JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 38, 517-529 (2022)
DOI: 10.6688/JISE.202205_38(3).0002

VAE+NN: Interpolation Composition by Direct Estimation of Encoded Vectors Against Linear Sampling of Latent Space*

PABLO LÓPEZ DIÉGUEZ AND VON-WUN SOO
Department of Computer Science
National Tsing Hua University
Hsinchu, 300 Taiwan
E-mail: pablo@gapp.nthu.edu.tw; soo@cs.nthu.edu.tw

In this paper, we introduce a machine learning technique to estimate the vector encoded by a Variational Autoencoder (VAE) model, without the need of explicitly sampling the vector from the VAE's latent space. The feasibility of our approach is evaluated in the field of music interpolation composition, by means of the *Hsinchu Interpolation MIDI Dataset* that was created. A novel dual architecture of VAE plus an additional neural network (VAE+NN) is proposed to generate a polyphonic harmonic bridge between two given songs, smoothly changing the pitches and dynamics of the interpolation. The interpolations generated by the VAE+NN model surpass a Random data baseline, a bidirectional LSTM model and the state-of-the-art interpolation approach in automatic music composition (VAE model with linear sampling of the latent space), in terms of reconstruction MSE loss. Furthermore, a subjective evaluation was done in order to ensure the validity of the metric-based results.

Keywords: VAE, variational autoencoders, interpolation, composition, polyphonic music, latent space, encoded vector

1. INTRODUCTION

In the 21st Century, Artificial Intelligence has brought innumerable innovations applicable to almost every field. Regarding music composition, although relatively successful results have been obtained in the past [1, 2], the generation constrains have almost always been the same: generating music conditioned on the immediate previous events [3, 4]. But, what if we wanted to condition the music composition on both the past and the future events? We call this the interpolation music composition problem. Until now, there existed very limited approaches that addressed this problem. This paper constitutes a further step in understanding the generative model-based available alternatives to create interpolations, as an extension of our conference paper "Variational Autoencoders for Polyphonic Music Composition" [5]. The motivation of interpolation music composition lies in the applications of musical transitions in the modern entertainment industry: music composition for transition of scenes in cinema, music composition between an advertisement and a TV series theme song, real-time music interpolation for musicians, *etc.* The practical demands of interpolation music composition are worth to be explored further. Our paper aims to use a type of Deep Generative Model (DGM) [6], the Variational Autoencoder (VAE), to solve the artistic and creative problem of generating a suitable interpolation music segment between two different songs. However, the interpolation music examples for machine lear-

Received January 28, 2021; accepted April 6, 2021.
Communicated by Po-Chyi Su.
* This research is supported in part by Taiwan MOST research project under the Grants 108-2221-E-007-076 and 109-2221-E-007-097.

ning are hard to collect in practice. To investigate the nature of the interpolation music composition, we considered specifically taking the last 10 seconds from the first song and the first 10 seconds from the second song, creating an interpolation song of 10 seconds too. Therefore, the ideal training dataset would contain 30 seconds tracks, separated in Begin, Interpolation and End tracks.

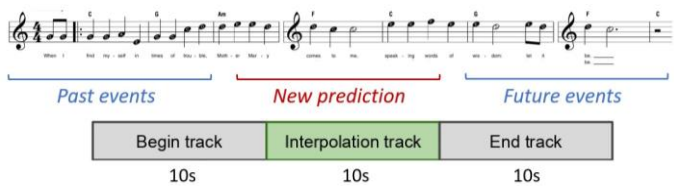

Fig. 1. Definition of interpolation.

2. RELATED WORK

Two papers for musical applications can be regarded as the state-of-the-art for their interpolation composition approach. The first one, "A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music" by Roberts *et al.*, 2018 [7], is perhaps the closest approach to generating an interpolating music piece between two given tracks using Variational Autoencoders. The main goal of this VAE with a novel hierarchical decoder model was not interpolation itself but modelling long-term sequences. It consisted of a naive and low level heuristic: direct sampling of the interpolation encoded vector from the latent space that could only generate monophonic music (one single note per timestep). Although the results obtained were positive, the training dataset was too big and it is not publicly available, containing around 1.5 million of unique files and making the construction of the model too computationally demanding. We focused on the interpolation problem of polyphonic music (several notes per timestep [8]) and on improving the resource and time constraints while keeping the positive results of the music generated. Furthermore, we also aimed to develop a higher level interpolation technique to improve the performance of the model training and its overall results.

