SURPRISENET: MELODY HARMONIZATION CONDITIONING ON USER-CONTROLLED SURPRISE CONTOURS

Yi-Wei Chen **Hung-Shin Lee** **Yen-Hsing Chen** **Hsin-Min Wang**
Institute of Information Science, Academia Sinica, Taiwan

ABSTRACT

The surprisingness of a song is an essential and seemingly subjective factor in determining whether the listener likes it. With the help of information theory, it can be described as the transition probability of a music sequence modeled as a Markov chain. In this study, we introduce the concept of deriving entropy variations over time, so that the surprise contour of each chord sequence can be extracted. Based on this, we propose a user-controllable framework that uses a conditional variational autoencoder (CVAE) to harmonize the melody based on the given chord surprise indication. Through explicit conditions, the model can randomly generate various and harmonic chord progressions for a melody, and the Spearman's correlation and p-value significance show that the resulting chord progressions match the given surprise contour quite well. The vanilla CVAE model was evaluated in a basic melody harmonization task (no surprise control) in terms of six objective metrics. The results of experiments on the Hooktheory Lead Sheet Dataset show that our model achieves performance comparable to the state-of-the-art melody harmonization model.

1. INTRODUCTION

In recent years, deep learning has developed rapidly and has become the main technology for automatic music generation. In this study, we focus on the task of automatic melody harmonization, in which a system needs to assign appropriate and harmonic chords to a given melody. From previous studies, we have seen that the models based on the bidirectional long short-term memory (BiLSTM) perform well in this task [1, 2]. Most of them can generate harmonic chords to harmonize a melody. In [3], by introducing blocked Gibbs sampling and class weighting, the model can further generate more reasonable and interesting chords, and is even comparable to human composers.

Based on these previous studies, we hope to further control the model to generate chords according to both melody conditions and user instructions. Latent representation learning is a powerful method that has been widely used in computer vision [4–6] and speech processing fields [7, 8] to learn semantically meaningful information of attributes. Such techniques have also been applied to music processing in several recent works [9–11]. Motivated by these latent variable models, we modify the variational autoencoder (VAE) [12] so that the chord-to-chord mapping can be trained under the conditions of a melody sequence and a user-defined temporal contour.

However, in the model, what are the most attractive and practical controllable conditions for users? Many high-level subjective emotions, such as happiness, sadness, surprisingness, and interestingness, can describe a musical sequence. These emotions seem to be abstract and difficult to quantify objectively. Fortunately, some of them can be calculated from the information dynamics [13]. Previous music theory studies have shown that perceptual qualities and subjective states, such as uncertainty, surprisingness, complexity, tension, and interestingness, are closely related to the measurement of information theory, such as relative entropy and mutual information. In [13], the authors explored this idea in the context of Markov chains using musical sequences, resulting a structural analysis, which is largely consistent with the views of professional human listeners. Therefore, we use the surprisingness metric, which is defined as the negative log transition probability in a Markov chain, to generate the time-varying surprisingness contour of a chord sequence.

Similar to Mellotron, a text-to-speech system that uses pitch and rhythm contours as conditions to synthesize speech [14], in the training stage, we concatenate a chord sequence with additional conditions, namely its corresponding melody and surprise contour, feed them into an encoder to convert them into latent variables. Then, the latent variables are concatenated with the melody and surprise contour again, and input into a decoder to reconstruct the chord sequence. The model is expected to learn the latent representations of chords when the melody and surprise contour are given. Owing to the sampling mechanism of the VAE-based model, in the inference stage, we can randomly sample latent variables from the standard normal distribution to generate a variety of chords, which cohere the input melody and propagate according to the required surprise contour. In addition, we extend the 96 chords used in [3] to all chord types in the Hooktheory Lead Sheet Dataset [15] to expand the chord selection of the model.

Our model is named SurpriseNet. It can harmonize a melody through user-controllable conditions. The highlights of SurpriseNet are two-fold. First, it relieves the

© Yi-Wei Chen, Hung-Shin Lee, Yen-Hsing Chen, Hsin-Min Wang. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Yi-Wei Chen, Hung-Shin Lee, Yen-Hsing Chen, Hsin-Min Wang, "SurpriseNet: Melody Harmonization Conditioning on User-controlled Surprise Contours", in *Proc. of the 22nd Int. Society for Music Information Retrieval Conf.,* Online, 2021.

tension between coherence and surprisingness caused by the melody and user-supplied condition, respectively. In general, it is easy to catch one factor (i.e., surprisingness) and lose another. Second, the results show that the vivid and harmonic chords generated by our model can not only correspond to the melody, but also strictly follow the given surprise contour. Several examples are available at `https://scmvp301135.github.io/SurpriseNet`.

