GAN-Enhanced InpaintNet for Music Inpainting on Limited Data

Komei Naemura*, Boyu Cao†, Ryotaro Nagase*, Ryoichi Takashima*, and Yoichi Yamashita*
* Ritsumeikan University, Japan
† South China University of Technology, China
E-mail: knaemura@spl.ise.ritsumei.ac.jp

***Abstract*—Music inpainting is the task of generating incomplete parts of a musical piece. Many deep neural network-based methods for music inpainting have been proposed; however, inpainting music with limited training data, such as works by deceased composers or from niche genres, remains a challenging task. A previous study reported that InpaintNet demonstrated relatively high performance in comparative evaluations of inpainting models under small-data conditions. However, even this model often produces unnatural results when trained on very limited data. To address this issue, we propose an inpainting method that integrates a generative adversarial network (GAN) into InpaintNet. In the proposed method, the GAN discriminator evaluates the entire musical piece by combining the inpainted section with the preceding and following segments, thereby encouraging more musically consistent outputs. Experiments using two different datasets demonstrated that the proposed method improves the performance of InpaintNet.**

I. Introduction

Music inpainting is the task of generating missing parts of a musical piece. This technique has been applied to music editing and the restoration of old scores. There are two main types of music inpainting approaches. One approach uses the symbolic representation of music [1]–[8], which encode musical elements, such as notes, harmony, and rhythm, using discrete symbols or numerical values. The musical instrument digital interface (MIDI) is one of the well-known symbolic representations. The other approach utilizes spectrograms derived from audio signals [9]–[12]. Both symbolic and spectrogram-based approaches have been studied in the context of monophonic and polyphonic music. In this study, we focus on the symbolic music inpainting for monophonic music.

Previous studies have proposed various deep neural network (DNN)-based methods for music inpainting. For example, Hadjeres et al. proposed a method that generates pitch sequences conditioned on structural musical metadata, such as fermatas and beat information, by combining a recurrent neural network (RNN) with pseudo-Gibbs sampling [1]. Huang et al. also proposed an inpainting method based on counterpoint, which simultaneously combines multiple independent melodies [2]. In addition, Hadjeres et al. proposed an RNN-based method that enables control over pitch and note duration in the inpainted segment [3], which was difficult in the previous method [1]. This method combines Constraint-RNN, which processes pitch and note duration conditions, with Token-RNN, which generates the inpainted segment.

Pati et al. proposed InpaintNet, which combines a variational autoencoder (VAE) [13] with a gated recurrent unit (GRU) [14] to inpaint a missing segment using representations obtained from the input, such as MIDI, using the VAE [4]. InpaintNet improved performance in terms of negative log-likelihood by up to 55% compared to the previous method [3], while achieving comparable results in the subjective evaluation. Chen et al. also proposed a model called Music SketchNet [5], which extracts latent representations of pitch and rhythm obtained from MIDI data and uses them to inpaint missing segments. This method improved the performance of generating pitch and rhythm compared to InpaintNet.

Recently, Transformer-based models [15] have been introduced. Chang et al. proposed a music inpainting method based on XLNet [6], which enables inpainting of variable-length segments. Furthermore, Guo et al. proposed a method that generates inpainted segment conditioned on not only pitch, but also note duration, pitch range per measure, and the number of notes. As demonstrated in previous studies, various methods have been proposed for music inpainting.

While increasing the number of parameters in neural networks can improves the performance of music inpainting, it also requires more training data. In data-driven music inpainting methods, the input music to be inpainted should ideally have a genre and style similar to those of the training data. However, the availability of music data is often limited for certain genres or styles, such as works by deceased composers or pieces from niche genres. A previous study reported that InpaintNet demonstrated relatively high performance in comparative evaluations of inpainting models under small-data conditions [16]. Nevertheless, even this model often produces unnatural results, such as repetitive generation of the same pitch, when trained on very limited data. To address this issue, we propose an inpainting method that integrates a generative adversarial network (GAN) into InpaintNet. In the proposed method, the GAN discriminator evaluates the entire musical piece by combining the inpainted section with the preceding and following segments, thereby encouraging more musically consistent outputs. We evaluate the effectiveness of the proposed method by comparing it with the conventional InpaintNet through experiments using two different datasets.

