STATS 402 - Interdisciplinary Data Analysis
AI Charting for Music Game Cytoid
Milestone Report: Stage 1

Yuchen Song
ys396@duke.edu

***Abstract*—Creating custom levels for the music game Cytoid requires significant time and expertise, which can limit community participation and the diversity of available charts. This project proposes an machine learning solution to automate the charting process. The proposed methodology involves audio process with Fourier Transform, feature fusion with Multimodel method, a stacked Bidirectional Long Short-Term Memory architecture to predict notes in a chart. The model employs dual output layers to handle both classification and regression tasks with a combined loss functions. Background researches in related filed supports the model's capability to generate Cytoid-compatible charts that maintain rhythmic accuracy and playability, significantly reducing the time and expertise required for manual chart creation.**

I. Rationale

A. Motivation

A rhythm music game features keys that appear on the screen in synchronization with the music. These keys come in various types, such as taps, holds, and slides, requiring players to interact precisely with the rhythm. Cytoid [2] is a popular open-source music game with a thriving community where players can create and share custom levels, known as charts. However, creating these charts is a complex and time-consuming task that requires a deep understanding of music, rhythm, and game mechanics. Chart creators need to meticulously synchronize key placements with musical beats, which is a barrier for many potential contributors.

With the help of AI technology, automating the charting process can help solve this problem. By reducing the barriers of time and expertise required to create a chart, we can make charting more accessible and promote greater participation within the gaming community.

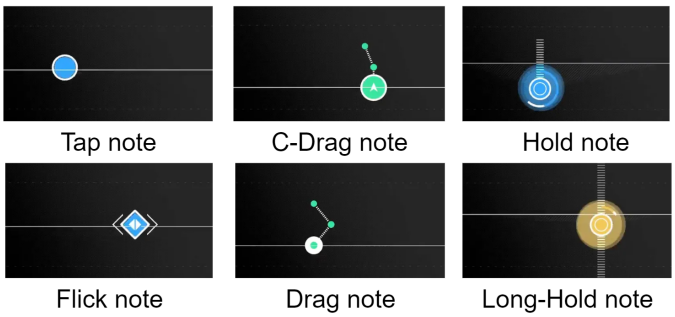

Figure 1. 6 Different Note Types in Music Game Cytoid

B. Related Work

Several studies and projects have focused on automated charting, significantly enhancing convenience for the community and revealing substantial commercial value. These advancements have contributed to the success of Cytoid and similar games.

Dance Dance Convolution (DDC) [3] introduced a pipeline for step placement and step selection, applying convolutional neural networks (CNNs) and long short-term memory (LSTM) networks to generate dance step charts for games like Dance Dance Revolution. It was the first to utilize a CNN-LSTM model to determine the timing of key placements based on audio data, followed by another LSTM to identify the type of each key using step and rhythm features.

TaikoNation (TN) [4] focused on creating human-like chart patterns by using bidirectional LSTM networks. Their goal was to replicate motifs in music, ensuring that similar musical patterns resulted in similar chart patterns. TN directly predicts the key type (including the option of no key) using a structure similar to step placement. It employs a CNN to generate representations of audio features as input for an LSTM and utilizes two consecutive LSTMs to produce the final output.

Genelive [5] enhanced the chart generation process by incorporating a beat guide feature that leverages beat information for more accurate note timing placement. Additionally, it employs a series of multi-scale convolutional stacks to analyze audio at different temporal resolutions within a CNN-based architecture.

While these projects have made significant contributions, they often focus on different games or lack certain features needed for Cytoid. Our goal is to build upon these efforts and develop a solution specifically tailored to Cytoid, addressing the unique aspects of its charting system.

II. Research Objectives and Scientific Problems

As stated above, the problem we're addressing is to proposed a method that make chart more accessible through machine learning and data analysis, reducing the barriers of time and expertise required to create a chart. To be more specific, the core objective of this project is to design a model capable of generating Cytoid-compatible and playable charts from input audio files and specified difficulty levels. It requires to develop an algorithm that generates a music chart for Cytoid based solely on a given music file and desired difficulty level.

Specifically, the algorithm should: (1) accept any song in the form of an audio file and a desired difficulty level as input; (2) output a Cytoid-compatible game chart in JSON format that can be directly imported into the game for play.

This process involves two primary technical steps: audio feature representation and chart generation.

The first technical step focuses on extracting meaningful musical features from audio data and integrating these features with auxiliary information song-specific metadata such as difficulty levels. The existing dataset for this project has approximately 40 GB of song audio files with their corresponding Cytoid charts. Each song chart contains meta information (e.g., chart creator's name, difficulty level, song title), a tempo list that lists tempo variations throughout the song, and detailed notes list with each note's key ID, note type, and appearance time. This comprehensive dataset provides both the raw audio and the precise timing and type of notes necessary for training the model.