2. RELATED WORK

2.1 Automatic Melody Harmonization

Automatic melody harmonization aims to establish a learning model that can generate chord sequences to accompany a given melody [16, 17]. In music, a chord is an arbitrary harmonic set consisting of three or more notes, which sounds as if these notes are sounding simultaneously [18]. Conventionally, methods based on hidden Markov models (HMMs) [19–21] and genetic algorithms (GA) [22] are commonly used to deal with the task.

Recently, some models based on deep learning have been proposed. For example, Lim *et al.* first proposed a model based on the BiLSTM [1]. The melody is input to the model to predict the simplified 24 chords(i.e., the major and minor triads) in the Wikifonia corpus. Based on the same model architecture in [1], Yeh *et al.* proposed a model called MTHarmonizer, in which the chord types were extended to 48 by considering major, minor, augment and diminished chords in a larger corpus (i.e., the Hooktheory Lead Sheet Dataset) [2]. In addition, they integrated information of chord functions [23] into the loss function to help chord label prediction [2]. However, there are several drawbacks in the above methods, such as overusing common chords and incorrect phrasing problems. Sun *et al.* tried to solve these problems and produced interesting but still reasonable chords by introducing the orderless NADE training techniques, class weights, and Blocked Gibbs sampling into their model. They also extended the chord space to 96 types, including major, minor, augment, diminish, suspend, major7, minor7, and dominant7 [3].

2.2 Controllable Music Generation

Music generation can be regarded as a conditional estimation problem defined as $p(\text{music}|\text{condition})$, where both "music" and "condition" are usually time-series features. Related tasks include melody-based chord generation [17] and chord-based melody generation [23, 24].

An alternative way is to learn the joint distribution $p(\text{music}, \text{condition})$, and then set the condition during the generation process. Related tasks include automatic music completion or accompaniment based on the melody or chords [25–29]. However, many abstract music factors, such as music texture, melody contour, or other high-level subjective perception, are difficult to be explicitly encoded.

Latent representation learning is an ideal solution to the above problem, because representation learning embeds discrete music and condition sequences into a continuous latent space, and accurately captures the latent information from the music. Recent research has used disentangled representation learning to achieve controllable music generation models for style transfer, texture variation, and accompaniment arrangement [11]. High-level subjective perception can also be captured in the latent space to generate music following the arousal condition [9]. These studies show that VAEs [30,31] are an effective framework for learning the representation of discrete music sequences. We incorporated this idea into our research, and expected the model to capture the latent information of chords when conditioned by melody sequences and surprise contours.

3. METHODOLOGY

In this section, we will introduce the calculation of surprise contours and the model architecture in detail. SurpriseNet is based on a conditional VAE, and its goal is to learn the representation of chords when conditioned by the melody sequences and surprise contours. In the inference process, the random latent variables, melody conditions, and surprise contours are provided to the decoder to produce harmonization.

3.1 Surprise Contour

Measures such as entropy and mutual information can be used to characterize random processes. One of the salient effects of listening to music is to create expectations for what is to come next, which may be fulfilled immediately, after a delay, or not at all depending on the situation. An essential aspect of this is that music is experienced as a phenomenon that "unfolds" over time, rather than being apprehended as a static object presented in its entirety [13].

Consider a snapshot of a stationary random process taken at a certain time: we can divide the timeline into the past and present parts, denoted as $t-1$ and t, respectively. Here we will consider one of the simplest random processes, a first-order Markov chain. Let S be a Markov chain with a finite state space $\{1, ..., N\}$ such that S_t is the random variable representing the t-th element of the sequence. We can establish a transition matrix $a \in \mathbb{R}^{N \times N}$ encoding the distribution of any element of the chord sequence given the previous element, that is $p(\text{chord}_t|\text{chord}_{t-1}) = a_{t,t-1}$. According to the definition in [13], the surprise contour can be derived as the negative log probability as

$$\text{Surprisingness} = -\log p(\text{chord}_t|\text{chord}_{t-1}). \quad (1)$$

Equation (1) is actually the definition of the information content in information theory. It means that in a chord sequence, the higher the surprise value at a certain time, the greater the amount of information at that time.

