on music. In the future, these human movements can then be translated into drone movements. Therefore, in this study we explore how a deep learning algorithm can generate 3D dance movements, based on a musical score and existing AKOB choreographies.

Different architectures of neural networks for generating human motions have been investigated in the literature, such as convolutional neural network [5], generative adversarial networks [6], restricted Boltzmann machines [7], recurrent neural networks [8], and transformers [9]. Recurrent networks and transformer networks are recently developed architectures [10]. In theory, they should be effective in generating unlimited motion; however, in reality the generated movements freeze or turn to abnormal movement after some iterations [9]. Bengio et al. (2015) propose to ease this problem by periodically using the network's outputs as inputs during training [11] (Fig. 1).

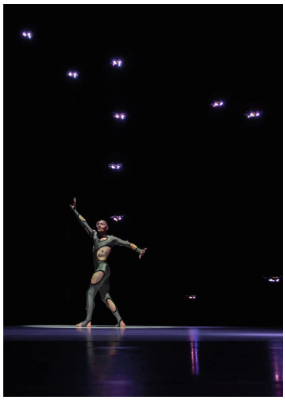

Fig. 1. Airman show performed by AKOB; drones are following dancer movements [12].

There are other studies that use music to generate 3D human motions. In these studies, different approaches are employed, such as long short-term memory networks [13], generative adversarial networks [14], recurrent neural networks [15], and convolutional sequence-to-sequence models [3]. However, these methods fail to predict the main pose when the same audio sample has multiple different corresponding motions, which often occurs in dance data. Li et al. (2021) eliminate this problem by using seed motion, which allows the generation of multiple motions from the same audio. They used the AIST dance video database [16], and the results show that their transformer model is better at retaining the correlation between music and dance movements than previous models. It generates plausible sequences with longer durations [16]. In this study, their transformer network will be trained with AKOB movement data. As far as we are aware, this is the first time such a model is trained with dedicated improvised dance data from a modern dance group. Also, we believe it's the first time the results were evaluated with a professional choreographer.

2 Data Acquisition

AKOB records dance and music sequences with a motion capture (MoCap) system for performance and research purposes. The company and dancers are well experienced with the MoCap system and have used it for various other projects previously [4]. AKOB utilizes multiple Optitrack cameras[1] installed around the studio to capture the motions. Dancers wear a Velcro suit with 54 reflective motion sensors, which are detected by the cameras. From the various 2D images, corresponding 3D movements are reconstructed using Motiv, the software for MoCap data recording and manipulation [17] (Fig. 2).

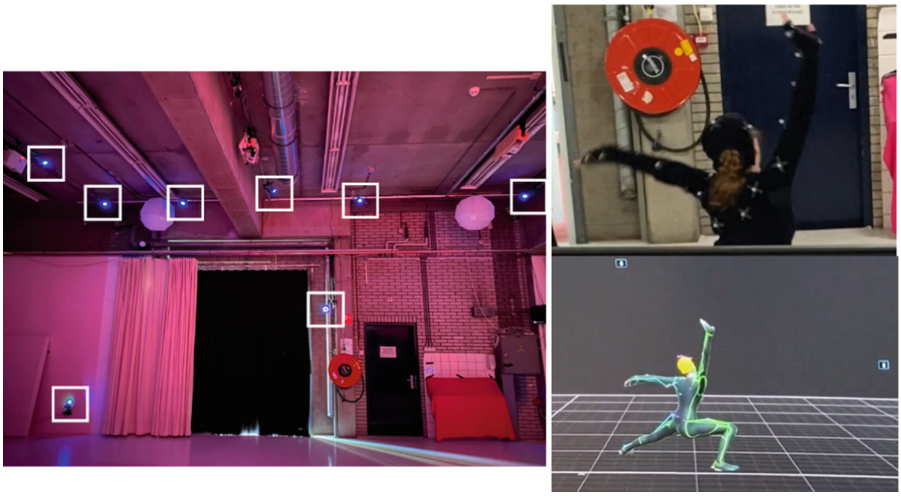

Fig. 2. Left: Optitrack cameras around the studio, right top: Markers on the dancer's suit, right down: MoCap recording in Motiv software.

