Electronic Preprint for Journal of Information Processing Vol.28

Regular Paper

A Symbol-level Melody Completion Based on a Convolutional Neural Network with Generative Adversarial Learning

Kosuke Nakamura[1,a)] Takashi Nose[1,b)] Yuya Chiba[1,c)] Akinori Ito[1,d)]

Received: July 5, 2019, Accepted: January 16, 2020

Abstract: In this paper, we deal with melody completion, a technique which smoothly completes partially-masked melodies. Melody completion can be used to help people compose or arrange pieces of music in several ways, such as editing existing melodies or connecting two other melodies. In recent years, various methods have been proposed for realizing high-quality completion via neural networks. Therefore, in this research, we examine a method of melody completion based on an image completion network. We represent melodies as images and train a completion network to complete those images. The completion network consists of convolution layers and is trained in the framework of generative adversarial networks. We also consider chord progression from musical pieces as conditions. From the experimental result, it was confirmed that the network could generate original melody as a completion result and the quality of the generated melody was not significantly worse than the result of a simple example-based melody completion method.

Keywords: melody completion, automatic music composition, convolutional neural networks, generative adversarial networks

1. Introduction

Automated music generation has been studied as a technology that helps people who do not have musical knowledge or skills to work on music creation [20]. Music generation is a technology that uses machines to automate music creation activities such as composition, arrangement, and performance without human intervention. In addition to lowering the threshold of music creation, research on music generation has been conducted with backgrounds such as human cost reduction and understanding of music creation.

The automatic composition is a sort of music generation. Using an automatic composition system, those who are not be able to compose can obtain new music with less effort and time.

Many of those methods generate musical pieces according to inputs such as lyrics, noise vectors, or some parameters such as note density. On the other hand, for people who do not have composition skills and want to create musical pieces with an atmosphere similar to specific existing pieces, a method to generate music based on these pieces can be useful.

Therefore, we examine melody completion, which receives a partially masked score and generates the masked part so that the generated part naturally compensates the masked part. The melody completion realizes partial editing of pieces by masking melodies and completing them. Moreover, by repeating masking and completing sequentially from the beginning of the pieces, we can generate a new piece that has the same structures and chord progression as the original music.

In this research, we propose a melody completion network that completes melodies considering both pitches and rhythms by applying image completion. In this research, we opt for applying image completion to complete melodies. Image completion is a technique that allows filling masked regions of images with alternative contents. Recently, various methods for high-quality completion via neural networks have been proposed. Thus we express a melody as an image and investigate a melody completion network similar to the image completion network [19]. We also consider the chord progression from musical pieces as conditions.

2. Related Works

2.1 Automatic Composition and Melody Completion

As soon as the automatic computing device was invented, the automatic composition was supposed to be an application of the computation [5]. Since then, there have been many attempts to realize automatic composition systems to date. Rule-based methods have been used in the 1990s [28], and the methods of composition were changed to that based on machine learning.

There have been many automatic composition methods; here are a part of those works. Biles et al. proposed an automatic composition system that generates a jazz solo using the genetic algorithm [1]. Roig et al. developed an automatic composition system that uses statistical models to model the rhythm patterns

[1] Graduate School of Engineering, Tohoku University, Sendai, Miyagi 980–8579, Japan
[a)] kosuke.nakamura.p6@spcom.tohoku.ac.jp
[b)] tnose@m.tohoku.ac.jp
[c)] yuya@spcom.ecei.tohoku.ac.jp
[d)] aito@spcom.ecei.tohoku.ac.jp

and pitch contour [25]. Fukayama et al. proposed an automatic composition system Orpheus, which generates melodies based on the prosody of Japanese lyrics [6]. In this system, the composition is regarded as the maximum likelihood search problem of the sound path, and by solving the problem under the constraint condition based on the prosody of the lyrics, the melody matching the lyrics is generated. Yang et al. generated natural melody using the Deep Convolutional Generative Adversarial Network (DCGAN), which has been extensively used for artificial image generation [30]. A similar method has been extended to generate multi-track melody generation [4]. Moreover, Columbo et al. developed an automated composition system based on the recurrent neural network (RNN) [2]. They used two RNNs to generate pitch and duration independently.

The melody completion is an application of the automatic composition, which automatically generates only a part of the melody given the surrounding context. There have been several methods to complete four-part Bach chorales using a convolutional neural network (CNN), long-short term memories (LSTM) and Gibbs sampling [10], [11], [17]. The problem of their methods is that this method can be applied to the four-part chorales, and thus it cannot be applied other genres such as pops.

2.2 Image Completion

Image completion or image inpainting [24] is a technique to complement the deleted or lost parts of an image naturally. Traditionally, image completion was based on pasting the image patches onto the missing region [16], [26]. In recent years, methods based on neural networks have been studied [12], [23], [29], [31]. Iizuka et al. proposed a method of image completion that results in images considering both local and global consistency [12]. In this method, the completion network is a CNN, trained using a generative adversarial network (GAN) technique [8] with global and local context discriminators.

3. Melody Completion

As described above, the melody completion is a technique to generate a partial melody given the surrounding melody. **Figure 1** shows an example of the melody completion. In this example, the second measure is masked, and then the new melody is generated so that the generated melody naturally fits the masked area.

In this paper, we develop a melody completion method based on Iizuka's work [12], where a melody is represented as an image, and the partially-masked melody image is inpainted using a convolutional neural network trained by the GAN framework.

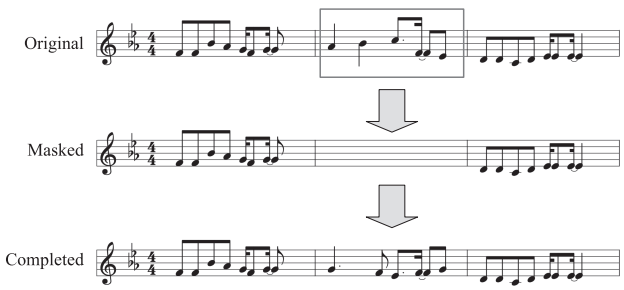


Fig. 1 An example of melody completion.

The reason why we employed a CNN and GAN based method is twofold. First, since many of neural-network-based music generation methods employ either RNN or CNN [2], [10], [15], [17], [30], it is natural to employ RNN or CNN. Among them, a combination of CNN and GAN is proved to be effective for symbolic music generation [30]. Second, the CNN+GAN method by Iizuka explicitly consider the local and global constraints, which are interpreted as the short-term and long-term constraints in the context of music generation. Besides, Iizuka's method gave the state-of-the-art performance for image completion. Thus we employed Iizuka's method for melody completion.