3.2 VAE and Conditional VAE

As described in [31], a common goal for various kinds of autoencoders is that they compress the salient information within the input sample into a lower-dimensional latent code. Ideally, this will force the model to produce compact

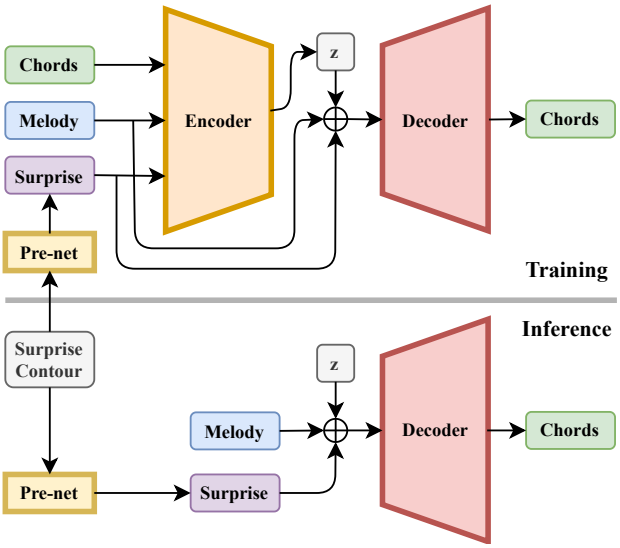


Figure 1. The structure of SurpriseNet. In the inference stage, only Decoder and Pre-net are used.

representations to capture the important factors of variation in the dataset. In pursuit of this goal, VAE [30, 32] introduces the constraint that the latent code $\mathbf{z}$ is a random variable distributed according to a prior $p(\mathbf{z})$.

The generative process of VAE is described as follows. A latent variable $\mathbf{z}$ is generated from the prior distribution $p(\mathbf{z})$, and the observation $\mathbf{x}$ is generated by the generative distribution $p(\mathbf{x}|\mathbf{z})$ conditioned on $\mathbf{z}$; that is, $\mathbf{z} \sim p(\mathbf{z})$ and $\mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$. A VAE consists of an encoder θ, which approximates the posterior $p(\mathbf{z}|\mathbf{x})$, and a decoder ϕ, which parameterizes the likelihood $p(\mathbf{x}|\mathbf{z})$. Following the framework of variational inference, we do posterior inference by minimizing the KL divergence between the approximate posterior (i.e., the output of the encoder) and the true posterior $p(\mathbf{z}|\mathbf{x})$ by maximizing the evidence lower bound (ELBO). The objective function of VAE with Gaussian latent variables is

$$\tilde{\mathcal{L}}_{VAE} = \mathbb{E}[\log p_\theta(\mathbf{x}|\mathbf{z})] - KL(p_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})). \quad (2)$$

In the common case where $p(\mathbf{z})$ is a diagonal-covariance Gaussian distribution, this can be circumvented by replacing $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma\mathbf{I})$ with

$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \epsilon, \quad (3)$$

where ϵ is a small random factor.

As for the conditional VAE [12], the conditional generative process of the model is given as follows. For a given condition $\mathbf{c}$, $\mathbf{z}$ is drawn from the prior distribution $p(\mathbf{z}|\mathbf{c})$ realized by a standard Gaussian distribution, and the output $\mathbf{x}$ is generated from the distribution $p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{c})$. Therefore, the training objective function can be expressed as

$$\tilde{\mathcal{L}}_{CVAE} = \mathbb{E}[\log p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{c})] - KL(p_\phi(\mathbf{z}|\mathbf{x}, \mathbf{c})||p(\mathbf{z}|\mathbf{c})). \quad (4)$$

Our work is divided into two parts. We first train the conditional VAE models for 96 or 633 types of chords, conditioned only on the melody, to observe the performance in completing the basic harmonization task. Next, we select the better conditional VAE to train the final SurpriseNet in combination with the surprise contours.

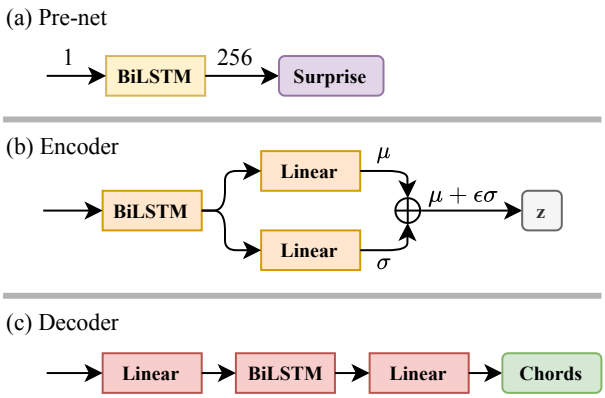


Figure 2. Three main components of SurpriseNet.