For the data acquisition, we were inspired by the AIST dance video database [18]. Dance movements were recorded with 9 dancers, who improvised modern dance movements. 20 different short choreographies were performed with 4 different pieces of music, with specific durations. Movements were recorded for two scenarios, basic and advanced. The basic recordings contain short dance movement sequences, while the advanced recordings contain more complex movements. By separating these situations, the neural network may be able to learn the separate movements with basic data. The advanced data should allow it to learn to integrate movements into sequences of dance motions. Each choreography was recorded at least 5 times under the same conditions. In total there were 183 tracks, 74 min of recording; 90% basic situations, and the rest advanced.

[1] https://optitrack.com/cameras/primex-13w/.

3 Data Preparation

To meticulously sync the start of the music track and the dance motions, 8 sound beats were added to the beginning of the music. The dancers moved immediately after these beats. Before training, the tracks were cut from the start of movement until the end of the music track. The number of frames per second for recordings is 60.

While analysing each joint's data, it was discovered that MoCap frequently lost track of the fingers' joints. Therefore, the data of the finger joints was removed, and only the general direction of the hand was kept, using the middle joint of the middle finger. This also reduced each frame's data size from 156 numbers to 66, and consequently reduced the complexity of the model and the necessary computation time.

For training the model, the motion and music data needed to be translated into features. A skinned multi-person linear (SMPL) is a 3D model of the human figure [19] and it is a form to represent realistic human motions data in machine learning applications [16]. Therefore, the motion features were SMPL inputs, which consists of pose and global translation parameters. The pose parameters are a 3D rotation matrix of each joint with respect to its parent joint (the joint to which it is attached), 72-dimension in total (24 joints $\times$ 3 dimensions), while the global translation is the root joint's position in space (3 dimensions) [19].

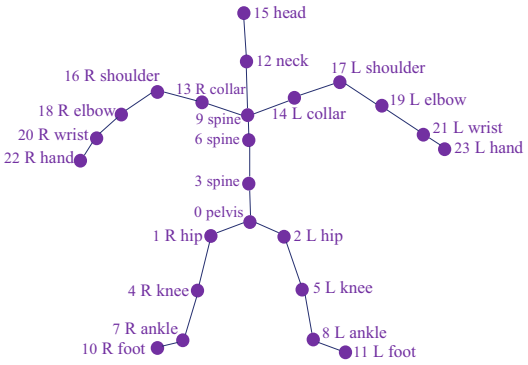

Fig. 3. Kinematic tree of the SMPL model [20].

The algorithm needs both movements and music data, so it can find the relation between the two of them, and generate movements based on music (as well as previous movements). In order to translate the music tracks into features, we used Librosa, a publicly available audio processing toolbox [21]. It is a Python package for audio and music signal processing, with various functions to retrieve musical information [22]. The following musical features were chosen based on Li et al. (2021): envelope, MFCC, Chroma, one-hot peaks, and one-hot beats [16]. These features were added as input to the model.

4 Model and Training

In music and dance, sequences are usually chronologically dependent in order to create aesthetically pleasing performances. Therefore, the model needs to account for these correlations within the input data. The transformer network is a viable solution to fulfill this requirement, as it is an architecture for investigating long-range dependencies in a sequence-to-sequence dataset [23]. One main difference of the transformer network compared to previous approaches is that the input sequence can be passed in parallel, which means all the motion and music sequences are passed simultaneously. Thus, translating from one sequence to the other is more accurate and faster compared to other architectures [23].

