Human-In-The-Loop Chord Progression Generator With Generative Adversarial Network

Yoshiteru Matsumoto, Hiroyoshi Ito, Hiroko Terasawa, Yuya Yamamoto, Yuzuru Hiraga, and Masaki Matsubara
University of Tsukuba, Ibaraki, Japan
E-mail: {s2221670@s, ito@slis, terasawa@slis, s2130507@s, hiraga@slis, masaki@slis}.tsukuba.ac.jp

***Abstract*—This paper proposes a framework for chord progression generation that uses human perception as a discriminator in a generative adversarial network model. By incorporating human perception as a discriminator, chords that do not exist in the distribution of training data can be treated as correct answers. However, since symbolic chord progressions cannot be calculated as numerical and continuous values, it is not clear how to compute the perturbation of a chord progression for human evaluation. Therefore, we formalized the perturbations of chord progressions in the embedding space based on existing music theory and devised a pairwise comparison task interface to collect human feedback for training the generative model. To verify the effectiveness of the proposed framework, we experimented to generate chord progressions based on crowd workers' evaluations as a discriminator and then asked musicians to evaluate the generated chord progressions. Consequently, the model generates significantly more natural and more diverse chord progressions, compared to the case where human perception is not incorporated.**

I. Introduction

Chord progressions are essential in harmony and a key element of music composition. One of the ways to compose music is to generate chord progressions as a first step and then write bass lines and melodies. Chord progressions determine the mood and impression of the song and can help make the melody. And even if no melody comes to your mind, you can compose music just with chord progressions, rhythms, and humming. However, chord progressions are difficult to come up with, especially for some beginners. Thus, chord builders have been developed to suggest chord progressions[1] and an automatic generation system can be applied as a data generation method to such support systems.

Various methods have been proposed for automatic music and chord progression generation systems based on probabilistic context-free grammar (PCFG) [1], deep neural network(DNN)-based methods like long short-term memory (LSTM) [2], [3], [4], [5], [6], generative adversarial networks (GAN) [8], [9], [10], and the system with a network for audio synthesis used in Magenta[2] [11]. Thus, DNN- and GAN-based methods have been realized on a practical level, including models such as in Magenta. However, these methods cannot generate data with features that do not exist in the dataset, because the generator in the basic GAN aims to fit its distribution with the real data's distribution.

By contrast, several methods have adopted the human-in-the-loop approach (e.g., emotional music generation using the interactive genetic algorithm by Zhu et al. [12], human-in-the-loop drum loop generation by Alain et al. [13], human-in-the-loop melody generation with Bayesian optimization by Zhou et al. [14]). The advantage of incorporating human evaluation is that data that does not exist in the dataset can be treated as correct answers.

Fujii et al. [15] proposed HumanGAN, which uses human perception as a discriminator in GAN for speech generation, and Chu [16] used HumanGAN for face image. However, there are no applications for automatic chord progression generation, and it is not clear how HumanGAN works on it. Since symbolic chord progressions cannot be calculated as numerical and continuous values, it is not easy to compute the perturbation of chord progressions for HumanGAN.

In this paper, we propose a framework for chord progression generation, which utilizes human perception as a discriminator in a GAN model (Figure 1). To achieve that, we formalize perturbations of chord progressions in the embedding space based on existing music theory (Tonal Pitch Space [17]) and devise a pairwise comparison task interface to collect human feedback for training the model. Our framework can be expected to generate more diverse and human-acceptable chord progressions than basic GAN.

The contributions of this paper are as follows:

- Framework: We implemented the GAN-based chord progression generator with human feedback. We also design the task interface to collect the listeners' subjective evaluations.
- Formalization: We defined the chord progression similarity and introduced a circle of fifths/chord embedding space. Then, we formalized perturbations of chord progressions based on music theory.
- Experiment: We experimented to generate chord progressions with crowdworkers evaluation as a discriminator, and asked musicians to evaluate the generated chord progressions. Experimental results showed that the model generates significantly more natural and more diverse chord progressions compared to the case where human perception is not incorporated.

[1] https://steinberg.help/cubase_pro_artist/v9/en/cubase_nuendo/topics/chord_pads/chord_pads_chord_assistant_c.html

[2] https://magenta.tensorflow.org/gansynth

II. Methodology

Model construction consists of two parts (Fig. 1); 1) Pre-training loop of the generative model and 2) Training loop with human feedback.

A. Pre-training Loop

The methods for model construction proposed in this paper can be divided into two main parts, as shown in Fig. 1. The former part is the pre-training loop, which is the first step for the chord progression generator. The latter part is the second step, a training loop based on human feedback. The latter part can also be divided into collecting and training loop sections.

We first trained the generative model with a DNN-based discriminator. This is because the generated data would look like white noises during the early iterations and it is difficult to acquire a large amount of human perceptual evaluations [16]. In this framework, we use SeqGAN[9], a generative model intended to generate token sequences such as natural languages and symbolic notated music, since we can regard chords as "letters" and chord progressions as "sentences."

B. Training Loop with Human Feedback

Next, we train the generative model with human feedback based on the HumanGAN, GAN with human-based discriminator. The HumanGAN trains a generator G to represent a perception distribution and replace a DNN-based discriminator with a human perceptual evaluation-based discriminator D. Human perception is tolerant to deviations from the real data; a human-acceptable distribution of this type is called a "perception distribution" [15]. When the perception distribution covers a wider range than the real data distribution, the basic GAN cannot represent the ranges. This is because the generator in the basic GAN aims to fit the distribution of G based on the real data distribution.