The main architecture of SurpriseNet and its key components are shown in Fig. 1 and Fig. 2, respectively. Our components of VAE follow the structures used in [33] and MusicVAE [31], and finally combine with the conditional part [12] to complete the harmonization task. In the training stage, the surprise contour is processed by the Pre-net implemented by a BiLSTM, as shown in Fig. 2(a), to extend the feature from a scalar to a 256-dimensional vector. Afterwards, the features of the chord, melody, and extended surprise contour are concatenated and fed into the encoder to generate a latent code $\mathbf{z}$ subject to a standard normal distribution. The encoder is also implemented by a BiLSTM, as shown in Fig. 2(b), where the size of each layer is 256 or 512. Different from the RNN-based VAE [33] and MusicVAE [31], the frame-wise outputs are further transformed by two linear layers to respectively generate the sequential latent variables, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, with dimensionalities of 16 or 64.

As for the decoder, instead of performing it in an autoregressive manner as in MusicVAE, we concatenate $\mathbf{z}$, melody, and extended surprise representation frame by frame as inputs of the decoder to reconstruct the chord sequence. The number of layers and hidden size in the decoder are the same as those in the encoder. Dropout was employed with a rate of 0.2 on each BiLSTM to prevent overfitting. The batch size was set to 64. Early stopping for 10 epoch patience was applied.

In the inference stage, given the melody and the surprise contour, the surprise contour is first processed by the Pre-net. Then, the latent variable is sampled from a standard normal distribution. Finally, the latent variable, melody, and extended surprise representation are input to the decoder to complete the harmonization process. The implementation details of the model are available at `https://github.com/scmvp301135/SurpriseNet`.

4. EXPERIMENTS

4.1 Datasets

We performed experiments on the Hooktheory Lead Sheet Dataset (HLSD) [15], which contains high-quality and human-arranged melodies with chord progressions. The

dataset is provided in two formats, event-based JSON files and MIDI files. Furthermore, there are many types of labels on chords, such as chord symbols and Roman numerals for reference. The dataset contains a total of 633 chord types, including 9th, 11th, 13th, half diminished, and slash chords. In previous studies conducted on the dataset [1–3], the number of chord types was simplified to 48 (major and minor triads) or 96 (major, minor, augment, diminish, suspend, major7, minor7, and dominant7). In this study, we experimented with two settings, 96 and 633 chord types.

We followed the data split in [3]; the training set contains 17,505 samples, the test set contains 500 samples. For each song, the melody and chords are aligned every two beats in a measure, and the chords are encoded into a one-hot format for training.

4.2 Objective Metrics

For objective evaluation, we used six different objective metrics proposed in [2]. The first three metrics measure the quality of the chord progression, and the others measure the harmonicity between the melody and the chords.

- **Chord histogram entropy (CHE):** The entropy of the chord histogram.
- **Chord coverage (CC):** The number of chord types in a chord sequence.
- **Chord tonal distance (CTD):** The tonal distance between two chords when they are represented by 6-D feature vectors [34].
- **Chord tone to non-chord tone ratio (CTnCTR):** The ratio of the number of chord tones to the number of non-chord tones.
- **Pitch consonance score (PCS):** The sum of the consonance scores between a melody note and each note in a given chord.
- **Melody-chord tonal distance (MCTD):** The tonal distance between a melody note and a chord when they are represented by 6-D feature vectors.

4.3 Surprise Contour Evaluation

The task of user-controlled melody harmonization has never been seen in the literature. Therefore, there is no baseline systems for comparison. We decided to use some statistical methods to evaluate the correlation or causation between a given contour and the generated sample, instead of subjective testing.

Unlike the pitch and rhythm error evaluation in Mellotron [14], we used Spearman's correlation, which is suitable for continuous and discrete ordinal values between two variables. The p-value was used to determine the significance of the results under the assumption that there is no significant correlation between the surprisingness trend of the predicted chord progression and the given surprise contour. A small p-value indicates strong evidence against the assumption, which means that the results are correlated to some extent.

The reason for not using error evaluation, such as the mean squared error (MSE), is that the harmonization result is jointly decided by the melody and the surprise contour. If the error between the generated trend and the given surprise contour is zero, it means that the model completely follows the surprise contour and ignores the melody conditions. Obviously, this is not acceptable and will lead to discordant harmonization results.

Table 1. Objective evaluation results with respect to various models. For the metrics related to Chord Progression, the higher value in CHE and CC means the higher diversity of the generated chords, and the lower value in CTD implies that the chord progression is smoother. As for the metrics related to Harmonicity, the higher value in CTnCTR and PCS and the lower value in MCTD indicate better harmonization results. The arrow denotes whether the metric is the larger the better or the lower the better.