The second paper, "Modeling Dynamics and Instrumentation of Music with Applications to Style Transfer" by Brunner *et al.*, 2018 [9], constitutes another approach that used exactly the same naive interpolation technique as in the first paper [7]. The main goal of the VAE proposed by Brunner *et al.* [9] was to perform a style transfer on symbolic music by automatically changing pitches, dynamics and instruments of a music piece from different styles. This approach modelled polyphonic music by using a GRU (Gated Recurrent Unit) based architecture, requiring a training of 48 hours using a single GTX 1080 GPU. Since the model could also interpolate linearly between short pieces of music, although requiring a style classifier for genre prediction mapped directly into the latent vector of the VAE as an extra term, we focused on improving the interpolation capability. Therefore, we propose a more advanced interpolation method that could at the same time get rid of the style classifier for genre estimation and lowering the computational requirements while maintaining a good performance on the interpolation generation. The interpolation method that we propose is based on the direct estimation of the interpolation latent vector with an additional data-based approach that uses machine learning.

3. BACKGROUND

3.1 Symbolic Music Representation

We use MIDI (Musical Instrument Digital Interface) [10] to represent the polyphonic music in a symbolic way, instead of using the raw waveform, which is computationally very expensive to manipulate. Each timestep of a song is represented as a vector of 64 binary elements in the range [28, 92] of MIDI pitches, where each binary element represents one piano key (or one note or pitch), meaning note on/off. The complete pianoroll track representation is a tensor of shape (number of timesteps × note value × hold on/off) = (number of timesteps × 64 × 2); where the number of timesteps is the length of a song.

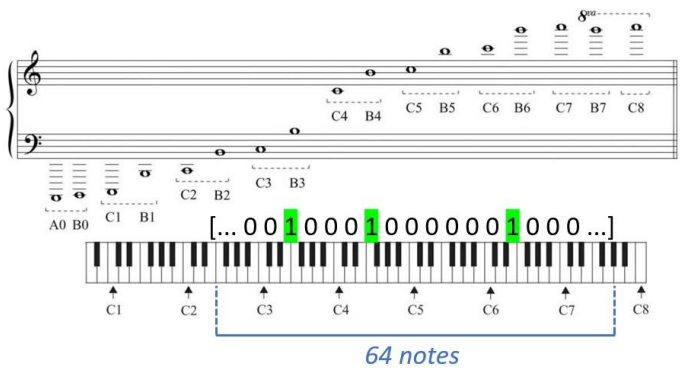

Fig. 2. Polyphonic modelling.

4. HSINCHU INTERPOLATION MIDI DATASET

We created the *Hsinchu Interpolation MIDI Dataset* based on the Lakh MIDI Dataset [11]. Firstly, all the piano tracks from the Lahk MIDI Dataset were extracted. Then, each piano track was divided in segments of 10 seconds, where the consecutive segments were put in groups of 3 tracks (Begin, Interpolation, End), constituting training examples of 30 seconds each. We were only interested in collecting the 30 seconds tracks which Begin and End track were as dissimilar as possible (high information value). Therefore, the dissimilarity of each pair of Begin and End tracks was evaluated by a simple 4-layer neural network, with sizes between 32 to 128 units, and ReLU as activation function. The output layer uses a sigmoid activation function, since we were solving a binary classification problem: prediction of two songs as similar or dissimilar.

This neural network was trained on 224 pairs of Begin and End songs labelled manually as similar or dissimilar with a subjective criteria based on: tempo, majority of single notes or chords, mode, time signature and melody similarity. We maximized the recall on the dissimilar label, since it was preferred to identify correctly most of the dissimilar tracks (high learnable information value) at a cost of including some similar examples (low learnable information value) in our dataset. AdaMax was chosen as optimizer, with parameters learning rate = 0.005, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and binary cross entropy was selected as loss function. After training, the recall of the model for the label of interest was 0.75. Besides,

the neural network utilized different features related to tempo, pitch and key of the songs, which were extracted from the MIDI files by means of the library *pretty midi* [12].

In the end, only the 30 seconds tracks (each track has Begin, Interpolation, End) whose Begin and End were predicted as dissimilar by the neural network were included in the newly created dataset. The final *Hsinchu Interpolation MIDI Dataset* contained 30,830 MIDI files of 30 seconds each (including Begin, Interpolation (ground truth) and End tracks); or 92,490 independent MIDI files of 10 seconds each.