There are a few aspects for evaluating the result of melody completion. Since the image completion is used for restoring the damaged image [22], naturalness is the most important metric for image completion. If the image completion method restores the original image, the method is regarded to be good. However, our purpose of melody completion is not recovering the original melody but obtaining a melody that naturally fits the masked part and it has some originality [3]. Therefore, we need to evaluate the result of melody completion from originality and naturalness points of view. The GAN-based neural network is often used for synthesizing a new image [21]. Therefore, we expect a GAN-based network to generate a new melody that does not exist in the training data.

4. Representation of Melody and Chord

In this section, we describe the ways to represent melodies and chords as images.

4.1 Melody Representation

There are several choices as a representation of melody. For example, DeepArtificialComposer [2] uses the symbolic representation (alphabet) that uniquely expresses pitch and duration. MidiNet [30] uses an image as a representation of a melody. Besides, DeepBach [10] uses a list of symbols of four notes of a chord. Among them, we expressed a melody as an image in a similar way to that of MidiNet, because an image-like representation is easily combined with CNN, and we can combine other information (such as the masked part) by overlaying them as different channels.

We extract four-measure melodies from musical pieces and converted them into images. The vertical direction of the image corresponds to the pitches of the notes, and the horizontal direction of the image corresponds to the beat/time. Each pixel value of the image is set to 1 if there is a note of the corresponding pitch at the corresponding time, otherwise 0. Then we obtain an image like a piano-roll. We refer to this image as a melody image. Note that we transposed all melodies and chords to C major or A minor, according to the original key.

We prepared two configurations of image size as shown in **Table 1**. The image width is determined so that an image contains four measures. In both configurations, we assume that the rhythm of a melody as four-four time. If we express a triplet of an eighth or shorter note, width of a quater note should be a multiple of 6 so that both a half of the note and a one-third of the note can be expressed. In configuration A, the width of a quarter note is 24

Table 1 Configurations of image width and height.

Conf.	Width of a quarter note	Width of a measure	Image width	Image height
A	24	96	384	60
B	8	32	128	32

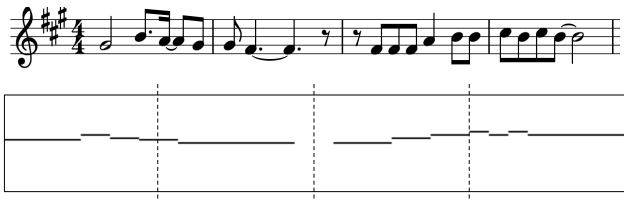


Fig. 2 An example of melody images. Above: original melody, below: image representation of the melody.

so that it can represent a triplet. For example, the width of a 32nd note is 3, and that of a triplet of eighth note is 8. In configuration B, we do not treat triplet and represent the width so that a 32nd note becomes one. The image height of configuration A is determined so that an image can represent a note from C2 to B6, which covers the height of most of the data. The image height of configuration B is 32, assuming that the too-high or too-low notes are excluded since the frequencies of those notes are small. **Figure 2** shows an example of a melody image of configuration A.

The reason why we employed two configurations is to investigate the balance between the power of expression of the representation and the training data. From the expressive power point of view, configuration A has more expressive than configuration B. However, considering that we have limited amount of training data, the less expressive representation could give the better result. Thus we compare the results of the two configurations to determine which is better for the current experimental condition.

4.2 Chord

We considered that the chord progression is useful for the completion network. In our problem, we have an original melody and its chord. If we can use the chord of the melody to be generated, we can expect that the system complements the melody so that the generated melody matches the given chord. The chord information is also used in several automatic composing methods [30]. Chords were expressed as images (chord images) of the same size as melody images. We can input a melody image and a chord image together into the network by just stacking them as different channels.

We converted chords into images in basically the same way as melody image. Each pixel value of a chord image is set if the corresponding pitch is a constituent of the chord at the corresponding time. **Figure 3** illustrates how to express C major in this way.

4.3 Dataset

We used MIDI files of 288 Japanese popular songs from the web store [*1] to build a dataset. Those files have both monophonic melody and chord information. **Table 2** shows the frequency of each kind of songs in the database. In addition, **Table 3** shows the

[*1] https://www3.ssw.co.jp/dl_data

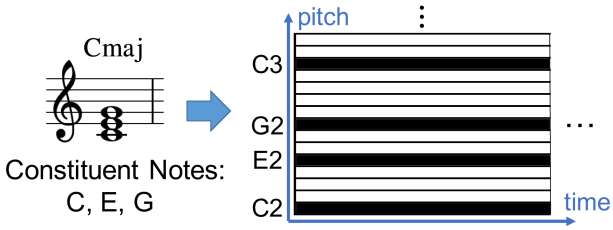


Fig. 3 Representation of C major chord.

Table 2 Type of songs in the database.

Type	frequency
J-POP/K-POP	214
70's/80's (*Kayōkyoku*)	11
Enka	6
Anime songs	57

Table 3 List of artists in the database that appear three or more times.

Name	Freq.	Name	Freq.
GACKT	10	SID	4
Hamasaki Ayumi	6	Yuzu	4
AKB48	5	Remioromen	4
Perfume	5	Nakagawa Shoko	4
Ikimono-gakari	5	BREAKERZ	3
Kobukuro	5	GIRL NEXT DOOR	3
EXILE	4	Fukuyama Masaharu	3
T.M.Revolution	4	Arashi	3
Jero	4		

Table 4 Data size.

Conf.	Training	Test
A	20,041	3,351
B	16,240	2,816

frequently-appear artists (three or more times) in the database.

We cut melodies out from the pieces by shifting a four-measure window measure by measure. For simplicity, we skipped sections that include any whole rests. We obtained 23,392 melody images and the same number of corresponding chord images. The images were divided so that melody images obtained from one song are included in either the training set or the test set, avoiding both sets share melodies of the same song. **Table 4** shows the number of samples. The reason why the number of samples is smaller in configuration B is that we needed to exclude the data that includes triplets and any too-high or too-low notes.

5. Melody Completion Network

In this section, we describe the architecture of the melody completion network and how we trained it.