Chord Progression	CHE↑	CC↑	CTD↓
Humans	1.266	4.344	0.628
Sun *et al.*, $\|S\| = 96$	1.280	4.900	0.730
CVAE, $\|S\| = 96$	1.210	4.712	**0.577**
CVAE weight, $\|S\| = 96$	1.670	7.360	0.620
CVAE, $\|S\| = 633$	1.644	7.074	0.730
CVAE weight, $\|S\| = 633$	**1.934**	**9.890**	0.649
M/C Harmonicity	CTnCTR↑	PCS↑	MCTD↓
Humans	0.726	0.515	1.276
Sun *et al.*, $\|S\| = 96$	**0.887**	**0.652**	**1.052**
CVAE, $\|S\| = 96$	0.851	0.611	1.110
CVAE weight, $\|S\| = 96$	0.841	0.523	1.190
CVAE, $\|S\| = 633$	0.767	0.530	1.229
CVAE weight, $\|S\| = 633$	0.705	0.476	1.290

5. RESULTS

In this section, we will first compare the conditional VAE models with Sun *et al.*'s model [3] and human performance in terms of six objective metrics. Then, we will illustrate some harmonization samples generated by SurpriseNet based on different surprise contours. The last part is the correlation analysis.

5.1 Objective Evaluation

The objective evaluation results are shown in Table 1. Compared with the results of humans and Sun *et al.*'s model, we can see that the vanilla CVAE model without using class weights (cf. CVAE, $|S| = 96$) achieved comparable results in all metrics. It performed better in CTD, indicating that the generated chord progression is smoother. After introducing chord balancing (cf. CVAE weight, $|S| = 96$), the CVAE model learned how to sample rare chords, thereby increasing the chord diversity, as shown in the results in CC and CHE.

We further expanded the chord space to 633 to maintain the integrity of the chords in the dataset. Due to the extension of the chord dimension, the deeper CVAE without using class weights (cf. CVAE, $|S| = 633$) obtained results

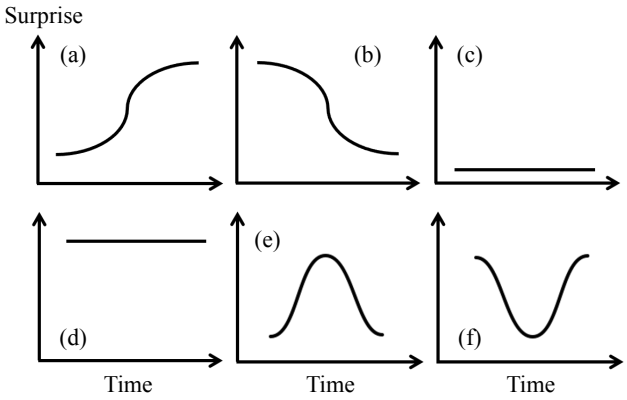


Figure 3. The six given surprise contours.

comparable to the vanilla CVAE model with chord balancing (cf. CVAE weight, $|S| = 96$). After introducing chord balancing (CVAE weight, $|S| = 633$), the deeper CVAE model achieved the best chord diversity, with an average of nearly 10 chord types in a musical sequence.

Despite the above improvements, trade-offs in other metrics (such as CTnCTR, PCS, and MCTD) can be observed. As pointed out in [3], rare chords, such as 7th, 9th, 11th, and 13th, will cause a degradation in the melody/chord harmonicity metrics, but this is mainly due to the definition of CTnCTR, PCS, and MCTD. In order to maintain the diversity of chords, we decided to train SurpriseNet with the conditional VAE architecture considering 633 chord types.

5.2 Generated Samples

We compared the chord generation results of SurpriseNet (based on CVAE, $|S| = 633$) and weighted SurpriseNet (CVAE weight, $|S| = 633$) based on 6 representative surprise contours, as shown in Fig. 3. The 6 contours were generated by the sigmoid function, plain line, and normal distribution, and their reversed profiles, respectively. We intended to check whether the generated chord progression really follows the surprise contour to harmonize the given melody.

To use the sigmoid function to represent the surprise contour, we first normalized it to match the maximum value in the surprise contours of the training data. This contour (cf. Fig. 3(1)) represents a song with lower chord variation in the first half and higher chord variation in the second half. The reverse sigmoid function leads to the opposite trend (cf. Fig. 3(2)). In the case of plain line, we used two sequences consisting of zero (cf. Fig. 3(3)) and the maximum value (cf. Fig. 3(4)) as the surprise contours. A surprise contour with all values being zero indicates that there should be no fluctuation in the chord sequence. In other words, it is expected to see that the given melody will always be harmonized with the same chord. As for the surprise contour with all values being the maximum, it indicates that there should be a lot of up-and-down changes in the resulting chord sequence. That is, it is expected to see that the given melody will be harmonized with various chords. These two cases are considered the most extreme cases. According to the normal distribution, we expect that the highest arousal will appear in the middle of the harmonization result (cf. Fig. 3(5)). As for its inverse profile (cf. Fig. 3(6)), it is expected to generate a plain and more predictable result in the middle of the chord sequence.