5. EXPERIMENTAL PROCEDURE: MODELS

5.1 Random Data Baseline

As a first approximation to the generation of an interpolation track, we aimed to use a non Machine Learning approach to construct random baseline data interpolations, by sampling a Bernoulli random variable with a parameter f to choose an element from either the initial or the end track for each timestep [7], for $f \in [0, 1]$. The probability p of the Bernoulli random variable varies linearly along the interpolation track that is being constructed: at the beginning of the interpolation, the chances of selecting a timestep from the Begin track are 100%; at the end, p is 0%. This is done to ensure the transition of the Interpolation track between the Begin and End tracks. This naive baseline technique was evaluated in 84 segments from the *Hsinchu Interpolation MIDI Dataset* using an objective MSE loss metric as in the reviewed literature [9]. We compared the generated interpolation segments with the ground truth composed by humans. Although the generated tracks using this method sounded much better than our initial expectations, they lacked musical coherence in most of the cases and therefore were not included into the subjective evaluation. A higher level approach was necessary, and we aimed to use Machine Learning based techniques to solve the interpolation problem.

5.2 Bi-Directional LSTM

The second baseline model is a bi-directional LSTM. Recurrent neural networks have been used in machine learning for music composition in the past, for their ability to capture both the local structure of melody and the long-term structure of a of a musical style [4]. The architecture of the bi-LSTM neural network consisted of 4 bi-directional LSTM layers with variable number of units (between 128 and 512), dropout and ReLU as activation function. Afterwards, the output dense layer of 128 units was added with sigmoid as activation function.

The tracks of the *Hsinchu Interpolation MIDI Dataset* were divided in timesteps in order to perform the training. 10 timesteps were used as input sequence, which corresponds with around 2.4 seconds of the MIDI track; the next timestep in the sequence (11^{th} timestep) would be predicted by the model. In total, almost 2 million training examples were gathered from the 30,830 songs contained in the dataset. The training was performed during 10 epochs using Adam optimizer with a learning rate of $5 \cdot 10^{-7}$. Binary cross entropy was chosen as loss function based on the multi-label multi-classification definition of the problem: several notes can be played during the same timestep.

5.3 VAE

The first model proposed in this paper to solve the interpolation problem is based on Variational Autoencoders. We developed a model with a non-traditional convolutional architecture that aims to utilize the state-of-the-art approach for music interpolation composition: the linear sampling of the latent space.

The probabilistic encoder neural network $q_\phi(z|x)$ consisted of 4 convolutional layers, after each of which batch normalization is performed [13]. Leaky ReLU is chosen as the activation function. 2 fully-connected layers are added after the last convolutional layer, to directly map the μ and σ vectors that form the compressed or latent space of the VAE. This compressed representation was chosen to have a dimensionality of 100 and, therefore, any encoded vector would have a shape of (1, 100).

1. Input 2D Convolutional layer of 32 units×(kernel 4×4), Leaky ReLU.
2. 2D Convolutional layer of 64 units×(kernel 4×4), Leaky ReLU.
3. 2D Convolutional layer of 128 units×(kernel 4×4), Leaky ReLU.
4. 2D Convolutional layer of 256 units×(kernel 4×4) (4096 units), Leaky ReLU.
5. 2 Fully-connected linear layers with 100 units each: mean and standard deviation vectors (latent code).

The probabilistic decoder $p_\theta(x|z)$ consisted of a fully-connected layer mapping the latent space to the 4 convolutional layers. Batch normalization was also applied after each of them:

1. Fully-connected linear layer with 4096 units (from the latent code).
2. 2D Convolutional layer of 256 units×(kernel 3×3), Leaky ReLU.
3. 2D Convolutional layer of 128 units×(kernel 3×3), Leaky ReLU.
4. 2D Convolutional layer of 64 units×(kernel 3×3), Leaky ReLU.
5. 2D Convolutional layer of 32 units×(kernel 3×3), sigmoid.