In this case, D is a perceptual evaluation-based discriminator that outputs the posterior probability $D = [0, 1]$ regarding how perceptually acceptable the input $\hat{x}_n$ is. The objective function for training V is reformulated as follows:

$$V(G, D) = \sum_{n=1}^{N} D(G(\boldsymbol{z_n})) \tag{1}$$

The model parameters of the generator G, θ_G, are learned to maximize (1). In this framework, θ_G are based on a gradient-based iterative method and are updated iteratively

$$\theta_G^{(new)} = \theta_G + \alpha \frac{\partial V(G, D)}{\partial \theta_G} = \theta_G + \alpha \frac{\partial V(G, D)}{\partial \hat{\boldsymbol{x}}} \cdot \frac{\partial \hat{\boldsymbol{x}}}{\partial \theta_G} \tag{2}$$

where α is the learning rate.

In the basic GAN, $\partial V(G, D)/\partial \hat{\boldsymbol{x}}$ is estimated through standard backpropagation because of the computational processes in G and D are differentiable. However, $\partial V(G, D)/\partial \hat{\boldsymbol{x}}$ cannot be estimated through backpropagation because D cannot not be differentiated in a HumanGAN. Therefore, we regard humans as black-box systems that output the differences between the posterior probabilities of the generated data. We also estimate $\partial V(G, D)/\partial \hat{\boldsymbol{x}}$ using an optimization algorithm for the black-box system.

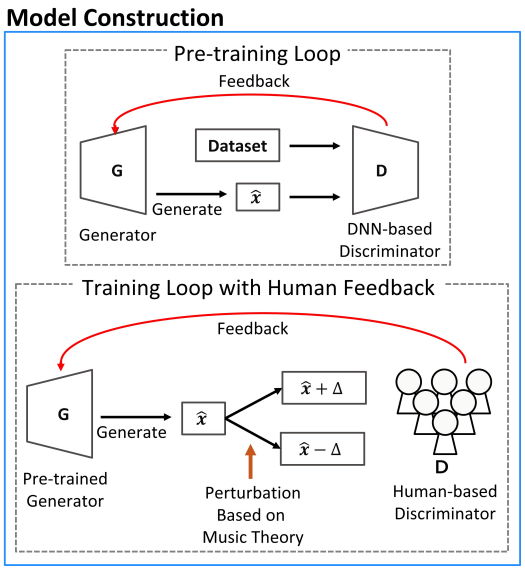

Fig. 1. An overview of model construction: Proposed framework utilizes human perception as the discriminator D in a GAN model. This model can be expected to generate more diverse and human-acceptable chord progressions than basic GAN.

In a HumanGAN, we use natural evolution strategies (NES) [18] to approximate the gradients using data perturbations. First, a perturbation $\Delta \boldsymbol{x_n^r}$ is generated from a multivariate Gaussian distribution $\mathcal{N}(\mathbf{0},\ \sigma^2\mathbf{I})$; here, σ is a constant value of the standard deviation, r is the index of the perturbation$(1 \leq r \leq R)$, and $\mathbf{I}$ is the identity matrix. A human then evaluates the differences between the posterior probabilities of two perturbed data $\{\hat{\boldsymbol{x}}_{\boldsymbol{n}} + \Delta \boldsymbol{x_n^r}, \hat{\boldsymbol{x}}_{\boldsymbol{n}} - \Delta \boldsymbol{x_n^r}\}$ as follows:

$$\Delta D\left(\boldsymbol{x_n^r}\right) \equiv D\left(\hat{\boldsymbol{x}}_{\boldsymbol{n}} + \Delta \boldsymbol{x_n^r}\right) - D\left(\hat{\boldsymbol{x}}_{\boldsymbol{n}} - \Delta \boldsymbol{x_n^r}\right) \tag{3}$$

where $\Delta D\left(\boldsymbol{x_n^r}\right) = [-1, 1]$. These perturbations and evaluation are iterated R times for $\hat{\boldsymbol{x}}_{\boldsymbol{n}}$. Thus, $\partial V(G, D)/\partial \hat{\boldsymbol{x}}$ for backpropagation is approximated as [18]:

$$\frac{\partial V(G, D)}{\partial \hat{\boldsymbol{x}}} = \left[\frac{\partial V(G, D)}{\partial \hat{\boldsymbol{x}}_{\mathbf{1}}}, \ldots, \frac{\partial V(G, D)}{\partial \hat{\boldsymbol{x}}_{\boldsymbol{N}}}\right] \tag{4}$$

$$\frac{\partial V(G, D)}{\partial \hat{\boldsymbol{x}}_{\boldsymbol{n}}} = \frac{1}{2\sigma R} \sum_{r=1}^{R} \Delta D\left(\boldsymbol{x_n^r}\right) \cdot \Delta \boldsymbol{x_n^r} \tag{5}$$

C. Perturbation Based on Music Theory

1) Basic Idea: In the NES, we need to add perturbations to generated data, as mentioned above, after (2) through (5). The perturbations can be taken as numerical additions and subtractions. And naturally, the smaller the perturbation, the smaller the deviation; the larger the perturbation, the larger the deviation from the original data (e.g., noisy images).