5.1 Architecture

We used the same structure as the networks in Ref. [12], as illustrated in **Fig. 4**. The completion network takes a melody image, a chord image corresponding to the melody, and a mask image that indicates the completion target region of the melody with binary values (1 for a pixel to be completed). The pixels of the completion target region were substituted to the mean pixel value of the training set. The completion network consists of standard convolutional layers, dilated convolutional layers, and deconvolutional layers. Dilated convolution uses filters that are spread out and can compute larger input areas [32]. Deconvolution layers, which are convolutional layers with fractional strides, increase

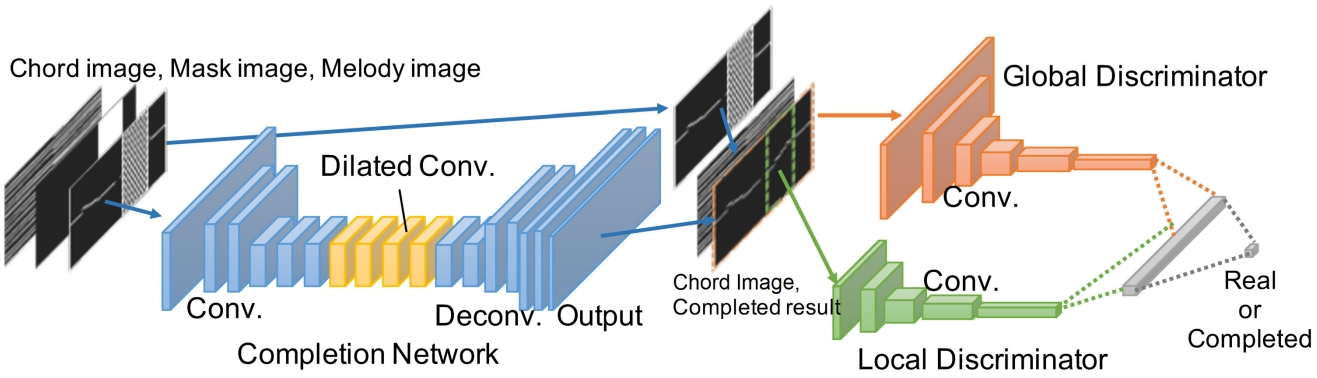


Fig. 4 Illustration of the networks.

the resolution of feature maps while normal stridden convolutions decrease the resolution [18]. A Rectified Linear Unit (ReLU) layer and a batch normalization layer [13] are inserted after each convolutional layer except the last one. A sigmoid function is employed for the output layer as the activation function. The network outputs a completed melody image. In this image, the region outside of the completion target region is replaced with the corresponding region in the input melody image.

The discriminator consists of a global context discriminator and a local context discriminator, and both consist of convolutional layers and fully-connected layers. The global discriminator takes the completed melody image or the original image with the corresponding chord image as an input. The local discriminator takes only the completion target region of the melody image with the same region of the chord image. In the case of the image is not a completed image, a random patch of the same size from the image is input. The outputs of these two discriminators are concatenated into a 2048-dimensional vector. The vector is input to a fully-connected layer with a sigmoid function to output a value which represents the probability that the input melody image is a real sample.

We set the number of output channels, size of filters, and strides as **Table 5** after the description in Iizuka's method [12].

5.2 Training

The completion network is trained in the framework of GAN [9]. As well as Ref. [12], we jointly used two loss functions. The first one is a Mean Squared Error (MSE) loss that is weighted by a mask image. Let $C(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{m_c})$ denote the output of the completion network with the input melody image $\boldsymbol{x}$, the input chord image $\boldsymbol{y}$, and the mask image $\boldsymbol{m_c}$. The MSE loss is defined as follows.

$$L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{m_c}) = \|\boldsymbol{m_c} \odot (C(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{m_c}) - \boldsymbol{x})\|^2 \quad (1)$$

Here, $\odot$ is the pixel-wise multiplication, and $\|\cdot\|$ is the Euclidian norm.

The other is the GAN loss. Let $D(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{m_d})$ denote the output of the discriminator, and $\boldsymbol{m_d}$ is a mask image which indicates the random region of the same size as the completion target region. The GAN loss is:

$$\log D(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{m_d}) + \alpha \log (1 - D(C(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{m_c}), \boldsymbol{y}, \boldsymbol{m_c})) \quad (2)$$

Table 5 The architecture of the networks. FC: fully-connected layer.

Completion Network

Type	Filter	Dilation	Stride	Outputs
conv.	5×5	1	1×1	64
conv.	3×3	1	2×2	128
conv.	3×3	1	1×1	128
conv.	3×3	1	2×2	256
conv.	3×3	1	1×1	256
conv.	3×3	1	1×1	256
dilated	3×3	2	1×1	256
dilated	3×3	4	1×1	256
dilated	3×3	8	1×1	256
dilated	3×3	16	1×1	256
conv.	3×3	1	1×1	256
conv.	3×3	1	1×1	256
deconv.	4×4	1	$1/2 \times 1/2$	128
conv.	3×3	1	1×1	128
deconv.	3×3	1	$1/2 \times 1/2$	64
conv.	3×3	1	1×1	32
output	3×3	1	1×1	1

Global context discriminator

Type	Filter	Stride	Outputs
conv.	5×5	2×2	64
conv.	5×5	2×2	128
conv.	5×5	2×2	128
conv.	5×5	2×2	256
conv.	5×5	2×2	512
conv.	5×5	2×2	512
conv.	5×5	2×2	512
FC	–	–	1024

Local context discriminator

Type	Filter	Stride	Outputs
conv.	5×5	2×2	64
conv.	5×5	2×2	128
conv.	5×5	2×2	128
conv.	5×5	2×2	256
conv.	5×5	2×2	512
conv.	5×5	2×2	512
FC	–	–	1024

Here, α is a hyperparameter that balances between the discriminator output of the natural melodies and the completed melodies. The larger value of this loss means that the discriminator more correctly distinguishes melody images.

The loss function which is the combination of those two loss functions becomes:

$$L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{m_c}) + \log D(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{m_d}) + \alpha \log (1 - D(C(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{m_c}), \boldsymbol{y}, \boldsymbol{m_c})) \quad (3)$$

The completion network and the discriminator is optimized to minimize and to maximize this, respectively. We use the

ADADELTA [33] optimizer.

For the sake of stability, the training is performed in three phases [12]. Firstly, the completion network is trained with only the MSE for T_C epochs. Next, the completion network is fixed, and the discriminator is trained for T_D epochs. Finally, both networks are trained jointly for T epochs.