Li et al. (2021) propose a transformer network architecture to learn music-motion correlation and generate realistic dance motion sequences [16]. They present a model with three transformers: an audio transformer, motion transformer, and a cross-modal transformer. The motion and audio transformers learn the relationship between sequences while decreasing the dimension of the input data. The cross-modal transformer captures the correlation between motion and music, and generates new motions. By using a cross-modal transformer, the mutual information between the sequence of motions and the sequence of music will be maximized at the sequence level rather than at the frame level [24] (Fig. 4).

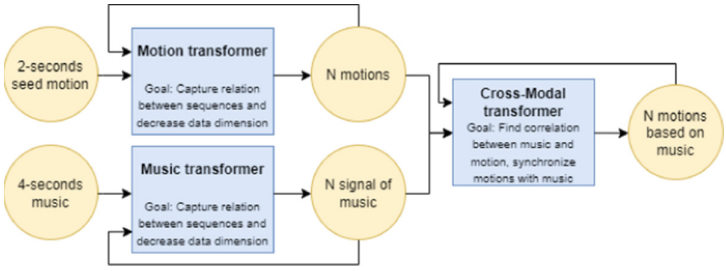

Fig. 4. The structure of the model, after Li et al. [17].

As seen in Fig. 3, the model receives as input: the 2 s seed motions (120 frames), and 4 s of music (240 frames). The output of the model is the data for 20 future frames. This output is the input of the next frames, as in an autoregressive model [16].

Li et al.'s model was retrieved from GitHub[2], and parameter values were set according to the specification provided by them. The data were split in a test and train dataset, with the music in the test dataset being different from the music in train dataset. Thus, we can investigate how the model responds to a new piece of music. In total, the test dataset includes 22 tracks, 15.5 min of recording and roughly 10% of the original dataset. The training dataset was used to train the model, and Adam optimizer was employed. The training phase took 25 days with HPC cloud and 1 GPU 2080 Ti. The learning rate started from 1e−4 and decreased to 1e−6. Afterwards, the trained model was used to

[2] https://github.com/google-research/mint.

generate new motions with the test dataset. Seed motion and music were used as input, and the output was the newly produced dance choreography. The results of the test data set were used for validation with human participants.

5 Validation

The results were analysed with three methods: assessing the quality of input data, specifically motion data; evaluating the training phase of the model with the loss function; and finally, investigating the generated motion quality, through an interview with a professional choreographer and a survey among non-professionals.

5.1 Input Quality

After converting the motion data to SMPL, animations were generated in the Blender software and compared to MoCap data, in order to investigate the quality of converted data (see Fig. 5). MoCap data can only be shown as a skeleton in Blender. The SMPL stick figures include pyramids above each joint, which shows the direction of each joint.

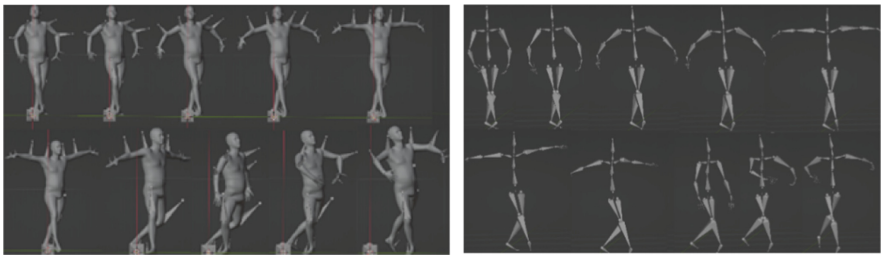

Fig. 5. Left: 10 frames of animated SMPL in Blender. Right: the same frames using MoCap.

We selected 35 (20% of recorded tracks) files at random for an input quality check. The results show that on average, there is an error in 15% of the hip motions in the converted files. By investigating the raw data, it was discovered that the error happened when the joint rotated more than 150 degrees, particularly when the rotation data jumped from 0 to 180 or −180 degrees for consecutive frames. However, the rest of the joints show the exact movements recorded with the MoCap. Therefore, there might be some error in the results for the hip motions, which shows the joints are rotating impossibly like Fig. 6.