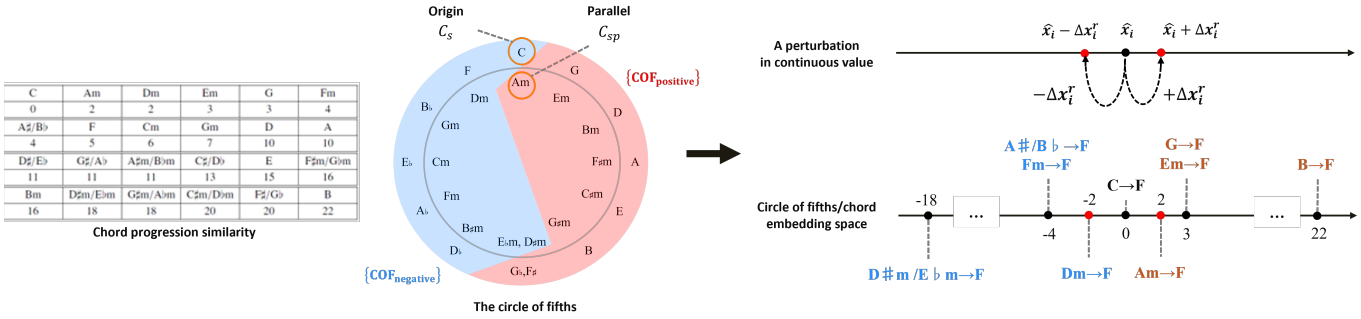

Fig. 2. An example of the circle of fifths/chord embedding space.
We regard chord progressions as the label of coordinates. The diagonal chord of C in the circle of fifths will be E♭m/D♯m and G♭/F♯.

There are two possible methods of adding perturbations to chord progressions. The first one is the method that considers a chord as a set of notes on a staff and moves the components of the chord on the staff according to the magnitude of the perturbation. However, this method may cause dissonance in a perturbed chord conflicting with the generation of a natural chord progression.

The next one is the method that treats the key and degree of chord symbolically and takes into account the function of each chord in the chord progression. This method could not generate dissonance, and we can expect to express the behavior of the perturbed data. However, symbolic chord progressions are not continuous values, unlike audio and images, and cannot calculate as numerical values. Therefore, we attempt to treat chord progressions as pseudo-continuous values by constructing a *circle of fifths/chord embedding space*, a continuous one-dimensional space like *word embedding* in natural language processing, based on Tonal Pitch Space (TPS) [17], [19]. And then we formulate perturbations for chord progressions to take into account the function in the tonality of the chords.

2) Chord Progression Similarity: We first define the chord progression similarity, the proximity between the previous chord progression and the candidate perturbed chord progression. In the TPS, we can calculate *chord proximity* and *regional proximity* as the similarity of two chords. Further, two progressions with close chord/regional proximity can be considered to have close functions in tonality[3]. For example, $\delta\left(\mathrm{I}/\mathbf{C} \to \mathrm{IV}/\mathbf{C}\right) = 5$ and $\delta\left(\mathrm{vi}/\mathbf{C} \to \mathrm{IV}/\mathbf{C}\right) = 7$. If the first chord on out of region $\mathbf{C}$, $\delta\left(\mathrm{I}/\mathbf{D} \to \mathrm{IV}/\mathbf{C}\right) = 15$, where $\mathrm{D}/\mathbf{R}$ is the chord of the degree D on region (key) $\mathbf{R}$ and $\delta\left(\mathrm{D_S}/\mathbf{R_S} \to \mathrm{D_D}/\mathbf{R_D}\right)$ is the chord/regional proximity of chord progression $\mathrm{D_S}/\mathbf{R_S} \to \mathrm{D_D}/\mathbf{R_D}$. The degree D is the position of note or chord on a scale ($\mathbf{R}$ in this time) relative to the tonic (the first note of the scale) and is written mainly in upper and lower case Roman numerals, such as I and vi. Thus, $\mathrm{I}/\mathbf{C}$ indicates C-major, $\mathrm{vi}/\mathbf{C}$ indicates A-minor, etc. The first two examples, I and vi on region $\mathbf{C}$, have the function of *tonic* in $\mathbf{C}$ and have close value, as you can see.

Hereafter, $\mathrm{D_x}/\mathbf{R_x}$ is denoted as C_x for sake of simplicity and the conversion from C_x to $\mathrm{D_x}/\mathbf{R_x}$ (and C_y to $\mathrm{D_y}/\mathbf{R_y}$) shall minimize $\delta\left(\mathrm{D_x}/\mathbf{R_x} \to \mathrm{D_y}/\mathbf{R_y}\right)$. In general, the proximity is smaller when interpreted without musical modulation (e.g., $\delta\left(\mathrm{I}/\mathbf{C} \to \mathrm{IV}/\mathbf{C}\right) < \delta\left(\mathrm{I}/\mathbf{C} \to \mathrm{I}/\mathbf{F}\right)$) [19]. We now define the chord progression similarity d as follows:

$$d\left(C_S \to C_D, C \to C_D\right) := \sqrt{\left(\delta\left(C_S \to C_D\right) - \delta\left(C \to C_D\right)\right)^2} \tag{6}$$

where C_S is the chord before adding the perturbation, C_D is the next chord, and C is any chord.

3) Circle of Fifths/Chord Embedding Space: To map chord progressions into the circle of fifths/chord embedding space (hereinafter called CCES), we assign a positive/negative sign to a chord progression similarity. Here, let COF be the set of chords in the circle of fifths (Fig. 2), as shown below:

$$\begin{aligned}\mathrm{COF} = \{&\mathrm{C}, \mathrm{Am}, \mathrm{G}, \mathrm{Em}, \mathrm{D}, \mathrm{Bm}, \mathrm{A}, \mathrm{F\sharp m}, \mathrm{E},\\ &\mathrm{C\sharp m}, \mathrm{B}, \mathrm{G\sharp m}, \mathrm{G\flat}, \mathrm{E\flat m}, \mathrm{D\flat}, \mathrm{B\flat m},\\ &\mathrm{A\flat}, \mathrm{Fm}, \mathrm{E\flat}, \mathrm{Cm}, \mathrm{B\flat}, \mathrm{Gm}, \mathrm{F}, \mathrm{Dm}\}\end{aligned} \tag{7}$$

First, we set the origin in the CCES. For any chord $C \in \mathrm{COF}$, let the chord progression $C = C_S$ hold up $d\left(C_S \to C_D, C \to C_D\right) = 0$ to be the origin in the CCES.

