Sequencing the musical sections with deep learning

Xuange Cui*†, Mingxue Liao†, Pin Lv† and Changwen Zheng†
*University of Chinese Academy of Sciences, Beijing, China
†Institute of Software, Chinese Academy of Sciences, Beijing, China
{xuange2017, mingxue, lvpin, changwen}@iscas.ac.cn

***Abstract*—Deep learning has become increasingly popular for sequence modeling in various domains. In this work, we address the musical order verification as a sequential pattern learning task. We present a more advanced self-supervised learning model, dubbed as the triple-wise similarity with recurrent neural networks (TSN-R) which is on the basis of state-of-the-art Similarity Embedding Network. We take triple-wise musical sections as the input instance, and leverage the temporal coherence as a supervisory signal. As a first step, we use the triplet Siamese network as the input layer, and assess the similarity of the triplet feature maps. After that, we choose the bidirectional LSTM to extract features along the time dimension. Finally, the subsequent fully-connected layers are used for order verification. The experiments show that our approach outperforms other competing models in terms of prediction accuracy. Additionally, this paper reveals the reasons for the good performance of our model through the 2d-visualization of features. All the source code, pre-trained models and the experiment results are available in our project page. https://github.com/ISCASTEAM/Sequencing-the-musical-sections**

I. Introduction

In the past few years, many researchers have designed the music sorting puzzle games, mostly for the education purposes [1]. These puzzle games require players to sort several musical sections and eventually restore the original song in the correct order. Cunningham et al. [2] performed an in-depth study on individual users who have played music sorting puzzle games, and concluded that the sequence sorting task is "more art than science". Indeed, these puzzle games is designed for people rather than machines. From a technical standpoint, how to train machines to learn sequential patterns and solve such games is interesting and challenging.

Neural networks, especially deep neural networks, have been increasingly used to solve sequence sorting task in various domains, such as text, video and audio [3]–[6]. A new unsupervised learning paradigm has recently emerged as self-supervised learning [7]–[9]. It is very universal to use the self-supervised method as either a sequence sorting task or an order verification especially when we do not have lots of manually annotated audio data. Within the deep neural networks, the self-supervised learning is to leverage the inherent structure of the raw data and develop a discrimination or reconstruction loss function to train the neural networks. Moreover, in the computer vision domain, Misra et al. [7] propose their work of self-learning model which used the temporal coherence of video frames as a supervisory signal. In audio domain, Yu-Siang Huang et al. [1] proposed the Similarity Embedding Network(SEN) which obtained the state-of-the-art performance in musical order verification. In a conclusion, these prior achievements confirm that it is very effective to train the models with self-supervised learning which are conducted on the unlabeled data.

A major challenge with musical sequence sorting is that if directly taking the N fragments as input, we need compute N! sequences. It is impossible if we want compare the arrangements of these N musical fragments. For example, if $N = 10,\ N! = 3,628,800$. When we sort 10 musical clips, our model needs to calculate about 3.6 million inputs, and this is only one instance from the training data. However, it is not necessary to put all the fragments in correct order all at once. The key idea of solving the challenge is to use pair-wise audio fragments as input which is much fewer than integral permutation. The output is the ordering of the pair fragments, and we decide the final sequence of the entire song by a simple heuristic rule [1]. Using the pair-wise fragments as input, the time complexity of our model is reduced from $N!$ to $N(N-1)$.

Moreover, the musical fragments sampled from the same song should be meaningful basic units, non-overlapping, multi-second clips and time-series data. Splitting a song into multi-second clips is practical, because the entire song consists of intro, verse, pre-chorus, chorus, bridge, elision, conclusion, instrumental solo [10]–[12]. Therefore, sampling multiple clips from the same song is permitted and meaningful. But the more the number of clips sampled from the song, the more difficult to sort the sequence. In addition, we can also sample from different songs, and the final sorted result is able to be regarded as automatic AI-DJ [13], [14] that the effects depend on the subjective discrimination of human beings.

From an application standpoint, our model has an important driving force for the AI DJ's remixes or mash-ups. Another benefit of our model is that we take the time-series data instead of images as input, so the proposed architecture is able to be applied to the sequence modeling in various domains.

The remainder of this paper is organized as follows: Section II provides existing works on musical sequence sorting. In section III, we describe the network architecture used in our experiments and review the other competing models. The outcome of experiments are shown in Section IV. Finally, in section V, we outline the limitations and conclude the paper.

