Emotion-Conditioned Chiptune Music Generation Using a Hybrid PatchTST-LSTM Model

Jing Yuan Sun, Roy Ma

***Abstract*—We propose and evaluate a hybrid deep learning model that combines Patch Time Series Transformers (PatchTST) with Long Short-Term Memory (LSTM) networks for symbolic music generation conditioned on emotional states. Using the YM2413-MDB dataset of annotated chiptune music, we map emotions into Russell's circumplex model (valence-arousal space) and assess the ability of three models—vanilla PatchTST, vanilla LSTM, and our hybrid architecture—to generate emotion-aligned music. Evaluation metrics include melodic coherence, rhythmic stability, harmonic richness, structural complexity, and a custom Emotion Alignment Score. Experimental results show that while the hybrid PatchTST-LSTM model achieved competitive performance, the vanilla LSTM slightly outperformed it in both validation loss and emotional alignment. The findings suggest that recurrent models remain highly effective for short symbolic music sequences, while Transformer-based approaches may require more complex datasets or longer compositions to demonstrate advantages. We discuss limitations of emotion encoding, evaluation methods, and dataset size, and outline directions for future research. Code is available at https://github.com/qwirty123/PatchTST-LSTM.**

***Index Terms*—Music Generation, Deep Learning, Emotion-Conditioned Generation, Patch Time Series Transformer (PatchTST), Long Short-Term Memory (LSTM), Chiptune, Symbolic Music, Circumplex Model**

I. INTRODUCTION

Significant advancements have been made in recent years in the applications of Artificial Intelligence (AI) in creative domains. In particular, automated music composition has undergone considerable evolution with the advent of deep learning models, enabling the creation of complex musical compositions. AI simplifies the process by automating compositional patterns, guided by both data-driven trends and underlying music theory.

Early successes in the field of AI music generation were primarily driven by Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) networks, which proved capable of capturing the temporal dependencies inherent in musical sequences, such as melodies and rhythms.

More recently, Transformer architectures have demonstrated the ability to handle long-range dependence and facilitate parallel processing, making them increasingly prominent in music generation. The progression to Transformers has allowed AI to capture complex, long-range dependencies, enabling the use of such models in more applications requiring real-time generation. Transformers, utilizing an attention and self-attention mechanism, excel at modeling sequential data by creating connections between elements across a sequence, regardless of temporal distance.

However, the initial application of transformers yielded notably weak performance, often being outperformed by simple autoregressive models due to the fact that while its self-attention mechanism excelled at correlating elements in a sequence, the permutation-invariant behavior of vanilla transformers proved insufficient as the temporal order of elements is often crucial in time-series applications [1].

This challenge was effectively addressed by PatchTST (Patch Time Series Transformer), which introduced an efficient design specifically for multivariate time series forecasting and representation learning. Its core innovations include segmenting time series into subseries-level patches and adopting a channel-independent architecture [2]. These adaptations are particularly valuable for capturing both local patterns and long-range dependencies, making PatchTST a compelling option for complex time-series tasks like music generation.

A key challenge in generating music lies in maintaining long-term coherence, a necessity for applications such as soundtracks that must play continuously or evolve over extended sessions. PatchTST offers a promising alternative by directly addressing the limitations of LSTMs and RNNs in long-range dependence, while also improving upon transformers by introducing manageable, subseries-level patches. These traits make PatchTST a promising approach for complex time-series tasks like music generation.

II. LITERATURE REVIEW

Music generation can be generally divided into audio-based and symbolic-MIDI based format. Although direct audio generation represents music more exactly without losing information due to playing back symbolic representations, MIDI based symbolic representations allow for more granular control over attributes like tempo, pitch, duration, and rhythm due to its ability to easily be vectorized [3].

Much research has been conducted on applying artificial neural networks to symbolic music generation. Early RNN based models [4] utilized deep recurrent neural networks with a bidirectional modeling approach to capture harmonic patterns. LSTM based RNN models like Performance RNN [5] improved on earlier sequence learning methods due to its

relative insensitivity to gap length. MusicVAE was among the first to apply a Variational Autoencoder to music. Its key innovation was creating a continuous latent space of melodies, demonstrating that music could be represented and manipulated as a high-dimensional vector [6]. MIDI-VAE utilized VAEs to generate music with a different input format, incorporating both instrumentation and music dynamics to show that these generative models could be applied effectively to different musical representations to mimic the structure of musical pieces and produce new, stylistically consistent compositions [7]. These early works laid the foundation for the transformer-based approaches that followed.