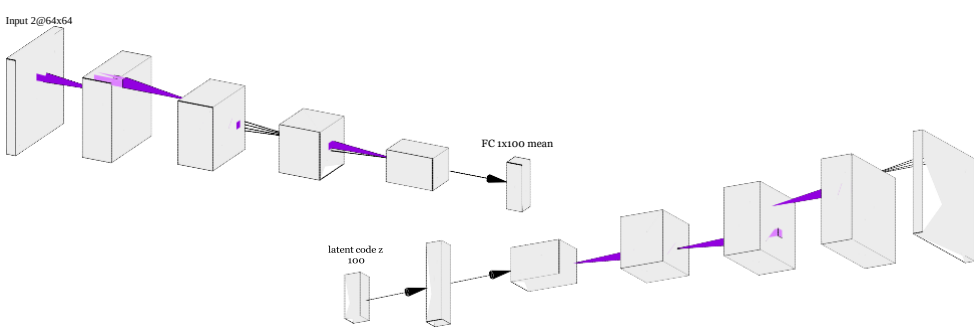

Fig. 3. CNN encoder and decoder architecture.

5.4 VAE + NN

The second model proposed in this paper to solve the interpolation problem is presented as an alternative to the previous VAE model and it is based on a new combined architecture of 1) a Variational Autoencoder (VAE) plus 2) a Neural Network (NN). The

base VAE model is exactly the same as the one explained in the previous section, with identical architecture of the probabilistic encoder $q_\varphi(z|x)$ and decoder $p_\theta(x|z)$. The main difference lies in the additional neural network added on top of the architecture of the VAE, as an alternative approach to the linear sampling of the latent space when constructing the interpolation. The neural network is used to directly estimate the Interpolation encoded vector ($z_{interpolation}$), taking Begin and End encoded vectors as inputs (z_{begin} and z_{end}, respectively). The architecture of the neural network consisted of 4 feed forward layers, the first of which had the particularity of being initialized with two weight matrices $H_{1,1}$ and $H_{1,2}$ for the inputs encoded Begin z_{begin} and encoded End z_{end} vectors, respectively, plus a bias vector b_1:

$$k_{hidden\_1} = z_{begin}H_{1,1} + z_{end}H_{1,2} + b_1, \tag{1}$$

$$a_{hidden\_1} = tanh(k_{hidden\_1}). \tag{2}$$

The first 3 layers (with 200, 100 and 100 units) were activated by a *tanh* function, whereas the last layer (100 units) used Leaky ReLU. The novelty of this VAE + NN model lies on both the new combined architecture and also the same efficient approach sought for training as in the previous model (VAE), due to the *Hsinchu Interpolation MIDI Dataset* again.

6. DIRECT ESTIMATION OF ENCODED INTERPOLATION VECTOR AGAINST LINEAR SAMPLING OF LATENT SPACE

In this section, we focus on the two interpolation methods performed by the VAE and VAE+NN models introduced in this paper, discussing the similarities and differences of each approach, and presenting the benefits of our interpolation method against the more traditional interpolation method that appears in previous works [7, 9]. Specifically, we introduce the direct estimation of the encoded interpolation vector performed by the VAE+NN model, as an alternative to the linear sampling of latent space performed by the VAE model. Furthermore, we would like to note that the interpolation approaches performed by our baseline models (Random data and Bi-directional LSTM) were set aside in this section, due to the intrinsic lack of innovation that these methods represented. A detailed comparison among the interpolations produced by all the models will be done in the Results section of this paper.

6.1 VAE: Linear Sampling of Latent Space

We developed a VAE model with a non-traditional convolutional architecture in the automatic composition field to solve the interpolation problem, utilizing the state-of-the-art approach for music interpolation composition: the linear sampling of the latent space.

6.1.1 Training

The VAE model was trained on the *Hsinchu Interpolation MIDI Dataset*, consisting of 92,490 MIDI files of 10 seconds each. The training was done on a single GTX 1080

GPU, taking less than 4 hours to converge. After the training, the VAE model was capable of encoding a 10 seconds MIDI song into a vector, and decoding it back to a reconstruction of the originally given MIDI song.

6.1.1 Interpolation method

The interpolation of the VAE model was performed by means of a linear sampling of the latent space, computing

$$z_{interpolation} = f z_{begin} + (1 - f) z_{end} \tag{3}$$

for $f \in [0, 1]$. That can be seen as, firstly, encoding the Begin and End tracks, in order to obtain z_{begin} and z_{end}, respectively. Secondly, averaging (if the weighting variable f is set to 0.5) as in

$$z_{interpolation} = (z_{begin} + z_{end})/2 \tag{4}$$

the encoded Begin and End tracks in order to obtain the Interpolation encoded vector $z_{interpolation}$. Finally, the Interpolation encoded vector would be decoded using the VAE to obtain the Interpolation track within its original dimensionality (a tensor transferable to MIDI format).