Also, we made use of One-sided label smoothing [7], [27] which is expected to regularize networks. It sets the label for real image to $1-\beta$ ($0 < \beta \ll 1$) instead of 1. By applying this, Eq. (2) becomes as follows.

$$(1-\beta)\log D(\boldsymbol{x},\boldsymbol{y},\boldsymbol{m_d}) + \beta\log\left(1 - D\left(\boldsymbol{x},\boldsymbol{y},\boldsymbol{m_d}\right)\right) \qquad (4)$$
$$+ \alpha\log\left(1 - D\left(C\left(\boldsymbol{x},\boldsymbol{y},\boldsymbol{m_c}\right),\boldsymbol{y},\boldsymbol{m_c}\right)\right)$$

6. Experiment

6.1 Experimental Condition

We trained the networks as follows. A completion target region was randomly set to the region corresponding to the entire second measure or the entire third measure of the input melody. We employed mini-batch training. **Table 6** describes mini-batch size, numbers of epochs, weighing hyperparameter α and β in Eq. (4). To evaluate the effect of using chord images, we did not train the networks only with chord images but also without chord images.

6.2 Results of Configuration A

Figure 5 (a) shows two examples of the completion result with closed inputs in each condition for configuration A. MSE and GAN in Fig. 5 (a) stand for the completion results of the completion network which is trained only with MSE for T_C epochs and the result of the completion network after the whole training, respectively.

The MSE results are very similar to the input images regardless of the use of chord images. On the other hand, the GAN results with chord are also similar to the input images, but there was noise over the completion target regions. Without using chord images, the GAN did not output any notes for most of the inputs. According to these results, it is considered that the training with GAN loss did not work adequately.

Figure 5 (b) shows two examples of the completion result with open inputs in each condition. With inputs from the test set, the network outputs roughly shorter lines as the completion results in most conditions while the networks could complete images correctly with inputs from the training set. The almost correct output for the closed input suggests the over-fitting of the network. As for the use of chord image, the MSE results gave similar results with and without chord image, while the GAN without chord image did not output any clear notes. These results suggest that use of chord images enhances the completion results.

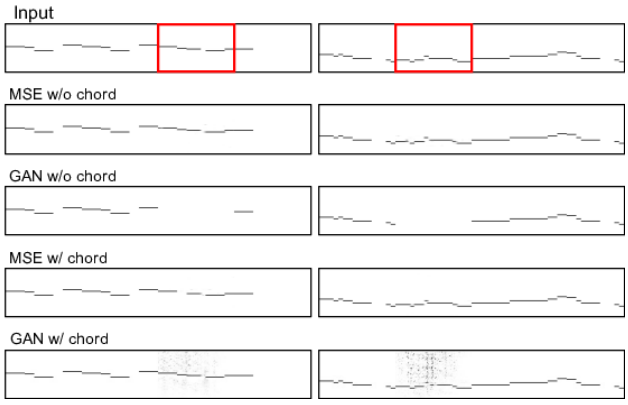
6.3 Results of Configuration B

Next, we observe the results of configuration B. In this experiment, chord images were always used, because the result of configuration A for inputs from the test set showed that use of chord images enhanced (or, at least, did not harm) the compensation results. Besides, we examined the effect of inserting the dropout layers after all the convolutional layers. Since the original network [12] did not use dropout, we employed dropout probability of 0.5 after the implementation of pix2pix [14].

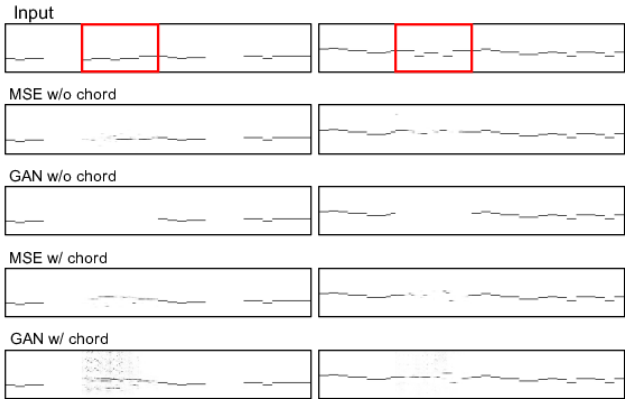
Figure 6 shows the examples obtained from the network with the MSE loss. Compared with Fig. 5, basic features are almost similar. Melodies of the closed data are almost completely reconstructed, and that of the open data is not properly generated. When we input open data, the network generated an unclear pattern. Use of the dropout did not improve the result; rather, the results were more and more unclear when dropout learning was used.

Table 6 Conditions of training.

mini-batch size	20	T_C	40
α	0.0004	T_D	20
β	0.1	T	40



(a) Inputs from the training set



(b) Inputs from the test set

Fig. 5 Examples of completion result (configuration A). Black and white are inverted for visibility. Regions inside of red frames are completion targets.

Fig. 6 Examples of completion result (configuration B, MSE loss). Left side: results for a closed input, Right side: results for an open input. Top: the input images, Middle: results without dropout, Bottom: results with dropout.

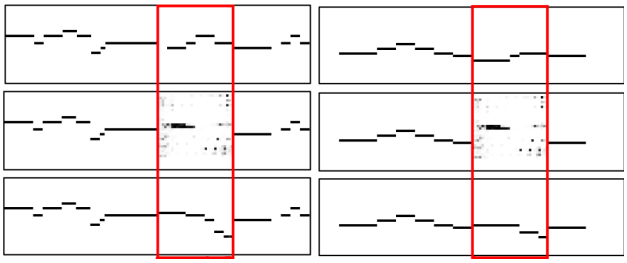


Fig. 7 Examples of completion result (configuration B, GAN loss). Left side: results for a closed input, Right side: results for an open input. Top: the input images, Middle: results without dropout, Bottom: results with dropout.

Figure 7 shows the examples obtained from the network with the GAN loss. The results without dropout were similar to the results of configuration A. The network did not generate any meaningful results regardless of closed or open data. However, when we introduced the dropout learning, the results became clear and the generated melodies were different from the original ones.

6.4 Evaluation of Originality of the Generated Melodies

From the results of the previous experiments, we found that the network could complete the masked melody under the combination of configuration B, GAN loss, and dropout. Thus, we evaluated the generated melody through two experiments.

Before the experiments, we converted an image into a MIDI format file with monophonic melody. The conversion was performed as follows. First, in each row (corresponding to the time) of the generated melody area of the completed melody image, the pixel with the maximum value was substituted with the value of 1, and all the other pixels were substituted with 0. With this substitution, the resulting image had only one pixel with the value 1 for each row and thus could be converted into a monophonic melody. We used the same rules used in converting score data to an image inversely to convert the image into notes.