Developments in the Transformer architecture, namely its self-attention mechanism [8], were applied to music generation to capture long-term dependencies in a musical piece [9], enabling the creation of music with a more cohesive and structured form than previous models, which often struggled with extended compositions. Transformer-XL builds on the standard Transformer by introducing a segment-level recurrence mechanism [10]. This allowed the transformer to reuse hidden states from previous segments, effectively extending its context window and enabling the generation of more coherent, long-form musical compositions. OpenAI's Jukebox used a Vector Quantised-Variational AutoEncoder (VQ-VAE) to compress raw audio into discrete codes, which were then used by a Transformer model to generate long sequences of music. Recently, MusicGen eliminated the need for cascading several models through its ability to condition a single transformer language model on both text and audio tokens [11].

The standard Transformer architecture, which has proven highly successful in natural language processing and computer vision, faces significant challenges when applied to time series forecasting due to the permutation invariant behavior of the self-attention mechanism. Furthermore, the original mechanism's quadratic time complexity with respect to input length is computationally expensive [1]. PatchTST addresses these limitations using its patching mechanism and channel-independent design. The patching mechanism divides the raw time series data into overlapping or non-overlapping segments, or patches. By analyzing patches rather than individual points, the model can consider local information within each patch before considering global dependencies while reducing computational complexity. For multivariate time series, PatchTST trains a single model for all variables, which are processed independently [2].

A hybrid PatchTST-LSTM music generation model could have promising applications in video game soundtracks, which requires changes in emotion but also long term stability across long sessions. Multiple algorithms and architectures have been employed in the specific niche of emotion based music generation in video games including Markov chains, RNNs, LSTMs, and Transformers. Early work largely utilised stochastic models to analyze rhythmic patterns from existing pieces to generate similar but novel patterns. Hierarchical Markov models proved to be able to generate original music in the style of a training piece in real time while requiring little processing power. [12] Further research focused on Recurrent Neural Networks (RNNs). Such models include Hutchings and McCormack's Adaptive Music System [13], which proposed a system based on a multi-agent algorithmic music composition and arranging system with separate harmony, melody, and percussive RNN agents combined with a knowledge activation model. More recently, LSTMs and Transformer models have gained prominence in video game music generation as both architectures offer better long-range dependencies in musical sequences. LSTM based soundtrack generation models have been tested on chiptunes [14] and various transformer architectures have been proposed [15] [16] and tested as applications of models such as MusicGen [17].

III. Methodology

A. Dataset

We chose to train our model with the YM2413-MDB dataset, a well annotated collection of 1980s games chiptune music [18]. This dataset was chosen due to its extensive labeling with emotion states, symbolic midi formatting, and use in previous works [14]. Furthermore, the short chiptune soundtracks are far less computationally expensive than expansive datasets.

B. Russell's circumplex model

To effectively generate music that aligns with specific emotional states and evaluate the long term music features, we employ Russell's circumplex model. Russel's model maps emotions to vectors in a 2D emotion space with axes representing valence and arousal [19].

1) Valence: representing the positive or negativity of the emotion or pleasure-displeasure, with higher values representing more positive emotions.

2) Arousal: representing the energy of the emotion, with higher values representing excitement and lower, negative values representing sleepiness.

We mapped the emotions annotated in the YM2413-MDB dataset to set vectors on the circumplex model as illustrated in Figure 1 based on quadrants described in the dataset [18] and existing literature [20] [14].

IV. Model Architecture

We propose a novel hybrid deep learning architecture. Our model combines a Patch Time Series Transformer (PatchTST) with a Long Short-Term Memory (LSTM) network. This dual-component design delegates distinct, complementary responsibilities: the PatchTST is tasked with establishing the high-level musical structure and long-range dependencies, while the LSTM focuses on generating the detailed, note-level symbolic music. The model's pipeline follows a sequential, multi-stage process. First, an initial musical seed sequence and a two-dimensional emotion vector are fed into the PatchTST component. This module processes the input in discrete patches to generate a structurally coherent but high-level musical outline. This output then serves as the input for the