6.2 VAE + NN: Direct Estimation of Encoded Interpolation Vector

As an alternative to the relatively naive method of obtaining the interpolation encoded vector by averaging the begin and end encoded vectors, or so-called linear sampling of the latent space, we developed a higher level estimation method of the interpolation encoded vector by means of a neural network architecture stacked onto the base VAE model. This novel machine learning approach to perform the interpolation exploits the ability of neural networks to learn "hidden" or "latent" features in music from the ground truth (interpolation songs composed by humans that would be encoded into vectors later), that could not be obtained with a simple averaging of the encoded inputs (z_{begin} and z_{end}) due to the sensitiveness of the features present in the latent code.

6.2.1 Training

The training of the combined architecture model that is proposed consisted on the Variational Autoencoder (VAE) training plus the training of the neural network (NN):

1. VAE: The base VAE model is the same as the previous VAE model and, therefore, was also trained on the *Hsinchu Interpolation MIDI Dataset* with the same resources (single GTX 1080 GPU) and the same metrics (less than 4 hours to converge).
2. NN: Prior to the training of the NN, certain pre-processing of the *Hsinchu Interpolation MIDI Dataset* had to be done, since the neural network is required to learn how to predict the Interpolation encoded vector ($z_{interpolation}$), taking the Begin and End encoded vectors (z_{begin} and z_{end}, respectively) as inputs. Therefore, all the files in the dataset were encoded using the base VAE, obtaining all the encoded vectors z_{begin}, $z_{interpolation}$ and

z_{end} from the Begin, Interpolation and End original tracks. Then, the training of the neural network was performed on the 30,830 encoded trio examples (92,490 consecutive tracks in groups of 3) on a single GTX 1080 GPU during 3500 epochs. Adam optimizer was chosen and the selected learning rate was 0.0001. The training was very fast due to the small dimensionality of the encoded vectors (1, 100) in comparison to the original tensor representation of each 10 seconds MIDI track (42, 64, 2). When predicting only 100 (1×100) elements versus predicting 5,376 (42×64×2) elements, the difference in terms of required computational power was very noticeable.

6.2.2 Interpolation method

Firstly, the base VAE model would be used to encode the Begin and End tracks into vectors, obtaining z_{begin} and z_{end}, respectively. Afterwards, z_{begin} and z_{end} would be input to the NN model, obtaining the encoded vector of the interpolation song. Finally, after the new $z_{interpolation}$ is obtained with the NN, it can be decoded utilising the base VAE again, returning the Interpolation encoded vector of size (1, 100) to its original tensor size (42, 64, 2), which can be converted to MIDI format. The full architecture of the VAE+NN model can be seen in Fig. 4.

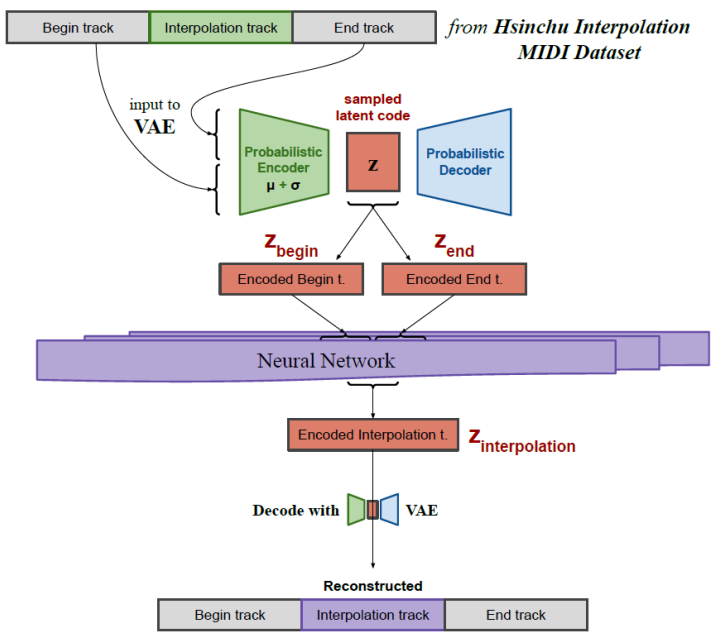

Fig. 4. VAE + NN architecture.