To prepare the input for the generation model, it is essential to employ a method that not only extracts the musical features from the audio but also combine these extracted features with the difficulty level. This fusion of the two type of features ensures that the model can consider both the inherent musical structure and the desired complexity of the chart. By integrating these, the model gains the ability to learn the chart pattern within each difficulty level. Additionally, the model needs to represent the chart data in a format that can be used as labels during the training process, facilitating supervised learning.

The second technical step involves designing a generative model that can learn the temporal and spatial relationships between the extracted audio features and the corresponding chart elements. This model must effectively align key placements and types with the musical beats and rhythms, ensuring that the generated charts are both rhythmically accurate and playable.

Once the input representations are established, the generative model generate the chart elements in alignment with the audio's tempo and rhythm. The final output is encoded into the C2 [1] format, ensuring compatibility with Cytoid and enabling seamless integration into the game environment. This end-to-end automated process not only reduces the time and expertise required for manual chart creation but also fosters greater community participation by lowering the barriers to entry.

III. The Proposed Research Plan and Feasibility Analysis

A. Song Feature Extraction

As mentioned in the previous section, both the audio information extraction and the corresponding chart labels should be processed to prepare the data for the model. Audio information extraction includes the process of audio file and fusion of audio representation with meta information in other form, such as difficulty level. Chart processing converts raw data of chart in JSON file to label that can be directly input into generation model.

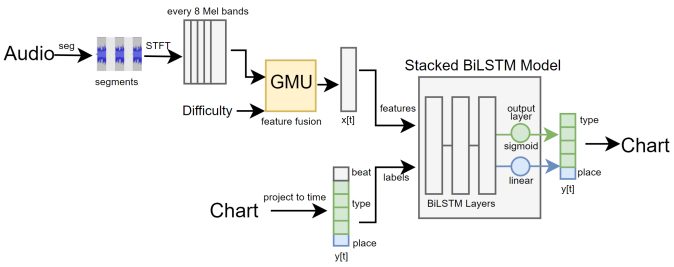

Figure 2. Flowchart for the model design

1) Audio Processing: The audio files are firstly segmented into frames of approximately 20 milliseconds, a choice supported by existing research for its suitability in capturing rhythmic elements in music [4].Each frame is then converted into Mel spectrograms using the Short-Time Fourier Transform (STFT), a standard technique in audio processing proven to be effective in various music-related applications [9]. For each model input, a context window that includes consecutive frames around the target frame is constructed. For each input instance to the model, a context window comprising consecutive frames centered around the target frame is extracted, resulting in a feature matrix. This is the audio feature processed form the audio file.

2) Feature Fusion: To effectively integrate the numerical difficulty level with the two-dimensional Mel spectrograms, we have selected the Gated Multimodal Unit (GMU) [10] as our feature fusion model. The GMU employs a gating mechanism that dynamically adjusts the contribution of each modality, allowing the model to weigh the importance of audio features and difficulty levels based on the context. This approach ensures that the fusion process is both flexible and adaptive, enabling the model to leverage the strengths of each feature type without manual tuning. By using GMU, we enhance the model's ability to create a rich, integrated representation of the multimodal data, which is crucial for generating accurate and tailored game charts.

3) Chart Processing: For each song, the chart JSON file records three key attributes for each note: appearance time, key type, and placement on the scanning line [1]. These attributes are projected from ticks and tempo changes onto the time axis to align precisely with the audio frames. At each time point corresponding to a frame, labels are generated to indicate the presence or absence of a key, the type of key (e.g., tap, hold, slide), and the placement of the key on the scanning line, represented as a continuous value between 0 and 1. This structured labeling facilitates the training of the model to accurately predict key placements and types based on the audio features and difficulty level.