II. Related work

In this work, we propose an advanced model on the basis of the state-of-the-art Similarity Embedding Network [1]. Before we provide the details of related works, let's

review the method of obtaining the correct order of the fragments. For example, there is an order verification about three fragments (P_a, P_b, P_c). The outcome of our model is $F(P_aP_b), F(P_aP_c), F(P_bP_c), F(P_bP_a), F(P_cP_a), F(P_cP_b)$. One of the arrangements for the score, such as $P_aP_bP_c$, is $F(P_aP_b) + F(P_bP_c)$. The arrangement which has the maximum score is taken as the correct order. To this end, we start to introduce several related models.

Yu-Siang Huang et al. [1] described a pair-wise architecture dubbed as Similarity Embedding Network(SEN) that was similar to the architecture used in this paper. The key idea of the SEN is calculating the similarity of the adjacent musical clips, and lately using a simple heuristic to decide the final ordering. The SEN uses Siamese network to extract features from pair-wise spectrogram of music [15] which is 2d feature rather than image. And then using a product layer to compute the inner product between two representations of the Siamese network [16], finally the model gets the outcome through fully-connected layer. The most important advantage of SEN is to distinguish the coherence between the pair-wise fragments through the cosine similarity. Additionally, the disadvantage of SEN is that the model does not take into account the potential information along the temporal dimension. And using the triplet Siamese network performs better than pair-wise Siamese network in some situation. In doing so, we improved the origin SEN network in these two aspects and get the state-of-the-art performance.

Wang et al. [17] provided a feature learning model dubbed as Concatenated-convolutions Siamese Network(CCSN) that consisted of convolution layers. The input layer is Siamese convolutional network, the middle layer concatenates feature maps along the depth dimension instead of calculating the similarity of the fragments which is the most different point from SEN model. Finally, the CCSN model gets results through fully-connected layer. The CCSN model's shortcomings are similar to SEN model, and we use CCSN as a competing model.

Misra et al. [7] proposed a Siamese network that used triplet data as input, the model dubbed as triplet Siamese network(TSN). The TSN model is the least complex compared to other models' architecture. The TSN model uses contrast loss to maximize the Euclidean distance between the categories. The model dose not use similarity and not use embedding space, just choose triplet Siamese network structure [16] as the input layer.

III. Model

In view of the challenges about sorting N musical clips, the time complexity should be taken into account which is mentioned in introduction. For N fragments of one song, there is no need to put all the clips in correct order all at once. In contrast, we only need prepare $N(N-1)(N-2)$ triplet-wise data as input. In order to simplify the model calculation, we use 3 fragments sampled from one song. That means 6 triplet-wise data as the input of model in the process of training. In the process of testing, we validate the prediction accuracy conducted on the different number of fragments.

A. Triplet Similarity with RNN Network

This work expands on the Similarity Embedding Network defined by Yu-Siang Huang et al. [1]. We take triple musical sections as the input instance, and leverage the temporal coherence as a supervisory signal.

As depicted in Figure 1. Firstly, we use the triplet Siamese network as input layer, because the Siamese network shares the same parameters and is bale to handle multiple inputs simultaneously. The Siamese network learns 2D-feature from spectrogram which is the representation of music features [2], [18], [19]. Lately, we assess the similarity of the triplet feature maps. Secondly, we extract the feature from the similarity matrix again by a few more convolutional layers as shown in