7. RESULTS

7.1 Metric-Based Results

We proposed a Variational Autoencoder model based on convolutional neural networks (VAE model) that achieves results comparable to the state-of-the-art when perform-

ing music interpolation. In addition, the model exhibits an improved efficiency in terms of computational and time resources during training due to the *Hsinchu Interpolation MIDI Dataset* that was created. While VAE models in previous works required a 48 hours training [9], our VAE model only needs 4 hours (both models being trained on a single GTX 1080 GPU). This is due to the training data reduction performed by the novel approach of the *Hsinchu Interpolation MIDI Dataset* created, which only includes transitions between songs with dissimilar musical characteristics, reducing the amount of invaluable training examples. The dataset itself is another result of the work presented in this paper, constituting, to the best of our knowledge, the first dataset for music interpolation.

The results of the second model proposed in this paper (VAE+NN) show that the novel combined architecture of 1) a Variational Autoencoder model based on convolutional neural networks plus 2) a feed-forward neural network for encoded vector estimation outperforms the state-of-the-art in terms of MSE loss. Furthermore, the combined architecture model also exhibits an improved efficiency regarding computational and time resources due to the *Hsinchu Interpolation MIDI Dataset*. The proposed novel interpolation method is proven to be not only feasible but also objectively better than the linear sampling technique under MSE loss computation. The MSE losses were computed on 84 unseen 30 seconds tracks in the form of pianoroll tensor, for each of the models developed: (1) Bi-LSTM baseline; (2) VAE with linear sampling of latent space and (3) VAE + NN with direct estimation of the encoded interpolation vector (see Table 1). The MSE values shown in the table are the average results after 30 computations. The τ column on the left of the table indicates the threshold value when setting the notes in the pianoroll tensor to a binary value (0: note off, 1: note on). The threshold can also regulate the amount of notes being played per timestep. As it can be observed in Table 1, the VAE model is superior to the bi-LSTM baseline model in 7 out of 9 thresholds; while the VAE + NN model outperforms the bi-LSTM in 8 out of 9 thresholds. VAE + NN is better than the VAE model in all the thresholds. Moreover, we compared the ML approaches with the Random data baseline by averaging the MSE losses along the thresholds. These results also show the superiority of the VAE + NN model with respect to all the models considered in this paper (see Table 2). In this evaluation, the bi-LSTM appears to be superior to the VAE model, but this false perception is only due to the effect of averaging the losses along the thresholds (bi-LSTM is only superior in the $\tau = 0.10$ threshold due to the MSE loss disadvantage of heavily weighting outliers when squaring each term).

Table 1. Pianoroll track MSE loss values for Bi-LSTM, VAE (linear sampling) and VAE+ NN (*z* prediction) models.

τ	Bi-LSTM (1)	VAE (2)	VAE+NN (3)	Improv. 3 vs 1	Improv. 3 vs 2
0.10	**0.0600**	0.0934	0.0840	−40%	10.06%
0.20	0.0554	0.0673	**0.0464**	16.25%	31.05%
0.30	0.0528	0.0523	**0.0384**	27.27%	26.58%
0.40	0.0479	0.0443	**0.0371**	22.55%	16.03%
0.50	0.0447	0.0375	**0.0364**	18.57%	2.93%
0.60	0.0434	0.0375	**0.0365**	15.90%	2.67%
0.70	0.0401	0.0374	**0.0364**	9.23%	2.67%
0.80	0.0398	0.0374	**0.0364**	8.54%	2.67%
0.90	0.0392	0.0373	**0.0364**	7.14%	2.41%

Table 2. Pianoroll track averaged MSE loss values for Random data baseline, Bi-LSTM, VAE (linear sampling) and VAE+NN (*z* prediction) models.

Rand. (1)	Bi-LSTM (2)	VAE (3)	VAE+NN (4)	Improv. 4 vs 1	Improv. 4 vs 2	Improve. 4 vs 3
0.0581	0.0470	0.0494	**0.0431**	25.82%	8.30%	12.75%

Furthermore, we computed the MSE loss of the encoded interpolation vectors in order to evaluate the tracks in the latent space, only for those models that were able to perform any sort of latent space manipulation: VAE and VAE+NN (see Table 3). According to this latent code evaluation, the VAE+NN model outperforms again the VAE model with the linear sampling approach.