Then, we create two sets to assign positive/negative signs to chord progression similarity and determine coordinates in the CCES. Of the origin $C = C_S$ and the chord C_{SP} in parallel key of C, let $\mathrm{D_{Smajor}}$ be the chord in a major key and $\mathrm{D_{Sminor}}$ be the chord in a minor key. The chords on the right of C_S in the circle of fifths and $\mathrm{D_{Sminor}}$ are elements of the set $\mathrm{COF_{positive}}$, and the chords on the left of C_S and $\mathrm{D_{Smajor}}$ are in $\mathrm{COF_{negative}}$. Note that these chords are separated on the diagonal of C_S to hold up $\mathrm{COF} \setminus \mathrm{COF_{positive}} = \mathrm{COF_{negative}}$. Now, we determine coordinates in the CCES d_s as follows:

$$\begin{aligned} &d_s\left(C_S \to C_D, C \to C_D\right) :=\\ &\quad (-1)^k d\left(C_S \to C_D, C \to C_D\right)\\ &k = \begin{cases} 0 & (C \in \mathrm{COF_{positive}})\\ 1 & (otherwise)\end{cases}\end{aligned} \tag{8}$$

[3]See the appendix for its formulation in https://giganticbite.github.io/hil-CPGenerator

4) Perturbation for Chord Progressions: Using the CCES defined in the previous section, we determine how to add a perturbation to a chord progression. Let T be the length of generated data $\hat{\boldsymbol{x}}_{\boldsymbol{n}}$, we first obtain a vector of random numbers as $\boldsymbol{S} = \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ ($\mathbf{0} \in \mathbb{R}^{T-1}$, $\mathbf{I}$ is the $(T-1)\times(T-1)$ identity matrix). From (3), we can make a pair of perturbed data $\left\{\hat{\boldsymbol{x}}_{\boldsymbol{n}} + \Delta\boldsymbol{x}_{\boldsymbol{n}}^{\boldsymbol{r}}, \hat{\boldsymbol{x}}_{\boldsymbol{n}} - \Delta\boldsymbol{x}_{\boldsymbol{n}}^{\boldsymbol{r}}\right\}$ with $\hat{\boldsymbol{x}}_{\boldsymbol{n}} = [C_1, \ldots, C_{T-1}, C_T]$.

As $\hat{x}_{n,i} + \Delta x_{n,i}^{r}$, we obtain the chord $C_{pp,i}$ associated with the coordinate $d_s(C_i \to C_{i+1}, C_{pp,i} \to C_{i+1})$ in the CCES closest to $s_i \in \boldsymbol{S}$, where i is the index of each element of $\hat{\boldsymbol{x}}_{\boldsymbol{n}}$ (i.e., chord) and $\Delta\boldsymbol{x}_{\boldsymbol{n}}^{\boldsymbol{r}}$ is the perturbation to $\hat{\boldsymbol{x}}_{\boldsymbol{n}}$. If there are multiple candidate chords, $C_{pp,i}$ is determined completely at random to avoid bias. In the same way, we obtain the chord $C_{pn,i}$ associated with the coordinate $d_s(C_i \to C_{i+1}, C_{pn,i} \to C_{i+1})$ in the CCES closest to $-s_i$ as $\hat{x}_{n,i} - \Delta x_{n,i}^{r}$. Through this process, $1 \leq i < T$, we finally get the pair of perturbed data:

$$\begin{aligned}\hat{\boldsymbol{x}}_{\boldsymbol{n}} + \Delta\boldsymbol{x}_{\boldsymbol{n}}^{\boldsymbol{r}} &= [C_{pp,1}, \ldots, C_{pp,T-1}, C_T] \\ \hat{\boldsymbol{x}}_{\boldsymbol{n}} - \Delta\boldsymbol{x}_{\boldsymbol{n}}^{\boldsymbol{r}} &= [C_{pn,1}, \ldots, C_{pn,T-1}, C_T]\end{aligned} \tag{9}$$

D. Perception Distribution

We make the perception distribution using feedback data to see the variety and the human-acceptable range of generated chord progressions. The perceptual distribution is calculated by weighting the transition probability of the perturbed chord progressions as follows:

1) As we mention in Section III-B2, the feedback scale is mapped to the differences between the posterior probabilities $\Delta D\left(\boldsymbol{x}_{\boldsymbol{n}}^{\boldsymbol{r}}\right)$.
2) Define the weight of the perturbed chord progressions $D\left(\hat{\boldsymbol{x}}_{\boldsymbol{n}} + \Delta\boldsymbol{x}_{\boldsymbol{n}}^{\boldsymbol{r}}\right), D\left(\hat{\boldsymbol{x}}_{\boldsymbol{n}} - \Delta\boldsymbol{x}_{\boldsymbol{n}}^{\boldsymbol{r}}\right)$.
3) Multiply the transition probability of each chord by the weights determined by 2., to create the perception distribution:

$$\begin{aligned}P_{pd}\left(C_{pp,j}|C_{pp,i}\right) &= P\left(C_{pp,j}|C_{pp,i}\right) \cdot D\left(\hat{\boldsymbol{x}}_{\boldsymbol{n}} + \Delta\boldsymbol{x}_{\boldsymbol{n}}^{\boldsymbol{r}}\right) \\ P_{pd}\left(C_{pn,j}|C_{pn,i}\right) &= P\left(C_{pn,j}|C_{pn,i}\right) \cdot D\left(\hat{\boldsymbol{x}}_{\boldsymbol{n}} - \Delta\boldsymbol{x}_{\boldsymbol{n}}^{\boldsymbol{r}}\right)\end{aligned} \tag{10}$$