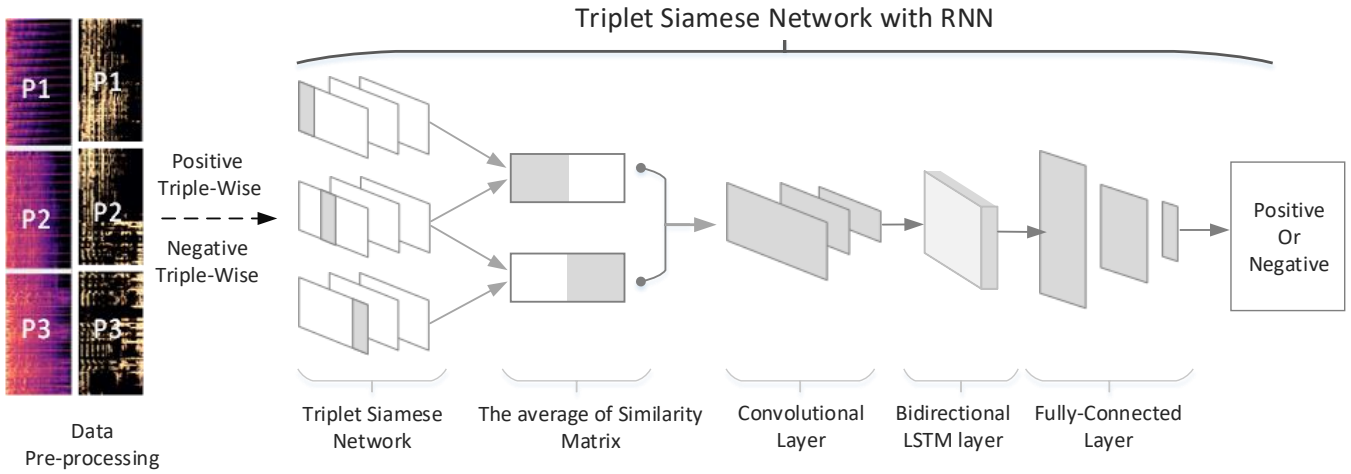

Fig. 1. TSN-R Network Architecture

the middle of Figure 1. And we choose the bidirectional LSTM to extract features alone the time dimension [20]–[23]. Finally, the fully-connected layers are used for order verification. The output of this model can be viewed as the classified result.

The input is triple-wise fragments which are divided into two labels, positive and negative. The classification is according to the inherent and intrinsic properties of the arrangement. It is the self-supervised approach that we mentioned in Section I.

In what follows, we present the mathematical formulation of the learning problem. For each instance, we get an arrangement of the fragments. The collection $P = \{P_1, P_2, \ldots, P_n\}$. The triple-wise data in correct order are considered as the positive data, and other possible triplet can be considered as the negative data.

$$P^+ = \{(P_i, P_{i+1}, P_{i+2})\} \tag{1}$$

$$P^- = \begin{cases} (P_i, P_{i+2}, P_{i+1}) \\ (P_{i+1}, P_i, P_{i+2}) \\ (P_{i+1}, P_{i+2}, P_i) \\ (P_{i+2}, P_i, P_{i+1}) \\ (P_{i+2}, P_{i+1}, P_i) \end{cases} \tag{2}$$

The collections of positive data are P^+, and the collections of negative data are P^- with $i \in \{1, \ldots, n-2\}$. The ratio of positive data and negative data,

$$P^+/P^- \approx \frac{1}{5}$$

Due to the unbalanced collections of positive data and negative data, we adapt a list of manual weights given to each class in the training process.

We provide a training set

$$S = \{(X, Y) \mid X \in P, Y \in \{0, 1\}\} \tag{3}$$

where P is the union of P^+ and P^- that mentioned before. Y is positive or negative label which is the inherent property of the given arrangement. If the input data belong to P+, Y is 0 otherwise Y is 1.

We need to learn the parameters θ of a neural network $F(\theta)$ by solving:

$$\min_{\theta} \sum_{X,Y \in S} L\big(F_\theta(X), Y\big) \; + \; R(\theta) \tag{4}$$

The $F_\theta(X)$ is the predicted label, and Y is the true label. And the L is a loss function, $R(\theta)$ is a regularization term. In the implementation, we choose the cross entropy as the loss function, and use batch size normalization, dropout, L2 regularization to avoid overfitting. When calculating the similarity, we apply the cosine similarity, because we want to capture the temporal correspondence between the triplet input fragments.

Finally, we get the global ordering by choosing the largest score in all the arrangements. For example, if n=4, one of the arrangement is (P_1, P_2, P_3, P_4) which is the correct order. The score of (P_1, P_2, P_3, P_4) is obtained by calculating $F(P_1P_2P_3) + F(P_2P_3P_4)$.

B. Implementation Details

We implement our approach and conduct all the experiments by using the Pytorch toolbox. As done in many previous works [1], [24]–[26], we calculate the spectrograms-like features by sampling the raw songs at 22,050 Hz and adopt the hop size 512 samples and the Hamming window of 2,048 samples. Lately, we transform the spectrograms into 128-bin log mel-scaled spectrograms [27], [28] as the input of the model. We choose zero-center as the pre-processing approach, and then normalize these data [29], [30]. The details are presented in Figure 2.