The remainder of this paper is organized as follows. In Section II describes InpaintNet, the GAN framework, and the proposed method. In Section III presents the experimental

979-8-3315-7206-8/25/$31.00 ©2025 IEEE

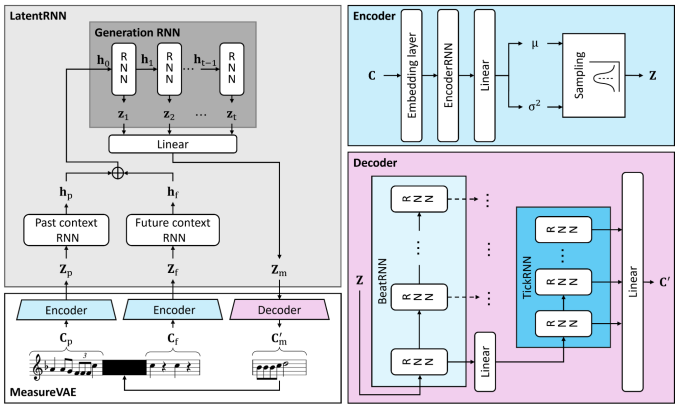


Fig. 1: InpaintNet

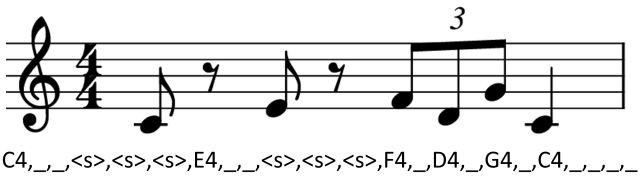


Fig. 2: Music information obtained from MIDI data

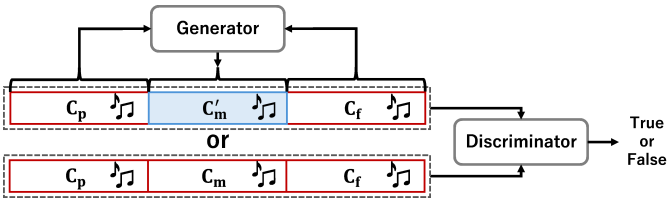


Fig. 3: GAN for music inpainting

setup and results. Finally, in Section IV concludes the paper and discusses future work.

II. Methodology

A. InpaintNet

In this study, music inpainting refers to the task of completing the missing segment of input music in which a certain section is absent. In other words, the inpainting model generates the missing segment based on the musical information from the sections preceding and following the missing part.

InpaintNet is composed of MeasureVAE, which reconstructs symbolic music, and LatentRNN, which generates latent representations of the inpainted segment. Fig. 1 shows the outline of InpaintNet.

MeasureVAE consists of an encoder and a decoder. The inputs of an encoder are symbolic music features obtained from MIDI data, as shown in Fig. 2. Symbolic music features are defined as discrete sequences per measure, in which each measure is converted into the smallest time unit, referred to as "ticks". Each tick is a 130-dimensional one-hot vector: 128 for note pitches (e.g., F1, E4), one for continuation ('_'), and one for rest (<s>). In this study, following the previous study [4], we represent each measure using 24 ticks. The encoder estimates the mean μ and variance σ^2 from the input data, and extracts latent representations $\mathbf{Z}$ by sampling a Gaussian distribution defined by these parameters. The decoder reconstructs the symbolic music features from the latent representations using BeatRNN and TickRNN. BeatRNN generates hidden states for each neat in a measure, while TickRNN generates those for each tick within a beat. Eventually, we input the TickRNN output into the fully connected layer, and obtain the symbolic music features $\mathbf{C}'$.

LatentRNN generates a latent representation of the inpainted segment $\mathbf{Z}_\mathrm{m}$, conditioned on the latent representations of both the preceding and following melodies. First, we feed the 130-dimensional one-hot vectors of the preceding and following melodies, $\mathbf{C}_\mathrm{p}$ and $\mathbf{C}_\mathrm{f}$, into the encoder to obtain their latent representations, $\mathbf{Z}_\mathrm{p}$ and $\mathbf{Z}_\mathrm{f}$. These are fed into the Past-Context-RNN and Future-Context-RNN, and the final hidden states $\mathbf{h}_\mathrm{p}$ and $\mathbf{h}_\mathrm{f}$ are extracted from them, respectively. Second, we feed the first hidden states $\mathbf{h}_0$ concatenated with $\mathbf{h}_\mathrm{p}$ and $\mathbf{h}_\mathrm{f}$ into the generation RNN. This model autoregressively generates the latent representations of the inpainted segment. Finally, we feed these representations into a fully connected layer, and obtain the latent representations $\mathbf{Z}_\mathrm{m}$. In the training procedure, MeasureVAE is first trained. Then, using the trained MeasureVAE to update only the parameters of LatentRNN.