Table 3. Encoded vector MSE loss values for VAE (linear sampling) and VAE+NN (*z* prediction) models.

VAE (3)	VAE+NN (4)	Improvement (2) vs (1)
1.3726	**1.0611**	22.69%

7.2 Listening Test Results

Besides the metrics taken into account for the objective evaluation, a subjective evaluation was done in the form of a quantitative user study. In the listening test, the preferences of 32 participants from 8 different countries distributed among Africa, America, Asia and Europe were gathered. 65.6% of the participants declared to have musical background or to play an instrument. Out of the 224 pair-wise comparisons performed, the bi-LSTM baseline obtained the lowest selection rate and the interpolations composed by humans were, in general, preferred over the other models' interpolations. The first model proposed (VAE) was selected in 75% of the cases over the bi-LSTM model, whereas the second model proposed (VAE + NN) was preferred over both the bi-LSTM and the VAE model. These findings fully support the results drawn from the objective evaluation. Further, a binomial test showed that 4 out of the 6 pair-wise comparisons exhibited a statistically significant difference between the models, with p-value = 0.05. The fact that the remaining 2 comparisons (VAE + NN vs bi-LSTM and VAE + NN vs human) did not show a statistically significant difference was due to the small sample size of the test. The details of the listening test are shown in the Table 4 and in Fig. 5 (asterisk for a pair-wise comparison indicates a statistically significant difference in the model selection) and Fig. 6 (total number of wins).

Table 4. Quantitative listening test results and statistical significance.

support of (1)	model (1)	model (2)	support of (2)	*p*-value
75%	**VAE**	bi-LSTM	25%	0.0035
31.25%	VAE	**VAE+NN**	68.75%	0.0251
31.25%	VAE	**human**	68.75%	0.0251
59.40%	**VAE+NN**	bi-LSTM	40.60%	0.1885
34.40%	VAE+NN	**human**	65.60%	0.0551
28.10%	bi-LSTM	**human**	71.90%	0.0100

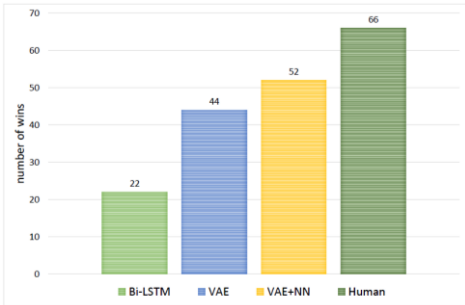

Fig. 5. Quantitative listening test results: number of wins per model.

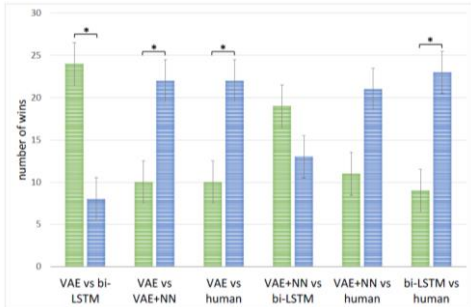

Fig. 6. Quantitative listening test results: number of total wins.

8. CONCLUSION

In this paper, two machine learning models based on Variational Autoencoders are proposed to generate a polyphonic interpolation piece between two songs, smoothly changing the pitches and dynamics of the interpolation to comply with the musical constraints of the MIDI tracks provided at the beginning and at the end of the interpolation.

Besides, the *Hsinchu Interpolation MIDI Dataset* was created, constituting, to the best of our knowledge, the first dataset for music interpolation. The first model is a Variational Autoencoder based on convolutional neural networks that uses a linear sampling of the latent space to perform the interpolation (VAE model), exhibiting an improved efficiency in terms of computational and time resources during training with respect to the state-of-the-art, due to the new dataset. This VAE model is superior to a Random data interpolation method and it also outperforms a bidirectional LSTM baseline model under MSE loss evaluations. The second proposed model is a novel combined architecture of (1) a Variational Autoencoder model based on convolutional neural networks and (2) a feed-forward neural network (VAE+NN model). The neural network directly estimates the interpolation encoded vector from the begin and end encoded vectors, constituting a novel method to perform music interpolations. In all the cases that were evaluated, the VAE+NN model outperformed the first model (VAE model) in terms of MSE loss in both the pianoroll space and the latent space, exhibiting the same efficiency in terms of computational and time resources during training. Also, the VAE + NN model surpassed the Random data and the bi-LSTM models.