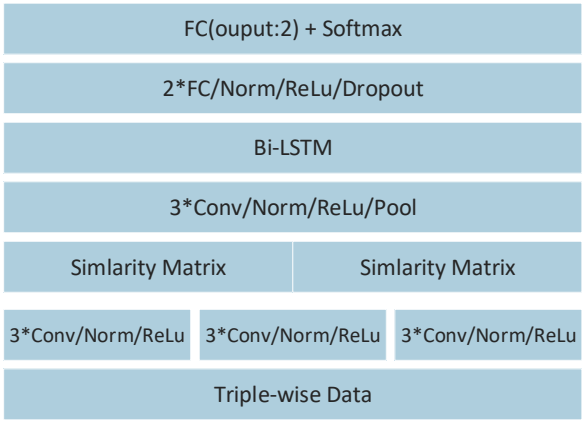

Fig. 2. TSN-R Details

We use 2D convolutional filters in the Siamese ConvNet which consists of three layers. The number of the filters is 128, 256 and 256. The length and width of the filters are [4,128], [4,1] and [4,1]. The stride size of the Siamese ConvNet is [1,128], [1,1] and [1,1]. We then compute the similarity of the triplet feature maps and get the average. The convolutional filters for the subsequent ConvNet are 64,128 and 256 that are followed by 3 by 3 maximum pooling and the convolutional filter size is 3 by 3. The stride size of the subsequent ConvNet is [1,1]. Then, we choose the bidirectional LSTM to extract features alone the time dimension. We stack double layers to form a stacked LSTM, with the second LSTM taking in the outputs of the first LSTM. The number of units in the hidden state is 128 and dropout set 0.1.

Finally, we use three fully-connected layers with 1024,1024 and 2 units. The dropout set as 0.5 in fully-connected layers. We use a softmax function to make the output to lie in the range of [0,1]. All the activation function is rectified linear unit(ReLu), and the batch size is 64. Moreover, we use Adam optimizer and the learning rate is 1e-3. And the weight decay is 1e-5 as the L2 penalty. Due to the unbalanced training set, we set 1 and 0.2 as their manual rescaling weights.

C. Baseline Model

These models have been mentioned in related works. We provide the difference and details in the Figure 3.

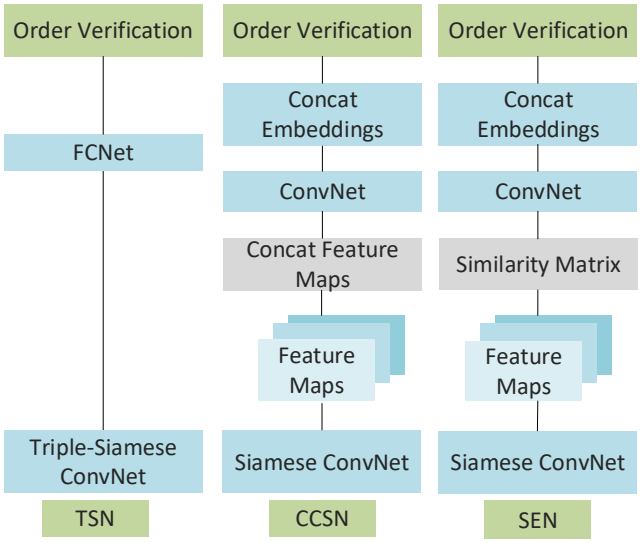

Fig. 3. Baseline Models

The Similarity Embedding Network(SEN) defined by Yu-Siang Huang et al. [1] had demonstrated the state-of-the-art performance in musical sequence sorting games. The key idea of the SEN is calculating the similarity of the adjacent musical clips. But the SEN model adapts pair-wise Siamese net rather than triplet Siamese net, and doesn't use Bi-LSTM architecture to exact time-series information which used in our TSN-R model.

Concatenated convolutions Siamese Network(CCSN) defined by Wang et al. [17] is the state-of-the-art network for image feature learning. The CCSN consists of convolution layers, and simply connect feature maps along the depth dimension instead of calculating the similarity. As show in Figure 3, the only difference between CCSN and SEN lines in the way of exacting features from the Siamese net.

Triplet Siamese Network(TSN) defined by Misra et al. [7]. Indeed, the key idea of TSN is simply using triple-wise input instead of pair-wise input, and using contrast loss function to maximize the Euclidean distance between the different class.

IV. Experiments

A. Dataset

Any music dataset is able to be used in our model, since we do not use any human annotations. In this work, we use the data crabbed from the famous music publisher, and we get 23,123 entire songs as our corpus that consist of different types and different languages. It is well known that the diverse data increase the complexity of the training process, but diverse data are useful to obtain a more robust model.

During the download process, we discard the songs that are less than 30 seconds, because the information of these songs is not rich enough, and the beat is not obvious. In the end, we get 23,123 valid songs, they are not limited in languages or types. We note that the SEN model is only conducted on the downbeat data [1] that downloaded form the Internet. By the way, out training dataset contains the English, Chinese, etc., and contains pop music, rock music, modern electronic dance music, and classical music [31]–[33]. Moreover, we randomly pick 18,000 songs as training dataset, and 5,213 songs for testing. For each song, we divided the first 24s into 3 segments, each segment is 8s. This is the same as the sampling way of the SEN model [1]. But we also compare the performance of different numbers of fragments, such as 10 clips pre song, then we intercept the first 80 seconds. In particular, we only train models on three clips pre song. In what follows, the experiments show the result of comparisons test and ablation analysis.