It is not easy to evaluate the automatically-generated melody. Thus, we examined two different approaches. In the first approach, we objectively tested the results. Since the network was trained using the training data, the generated melodies could be similar to the melody included in the training data.

First, we defined the similarity between the two melodies. Let $A = \{a_{ij}\}$ and $B = \{b_{ij}\}$ be the matrices of melody images with the same size, where the value of a specific row and column corresponds to the value of a pixel. We assume $a_{ij}, b_{ij} \in \{0, 1\}$. Then the similarity between the two melody images is as follows.

$$S(A, B) = \frac{A \odot B}{\max\left(\sum_{i,j} a_{ij}, \sum_{i,j} b_{ij}\right)} \tag{5}$$

Here, $\odot$ is the element-wise product. This value is the fraction of coinciding notes in the two melodies, and it becomes one when the two melodies are identical.

Using this similarity, we calculated the originality of melody A with respect to the training data $\mathcal{X}$ as follows.

$$O(A, \mathcal{X}) = 1 - \max\{S(A, X_{i,m}) | X_{i,m} \in \mathcal{X}\} \tag{6}$$

where $X_{i,m}$ is the image of m-th measure in the i-th melody. In this experiment, $1 \leq i \leq 16240$ and $1 \leq m \leq 4$ because one melody image has four measures. $O(A, \mathcal{X}) = 0$ means that the melody A is included in the training data.

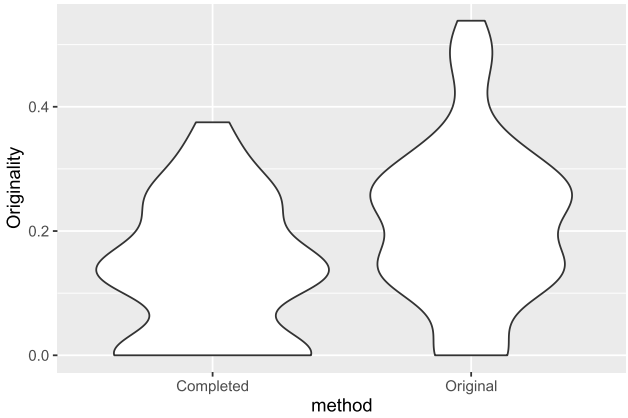


Fig. 8 Distribution of originality.

Table 7 Evaluation criteria.

Score	Evaluation
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

Figure 8 shows the distribution of originality for the original melodies and completed ones using a violin plot. We can see that frequency of the low originality of the completed melody is higher than that of the original melody, that suggests that the completion network tends to use the melody in the training data to complete the missing melody. However, there are completed melodies with higher originality, that means the network could generate new melodies that were not included in the training data.

6.5 Subjective Evaluation

Next, we conducted a subjective evaluation experiment. The subject listened to the original melody and the completed melody and evaluated the naturalness in the five-scale grade.

We asked the subjects to pay attention to "whether the generated melody connects the surrounding melodies naturally." In addition, we also instructed the subject to consider "whether the generated melody is in harmony with the chord reproduced as an accompaniment" as a secondary criterion. Finally, the subject scored each of generated melodies according to the evaluation words shown in **Table 7**.

For comparison, we examined an example-based melody completion method. This method chooses a measure in the training data that appears in the same chord progression as the melody to be completed. We determined the measure used for completion as follows. The system receives the four-measure melody image and the corresponding chord image as well as the melody completion network, and the information of the measure to be interpolated. From the database, the system searches for code images that match the code image for which the chord progression of one measure to be complemented is input and list them up. Next, for each melody image corresponding to these code images, the similarity to the input melody image is calculated for the three measures other than the measure to be completed. This

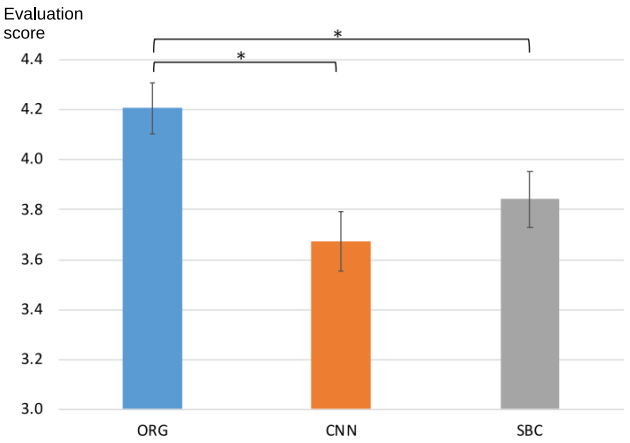


Fig. 9 Result of subjective evaluation. * denotes 5%-significant difference.

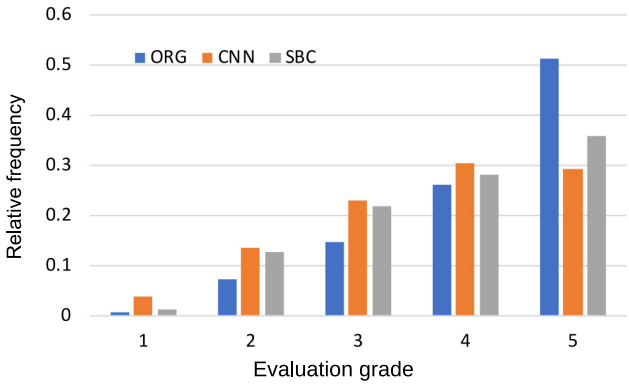


Fig. 10 Distribution of evaluation grade.

similarity is calculated by the method proposed in the previous section. Then we choose the melody image with the highest similarity, and the corresponding measure to the masked part is used for completion.

This method assures that the complement result perfectly matches to the code progression, but the result has no originality.

The sample presented to the subject was prepared as follows. First, out of 2,816 melodies included in the test data, we chose 414 samples where the four measures consisted of one phrase. Next, the chord progression patterns of the second measure in these samples were counted, then we chose 90 melodies by stratified sampling so that the chose samples had an almost uniform distribution of chord progression. The second measure of these 90 melodies was edited using the proposed method and the similarity-based complement method. Finally, a total of 270 samples were obtained by combining the original melodies and the generated melodies.

For these samples, we created MIDI data in which accompaniment corresponding to the melody are played simultaneously, and synthesized waveform data using Musescore2. In order to allow the subject to pay attention to the melody itself, we used a square wave as the tone of the melody and the piano as the accompaniment. The tempo of all melodies was 120 BPM.