B. GAN

GAN is a framework in which a generator and a discriminator are trained alternately. This approach enables natural data generation by learning without assuming a predefined distribution of the generated data. GANs have been widely used in various tasks such as image generation and speech synthesis, and have also been applied in previous studies on music inpainting. For example, Andres et al. [9] incorporated a GAN into a music inpainting model to capture musical complexity and length of missing segments. Pirmin et al. [11] introduced Wasserstein GAN to stablilize GAN training for audio inpainting. The main difference between these previous studies and the present work is that this study focuses on scenarios with limited training data and demonstrates the effectiveness of integrating a GAN into InpaintNet under such conditions.

C. InpaintNet with GAN

In VAE-based generative models such as InpaintNet, the encoder samples latent variable $\mathbf{Z}$ from a probability distribution, typically a normal distribution, and the decoder reconstructs the data from the sampled latent variable. In general, VAE-based models tend to produce average outputs that reflect the overall distribution of the training data. Furthermore, since InpaintNet generates features for the missing segment in an autoregressive manner, it may produce unnatural results, such as repetitive generation of the same pitch or outputs that lack consistency with the preceding and following segments, especially when the amount of training data is limited. In this

study, we incorporate a GAN framework into InpaintNet to suppress such unnatural outputs and generate more musically consistent results.

Fig. 3 illustrates the training process of LatentRNN using the GAN framework. Here, $\mathbf{C}'_{\mathrm{m}}$ denotes the music sequence generated by the InpaintNet decoder, while $\mathbf{C}_{\mathrm{m}}$ represents the ground truth sequence. In the proposed method, the GAN discriminator is given a sequence that combines the inpainted segment with the preceding and following musical segments (either $[\mathbf{C}_{\mathrm{p}},\ \mathbf{C}'_{\mathrm{m}},\ \mathbf{C}_{\mathrm{f}}]$ or $[\mathbf{C}_{\mathrm{p}},\ \mathbf{C}_{\mathrm{m}},\ \mathbf{C}_{\mathrm{f}}]$). This allows the discriminator to evaluate not only the similarity between the inpainted output and the ground truth, but also the consistency of the output with the surrounding musical context.

$$\mathrm{V}(\mathrm{D},\mathrm{G}) = \mathbb{E}_{\mathbf{x}\sim\mathbf{P}_{\mathbf{x}}}[\log \mathrm{D}(\mathbf{x})] + \mathbb{E}_{\mathbf{x}'\sim\mathbf{P}_{\mathbf{x}'}}[\log(1-\mathrm{D}([\mathrm{G}(\mathbf{x}'),\mathbf{x}'])))]. \quad (1)$$

Here, $\mathbb{E}$ denotes the expected value, and D and G represent the discriminator and generator, respectively. Let $\mathbf{x} = [\mathbf{C}_{\mathrm{p}}, \mathbf{C}_{\mathrm{m}}, \mathbf{C}_{\mathrm{f}}]$ denotes the ground truth music sequence, and $\mathbf{x}' = [\mathbf{C}_{\mathrm{p}}, \mathbf{C}_{\mathrm{f}}]$ the sequence with the middle part missing. Then, $\mathrm{G}(\mathbf{x}') = \mathbf{C}'_{\mathrm{m}}$ denotes the inpainted result. The generator is trained to minimize Eq. (1), while the discriminator is trained to maximize it, following a min-max optimization strategy.

III. Experiments

A. Dataset

In this study, when training InpaintNet, we use relatively large training data for MeasureVAE and limited data for LatentRNN. Our preliminary experiments have confirmed that by using a pretrained model for MeasureVAE, the output results are stable even when the training data is small. For the same reason, we used a pretrained model for MeasureVAE in this experiment. In this study, we used two music datasets: the Irish folk dataset and Johann Sebastian Bach (JSB) chorales dataset.