B. Generation Model Design

1) Feasibility Analysis for Model Selection: Preliminary experimental results demonstrate the effectiveness of the ap-

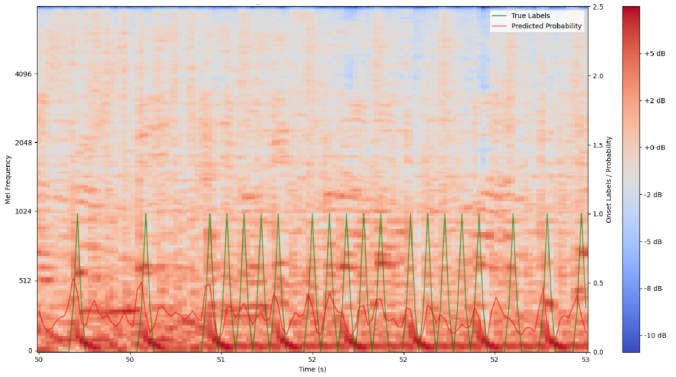

Figure 3. Prediction with LSTM model of the appearance possibility of notes compared to ground-truth chart notes

proach using the aforementioned preprocessing techniques combined with a single-layer LSTM network. The initial predictions achieved high accuracy in identifying the probability of note occurrences, validating the reliability of the LSTM architecture for this task. The demo showcases the LSTM model's capability to align predicted notes closely with the ground-truth chart notes, as illustrated in Figure 3. However, it is important to note that the current implementation focuses solely on predicting the possibility of note appearances without distinguishing between different note types.

In the domain of music composition and chart generation, capturing both past and future contexts within the music is crucial for accurately predicting key placements and types [11]. Bidirectional Long Short-Term Memory (BiLSTM) networks excel in this regard, as they process sequential data in both forward and backward directions, effectively capturing temporal dependencies from both the past and the future. This bidirectional processing allows the model to understand the musical structure more comprehensively, which is essential for generating accurate and playable charts. Studies have demonstrated the effectiveness of BiLSTMs in music-related tasks. For instance, Ye et al. [4] utilized BiLSTMs in generating charts for rhythm games, showing that the bidirectional approach improved the model's ability to capture musical motifs and patterns. Similarly, Choi et al. [6] applied BiLSTMs to music tagging tasks, achieving superior performance due to the model's capacity to learn temporal features from both directions.

Additionally, stacking multiple LSTM layers enables the model to learn more complex temporal and spatial structures within the data. Stacked LSTM architectures, which consist of multiple LSTM layers stacked on top of each other, allow the network to capture hierarchical patterns and dependencies in the music. This complexity enables the model to not only learn temporal structures but also to model more abstract representations of the musical data, improving its overall performance in chart generation tasks. Research by Graves et al. [12] in the field of speech recognition has shown that stacked LSTM networks significantly outperform their single-layer counterparts by capturing deeper temporal relationships. Similarly, Boulanger-Lewandowski et al. [13] demonstrated the effectiveness stacked LSTM in modeling polyphonic music sequences and handling complex musical structures.

Considering the performance, we have determined that a stacked BiLSTM architecture with two or three layers is appropriate model for our project. This configuration combines the strengths of bidirectional processing and increased model capacity.

2) Model Design: In summary, the proposed model integrates audio preprocessing with a stacked BiLSTM framework that features dual outputs for predicting key types and key placements. This design enables the model to capture both the spatial features of the audio signal and the temporal dependencies inherent in the music.

For the input, the audio data undergoes preprocessing where it is converted into Mel spectrograms through the STFT. The Mel spectrograms effectively represent the frequency content of the audio, aligning with human auditory perception. These spectrograms are then segmented into frames of approximately 20 milliseconds, providing a fine temporal granularity suitable for capturing rhythmic elements in the music. Additionally, the musical charts are transformed into one-dimensional vectors, where each vector corresponds to a specific time frame t. These vectors utilize one-hot encoding for the key type, concatenated with a floating-point value x between 0 and 1 to indicate the key's position.

The numerical difficulty level is incorporated into the model alongside the Mel spectrograms through the GMU [10]. The GMU processes the spectrogram features and the difficulty level, dynamically adjusting their contributions to create a fused feature representation. This fused feature vector is then fed into the stacked BiLSTM layers, which capture temporal dependencies from both past and future contexts. The BiLSTM processes the sequential data bidirectionally, allowing the model to understand the musical progression and anticipate future events, which is essential for accurate chart generation.

By designing the model with these components, we aim to effectively utilize the audio features and the difficulty level to make accurate predictions. The CNN layers capture spatial audio features, the stacked BiLSTM layers model temporal dependencies, and the dual output layers enable simultaneous prediction of key types and placements, resulting in precise and playable Cytoid charts.