where $P\left(C_j|C_i\right)$ is a transition probability from a chord C_i to a chord C_j (indicating each cell of Fig. IV-A), $C_{pp,i}$ and $C_{pp,j}$ are the chords extracted from $\hat{\boldsymbol{x}}_{\boldsymbol{n}} + \Delta\boldsymbol{x}_{\boldsymbol{n}}^{\boldsymbol{r}}$, $C_{pn,i}$ and $C_{pn,j}$ are the chords extracted from $\hat{\boldsymbol{x}}_{\boldsymbol{n}} - \Delta\boldsymbol{x}_{\boldsymbol{n}}^{\boldsymbol{r}}$, and $P_{pd}\left(C_j|C_i\right)$ is a transition probability of perception distribution (indicating each cell of Fig. IV-A). Thus, if you have $\hat{\boldsymbol{x}}_{\boldsymbol{n}} + \Delta\boldsymbol{x}_{\boldsymbol{n}}^{\boldsymbol{r}} = [\mathrm{C}, \mathrm{F}, \mathrm{G}]$, you can calculate transition probabilities of $(C_{pp,i}, C_{pp,j}) = (\mathrm{C}, \mathrm{F}), (\mathrm{F}, \mathrm{G})$. We also make a distribution with the transition frequency $F\left(C_j|C_i\right)$ used in Section IV-A.

III. Experiment

A. Pre-training Loop and Dataset

We set the output length of the generative model $T = 9$ (mentioned below), the learning rate of the optimization algorithm $\alpha = 0.001$, the number of embedding layers to 32 for the generator and 64 for the discriminator, and the rest of the parameters are the same as the PyTorch version[4].

We also obtained the dataset for the pre-training from HookTheory[5] that is a database of chord progressions and melodies. This dataset comprises 19,664 phrases (e.g., Intro and Chorus), and we only used chord progressions in the dataset, all of which were transposed into the key of C.

Many pieces of music consist of multiple 8 or 4 bars, and there is a method generating 8 bars of music as an experiment [20]. In the application of automatic generation, however, 4 bars chord progressions would reduce the involvement and variation of the chord progressions. And 16 bars would increase the burden on the listeners and make it difficult to catch the whole impression. Thus we decided to generate $T = 8$ chord progressions (one chord per bar) in this study.

For all of the lengths of each chord progression $T = 8$, we add *EOC* (End Of Chord progression) to the end of each chord progression as a terminal symbol. This is the same as the output length of the generative model. If the length of a chord progression $T < 8$, we did not use it; if $T > 8$, we cut it out from the back via $T = 8$ to give the chord progression a cadence (a sense of resolution). Through this process, the dataset comprised 14,776 chord progressions in the end. In addition, we classified the 704 chords in the dataset into 24 major and minor triads, to reduce the number of chords, and the vocabulary size of the generative model (Fig. 3).

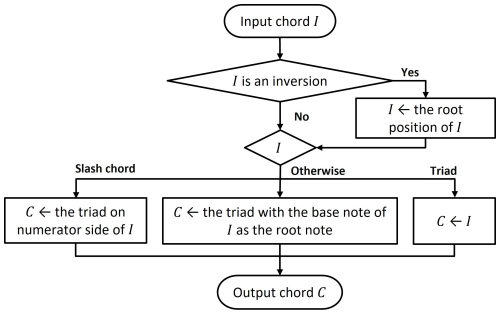

Fig. 3. The flowchart of chord classification.
This procedure makes chords into triads and the vocabulary size of the generative model less.

B. Collecting Feedback from Crowdworkers

In order to collect listeners' subjective feedback as (3) in various countries, we used Lancers[6] in Japan and Amazon Mechanical Turk[7] in English-speaking countries. This feedback consisted of two parts: a music screening and a feedback task. The total number of listeners was 120, 60 each from Japan and English-speaking countries. Hereafter, the crowdworkers in the feedback are referred to as "listeners."

[4]https://github.com/ZiJianZhao/SeqGAN-PyTorch
[5]https://www.hooktheory.com/theorytab
[6]httpss://www.lancers.jp/
[7]https://www.mturk.com/

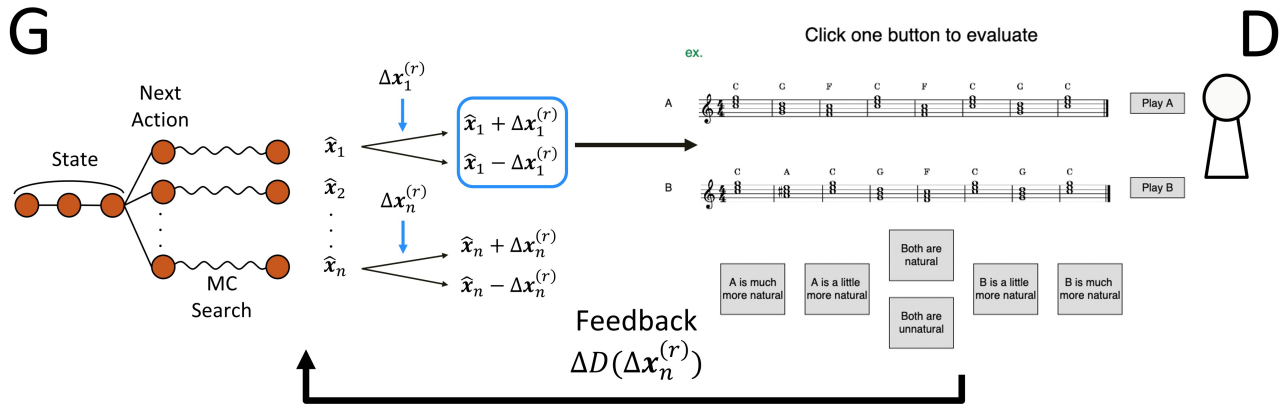

Fig. 4. Experiment overview.
The training loop comprises collecting perceptual evaluations of pairs of chord progressions as feedback and training using collected feedback. The right part of this image is the interface to feedback on the differences between the posterior probability of chord progressions.