Table 2. Spearman's ρ and p-value significance of SurpriseNet and weighted SurpriseNet.

Method	**Spearman's ρ**	**p-value**
SurpriseNet	0.517	< 0.001
Weighted SurpriseNet	0.406	< 0.001

Fig. 4 and Fig. 5 show the harmonization results of SurpriseNet and weighted SurpriseNet for a 4-measure melody, respectively. They are displayed in the same order as the function types in Fig. 3. From Fig. 4(1), as expected, we can see that SurpriseNet generated continuous C chords for the first two measures at the beginning, and then generated varying chords for the last part of the song, according to the given sigmoid-like surprise contour in Fig. 3(1). From Fig. 4(2), we can also see that SurpriseNet generated different chords at the beginning, and then generated more C and G chords that appeared previously, following the given reverse sigmoid-like surprise contour in Fig. 3(2). As for the results of weighted SurpriseNet (see Figs. 5(1) and 5(2)), the chord progressions generated were not exactly as expected, but some surprising and complicated chords were brought in for users' reference. Next, given the all-zero surprise contour in Fig. 3(3), it is obvious that both models followed the condition to generate only one type of chord in the results (see Figs. 4(3) and 5(3)). As for the all-maximum surprise contour in Fig. 3(4), as shown in Figs. 4(4) and 5(4), it is also obvious that the results generated by the two models changed in the chord type almost every two beats, which is the minimum time unit for changing the chord in the training data. For the normal distribution contour in Fig. 3(5), the result generated by SurpriseNet is roughly as expected, with more chord changes in the middle of the song (see Figs. 4(5)). But the result of weighted SurpriseNet is quite different from expectations (see Figs. 5(5)). For the inverse normal distribution contour in Fig. 3(6), the results of both models are not in line with our expectations. But we can see that they are similar to the results generate based on the reverse sigmoid-like surprise contour in the beginning part of the song. When the model is initially assigned a high surprise value, the trend in the first part of the output is similar.

In summary, the above samples generated by SurpriseNet are almost in good agreement with our expectations. The model can indeed generate chords that have a tendency to follow a given surprise contour. However, weighted SurpriseNet seems to over concentrate on using more complicated or rare chords to harmonize the melody due to the class penalty (i.e., weights), so that the trend is not clearly consistent with the given contour.

5.3 Correlation Measurement

Because there is no existing model for this task, we use Spearman's ρ and p-value to evaluate the correlation and
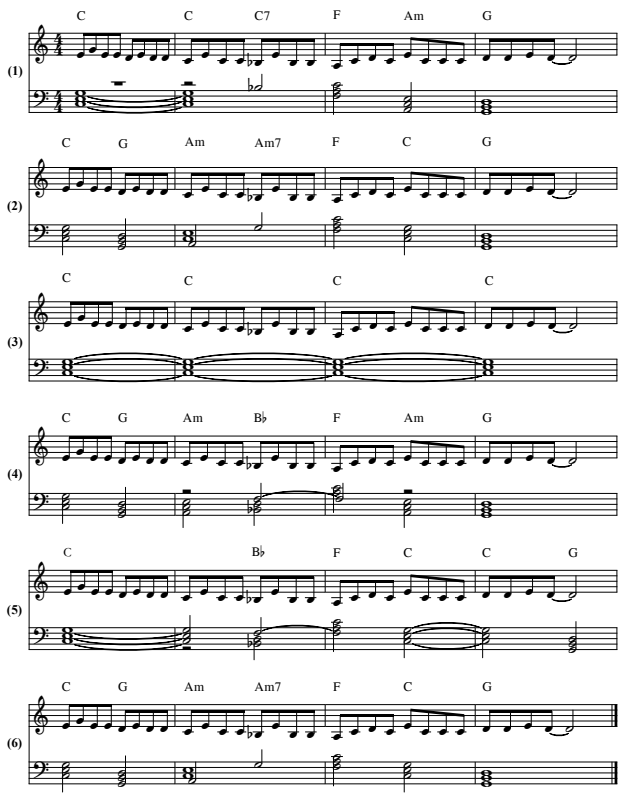