5.2 Model Quality

To see if the model was trained reliably, the loss function was examined during the training phase. The loss function is a measure of the difference between the training output data (ground truth) and the generated data. The L2 loss function was used, which

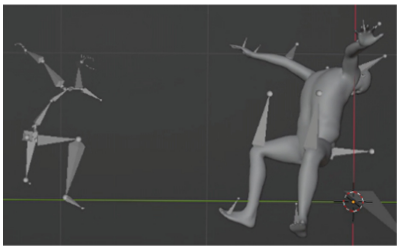

Fig. 6. A sample of common error in SMPL hip movements (right) versus MoCap (left).

calculates the sum of squared differences between predicted values in the generated motions and the actual movement values.

$$L2 = \sum_{i=1}^{n} \left(y_{true} - y_{predicted} \right)^2 \tag{1}$$

It is expected that losses decrease with more training steps, as the model learns how to generate new movements. The loss function trend with the number of training steps is demonstrated in the table below. The loss is decreasing with more training steps; thus, the algorithm is better able to predict the output values in the train dataset. However, the fact that the model is able to reproduce motions it has seen, does not necessarily mean that newly generated motions will be adequate. For this, we evaluated the test dataset with human participants (see 5.3) (Table 1).

Table 1. Loss function trend over training steps.

Number of training steps	Loss function
20	0.756355
50	0.456406
100	0.279331
5000	0.056492
100,000	0.003793
200,000	0.000244
500,000	0.000146
2,400,000	0.0000498

5.3 Motion Quality

To explore the quality of the generated motions, a survey was done with 20 candidates in parallel to an interview with a choreographer. The generated motion files were converted to an animations for two different models, one trained with 500K steps and the other

with 2.4 million. By comparing these models, we can establish both the quality of the generated movements, as well as the effect of longer training time.

Figure 7 show the sequences of one of the generated movements with 2.4 million steps. The first figure is the seed motion, and the rest are the generated movements for every 20 frames.

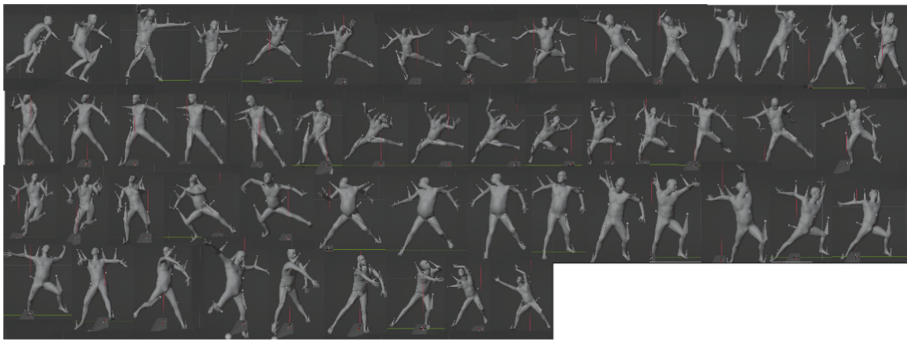

Fig. 7. Sequences of generated motions with 2.4 million training steps.

Study

The evaluation of generated choreography is complicated, as there is no unique way to dance to the identical music, and the same movements can be used for distinct music. In order to get a general impression of the naturalness of our generated movements, we asked 20 participants to rate the generated choreographies. Two sets of videos, including three videos of each model (with 500K and 2.4 million training steps) were provided in pseudo random order, and were sent to participants. We asked these participants to rate the correspondence of the movements to the music and the degree to which they resembled real human dancing, both on a scale from 1–10 (1 meaning "not at all", 10 meaning "perfectly"). The scores for the three videos were averaged for each candidate and the results were examined statistically.