1) Music Screening: The listeners took a test about their musical background and musical experience using the Gold-smiths Musical Sophistication Index (Gold-MSI) [21], [22]. In this study, we made a questionnaire form for two of the five Gold-MSI indices: perceptual abilities for listening to music, and musical training, for music and musical instrument training.

2) Feedback Task: The listeners evaluated the naturalness of the chord progressions and gave feedback on the difference between the posterior probabilities to the generative model. The stimuli in the task were a pair of two perturbed chord progressions [8] , and the listener worked with 25 sets of this stimulus and feedback as one set, as shown in Fig. 4.

The presented chord progressions were generated from the generative model pre-trained for 5, 10, and 15 epochs. Since the pre-training loop showed a tendency toward over-fitting around epoch 15, we used models pre-trained before epoch 15. We prepared the generated $N = 200$ chord progressions respectively and added $R = 5$ perturbations per data set. And also we decide the standard deviation $\sigma = 3$ of random noise S so that most of the random noise will be $\pm 3\sigma = \pm 9$ on the normal distribution since a perturbed chord can lose its sense of tonality when a random noise, a value of the circle of fifths/chord embedding space is greater than 10 or less than -10. Thus, we collected $200 \times 5 = 1000$ feedback data for each model.

In this framework, the listeners' feedback is the differences between the posterior probabilities of the generated data, instead of a discriminator. If this were applied to the SeqGAN, it would be necessary to gather feedback on all posterior probabilities of the token sequences, chord progressions in this case, generated by a Monte Carlo (MC) search, as shown in Fig. 4. However, if the length of token sequence T, the number of MC search iterations for a token sequence of length t is $(number\_of\_tokens)^{(T-t)}$ and the generator repeats it for $T-1$ times (i.e., $\sum_{t=1}^{T-1} 24^{T-t} = 4,785,883,224$ feedback data needed at this time). This is difficult to do, considering the time required and the burden on the listeners, so the listeners only evaluated the last generated token sequence of length $T = 8$ in this study.

The subjective feedback scales have 6-point for evaluating the naturalness of a pair of chord progressions, A and B: A is much more natural, A is a little more natural, Both are natural, Both are unnatural, B is a little more natural, and B is much more natural. We also mapped these scales to the difference between the posterior probabilities:

$$\Delta D\left(\boldsymbol{x_n^r}\right) = (1.0, 0.5, 0.0, 0.0, -0.5, -1.0) \tag{11}$$

Then, we could get the weight of the perturbed chord progressions mentioned in Section II-D. Since the weighting multiplies 0.0 for *unnatural* chord progressions and 1.0 for *natural* chord progressions, we can assume that the perception distribution consists of chord progressions that humans evaluate as *natural*.

$$D\left(\hat{\boldsymbol{x}}_{\boldsymbol{n}} + \Delta \boldsymbol{x_n^r}\right) = (1.0, 0.75, 1.0, 0.0, 0.25, 0.0) \tag{12}$$
$$D\left(\hat{\boldsymbol{x}}_{\boldsymbol{n}} - \Delta \boldsymbol{x_n^r}\right) = (0.0, 0.25, 1.0, 0.0, 0.75, 1.0) \tag{13}$$

C. Training Loop with Feedback Data

We finally trained the generative model with feedback data according to (2), (4), (5). Here, the learning rate of the optimization algorithm α_h was set to be the same as for the pre-training loop $\alpha_h = \alpha = 0.001$, and we updated the model parameters θ as follows:

$$\theta \leftarrow \theta + \alpha_h \frac{\partial V\left(G, D\right)}{\partial \hat{\boldsymbol{x}}} \cdot \frac{\partial \hat{\boldsymbol{x}}}{\partial \theta_G} \tag{14}$$

where $\partial V\left(G, D\right) / \partial \hat{\boldsymbol{x}}$ is the gradient of the objective function of HumanGAN ((4)).

[8]Note that all the generated chord progressions are in root position.

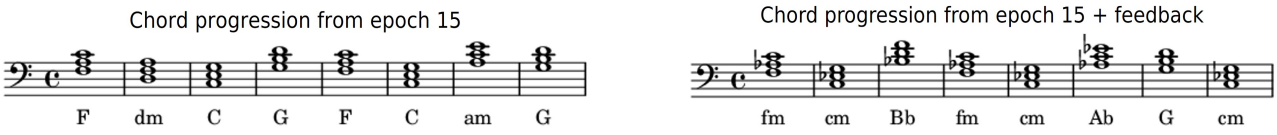

Fig. 5. The most natural chord progressions generated from 15 (left) and 15 + feedback (right) rated in Section III-D.

D. Qualitative Evaluation

To verify the effectiveness of the generative model by the training loop with human feedback, we experimented with a qualitative evaluation approach. In this evaluation, we will use the most trained and optimized generative models: pre-trained for 15 epochs and 15 epochs + feedback (e.g., Fig. 5).