Figure 4. Samples generated by SurpriseNet.
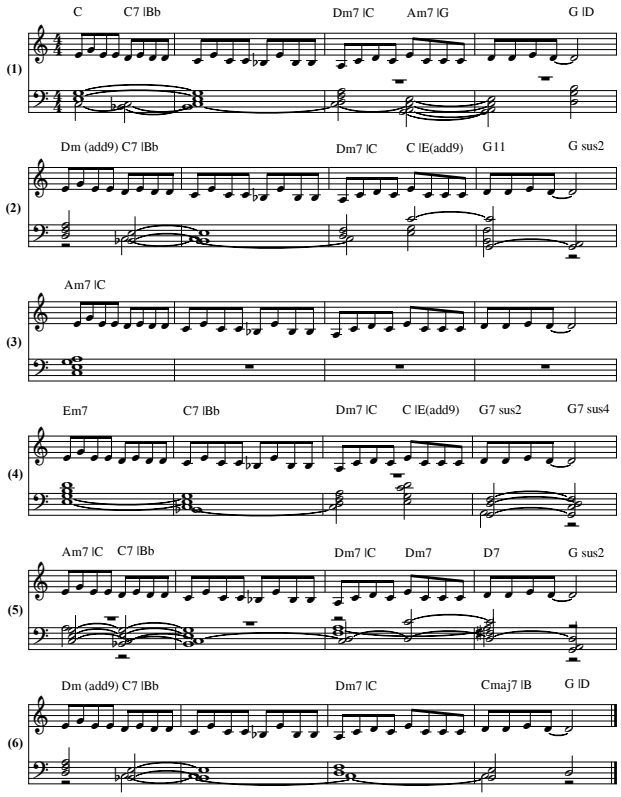


Figure 5. Samples generated by weighted SurpriseNet.

significance between the surprisingness values in the given surprise contour and the surprisingness values in the generated chord progression. From Table 2, we can find that there is indeed a certain correlation between the given surprise contour and the generated chord progression. Furthermore, SurpriseNet seems to be more controllable than weighted SurpriseNet, with a higher Spearman's ρ. The p-value shows that there is no significant difference between the surprisingness trend of the given surprise contour and the surprisingness trend of the generated chord progression for the two models. The result implicitly confirms that given a surprise contour and a melody, these models can generate the corresponding chords as instructed to complete the melody harmonization task, thereby achieving a user-controlled model.

6. FUTURE WORK

Our demo website will be available soon. Users can draw any surprise contour to control the model to generate corresponding chords for a given melody. The link will be provided at `https://github.com/scmvp301135/SurpriseNet`.

The HLSD dataset contains rich intonation data, such as Roman, symbol, secondary, and mode data. We can introduce some approaches from the NLP field, such as modeling these data by referring to various language models. Moreover, in this work, the rhythmic type of chords is simplified and restricted to two beats in a measure. But in fact, there are various rhythmic types in the dataset, such as syncopation and tuplets. Perhaps a disentangled representation learning model can be used to capture this information, so that we can implement an omni model with more complicated rhythms.

In addition, surprise is still an open for discussion topic. In this work, we only consider the surprise in the chord sequence. In the future, can also consider the surprise in the melody sequence at the same time. Moreover, the surprise can be considered not only as the conditional probability of past chord events but also as the conditional probability of the melody sequence at that time. These different considerations will bring different meanings to the surprise.

7. CONCLUSIONS

In this paper, we proposed SurpriseNet, which is based on a conditional VAE model and combines a surprise contour from the transition probability in a Markov chain, to achieve a user-controlled melody harmonization task. From the generated samples, we observed that the model could accurately generate various harmonic chord progressions according to the given surprise contours. The Spearman's correlation and p-value significance show that there is a positive correlation between the given surprise contour and the generated chord progression.

The vanilla conditional VAE model was evaluated in the basic melody harmonization task. The objective evaluation results show that the conditional VAE model could achieve performance comparable to the state-of-the-art melody harmonization model [3]. We expanded the chord space from 96 to 633 to broaden the range of chord selection for the model. The conditional VAE model could generate more types of chords, such as 7th, 9th, 11, 13th, and slash chords, resulting in vivid and harmonic results.

8. REFERENCES

[1] H. Lim, S. Rhyu, and K. Lee, "Chord generation from symbolic melody using BLSTM networks," in *Proc. ISMIR*, 2017.

[2] Y. C. Yeh, W. Y. Hsiao, S. Fukayama, T. Kitahara, B. Genchel, H. M. Liu, H. W. Dong, Y. Chen, T. Leong, and Y. H. Yang, "Automatic melody harmonization with triad chords: A comparative study," 2020. [Online]. Available: https://arxiv.org/abs/2001.02360

[3] C.-E. Sun, Y.-W. Chen, H.-S. Lee, Y.-H. Chen, and H.-M. Wang, "Melody harmonization using orderless NADE, chord balancing, and blocked Gibbs sampling," in *Proc. ICASSP*, 2020.

[4] H. Kim and A. Mnih, "Disentangling by factorising," in *Proc. MLR*, 2018.

[5] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. NIPS*, 2016.