B. Result On 3-piece

As the first experiment, we consider the fragments=3 sorting task. We use the same dataset for our model TSN-R and other competing models. Accordingly, we use the global accuracy as our performance metrics. For example, the ordering set(P_1, P_2, P_3) as $P_1P_2P_3$ would be the only correct order.

TABLE I
The Accuracy on Four Kinds of Models

Model	Accuracy	Input	Similarity
TSN	36.60%	Triplet	No
CCSN	58.60%	Pair	No
SEN	67.00%	Pair	Yes
TSN-R	**75.80%**	**Triplet**	**Yes**

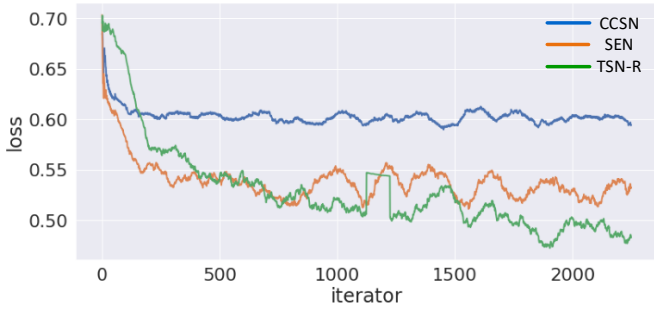

Fig. 4. The Cross Entropy Loss

The analysis of the results of Table I shows that the performance seems to be positively related to the complexity of the model. Although SEN works best in the competing models, our TSN-R model works better than SEN about 8.8%.

The baseline TSN model uses the contrast loss instead of cross entropy. Thus, only the loss curves of CCSN,SEN,TSN-R are shown in the Figure 4. It can be found that the loss curve of the TSN-R is significantly lower than other competing models.

The reason is that triplet-wise input offers more information. More importantly, the performance of TSN-R reveals that

TABLE II
THE ACCURACY ON DIFFERENT NUMBER OF FRAGMENTS

Model\Length	3	4	5	6	7	8	9	10
SEN	67%	43%	32%	17%	11%	5%	3%	0%
TSN-R	75.80%	69%	53%	44%	37%	32%	29%	24%
Time(SEN)	0.05s	0.08s	0.25s	1.16s	8.83s	78.84s	12.96m	2.43h
Time(TSN-R)	0.04s	0.06s	0.24s	1.06s	8.97s	82.14s	13.7m	2.58h

triple-wise input and Bi-LSTM are useful architectures.In particular, this paper reveals the reasons for the good performance of our model through the 2d-visualization of features in the follows.

C. Result On Multi-piece

Next, we take into account the multi-piece sorting task. We don't consider TSN and CCSN here, for their demonstrated poor performance in fragments=3 sorting task. To simplify the calculation, we only compare the performance of the SEN and TSN-R. We randomly choose 100 songs for testing with pre-trained model SEN and TSN-R. The criterion is the performance for different number of segments for 3,4,5,6,7,8,9,10, respectively.

The analysis of the results of Table II shows that the correct rate of SEN and TSN-R is declining, but TSN-R is declining slower than SEN. This shows that the new model of this paper is more robust than SEN model.

The time-consuming metrics are the results of training models on double GPU and single CPU which is occupied by only one the python process. Due to the triple-wise input and more complicated architecture, TSN-R consumes more time what is expected. In addition, the time-consuming on the CPU is only the problem of finding max score in all of the arrangements, and this part is able to be computed in parallel, so it is not necessary to worry too much about the time-consuming problem.

D. Ablation Analysis

In the Table III, we evaluate the impact of the network structure on the TSN-R model.

TABLE III
THE RESULT OF A FEW ABLATED ANALYSIS

Replace	Accuracy
Inner-product	70.10%
Pair-Wise	68.40%
No-LSTM	75.50%

We transform triplet-wise input to pair-wise input, transform cosine similarity to inner product, and the bidirectional LSTM architecture is deleted.

We note that using inner-product or using pair-wise input caused the lower accuracy than origin model. The former cannot guarantee that the similarity scores lie in the range of [0,1], and the later provides less information which are the reasons of causing the poor performance. Without the Bi-LSTM, the accuracy is slightly lower than origin TSN-R model. The reason is that the testing fragments are 8 seconds which means triplet-wise input is only 24s occupying 11% of an entire song(the full song is usually 3.5min, or 210 seconds). We believe that the longer fragments contain more information which would be more dependent on Bi-LSTM.