The Irish folk dataset [17] contains 45,849 MIDI files of Irish folk music, and is separated training, validation, and testing subsets. In our experiments, we used 17,538 MIDI files of monophonic music with more than 17 measures in 4/4 time for building MeasureVAE and LatentRNN. Note that for the pretraining of MeasureVAE, 14,030 songs were used as training data and 1,753 songs as validation data. For the training of LatentRNN, 468 songs were used as training data, 175 as validation data, and 1,112 as test data. We refer to the dataset used for training LatentRNN as the "small Irish folk dataset" throughout this paper.

The JSB chorales dataset [18] contains 408 MIDI files of the chorus score composed by Johann Sebastian Bach. These scores are mixed choruses consisting of soprano, alto, tenor, and bass parts, and each part is stored independently as a separate track. In our experiments, we used 173 MIDI files longer than 16 measures to build LatentRNN. Note that for the training of LatentRNN, 46 songs were used as training data, 17 as validation data, and 110 as test data.

B. Model

The model structure of InpaintNet is that the encoder consists of an embedding layer, a two layer bidirectional gated recurrent unit (BiGRU)-based EncoderRNN, and two fully connected layers. The decoder consists of a two layer GRU-based BeatRNN and TickRNN, and a fully connected layer. The LatentRNN consists of a two layer BiGRU-based Past-Context-RNN and Future-Context-RNN, a two layer GRU-based GenerationRNN, and a fully connected layer.

For training MeasureVAE, 16 bar MIDI sequences were used as input. The number of epochs to 100, the batch size to 64 and the learning rate to 0.0001, The loss function consisted of cross-entropy loss and Kullback-Leibler (KL) divergence loss. We use the Adam optimizer and the number of repeats to 10. Note that the number of repeats indicates how much learning data is extracted from each MIDI data. For example, if the number of repeats is 10, 10 random segments are extracted from each MIDI data and used as training samples for that epoch. Each extracted sample consists of 16 measures, which are divided into 6, 4 and 6 measures, respectively and used as the preceding, middle, and following musical contexts denoted as $\mathbf{C}_{\mathrm{p}}$, $\mathbf{C}_{\mathrm{m}}$, and $\mathbf{C}_{\mathrm{f}}$. Note that the tempo is set to 120 beat per minute (BPM), and the time signature is 4/4.

For training LatentRNN, we ran the training both with and without GAN. The number of epochs to 10, the number of repeats to 1,000 and using cross-entropy loss as the loss function. The batch size, learning rate, and optimization method were the same as those used in MeasureVAE trained.

For training discriminator, we ran the training using four different methods that change the model structure and input information. In this study, we used a CNN-based discriminator composed of five convolutional layers, and a GRU-based discriminator consisting of one bidirectional GRU (BiGRU) layer and one fully connected layer. In the proposed method, the discriminator takes the musical contexts $\mathbf{C}_{\mathrm{p}}$, $\mathbf{C}_{\mathrm{f}}$, and $\mathbf{C}_{\mathrm{m}}$ as input (DR: data representation). For comparison, we also evaluated a setting where the latent variables $\mathbf{Z}_{\mathrm{p}}$, $\mathbf{Z}_{\mathrm{f}}$ and $\mathbf{Z}_{\mathrm{m}}$ were used instead (LV: latent variable). The proposed method is denoted in the format Prop-DR-CNN, where the first element indicates the proposed method, the second represents the input information to the discriminator, and the third denotes the architecture of the discriminator. None that the number of epochs, batch size, learning rate, and optimization method were the same as those used in LatentRNN. generative adversarial loss is used as the loss function.

C. Metrics

Following the privious study[16], we used the six evaluation metrics, each of which is described in detail bellow.

Position score ($\mathrm{pos}_{\mathrm{F}_1}$): This metric measures how well the pitch onset position of the prediction aligns with that of the ground truth, and is calculated as the harmonic mean $\mathrm{pos}_{\mathrm{F}_1}$ of the Recall and $\mathrm{Precision}$ for the pitch onset position, as defined in Eq. (2).

$$\mathrm{pos}_{\mathrm{F}_1} = 2 \cdot \frac{\mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \quad (2)$$

A higher $\mathrm{pos}_{\mathrm{F}_1}$ indicates that the prediction of pitch onset positions is more accurately.