The subjects were seven university faculty members and professional composers with academic degrees in music composition. We choose the semantic differential (SD) method with a 5-point Likert scale (1: disagree – 5: agree) for seven evaluation scales: smoothness, evolvement, the beauty of the sound, simplicity, fun, naturalness, and cadence. In concrete, the subjects evaluated a total of eight chord progressions, four each from the models.

E. Results

Since both models were pre-trained at the same epochs and the values were rating scales, we used the Wilcoxon signed-rank test, a nonparametric test of the two matched samples. Here, the significance level of the Wilcoxon signed-rank test was $\alpha_t = 0.05$, the null hypothesis H_0: the evaluation value of chord progressions from the only pre-trained generator was the same as that from the pre-training + feedback generator, and the alternative hypothesis H_1: the evaluation value of chord progressions from the pre-training + feedback generator was higher than that from only the pre-trained generator, thus making it a one-sided test.

Tab. IV-A shows the test results. Focusing on the evaluation of naturalness, the result showed that the null hypothesis was rejected for epoch 15 and epoch 15 + FB, since $p < 0.05$, and it was statistically significant that the evaluation of naturalness increased with the training loop with human feedback. Similarly, it was statistically significant for the evaluation of cadence and the beauty of the sound.

TABLE I
THE RESULTS OF THE WILCOXON SIGNED-RANK TEST. 15 AND 15+FB INDICATE THE TRAINING EPOCH OF THE GENERATIVE MODEL, PRE-TRAINED FOR 15 EPOCH AND 15 EPOCH + FEEDBACK.

	15	15+FB		
	average		statistic	p-value
smoothness	3.04	**3.54**	229.0	0.08
cadence	2.04	**2.79**	176.0	**<0.05**
naturalness	2.79	**3.75**	223.5	**<0.01**
fun	2.54	**2.71**	137.5	0.36
simplicity	3.11	**3.68**	156.0	0.08
beauty of sound	3.21	**3.93**	195.5	**<0.05**
evolvement	**2.96**	2.79	101.0	0.70

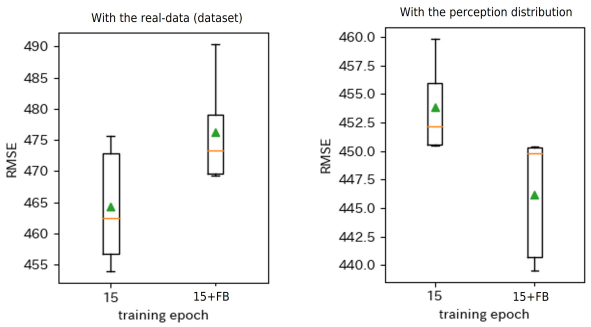

Fig. 6. The comparison on the average of five attempts for the RMSE of transition frequencies.
Each caption indicates the ground truth, and triangles in the box plots indicate the average of each RMSE. The dataset: the larger the RMSE, the more various the chord progression. The perception distribution: the smaller the RMSE, the more natural the chord progression.

IV. Discussion and Limitations

A. Discussion

Based on the result, we found that the training loop with human feedback on the naturalness of chord progressions had a positive effect on the naturalness of the generated data (i.e., the training increased the value of the naturalness of the data generated from the model that had been relatively pre-trained, and the same was true for cadence and beauty of sound).

We also checked the diversity of the output chord progressions. Fig. 6 shows the root mean square errors (RMSE) of the transition frequency $F\left(C_j|C_i\right)$ from a chord C_i to a chord C_j of the dataset and the perception distribution (formulated in Section II-D and showed in Fig. 7) as the ground truth. When the dataset is the ground truth, the larger the RMSE, the better the model can generate the chord progression that represents a feature not found in the dataset, and when the perception distribution is the ground truth, the smaller the RMSE, so the more natural chord progressions can be generated for humans. From Fig. 6 and Fig. 5, it can be said that our framework generates significantly more natural and more diverse chord progressions, compared to the case where human feedback is not incorporated.

B. Limitation

For vocabulary reduction of the generative model and the dataset, we classified chords into triads, as in Section III-A. However, this operation leads to reducing the feature of chord progressions that cannot be expressed by triads (e.g., seventh chords and ninth chords, which are found in jazz). And this system cannot be used when a user wants to generate other

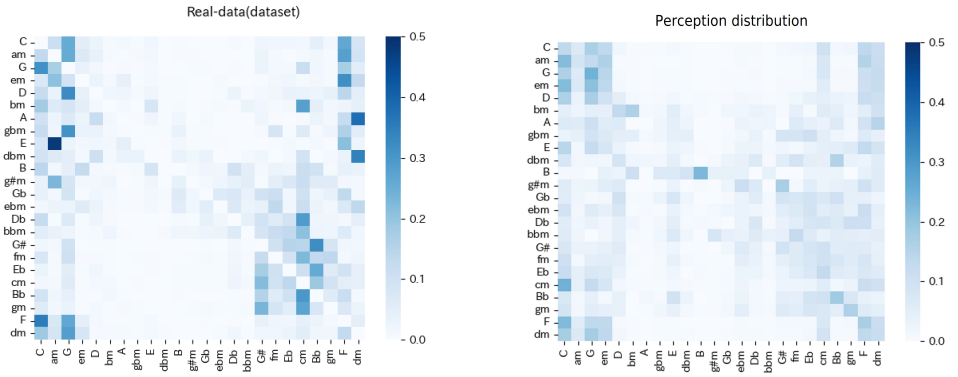

Fig. 7. The transition probabilities matrices of chord progressions.
The real-data on the left and the perception distribution on the right. The vertical axis indicates a from-chord C_i and horizontal axis indicates a to-chord C_j, $C_i \rightarrow C_j$. The perception distribution has some peaks that don't or a little exist in the real-data (e.g., C-minor to C-major and C-minor to E-minor).

than 8 bars because we limited $T = 8$ for generating chord progressions. It also reduces the number of available chord progressions in dataset and leads to decrease a cadence since we need to cut out chord progressions (see Section III). We will take on this issue in future work and consider how to keep the model small without this operation.