A t-test comparing the results for the 500K and the 2.4 million steps models showed that the average correspondence of the movements to the music for the 2.4 million steps model ($M = 5.2$, $SD = 1.79$) was significantly more than the 500K steps model ($M = 3.75$, $SD = 2.09$) conditions; $t(37) = 2.36$, $p = 0.012$. Moreover, the average rate of resembling real human dance for the 2.4 million steps model ($M = 5.12$, $SD = 1.95$) was significantly more than the 500K steps model ($M = 3.73$, $SD = 1.65$) conditions; $t(36) = 2.42$, $p = 0.010$.

Expert Interview

The interview was done with David Middendorp, experienced choreographer and AKOB's director. At the beginning of the discussion, three generated videos from the 2.4 million steps model were shown, and a couple of general questions were asked. Then we went through one of the video's movements in more detail.

In general, David thought the movements did not resemble natural human motions because they were wobbling a lot, did not consider gravity and inertia, and did not correspond to the floor surface. He used words such as 'spooky', 'unhuman', and 'not smooth' to describe the movements. However, he believed there was some coherence in the motion sequences that shows they are not random. It was hard to judge the relationship between music and movements for him. He mentioned, 'It is pretty a matter of taste, are the movements right and what is right'. He said the generated motions were so fast while the music was slow, and it seemed they might be correlated in a higher frequency of the music. Meanwhile, he can see some relationships in arm motions, and he concluded: 'The movements are a little good to be random, but I am not sure.' In the end, he mentioned the current generated movements were not usable in his art shows because he preferred to have the real human motions first, and then change them in his desired way. However, he believed 'They are not completely unusable. They have their weirdness that can be usable.'

By going through one of the video sequences, Fig. 7, David agreed that most positions are possible for a human, but the transition speed from one spot to another is not feasible, and the movements do not correspond to gravity and the floor surface correctly. He liked some generated positions, Fig. 8, and said they could not happen in real life for too long, but they are 'cool' positions. Moreover, he vividly saw a proper correlation between neck and arms movements in some motion sequences, Fig. 9. The neck moved smoothly with the arms motion, which is aesthetically pleasing when the video ran slowly.

Fig. 8. Interesting positions in generated dance chosen by the expert.

Fig. 9. Appropriate relation between the neck and the arms motions.

6 Discussion

In this study, Li et al. [16] model is used to generate new dance movements for the AKOB dance group. The data is managed precisely, from recording data to converting it into SMPL and validating the new format. Although there are still a few errors in hip

motions, their effects are not seen in the results. It is worthwhile to examine the reason behind it more. Also, data splitting is done meticulously to assure the newly generated movements do not overlap with the training dataset.

The current results show that with more training steps, the loss function decreases, and the quality of generated motions gets better. The improvement in the quality of generated dances with more training steps is validated with a survey. This approach was chosen because there is no standard for dance movements, and any type of movement might be pleasant for some people. However, the relatively low scores for the 2.4 million steps model for both musical correspondence ($M = 5.2$, $SD = 1.79$) and human-like quality ($M = 5.12$, $SD = 1.95$) show that even for the fully trained models, participants did not perceive them as particularly natural. Besides the study, an interview is done with an expert choreographer. He did not deny that the current results might be pleasant for the audience, but he believed the crucial issue is syncing the speed of music and motions, which is not yet satisfying. Moreover, the professional mentioned the movements are uncanny and do not consider the floor surface and gravity. By providing more datasets for training, these drawbacks might be mitigated.

In the next steps, with iteratively running the model, it can be defined if there is a need for modifying the model or tuning the parameters. The current music features are chosen based on Li et al. [16] study. However, the genres in the literature differ from the AKOB dataset. So, it is needed to examine if the relation between movements and music improves with additional modifications in music features.

Acknowledgements. We would like to express our gratitude to David Middendorp and the Another Kind of Blue crew for their participation in this project. We would also like to thank Bas van der Linden for his technical advice.