E. 2d Visualization Of Features

In order to examine our main premise, there are the representations with meaningful properties [29], [34], [35]. We use PCA to project the embedding into 2d Euclidean space which can be intuitively visualized. The testing dataset is a sampling collection of 1,000 songs. Due to triplet-wise input, we need calculate 6,000 triplet samples. Lately, we exact the outcome in the second to the last layer which has 1024 units. It is obvious that there are too many features, so we use the PCA to reduce the dimension to 50-dimensions. According to the assessment, these 50 features already represent 99.8% of the origin information. Finally, we use t-SNE to get 2D space for visualization which is best viewed in color.

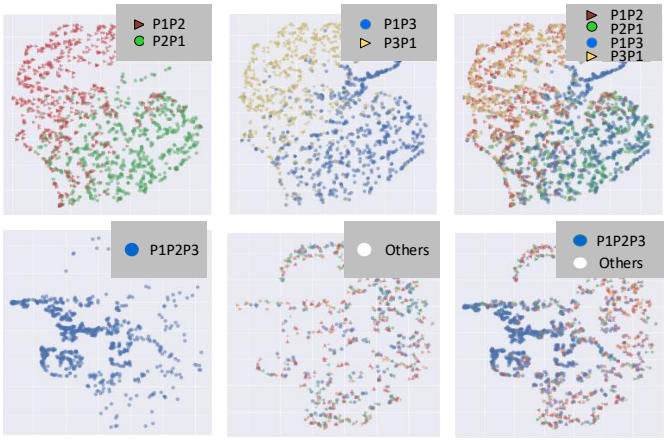

Fig. 5. TSN-R Network Architecture

As shown in the Figure 5, the distance between P1P3 and P3P1, the distance between P_1P_2 and P_2P_1 are far away. But the distance between P_1P_2 and P_1P_3, the distance between P_3P_1 and P_3P_2 are nearby. It reveals that SEN model is not able to make a good distinction between P_1P_2 and P_1P_3 which are not nearby orders. It is the shortcoming of the SEN model.

In contrast, the TSN-R model takes triplet-wise data as input and make the location of $P_1P_2P_3$ far away from other arrangements such as $P_1P_3P_2, P_2P_1P_3, P_2P_3P_1, P_3P_1P_2, P_3P_2P_1$. The result intuitively confirms the superiority and the effect of the TSN-R model.

V. Conclusion & Discussion

In this work, we have presented a novel triplet Siamese network dubbed as triple-wise similarity with recurrent neural network (TSN-R) for learning sequential patterns in a self-supervised way. The experiments show that TSN-R has been the state-of-the-art competed with other models that includes the SEN [1],CCSN [17] ,TSN [7] network. In addition, we reveal the reasons for the good performance of TSN-R network through the 2d-visualization of features.

As the section IV mentioned, we choose the fragments of the fixed length from the same song. This is the same way of sampling as the previous work identified by the SEN approach [1]. In the future, we intend to investigate picking fragments from different songs and cross-fade [36] to improve the transition between fragments. And we will explore the s automatic AI-DJ [9] by sorting the clips form the cross songs. Another research direction is to explore how to apply our network architecture to the sequence modeling in other domains.

References

[1] Y. Huang, S. Chou, and Y. Yang, "Generating music medleys via playing music puzzle games," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 2018, pp. 2281–2288. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16174

[2] J. Cunningham, A. Hapsari, P. Guilleminot, A. Shafti, and A. A. Faisal, "The supernumerary robotic 3rd thumb for skilled music tasks," in *7th IEEE International Conference on Biomedical Robotics and Biomechatronics, BioRob 2018, Enschede, The Netherlands, August 26-29, 2018*, 2018, pp. 665–670. [Online]. Available: https://doi.org/10.1109/BIOROB.2018.8487609

[3] H. Lee, J. Huang, M. Singh, and M. Yang, "Unsupervised representation learning by sorting sequences," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017, pp. 667–676. [Online]. Available: https://doi.org/10.1109/ICCV.2017.79

[4] Y. Liu, Q. Wu, L. Tang, and L. Xu, "Self-supervised learning of video representation for anticipating actions in early stage," *IEICE Transactions*, vol. 101-D, no. 5, pp. 1449–1452, 2018. [Online]. Available: https://doi.org/10.1587/transinf.2018EDL8013

[5] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. [Online]. Available: https://doi.org/10.1038/nature14539

[6] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, 2015, pp. 1422–1430. [Online]. Available: https://doi.org/10.1109/ICCV.2015.167