We randomly presented the samples to the subjects so that a subject did not know how a melody was prepared. We made a group of subjects with three subjects. A total of 90 samples were presented to one subject in a group, and evaluation was made sample by sample. All of 270 samples were presented to three subjects in a group so that one sample was presented to only one subject in a group. On the evaluation, we disclosed to the subjects that the second measure may be completed.

We employed twelve subjects (four groups) who were students in their 20s (two males and ten females). Seven of them had music experience, of whom 3 had experience in writing and arranging.

Figure 9 shows the result of subjective evaluation. ORG, CNN and SBC are the original, completion by the proposed method, and completion by similarity, respectively. Welch's t-test with Bonferroni correction was performed on the results at a significance level of 5%. As a result, we observed a significant differences between ORG and CNN and between ORG and SBC. We did not find significant difference between CNN and SBC. Although the melody by the proposed method is somewhat unnatural compared to the original melody, the average of the evaluation values is between "normal" and "good." This result suggests that the proposed method gave a reasonably good result.

Figure 10 shows the ratio of each evaluation value to the total number of evaluations for each method. Looking at the CNN results, the ratio of the evaluation values 1 and 4 is larger than that of ORG and SBC, and the ratio of 5 is smaller.

7. Discussion

7.1 Analysis of Unnatural Completion Results

In this section, we examine the results for making further discussion. **Figure 11** shows examples of the well-completed results. The top result is an example of completing the third measure. The completed melody sounds natural to the chord in the measure and the surrounding melody. The bottom result is an example of completing the second measure. The first three notes are the same as the original ones, and the following notes have a similar melody with a different rhythm.

Next, let us look at examples with low evaluation scores. There are several reasons why the completion result sounds unnatural, such as:

(1) Extremely short notes and short rests appear,
(2) Dissonant notes for the chord are used, and
(3) Rhythm is different from that of the surrounding melody.

Figure 12 is an example of (1) and (2). The circled part of the score is an example of the problem (1). This result happens because the completion result is not a clear line but a blurred pattern. Since we employed an algorithm explained in section 6.4, a note in a specific time is determined considering only the pixel value of that time. Thus, when the generation result is not a clear line, the resulting melody fluctuates. This problem could be improved by improving the algorithm to convert the image into notes so that the temporal naturalness is considered.

Moreover, in the example of Fig. 12, notes of F appear in the first and third beats. However, the chord of this measure (the second measure) is D7 whose components are D, F#, A, and C. Thus, the F notes are dissonant to this chord. This result could happen because the frequency of this chord is low. This problem could be treated by increasing the amount and variation of the training

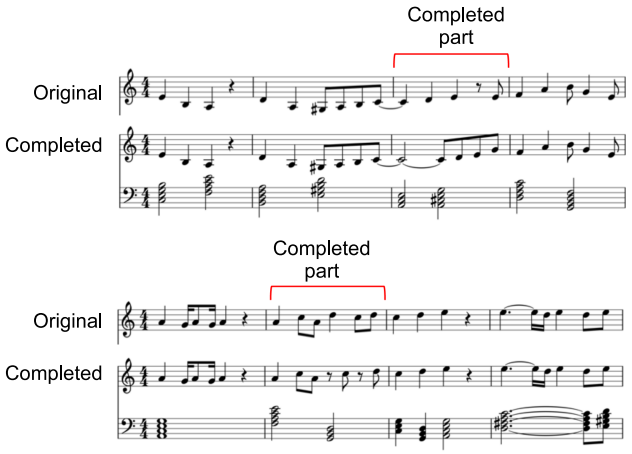


Fig. 11 Examples of good completion results.

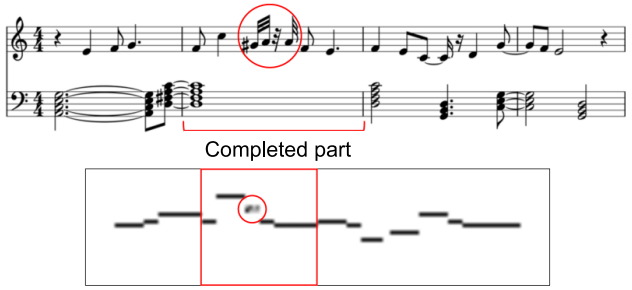


Fig. 12 Examples of unnatural completion result (fragmented and dissonant notes).

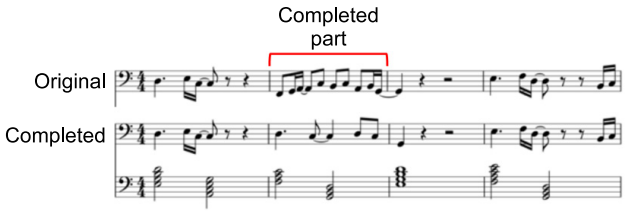


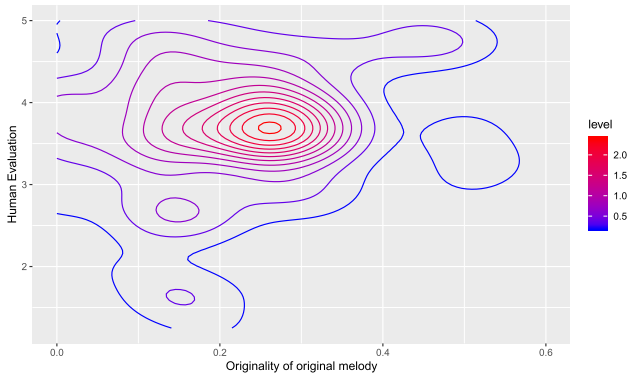
Fig. 13 Examples of unnatural completion result (unnatural rhythm).

data.

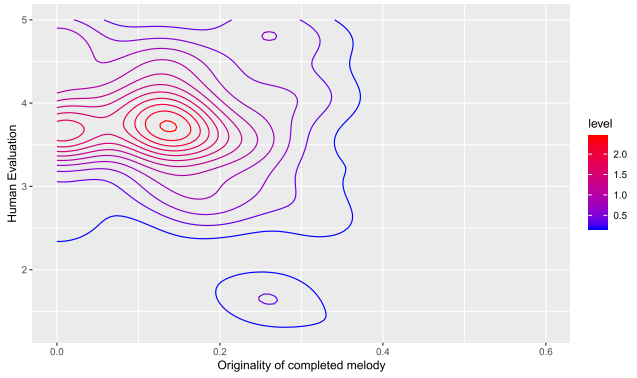
We observed only a few examples of the problem (3). These results happen when the length of the notes in the completed measure and that of the surrounding measures are different. **Figure 13** shows such an example. In this example, 16th notes frequently appear in the 1st and 4th measure, while no 16th note appears in the completed measure (the 2nd measure). As a result, the completed measure and other measures have a different rhythm, and the result sounds unnatural. This could be improved to consider the average length of a note when converting the generated image to notes. Currently, we regard a continuous line in the image as one note. However, considering the rhythm pattern, we can split the line into multiple notes so that it fits the predicted rhythm.