References

1. Kale, G.V., Patil, V.H.: A study of vision based human motion recognition and analysis. Int. J. Ambient Comp. Intellig. **7**(2), 75–92 (2016)
2. Mallick, T., Das, P.P., Majumdar, A.K.: Posture and sequence recognition for Bharatanatyam dance performances using machine learning approach. arXiv (2019)
3. Ahn, H., Kim, J., Kim, K., Oh, S.: Generative autoregressive networks for 3D dancing move synthesis from music. IEEE Robutics and Automation Letters **5**(2), 3501–3508 (2020)
4. Another Kind of blue (2022). [Online]. Available: https://www.anotherkindofblue.nl/en/
5. Holden, D., Saito, J., Komura, T.: A deep learning framework for character motion synthesis and editing. ACM Transactions on Graphics (TOG) **35**(4), 1–11 (2016)
6. Hernandez, A., Gall, J., Moreno-Noguer, F.: Human motion prediction via spatio-temporal inpainting. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 7134–7143 (2019)
7. Taylor, G.W., Hinton, G.E.: Factored conditional restricted boltzmann machines for modeling motion style. In: Proceedings of the 26th annual international conference on machine learning, pp. 1025–1032 (2009)
8. Du, X., Vasudevan, R., Matthew, R.J.: Bio-lstm: A biomechanically inspired recurrent neural network for 3-d pedestrian pose and gait prediction. IEEE Robotics and Automation Letters **4**(2), 1501–1508 (2019)

9. Aksan, E., Cao, P., Kaufmann, M., Hilliges, O.: A spatio-temporal transformer for 3d human motion prediction. arXiv preprint arXiv:2004.08692 (2020)
10. Leijnen, S., Veen, F.V.: The Neural Network Zoo. In: Proceedings, vol. 47(1), no. 9 (2020)
11. Bengio, S., Vinyals, O., Jaitly, N., Shazeer, N.: Scheduled sampling for sequence prediction with recurrent neural networks. In: Advances in neural information processing systems (2015)
12. Theater de storm (2022). [Online]. Available: https://www.theaterdestorm.nl/voorstelling/another-kind-of-blue/
13. Alemi, O., Franc¸oise, J., Pasquier, P.: Real-time music-driven dance movement generation using artificial neural networks. Networks **8**(17), 26 (2017)
14. Ginosar, S., Bar, A., Kohavi, G., Chan, C., Owens, A., Malik, J.: Learning individual styles of conversational gesture. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3497–3506 (2019)
15. Huang, Y., Kaufmann, M., Aksan, E., Black, M.J., Hilliges, O., Pons-Moll, G.: Deep inertial poser learning to reconstruct human pose from sparseinertial measurements in real time. ACM Transactions on Graphics **37**(6), 1–15 (2018). 185
16. AIST Dance Video Database (May 2022). [Online]. Available: https://aistdancedb.ongaaccel.jp/
17. Li, R., Yang, S., Ross, D.A., Kanazawa, A.: AI Choreographer: Music conditioned 3D dance generation with AIST++. In: IEEE/CVF International Conference on Computer Vision (ICCV), pp. 13381–13392 (2021)
18. Optitrack Wiki (May 2022). [Online]. Available: https://v30.wiki.optitrack.com/index.php?title=Quick_Start_Guide:_Getting_Started
19. Loper, M., Mahmood, N., Romero, J., Moll, G.P., Black, M.J.: SMPL: a skinned multi-person linear model. ACM transactions on graphics (TOG) **34**(6), 1–16 (2015)
20. Codetd: Sorting of joint points in common human hinge joint point data sets (2020). [Online]. Available: https://www.codetd.com/en/article/11819052. Accessed July 2022
21. Librosa (2022). [Online]. Available: https://librosa.org/
22. McFee, B., et al.: librosa, Audio and music signal analysis in python. In: Proceedings of the 14th python in science conference, pp. 18–25 (2015)
23. Vaswani, A., et al.: Attention is all you need. Advances in neural information processing systems 30 (2017)
24. Sun, C., Baradel, F., Murphy, K., Schmid, C.: Learning video representations using contrastive bidirectional transformer. arXiv preprint arXiv:1906.05743 (2019)