Pitch accuracy (pAcc): This metric measures how well the predicted pitches matches the ground truth, given that their pitch onset position are aligned. pAcc is defined by Eq. (3).

$$\mathrm{pAcc} = \frac{|\mathrm{Y}_{\mathrm{pos}} \cap \mathrm{Y}_{\mathrm{pitch}}|}{|\mathrm{Y}_{\mathrm{pos}}|} \tag{3}$$

Note that $\mathrm{Y}_{\mathrm{pos}}$ donates the set of ticks where the pitch onset position of the prediction matches that of the ground truth, and $\mathrm{Y}_{\mathrm{pitch}}$ donates the set of ticks where the predicted pitch matches the ground truth. A higher pAcc indicates that the prediction of pitches is more accurately.

Rhythm accuracy (rAcc): This metric measures how well the predicted note duration matches the ground truth, given that their pitch onset position are aligned. rAcc is defined by Eq. (4).

$$\mathrm{rAcc} = \frac{|\mathrm{Y}_{\mathrm{pos}} \cap \mathrm{Y}_{\mathrm{duration}}|}{|\mathrm{Y}_{\mathrm{pos}}|} \tag{4}$$

$\mathrm{Y}_{\mathrm{duration}}$ is the set of ticks where the note duration of the prediction matches that of ground truth. A higher rAcc indicates that the prediction of rhythm is more accurately.

Silence density divergence ($\mathrm{S}_{\mathrm{div}}$): This metric measures how well the distribution of rests in the prediction aligns with that in the ground truth. $\mathrm{S}_{\mathrm{div}}$ is defined by Eq. (5).

$$\mathrm{S}_{\mathrm{div}} = \mathrm{JS}_{\mathrm{div}}(p_{\mathrm{true,rest}} || p_{\mathrm{pred,rest}}) \tag{5}$$

$p_{\mathrm{pred,rest}}$ and $p_{\mathrm{true,rest}}$ donate the distribution of rests in the prediction and the ground truth, respectively. Also, JS represents Jensen-Shannon (JS) divergence. A lower $\mathrm{S}_{\mathrm{div}}$ indicates that the number of rests in the prediction more closely matches that in the ground truth.

Pitch class divergence ($\mathrm{H}_{\mathrm{div}}$): This metric measures how well the onset distribution of the 12 pitch classes (C, C#, $\cdots$, A#, B) in the entire prediction matches that in the entire ground truth. First, the pitch class histogram entropy per measure over the 12 pitch classes is defined as shown in Eq. (6).

$$\mathcal{H}_m = -\sum_{i=0}^{11} p_i^{\mathrm{pch}} \log_2(p_i^{\mathrm{pch}}) \tag{6}$$

Second, the average of pitch difference between the inpainted segment and the preceding and following segments is calculated as shown in Eq. (7).

$$\mathrm{H} = \frac{1}{M_1 M_2} \sum_{m_1=0}^{M_1} \sum_{m_2=0}^{M_2} |\hat{\mathcal{H}}_{m_1} - \mathcal{H}_{m_2}| \tag{7}$$

Finally, the similarity between the entropy distributions of the entire prediction and the entire ground truth are evaluated as shown in Eq. (8).

$$\mathrm{H}_{\mathrm{div}} = \mathrm{JS}_{\mathrm{div}}(p_{\mathrm{H}_{\mathrm{true}}} || p_{\mathrm{H}_{\mathrm{pred}}}) \tag{8}$$

M_1 is the number of measures in the inpainted segment, and M_2 is the total number of measures in the preceding and following segments. p_i^{pch} donates the probability of the i-th pitch class. $\hat{\mathcal{H}}_m$ and $\mathcal{H}_m$ represent the entropy of the pitch class distribution of the m-th measure in the inpainted segment and the surrounding segments, respectively. Also, H denotes the average pitch class entropy per measure in the inpainted segment and the surrounding segments. $p_{\mathrm{H}_{\mathrm{pred}}}$ and $p_{\mathrm{H}_{\mathrm{true}}}$ represent the distributions of the average difference in pitch class entropy between the entire prediction and the ground truth. A lower $\mathrm{H}_{\mathrm{div}}$ indicates that the pitch class distribution in the prediction more closely matches that of the ground truth.