7.2 Originality and Human Evaluation Score

As shown in Fig. 8, completed melodies have lower originality than the original melody. On the other hand, as shown in Fig. 10, the completed melodies have fair evaluation scores. Here, we investigate relationship between the originality and human evaluation score. If the originality of a melody is low, that means the completed melody exists in the training data, thus it may sound more natural.



(a) Originality and human evaluation score of the original melodies.



(b) Originality and human evaluation score of the completed melodies.

Fig. 14 Smoothed 2-D distribution of originality and human evaluation score.

Table 8 Statistical test of correlation between originality and human evaluation.

Method	Correlation coefficient	p-value
Original	0.11	0.298
Completed	−0.21	0.0502

Figure 14 shows the 2-D smoothed distribution contour of originality and human evaluation score for the original melody and the completed melody. Figure 14 (a) is the distribution of the original melodies, and Fig. 14 (b) is that of the completed melodies. It is clear that the originality of the original melodies is higher than that of the completed melody. We see a cluster of higher originality (around 0.25) and lower evaluation score (around 1.7) in Fig. 14 (b), which may correspond to the bad completion results explained in the previous section. However, most of the completed melodies have a distribution where originality and evaluation score look independent.

Table 8 shows the Pearson's correlation coefficients between originality and human evaluation score for the original and completed melodies. The correlation coefficients of the original and completed melodies seems to have positive and negative correlations respectively. However, according to t-test of correlation coefficients, we could not find any statsitically significant correlation at 5% level for either melodies. It does not necessarily imply that the originality and human evaluation is independent, but we can say the relationship between those variables would be weak even if it existed.

7.3 Amount of Training Data and Melody Image Configuration

Next, we discuss the amount of training data and the completion result. We conducted experiments for configurations A and B, and obtained better result for configuration B. That result seems to be related to the amount of training data. The original image inpainting network was trained using eight million images, while we used only 16 thousands to 20 thousands of melody images. With more training data, we expect that we can train better model for larger melody image such as configuration A.

7.4 Melody Completion and Image Inpainting

Finally, we discuss the difference between our task and image inpainting to which Iizuka's original network was applied. There are several differences between natural images and melody images. The pixel values of melody images are almost zero, and chord image is superimposed into the melody image. Besides the difference of input images, our network has several differences from the original one. First, size and position of mask images are very limited compared with the image inpainting task. Second, we introduced the label smoothing and dropout that were not used in the original network.

Although there are several differences, the basic concept of our network is same as the one for image inpainting. The success of the network for melody completion seems to show the universal ability of the completion network using CNN and GAN.

8. Conclusion

We proposed a melody completion network which completes melody by applying an image completion network which is based on CNN and is trained in the framework of GAN. From the results, it was confirmed that the network could complete reasonably the melody images under a certain condition. Through the evaluation experiments, we confirmed that the network could generate original melodies that were not included in the training data, and the quality of the completed melody is not significantly different from the result of the example-based method.

As described in the discussion, we still have several problems, especially the algorithm to convert the resulting image into actual notes. This problem should be solved in the future.

References

[1] Biles, J.A.: GenJam: A genetic algorithm for generating jazz solos, *Proc. Int. Computer Music Conf.*, pp.131–137 (1994).

[2] Colombo, F., Seeholzer, A. and Gerstner, W.: Deep artificial composer: A creative neural network model for automated melody generation, *International Conference on Evolutionary and Biologically Inspired Music and Art*, pp.81–96, Springer (2017).

[3] de Freitas, A.R., Guimaraes, F.G. and Barbosa, R.V.: Ideas in automatic evaluation methods for melodies in algorithmic composition, *Sound and Music Computing Conference* (2012).

[4] Dong, H.-W., Hsiao, W.-Y., Yang, L.-C. and Yang, Y.-H.: MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment, *32nd AAAI Conference on Artificial Intelligence* (2018).

[5] Fuegi, J. and Francis, J.: Lovelace & Babbage and the creation of the 1843'notes', *IEEE Annals of the History of Computing*, Vol.25, No.4, pp.16–26 (2003).

[6] Fukayama, S., Nakatsuma, K., Sako, S., Yonebayashi, Y., Kim, T.H., Qin, S.W., Nakano, T., Nishimoto, T. and Sagayama, S.: Orpheus: Automatic Composition System Considering Prosody of Japanese Lyrics, *Proc. Int. Conf. Entertainment Computing* (*ICEC 2009*), Natkin, S. and Dupire, J. (Eds.), pp.309–310, Springer (2009).

[7] Goodfellow, I.: NIPS 2016 Tutorial: Generative Adversarial Networks (2017).

[8] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y.: Generative adversarial nets, *Advances in Neural Information Processing Systems*, pp.2672–2680 (2014).

[9] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y.: Generative Adversarial Nets, *Advances in Neural Information Processing Systems 27*, pp.2672–2680, Curran Associates, Inc. (2014).

[10] Hadjeres, G., Pachet, F. and Nielsen, F.: DeepBach: A Steerable Model for Bach Chorales Generation, *Proc. 34th International Conference on Machine Learning*, *ICML'17*, JMLR.org, pp.1362–1371 (2017) (online), available from ⟨http://dl.acm.org/citation.cfm?id=3305381.3305522⟩.

[11] Huang, C.-Z.A., Cooijmans, T., Roberts, A., Courville, A.C. and Eck, D.: Counterpoint by Convolution, *Proc. 18th International Society for Music Information Retrieval Conference*, pp.211–218 (2017).

[12] Iizuka, S., Simo-Serra, E. and Ishikawa, H.: Globally and locally consistent image completion, *ACM Trans. Graphics* (*ToG*), Vol.36, No.4, p.107 (2017).

[13] Ioffe, S. and Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, arXiv prepreint arXiv:1502.03167 (2015).

[14] Isola, P., Zhu, J.-Y., Zhou, T. and Efros, A.A.: Image-to-image translation with conditional adversarial networks, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp.1125–1134 (2017).