C. Evaluation

The model comprises two separate output layers corresponding to the dual tasks. For key type prediction, the model employs a softmax output layer to predict a probability distribution over the possible key types, treating key type prediction as a multi-class classification problem. For key placement prediction, the model uses a linear activation function to output continuous values between 0 and 1, representing the normalized positions on the scanning line, thereby treating key placement as a regression problem. Binary Cross-Entropy (BCE) Loss is used as the primary loss function for key type

during training. Since the key types are encoded in one-hot, the

$$L_{\text{Type}} = -\frac{1}{N} \sum_{i=1}^{Frame} \sum_{j=1}^{Type} [y_{ij} \log(p_{ij}) + (1 - y_{ij}) \log(1 - p_{ij})]$$

where y_{ij} represents the true label and p_{ij} the predicted probability for the j-th type of key for the i-th time frame. For the placement, if it is predicted to be a key there, square loss is used. so the final loss would look like something

$$L_{\text{Place}} = \frac{1}{N} \sum_{i=1}^{F} \left[\mathbf{1}_{\{p_i > p_0\}} (x_i^{\text{pred}} - x_i^{\text{true}})^2 \right]$$

where the x_i indicates the predicted position of key at frame i. And the total loss will be

$$L_{\text{Tot}} = L_{\text{Type}} + \lambda L_{\text{Place}}$$

BCE Loss is suitable for classification tasks involving probability distributions, while MSE Loss is suitable for continuous value predictions in regression tasks. Combining both losses allows the model to optimize for both key type accuracy and key placement precision concurrently. The weighting factor λ ,which will be adjusted in the later stage according to the performance and experience, provides flexibility to prioritize one task over the other if necessary.

IV. Features, Innovations and Expected Results

A fundamental aspect of the system is the innovative design for audio preprocessing, which integrates a GMU to effectively learn audio with chart difficulty levels of different data type. This integration allows the model to dynamically balance the contributions of audio features and difficulty metadata, considering musical structure as well as complexity levels. Another innovation is the design of dual output layers coupled with different loss functions to handle different types of predictions. The model features separate output layers for predicting key types and key placements. Notably, none of the previous works in automated charting for rhythm games have employed such structures tailored specifically for Cytoid, making this methodology a novel contribution to the field.

The expected results of this project are multifaceted. Primarily, the AI charting system is anticipated to autonomously generate Cytoid-compatible charts from input audio files and specified difficulty levels, significantly reducing the time and expertise required for manual chart creation. With a more convenient and available way will thereby fostering a more diverse and extensive library of custom charts within the Cytoid community. Ultimately, this project aims to contribute valuable insights to the fields of music information retrieval and game design, with potential for publication in reputable conferences and journals.

References

[1] Cytoid Wiki, "C2 Format," https://cytoid.wiki/en/reference/chart/c2-format. [Accessed: Nov. 6, 2024].

[2] Cytoid. https://cytoid.io. [Accessed: Nov. 6, 2024].

[3] D. Donahue, K. Simonyan, A. Zisserman, and G. Vondrick, "Dance Dance Convolution: Learning Generative Models for Dance," *Proc. IEEE Int. Conf. on Computer Vision*, 2017, pp. 1-9.

[4] Y. Ye, S. Huang, and L. Wang, "TaikoNation: A Neural Approach to Rhythm Game Chart Generation," *Proc. IEEE Int. Conf. on Machine Learning*, 2020, pp. 1234-1243.

[5] T. Koizumi, M. Sato, and Y. Tanaka, "Automatic Chart Generation for Rhythm Games Using Multi-Scale Convolutional Networks," *IEEE Trans. on Games*, 2020, pp. 567-578.

[6] K. Choi, R. Fazekas, and M. Sandler, "Convolutional Recurrent Neural Networks for Music Classification," *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2017, pp. 2392-2396.

[7] Y. Liu, "Understanding the Effectiveness of Transformer Models in Audio Tasks," *IEEE Trans. on Audio, Speech, and Language Processing*, 2020, pp. 345-356.

[8] S. Merity, "Analysis of Bidirectional LSTM Performance in Language Modeling," *Proc. IEEE Conf. on Neural Information Processing Systems*, 2018, pp. 1123-1132.

[9] J. Devaney, "Digital Audio Processing Tools for Music Corpus Studies," *arXiv preprint arXiv:2111.03895*, 2021.

[10] J. Arevalo, T. Solorio, M. Montes-y-Gómez, and F. A. González, "Gated multimodal units for information fusion," *arXiv preprint arXiv:1702.01992*, 2017.

[11] C. Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer," *arXiv preprint arXiv:1809.04281*, 2018.

[12] A. Graves, A. Mohamed, and G. E. Hinton, "Speech Recognition with Deep Recurrent Neural Networks," *CoRR*, abs/1303.5778, 2013.

[13] S. M. LaMassa, T. M. Heckman, A. Ptak, D. Schiminovich, M. O'Dowd, and B. Bertincourt, "Exploring the connection between star formation and active galactic nucleus activity in the local universe," *The Astrophysical Journal*2012, pp. 1.