[7] I. Misra, C. L. Zitnick, and M. Hebert, "Shuffle and learn: Unsupervised learning using temporal order verification," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I*, 2016, pp. 527–544. [Online]. Available: https://doi.org/10.1007/978-3-319-46448-0_32

[8] B. Battey, M. Giannoukakis, and L. Picinali, "Haptic control of multistate generative music systems," in *Looking Back, Looking Forward: Proceedings of the 41st International Computer Music Conference, ICMC 2015, Denton, TX, USA, September 25 - October 1, 2015*, 2015. [Online]. Available: http://hdl.handle.net/2027/spo.bbp2372.2015.018

[9] J. Lin, W. Wei, and H. Wang, "Automatic music video generation based on emotion-oriented pseudo song prediction and matching," in *Proceedings of the 2016 ACM Conference on Multimedia Conference, MM 2016, Amsterdam, The Netherlands, October 15-19, 2016*, 2016, pp. 372–376. [Online]. Available: https://doi.org/10.1145/2964284.2967245

[10] V. Lostanlen and C. Cella, "Deep convolutional networks on the pitch spiral for music instrument recognition," in *Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016, New York City, United States, August 7-11, 2016*, 2016, pp. 612–618. [Online]. Available: https://wp.nyu.edu/ismir2016/wp-content/uploads/sites/2294/2016/07/093_Paper.pdf

[11] A. Ycart and E. Benetos, "A study on LSTM networks for polyphonic music sequence modelling," in *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*, 2017, pp. 421–427. [Online]. Available: https://ismir2017.smcnus.org/wp-content/uploads/2017/10/60_Paper.pdf

[12] S. Oramas, O. Nieto, F. Barbieri, and X. Serra, "Multi-label music genre classification from audio, text and images using deep features," in *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*, 2017, pp. 23–30. [Online]. Available: https://ismir2017.smcnus.org/wp-content/uploads/2017/10/126_Paper.pdf

[13] M. E. P. Davies, P. Hamel, K. Yoshii, and M. Goto, "Automashupper: automatic creation of multi-song music mashups," *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 22, no. 12, pp. 1726–1737, 2014. [Online]. Available: https://doi.org/10.1109/TASLP.2014.2347135

[14] R. de Lima Aguiar, Y. M. G. Costa, and C. N. S. Jr., "Exploring data augmentation to improve music genre classification with convnets," in *2018 International Joint Conference on Neural Networks, IJCNN 2018, Rio de Janeiro, Brazil, July 8-13, 2018*, 2018, pp. 1–8. [Online]. Available: https://doi.org/10.1109/IJCNN.2018.8489166

[15] Y. M. G. Costa, L. S. Oliveira, and C. N. S. Jr., "An evaluation of convolutional neural networks for music classification using spectrograms," *Appl. Soft Comput.*, vol. 52, pp. 28–38, 2017. [Online]. Available: https://doi.org/10.1016/j.asoc.2016.12.024

[16] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Similarity-Based Pattern Recognition - Third International Workshop, SIMBAD 2015, Copenhagen, Denmark, October 12-14, 2015, Proceedings*, 2015, pp. 84–92. [Online]. Available: https://doi.org/10.1007/978-3-319-24261-3_7

[17] K. F. Hansen, R. Hiraga, Z. Li, and H. Wang, "Music puzzle: An audio-based computer game that inspires to train listening abilities," in *Advances in Computer Entertainment - 10th International Conference, ACE 2013, Boekelo, The Netherlands, November 12-15, 2013. Proceedings*, 2013, pp. 540–543. [Online]. Available: https://doi.org/10.1007/978-3-319-03161-3_48

[18] Y. Huang, X. Huang, and Q. Cai, "Music generation based on convolution-lstm," *Computer and Information Science*, vol. 11, no. 3, pp. 50–56, 2018. [Online]. Available: https://doi.org/10.5539/cis.v11n3p50

[19] Y. Ma, X. Li, M. Xu, J. Jia, and L. Cai, "Multi-scale context based attention for dynamic music emotion prediction," in *Proceedings of the 2017 ACM on Multimedia Conference, MM 2017, Mountain View, CA, USA, October 23-27, 2017*, 2017, pp. 1443–1450. [Online]. Available: https://doi.org/10.1145/3123266.3123408

[20] Q. Lyu, Z. Wu, and J. Zhu, "Polyphonic music modelling with LSTM-RTRBM," in *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference, MM '15, Brisbane, Australia, October 26 - 30, 2015*, 2015, pp. 991–994. [Online]. Available: https://doi.org/10.1145/2733373.2806383