[15] Koh, E.S., Dubnov, S. and Wright, D.: Rethinking Recurrent Latent Variable Model for Music Composition, *2018 IEEE 20th International Workshop on Multimedia Signal Processing* (*MMSP*), pp.1–6, IEEE (2018).

[16] Le Meur, O., Gautier, J. and Guillemot, C.: Examplar-based inpainting based on local geometry, *Proc. 18th IEEE Int. Conf. Image Processing*, pp.3401–3404, IEEE (2011).

[17] Liang, F.T., Gotham, M., Johnson, M. and Shotton, J.: Automatic Stylistic Composition of Bach Chorales with Deep LSTM, *ISMIR*, pp.449–456 (2017).

[18] Long, J., Shelhamer, E. and Darrell, T.: Fully Convolutional Networks for Semantic Segmentation, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp.3431–3440 (2015).

[19] Nakamura, K., Nose, T., Chiba, Y. and Ito, A.: Melody Completion Based on Convolutional Neural Networks and Generative Adversarial Learning, *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp.116–123, Springer (2018).

[20] Nierhaus, G.: *Algorithmic composition: paradigms of automated music generation*, Springer Science & Business Media (2009).

[21] Odena, A., Olah, C. and Shlens, J.: Conditional image synthesis with auxiliary classifier gans, *Proc. 34th International Conference on Machine Learning-Volume 70*, pp.2642–2651 (2017).

[22] Oncu, A.I., Deger, F. and Hardeberg, J.Y.: Evaluation of Digital Inpainting Quality in the Context of Artwork Restoration, *Computer Vision – ECCV 2012*, *Workshops and Demonstrations*, Fusiello, A., Murino, V. and Cucchiara, R. (Eds.), pp.561–570, Springer Berlin Heidelberg (2012).

[23] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T. and Efros, A.A.: Context encoders: Feature learning by inpainting, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp.2536–2544 (2016).

[24] Pushpalwar, R.T. and Bhandari, S.H.: Image inpainting approaches-a review, *IEEE 6th Int. Conf. Advanced Computing* (*IACC*), pp.340–345, IEEE (2016).

[25] Roig, C., Tardón, L.J., Barbancho, I. and Barbancho, A.M.: Automatic melody composition based on a probabilistic model of music style and harmonic rules, *Knowledge-Based Systems*, Vol.71, pp.419–434 (2014).

[26] Ružić, T. and Pižurica, A.: Context-aware patch-based image inpainting using Markov random field modeling, *IEEE Trans. Image Processing*, Vol.24, No.1, pp.444–456 (2014).

[27] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X. and Chen, X.: Improved Techniques for Training GANs, *Advances in Neural Information Processing Systems 29*, pp.2234–2242, Curran Associates, Inc. (2016).

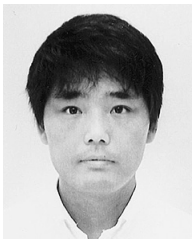
[28] Supper, M.: A few remarks on algorithmic composition, *Computer Music Journal*, Vol.25, No.1, pp.48–53 (2001).

[29] Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O. and Li, H.: High-resolution image inpainting using multi-scale neural patch synthesis, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp.6721–6729 (2017).

[30] Yang, L.-C., Chou, S.-Y. and Yang, Y.-H.: MidiNet: A Convolutional Generative Adversarial Network for Symbolic-domain Music Gener-

ation, *18th Int. Society for Music Information Retrieval Conference* (2017).
[31] Yeh, R.A., Chen, C., Lim, T.Y., Schwing, A.G., Hasegawa-Johnson, M. and Do, M.N.: Semantic image inpainting with deep generative models, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp.5485–5493 (2017).
[32] Yu, F. and Koltun, V.: Multi-Scale Context Aggregation by Dilated Convolutions, arXiv preprint arXiv:1511.07122 (2015).
[33] Zeiler, M.D.: ADADELTA: An Adaptive Learning Rate Method, arXiv preprint arXiv:1212.5701 (2012).

Kosuke Nakamura received his B.E. and M.E. degrees from School of Engineering, Tohoku University, Sendai, Japan in 2017 and 2019, respectively. During that period he researched the music information processing. His research interests include the deep learning and knowledge engineering.



Takashi Nose received his B.E. degree in electronic information processing, from Kyoto Institute of Technology, Kyoto, Japan, in 2001. He received his Dr.Eng. degree in information processing from Tokyo Institute of Technology, Tokyo, Japan, in 2009. He was a Ph.D. researcher of The 21st Century Center of Excellence (COE) program and Global COE program in 2006 and 2007, respectively. He was an Intern Researcher at ATR spoken language communication Research Laboratories (ATR-SLC) from July 2008 to January 2009. He became an Assistant Professor of the Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology in 2009. He became a Lecturer of the Graduate School of Engineering, Tohoku University, Sendai, Japan in 2013. He is currently a Associate Professor of the Graduate School of Engineering, Tohoku University. His research interests include speech synthesis, speech recognition, spoken dialogue system, and music information processing. Prof. Nose is a member of IEEE, ISCA, IEICE, IPSJ, and ASJ.

Yuya Chiba received B.E., M.E. and Ph.D. degrees in engineering from School of Engineering, Tohoku University, Sendai, Japan in 2010, 2012, and 2015, respectively. He is currently an Assistant Professor at the Graduate School of Engineering, Tohoku University, Sendai, Japan. His research interests include spoken dialogue systems, multimodal dialogue systems, and human-centric interfaces. Prof. Chiba received the IEICE ISS Young Researcher's Award in Speech Field in 2014. He is a member of ISCA, ACL, IEICE, and ASJ.

Akinori Ito was born in Yamagata, Japan in 1963. He received his B.E., M.E. and Ph.D. degrees from Tohoku University, Sendai, Japan, in 1984, 1986 and 1992 respectively. He joined the Research Center for Applied Information Sciences, Tohoku University as an Assistant Professor in 1992, and then moved to Education Center for Information Processing, Tohoku University in 1993. In 1995, he joined Faculty of Engineering, Yamagata University, Japan as a Lecturer. He is now a Professor of Graduate School of Engineering, Tohoku University, Sendai, Japan. He has engaged in spoken language processing, speech signal processing and music information processing. Prof. Ito is currently the president of the Acoustical Society of Japan (ASJ), fellow of the Institute of Electronics, Information and Communication Engineers (IEICE), and a member of the Information Processing Society Japan, Human Interface Society and the IEEE.