Besides the objective metric-based results, a quantitative listening test was done. The results of the listening test fully support the conclusions drawn from the objective evaluation, proving the improvement in the quality of the interpolation generated by the VAE + NN model with respect to the VAE and the bi-LSTM approaches.

To facilitate future research on polyphonic music interpolation, we make our code[1] and data[2] publicly available.

8.1 Future Work

In this paper, Variational Autoencoder based models have been proven to be very

[1] https://github.com/pablomp3/ML-interpolation-Master-Thesis
[2] https://github.com/pablomp3/HsinchuInterpolationMIDIDataset

powerful generative models for applications in the music composition field. In the future, we would like to address the interpolation problem solved in this work with different VAE models and architectures. Also, the use of Flow-based deep generative models and even the improvement of VAEs with Conditional Flows could be considered. Furthermore, keep improving the quality and size of the *Hsinchu Interpolation MIDI Dataset* would be desirable if we aim to make it standing out as a reference dataset in the Music Information Retrieval field. Finally, one important implication of this paper for future research is the novel interpolation method proposed, which deserves further study and may be applicable to different domains that demand solving interpolation tasks, such as biological sequence design and NLP. We believe that the use of other machine learning techniques to perform the estimation and manipulation of the latent code generated by VAEs could lead to new research paths in the field of machine and deep learning, along with applications with huge potential, which are definitely worth exploring further.

REFERENCES

1. R. K. Zaripov, "An algorithmic description of a process of musical composition," *Doklady Akademii Nauk SSSR*, Vol. 132, 1960, pp. 1283-1286.
2. S. I. Mimilakis, E. Cano, J. Abeßer, and G. Schuller, "New sonorities for jazz recordings: Separation and mixing using deep neural networks," in *Proceedings of Audio Engineering Society Workshop on Intelligent Music Production*, Vol. 140, 2016.
3. S. Dai, Z. Zhang, and G. G. Xia, "Music style transfer: A position paper," *arXiv Preprint*, 2018, arXiv:1803.06841.
4. D. Eck and J. Schmidhuber, "A first look at music composition using LSTM recurrent neural networks," Technical Report, Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale, 2002.
5. P. L. Dieguez and V. W. Soo, "Variational autoencoders for polyphonic music interpolation," in *Proceedings of International Conference on Technologies and Applications of Artificial Intelligence*, 2020, pp. 56-61.
6. A. Karpathy, P. Abbeel, G. Brockman, *et al.*, "Generative models," https://openai.com /blog/generative-models/, 2016.
7. A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A hierarchical latent vector model for learning long-term structure in music," in *Proceedings of the 35th International Conference on Machine Learning*, 2018.
8. B. Benward and M. N. Saker, *Music: In Theory and Practice*, McGrawHill, Boston, 2003.
9. G. Brunner, A. Konrad, Y. Wang, and R. Wattenhofer, "Midi-VAE: Modeling dynamics and instrumentation of music with applications to style transfer," in *Proceedings of the 19th International Society for Music Information Retrieval Conference*, 2018.
10. D. M. Huber, *The MIDI Manual*, SAMS Carmel, Indiana, 1991.
11. C. Raffel, "Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching," PhD Thesis, Graduate School of Arts and Sciences, Columbia University, 2016.
12. C. Raffel and D. P. W. Ellis, "Intuitive analysis, creation and manipulation of midi data with pretty midi," in *the 15th International Conference on Music Information Re-*

trieval Late Breaking and Demo Papers, 2014.
13. X. Hou, K. Sun, L. Shen, and G. Qiu, "Deep feature consistent variational autoencoder," in *Proceedings of IEEE Winter Conference on Applications of Computer Vision*, 2017, 1133-1141.

Pablo López Diéguez received the B.S. degree in Electronic and Automation Engineering from the University of Oviedo, Spain, and the M.S. degree from the Institute of Information Systems at National Tsing Hua University, Taiwan, in 2018 and 2020, respectively. His research interests include deep generative models and automatic music composition.

Von-Wun Soo received his B.S. degree from the Department of Electrical Engineering, National Taiwan University, Taiwan, and his Ph.D. degree from the Department of Computer Science, Rutgers University, USA. He is currently a Professor at National Tsing Hua University. His research interests include virtual singers, multi-drone coordination and text generation.