[21] S. Böck, F. Krebs, and G. Widmer, "Joint beat and downbeat tracking with recurrent neural networks," in *Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016, New York City, United States, August 7-11, 2016*, 2016, pp. 255–261. [Online]. Available: https://wp.nyu.edu/ismir2016/wp-content/uploads/sites/2294/2016/07/186_Paper.pdf

[22] J. Mueller and A. Thyagarajan, "Siamese recurrent architectures for learning sentence similarity," in *Proceedings of the Thirtieth*

AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA., 2016, pp. 2786–2792. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12195

[23] S. Purushwalkam and A. Gupta, "Pose from action: Unsupervised learning of pose features based on motion," *CoRR*, vol. abs/1609.05420, 2016. [Online]. Available: http://arxiv.org/abs/1609.05420

[24] R. M. Bittner, M. Gu, G. Hernandez, E. J. Humphrey, T. Jehan, H. McCurry, and N. Montecchio, "Automatic playlist sequencing and transitions," in *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*, 2017, pp. 442–448. [Online]. Available: https://ismir2017.smcnus.org/wp-content/uploads/2017/10/86_Paper.pdf

[25] T. Li, M. Choi, K. Fu, and L. Lin, "Music sequence prediction with mixture hidden markov models," *CoRR*, vol. abs/1809.00842, 2018. [Online]. Available: http://arxiv.org/abs/1809.00842

[26] C. Laurier, J. Grivolla, and P. Herrera, "Multimodal music mood classification using audio and lyrics," in *Seventh International Conference on Machine Learning and Applications, ICMLA 2008, San Diego, California, USA, 11-13 December 2008*, 2008, pp. 688–693. [Online]. Available: https://doi.org/10.1109/ICMLA.2008.96

[27] W. Lotter, G. Kreiman, and D. D. Cox, "Deep predictive coding networks for video prediction and unsupervised learning," *CoRR*, vol. abs/1605.08104, 2016. [Online]. Available: http://arxiv.org/abs/1605.08104

[28] B. Bemman and D. Meredith, "Generating new musical works in the style of milton babbitt," *Computer Music Journal*, vol. 42, no. 1, 2018. [Online]. Available: https://doi.org/10.1162/comj_a_00451

[29] J. Schlüter and S. Böck, "Improved musical onset detection with convolutional neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014*, 2014, pp. 6979–6983. [Online]. Available: https://doi.org/10.1109/ICASSP.2014.6854953

[30] J. Schlüter and T. Grill, "Exploring data augmentation for improved singing voice detection with neural networks," in *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR 2015, Málaga, Spain, October 26-30, 2015*, 2015, pp. 121–126. [Online]. Available: http://ismir2015.uma.es/articles/264_Paper.pdf

[31] A. Salgian, D. Vickerman, and D. Vassallo, "A smart mirror for music conducting exercises," in *Proceedings of the on Thematic Workshops of ACM Multimedia 2017, Mountain View, CA, USA, October 23 - 27, 2017*, 2017, pp. 544–549. [Online]. Available: https://doi.org/10.1145/3126686.3129333

[32] Y. Lin, I. Liu, J. R. Jang, and J. Wu, "Audio musical dice game: A user-preference-aware medley generating system," *TOMCCAP*, vol. 11, no. 4, pp. 52:1–52:24, 2015. [Online]. Available: https://doi.org/10.1145/2710015

[33] J. George and L. Shamir, "Computer analysis of similarities between albums in popular music," *Pattern Recognition Letters*, vol. 45, pp. 78–84, 2014. [Online]. Available: https://doi.org/10.1016/j.patrec.2014.02.021

[34] L. Nanni, Y. M. G. Costa, A. Lumini, M. Y. Kim, and S. Baek, "Combining visual and acoustic features for music genre classification," *Expert Syst. Appl.*, vol. 45, pp. 108–117, 2016. [Online]. Available: https://doi.org/10.1016/j.eswa.2015.09.018

[35] G. Yu and J. E. Slotine, "Audio classification from time-frequency texture," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2009, 19-24 April 2009, Taipei, Taiwan*, 2009, pp. 1677–1680. [Online]. Available: https://doi.org/10.1109/ICASSP.2009.4959924

[36] A. Haron, S. X. Yong, and W. C. Onn, "Cellular music: An interactive game of life sequencer," in *2018 ACM Multimedia Conference on Multimedia Conference, MM 2018, Seoul, Republic of Korea, October 22-26, 2018*, 2018, pp. 2081–2083. [Online]. Available: https://doi.org/10.1145/3240508.3264577