Groove similarity divergence ($\mathrm{GS}_{\mathrm{div}}$): This metric measures how well the rhythm in the entire prediction matches that in the entire ground truth. First, the average of grooving pattern similarity [19] of the onset positions in the inpainted segment and the surrounding segments is calculated as shown in Eq. (9) and Eq. (10).

$$\mathrm{GS} = \frac{1}{M_1 M_2} \sum_{m_1=0}^{M_1} \sum_{m_2=0}^{M_2} \mathcal{GS}(\hat{\mathbf{g}}_{m_1}, \mathbf{g}_{m_2}) \tag{9}$$

$$\mathcal{GS}(\hat{\mathbf{g}}, \mathbf{g}) = 1 - \frac{1}{T} \sum_{t=0}^{T-1} \mathrm{XOR}(\hat{\mathbf{g}}_t, \mathbf{g}_t) \tag{10}$$

After that, the similarity between the distributions of grooving pattern similarity for the entire prediction and the entire ground truth is evaluated as shown in Eq. (11).

$$\mathrm{GS}_{\mathrm{div}} = \mathrm{JS}_{\mathrm{div}}(p_{\mathrm{GS}_{\mathrm{true}}} || p_{\mathrm{GS}_{\mathrm{pred}}}) \tag{11}$$

T is the number of ticks per measure, $\hat{\mathbf{g}}_m$ and $\mathbf{g}$ are the pitch sequences per measure in the prediction and ground truth, respectively. $\mathrm{XOR}(\hat{\mathbf{g}}_t, \mathbf{g}_t)$ denotes an XOR function that returns zero if the presence or absence of pitch onsets match at the t-th tick, and one otherwise. $\mathcal{GS}(\hat{\mathbf{g}}, \mathbf{g})$ represents how well the presence or absence of pitch onsets matches between the inpainted segment and the surrounding segments. Also, GS represents the average similarity of the pitch notes per measure in the inpainted segment and the surrounding segments, $p_{\mathrm{GS}_{\mathrm{pred}}}$ and $p_{\mathrm{GS}_{\mathrm{true}}}$ represent the distributions of the average similarity of the pitch onsets in the prediction and ground truth, respectively. A lower $\mathrm{GS}_{\mathrm{div}}$ indicates that the rhythm in the prediction more closely matches that of the ground truth.

D. Quantitative evaluation result

The quantitative evaluation results on the JSB chorales dataset and the small Irish folk dataset are shown in Table I and Table II, respectively. In Table I, Prop-DR-CNN outperformed the Baseline across all results. In particular, the 2.5-point improvement in pAcc indicates that the proposed method is able to generate pitch sequences more accurately.

Furthermore, the fact that $\mathrm{GS}_{\mathrm{div}}$ improves by approximately 2.5 times compared to the baseline model suggests that the proposed method performs inpainting with grooving patterns more similar to those of the surrounding segments. This can be attributed to the fact that the proposed method inputs the combined musical features of the inpainted segment and its surrounding segments into the GAN discriminator. As a result,

TABLE I: Quantitative evaluation result (JSB chorales)

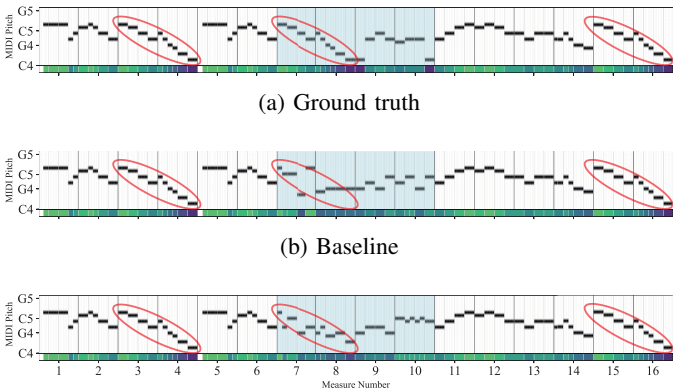
Methods	$\text{pos}_{\text{F}_1}\uparrow$	pAcc↑	rAcc↑	$\text{S}_{\text{div}}\downarrow$	$\text{H}_{\text{div}}\downarrow$	$\text{GS}_{\text{div}}\downarrow$
Baseline	82.9	22.7	69.6	0.079	0.222	0.057
Prop-DR-CNN	**83.4**	25.2	**69.8**	0.076	0.203	**0.022**
Prop-DR-GRU	81.6	23.4	67.3	**0.071**	0.262	0.052
Prop-LV-CNN	79.5	**26.7**	65.5	0.077	0.274	0.034
Prop-LV-GRU	81.8	10.4	66.7	0.082	**0.154**	0.056