[6] Y. Li and S. Mandt, "Disentangled sequential autoencoder," 2018. [Online]. Available: https://arxiv.org/abs/1709.01620

[7] W. N. Hsu, Y. Zhang, and J. Glass, "Unsupervised learning of disentangled and interpretable representations from sequential data," in *Proc. NIPS*, 2017.

[8] Y. Wang, D. Stanton, Y. Zhang, R. J. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *Proc. ICML*, 2018.

[9] H. H. Tan and D. Herremans, "Music fadernets: controllable music generation based on high-level features via low-level feature modelling," in *Proc. ISMIR*, 2020.

[10] G. Brunner, A. Konrad, Y. Wang, and R. Wattenhofer, "MIDI-VAE: Modeling dynamics and instrumentation of music with applications to style transfer," in *Proc. ISMIR*, 2018.

[11] Z. Wang, D. Wang, Y. Zhang, and G. Xia, "Learning interpretable representation for controllable polyphonic music generation," in *Proc. ISMIR*, 2020.

[12] K. Sohn, X. Yan, and H. Lee, "Learning structured output representation using deep conditional generative models," in *Proc. NIPS*, 2015.

[13] S. Abdallah and M. Plumbley, "Information dynamics: Patterns of expectation and surprise in the perception of music," *J. Connection Science*, vol. 21, 2009.

[14] R. Valle, J. Li, R. Prenger, and B. Catanzaro, "Mellotron: Multispeaker expressive voice synthesis by conditioning on rhythm, pitch and global style tokens," in *Proc. ICASSP*, 2020.

[15] C. Anderson, D. Carlton, R. Miyakawa, and D. Schwachhofer, "Hooktheory." [Online]. Available: https://www.hooktheory.com

[16] C.-H. Chuan and E. Chew, "A hybrid system for automatic generation of style-specific accompaniment," in *Proc. IJWCC*, 2007.

[17] I. Simon, D. Morris, and S. Basu, "MySong: automatic accompaniment generation for vocal melodies," in *Proc. CHI*, 2008.

[18] D. Makris, I. Kayrdis, and S. Sioutas, "Automatic melodic harmonization: An overview, challenges and future directions," in *Trends in Music Information Seeking, Behavior, and Retrieval for Creativity*. IGI Global, 2016, pp. 146–165.

[19] J. F. Paiement, D. Eck, and S. Bengio, "Probabilistic melodic harmonization," in *Proc. Canadian AI*, 2006.

[20] H. Tsushima, E. Nakamura, K. Itoyama, and K. Yoshii, "Function- and rhythm-aware melody harmonization based on tree-structured parsing and split-merge sampling of chord sequences," in *Proc. ISMIR*, 2017.

[21] ——, "Generative statistical models with self-emergent grammar of chord sequences," *J. New Music Res.*, 2018.

[22] T. Kitahara, S. Giraldo, and R. Ramírez, "JamSketch: Improvisation support system with GA-based melody creation from user's drawing," in *Proc. CMMR*, 2018.

[23] T.-P. Chen and L. Su, "Functional harmony recognition of symbolic music data with multi-task recurrent neural networks," in *Proc. ISMIR*, 2018.

[24] L. C. Yang, S. Y. Chou, and Y. H. Yang, "MidiNet: A convolutional generative adversarial network for symbolic-domain music generation," in *Proc. ISMIR*, 2017.

[25] G. Hadjeres, F. Pachet, and F. Nielsen, "DeepBach: A steerable model for bach chorales generation," in *Proc. ICML*, 2017.

[26] H. Zhu, Q. Liu, N. J. Yuan, C. Qin, J. Li, K. Zhang, G. Zhou, F. Wei, Y. Xu, and E. Chen, "Xiaoice band: A melody and arrangement generation framework for pop music," in *Proc. ACM SIGKDD*, 2018.

[27] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," in *Proc. AAAI*, 2018.

[28] C. Donahue, H. H. Mao, Y. E. Li, G. W. Cottrell, and J. McAuley, "LakhNES: Improving multi-instrumental music generation with cross-domain pre-training," in *Proc. ISMIR*, 2019.

[29] I. Simon, A. Roberts, C. Raffel, J. Engel, C. Hawthorne, and D. Eck, "Learning a latent space of multitrack measures," 2018. [Online]. Available: https://arxiv.org/abs/1806.00195

[30] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proc. ICLR*, 2014.

[31] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A hierarchical latent vector model for learning long-term structure in music," in *Proc. ICML*, 2018.

[32] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proc. ICML*, 2014.

[33] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," in *Proc. SIGNLL*, 2016.

[34] C. Harte, M. Sandler, and M. Gasser, "Detecting harmonic change in musical audio," in *Proc. AMCMM*, 2006.