V. Conclusion

In this study, we propose a framework for chord progression generation that uses human perception as a discriminator in a generative adversarial network model. The qualitative evaluation results show the positive effect on naturalness, cadence, and evolvement of the chord progressions generated by the relatively pre-trained model, and the model generates significantly more natural and more diverse chord progressions, compared to the case without human perception.

Acknowledgment

This study was partially supported by JSPS KAKENHI Grant Number 21H03552 and 22K17944.

References

[1] H. Tsushima, E. Nakamura, K. Itoyama, and Y. Kazuyoshi. "Function- and Rhythm-Aware Melody Harmonization Based on Tree-Structured Parsing and Split-Merge Sampling of Chord Sequences," in *Proc. of the International Society for Music Information Retrieval Conference*, pp. 502–508, 2017.

[2] K. Choi, G. Fazekas, and M. Sandler. "Text-based LSTM networks for Automatic Music Composition," in *1st Conference on Computer Simulation of Musical Creativity*, 2016.

[3] C. Marc and K. Itou. "Generating Homophonic Music with LSTMs Dedicated to Melody and Harmony," in *The 83rd National Convention of IPSJ*, ID. 2P–07, pp. 267–268, 2021.

[4] H. Lim, S. Rhyu, and K. Lee. "Chord Generation from Symbolic Melody Using BLSTM Networks," in *Proc. of the International Society for Music Information Retrieval Conference*, pp. 621–627, 2017.

[5] W. Yang, P. Sun, Y. Zhang, and Y. Zhang. "CLSTMS: A Combination of Two LSTM Models to Generate Chords Accompaniment for Symbolic Melody," in *Proc. of International Conference on High Performance Big Data and Intelligent Systems*, pp. 176–180, 2019.

[6] C. Zhuang and Y. Jinming. "GCA: A chord music generation algorithm based on double-layer LSTM," in *Proc. of International Conference on Advances in Computer Technology, Information Science and Communication*, pp. 57–61, 2021.

[7] I. Goodfellow, J. Abadie, M. Mirza, B. Xu, D. WardeFarley, S. Ozair, A. Courville, and Y. Bengio. "Generative adversarial nets," in *Proc. of Neural Information Processing Systems*, pp. 2659–2665, 2017.

[8] H. Dong, W. Hsiao, L. Yang, and Y. Yang. "MuseGAN: Multi-Track Sequential GenerativeAdversarial Networks for Symbolic Music Generation and Accompaniment," in *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, pp. 34–41, 2018.

[9] L. Yu, W. Zhang, J. Wang, and Y. Yu. "SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient," in *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, pp. 2852–2858, 2017.

[10] S. Lee, U. Hwang, S. Min, and S. Yoon. "Polyphonic Music Generation with Sequence Generative Adversarial Networks," in *arXiv abs/1710.11418*, 2018.

[11] J. Engel, K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts. "GANSynth: Adversarial Neural Audio Synthesis," in *Proc. of International Conference on Learning Representations*, 2019.

[12] Z. Wang, H. Zhu, and S. Wang. "Emotional Music Generation Using Interactive Genetic Algorithm," in *Proc. of IEEE International Conference on Computer Science and Software Engineering*, pp. 345–348, 2008.

[13] G. Alain, M. Boisvert, F. Osterrath, and R. Taillefer. "DeepDrummer : Generating Drum Loops using Deep Learning and a Human in the Loop," in *Proc. of Joint Conference on AI Music Creativity*, pp. 11, 2020.

[14] Y. Zhou, Y. Koyama, M. Goto, and T. Igarashi. "Generative Melody Composition with Human-in-the-Loop Bayesian Optimization," in *Proc. of Joint Conference on AI Music Creativity*, pp. 10, 2020.

[15] K. Fujii, Y .Saito, S. Takamichi, Y. Baba, and H. Saruwatari. "HumanGAN: generative adversarial network with human-based discriminator and its evaluation in speech perception modeling," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6239–6243, 2020.

[16] L. Chu. "HumanGAN for Human Faces," https://github.com/lanstonchu/HumanGAN-Faces, (accessed 2022-09-12).

[17] F. Lerdahl, *Tonal Pitch Space*. Oxford University Press, 2001.

[18] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin. "Black-box Adversarial Attacks with Limited Queries and Information," in *Proc. of International Conference on Machine Learning*, vol. 2m, pp. 2137–2146, 2018.

[19] S. Sakamoto and S.Tojo. "Harmony Analysis of Music in Tonal Pitch Space," in *IPSJ SIG Technical Report*, Vol. 2009-MUS-80, No. 9, pp. 1–6, 2009, (in Japanese).

[20] L. Yang, S. Chou, and Y. Yang. "MidiNet: A Convolutional Generative Adversarial Network for Symbolic-domain Music Generation," in *Proc. of the International Society for Music Information Retrieval Conference*, pp. 324–331, 2017.

[21] D. Müllensiefen, B. Gingras, J. Musil, and L. Stewart. "The musicality of non-musicians: An index for assessing musical sophistication in the general population," PloS One, 9, e89642, 2014.

[22] M. Sadakata, Y. Yamaguchi, C. Ohsawa, M. Matsubara, H. Terasawa, A. von Schnehen, D. Müllensiefen, and K. Sekiyama. "The Japanese translation of the Gold-MSI: Adaptation and validation of the self-report questionnaire of musical sophistication," in *Musicae Scientiae*, 2022.