TABLE II: Quantitative evaluation result (small Irish folk)

Methods	$\text{pos}_{\text{F}_1}\uparrow$	pAcc↑	rAcc↑	$\text{S}_{\text{div}}\downarrow$	$\text{H}_{\text{div}}\downarrow$	$\text{GS}_{\text{div}}\downarrow$
Baseline	89.3	44.8	83.1	0.022	0.079	0.015
Prop-DR-CNN	**89.7**	**45.5**	**84.0**	0.029	0.083	**0.014**
Prop-DR-GRU	89.5	44.5	83.7	0.023	0.078	0.016
Prop-LV-CNN	89.4	45.4	83.6	**0.019**	**0.073**	**0.014**
Prop-LV-GRU	80.4	12.5	66.9	0.056	0.401	0.034

the model is trained to consider the consistency between the inpainted segment and the preceding and following segments.

On the other hand, the effectiveness of Prop-LV-CNN and Prop-LV-GRU was limited. In these methods, the latent variable $\mathbf{Z}$ is used as input to the GAN discriminator. However, since $\mathbf{Z}$ represents the feature sequence before the decoder upsamples it from one measure to 24 ticks, its sequence length is only 1/24 that of the musical information sequence $\mathbf{C}$. As a result, the GAN may not have functioned effectively due to the lack of sufficient sequential information. In table II, however, Prop-LV-CNN outperformed the Baseline across all metrics. In the case of the small Irish folk dataset, the amount of training data was approximately 10 times larger than that of the JSB chorales dataset. It is therefore considered that the model was sufficiently trained to distinguish between real and generated data based on the latent variables, leading to improved output from the generator.



(a) Ground truth

(b) Baseline

(c) Prop-DR-CNN

Fig. 4: Examples of piano rolls generated from the ground truth and predicted pitch sequences on the JSB chorales dataset

E. Qualitative evaluation result

Fig. 4 shows the examples of piano rolls generated from the ground truth and predicted pitch sequences on the JSB chorales dataset. The blue-highlighted regions indicate the inpainted segments. The colormap below each piano roll represents pitch values, where darker colors correspond to lower pitches and brighter colors to higher pitches. As shown in the area enclosed by the red ellipse in Fig. 4a, the preceding, following, and inpainted segments all contain melodic lines in which the pitch gradually descends. In Fig. 4b, the segment inpainted by the baseline model fails to reproduce this descending melodic line. In contrast, as shown in Fig. 4c, the segment inpainted by the proposed method successfully reproduces the descending melodic line, suggesting that the proposed method achieves more accurate and musically consistent inpainting with respect to the surrounding segments.

Focusing on the pitch color map of the following segment (measures 11 to 16) in Fig. 4, we observe a tendency in which the pitch occurrence frequency is low in the early part and increases toward the latter part. In the inpainted segments (measures 7 to 10) shown in Fig. 4b and Fig. 4c, the result generated by the baseline model exhibits a generally low pitch occurrence frequency, showing no clear resemblance to the following segment. In contrast, the result produced by the proposed method displays a similar trend to that of the following segment. This suggests that the grooving pattern of the inpainted segment generated by the proposed method is similar to that of the following segment, which supports the low GS_{div} value reported in Table I.

Based on the above discussion, incorporating the GAN framework into the training of LatentRNN in InpaintNet appears to enable the generation of musically consistent sequences with pitch patterns similar to those of the preceding and following segments.

IV. Conclusions

In this study, we proposed a novel method that introduces GAN to the previous method InpaintNet. In training the LatentRNN with limited data, we compared the outputs obtained by changing the input information and using two discriminators with different architectures to those of the Baseline. As a result, it became clear that by performing adversarial learning

during LatentRNN training, it becomes possible to perform completion that considers the pitch onset position and pitch type in the surrounding segment. In future work, we aim to further improve performance with limited training data.

REFERENCES

[1] G. Hadjeres, F. Pachet, and F. Nielsen, "DeepBach: A steerable model for Bach chorales generation," in *Proceedings of the 34th International Conference on Machine Learning*, D. Precup and Y. W. Teh, Eds., ser. Proceedings of Machine Learning Research, vol. 70, International Convention Centre, Sydney, Australia: PMLR, Jun. 2017, pp. 1362–1371.

[2] C.-Z. A. Huang, T. Cooijmans, A. Roberts, A. Courville, and D. Eck, "Counterpoint by convolution," in *International Society for Music Information Retrieval (ISMIR)*, 2017.

[3] G. Hadjeres and F. Nielsen, "Anticipation-rnn: Enforcing unary constraints in sequence generation, with application to interactive music generation," *Neural Computing and Applications*, Nov. 2018, ISSN: 1433-3058. DOI: 10.1007/s00521-018-3868-4.

[4] A. Pati, A. Lerch, and G. Hadjeres, "Learning to traverse latent spaces for musical score inpaintning," in *20th International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, Oct. 2019, pp. 343–351. DOI: 10.5281/zenodo.3527814.

[5] K. Chen, C.-i Wang, T. Berg-Kirkpatrick, and S. Dubnov, "Music sketchnet: Controllable music generation via factorized representations of pitch and rhythm," in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR*, 2020.

[6] C.-J. Chang, C.-Y. Lee, and Y.-H. Yang, "Variable-length music score infilling via XLNet and musically specialized positional encoding," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR)*, Online, 2021, pp. 123–130.

[7] L. Hadjeres and L. Crestel, *The piano inpainting application*, arXiv.preprint, arXiv:2107.05944, 2021.

[8] R. Guo, I. Simpson, C. Kiefer, T. Magnusson, and D. Herremans, *Musiac: An extensible generative framework for music infilling applications with multi-level control*, arXiv.preprint, arXiv:2202.05528, 2022.

[9] A. Marafioti, P. Majdak, N. Holighaus, and N. Perraudin, "Gacela: A generative adversarial context encoder for long audio inpainting of music," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 1, pp. 120–131, Jan. 2021, ISSN: 1941-0484. DOI: 10.1109/jstsp.2020.3037506.

[10] K. Liu, W. Gan, and C. Yuan, "Maid: A conditional diffusion model for long music audio inpainting," *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023.

[11] P. P. Ebner and Amr Eltelt, *Audio inpainting with generative adversarial network*, arXiv preprint arXiv:2003.07704, 2020.

[12] A. Marafioti, N. Holighaus, P. Majdak, and N. Perraudin, *Audio inpainting of music by means of neural networks*, 2022. arXiv: 1810.12138 [cs.SD]. [Online]. Available: https://arxiv.org/abs/1810.12138.

[13] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *ICLR 2014 - 2nd International Conference on Learning Representations, April 14-16, 2014, Conference Track Proceedings*, Banff, AB, Canada, Apr. 2014.

[14] K. Cho, B. van Merriënboer, C. Gulcehre, *et al.*, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1724–1734. DOI: 10.3115/v1/D14-1179.

[15] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[16] M. Araneda-Hernandez, F. Bravo-Marquez, D. Parra, and R. F. Cádiz, "MUSIB: Musical score inpainting benchmark," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2023, pp. 1–15, 2023.

[17] B. L. Sturm, J. F. Santos, O. Ben-Tal, and I. Korshunova, *Music transcription modelling and composition using deep learning*, arXiv.preprint, arXiv:1604.08723, 2016.

[18] M. Allan and C. Williams, "Harmonising chorales by probabilistic inference," in *Advances in Neural Information Processing Systems*, vol. 17, 2004.

[19] S.-L. Wu and Y.-H. Yang, "The jazz transformer on the front line: Exploring the shortcomings of ai-composed music through quantitative measures," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, (Montreal, Canada), ISMIR, Oct. 2020, pp. 142–149. DOI: 10.5281/zenodo.4245390.