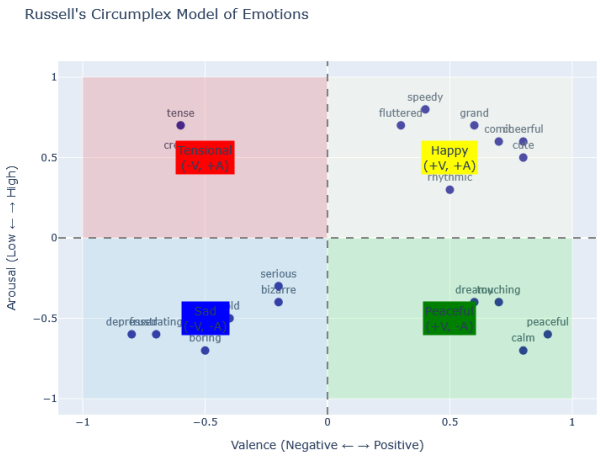


Fig. 1. Corresponding Valence and Arousal values for emotion annotations in YM2413-MDB

LSTM detail generator, which, also conditioned on the emotion vector, populates this structure with detailed, sequential note information. Finally, a fusion layer integrates the outputs from both components to produce the final, rich chiptune audio sequence. This hierarchical approach ensures that the generated music is not only locally coherent but also possesses a globally consistent and logical structure, all while adhering to the specified emotional context.

A. PatchTST

The primary role of the MusicalPatchTST is to learn and generate the macroscopic structure of a musical piece. It leverages the core strengths of the Transformer architecture to capture long-range dependencies across the temporal dimension of the music. To adapt the Transformer for this task, we employ a patching mechanism, which is a key feature of the PatchTST model. The input musical sequence, which comprises data for 15 distinct instrumental tracks, is first divided into a series of non-overlapping patches.

A critical aspect of our implementation of PatchTST is the principle of channel independence. We treat each of the 15 tracks as an independent channel. The patching process is applied individually to each channel, preserving the unique melodic and rhythmic integrity of each instrument. This prevents the model from conflating the distinct musical lines and allows it to learn the characteristic patterns of each instrument separately.

Once patched, each segment is flattened and projected into a high-dimensional embedding space via a linear layer. As the Transformer architecture is inherently permutation-invariant, positional encoding is added to these embeddings to supply the model with information about the relative order of the patches.

To infuse the generative process with emotional context, the two-dimensional emotion vector (valence, arousal) is projected to match the dimension of the patch embeddings. This projected emotion vector is then added to each patch embedding. This step informs the model's initial understanding of the musical structure, guiding the subsequent generation to align with the desired emotional target.

The conditioned patch embeddings are then processed by a multi-layer Transformer encoder. Operating independently on each channel, the encoder's self-attention mechanism enables the model to weigh the significance of different patches relative to one another. This is where the model learns the long-range dependencies. For instance, how a melodic idea introduced in an early patch should relate to a harmonic progression in a later one.

After passing through the Transformer encoder layers, the processed patch embeddings are projected back to their original flattened patch dimension. Finally, these processed patches are concatenated in their original order to reconstruct the full-length musical sequence for each channel. The output of this stage is a musical sequence that possesses a coherent, high-level structure but lacks fine-grained detail.

B. LSTM Detail generator

While the PatchTST component provides the structural skeleton, the LSTM based detail generator is responsible for adding the note-by-note details. It leverages the inherent strengths of LSTMs in modeling sequential data to generate locally coherent and melodically plausible passages.

The LSTM takes the structurally-aware sequence generated by the PatchTST component as its primary input. This ensures that the details it generates are grounded in the global structure established by the Transformer. The model is composed of a multi-layered LSTM network that processes the sequence for each of the 15 channels independently. The LSTM layers capture the local dependencies between adjacent notes, which is essential for generating smooth melodic lines and consistent rhythmic patterns.

Similar to the PatchTST component, the LSTM is also conditioned on the same emotion vector. This ensures that the details, such as variations in timing or velocity, are congruent with the overarching emotional theme. Furthermore, our LSTMDetailGenerator incorporates a multi-head attention mechanism. This allows the LSTM, at each step of its generation process, to selectively focus on the most relevant segments of the structural input from the PatchTST. This helps to maintain alignment between the generated details and the foundational structure.

C. Hybrid Fusion Layer

The final stage of our architecture resolves the architectural tension between the global structure provided by the PatchTST module and the local detail generated by the LSTM detail module. To combine the two layers while preserving high-level musical form and low-level melodic nuance, we introduce a dedicated fusion layer, implemented as a small multi-layer perceptron (MLP), which learns to intelligently synthesize the two streams into a single, cohesive output.

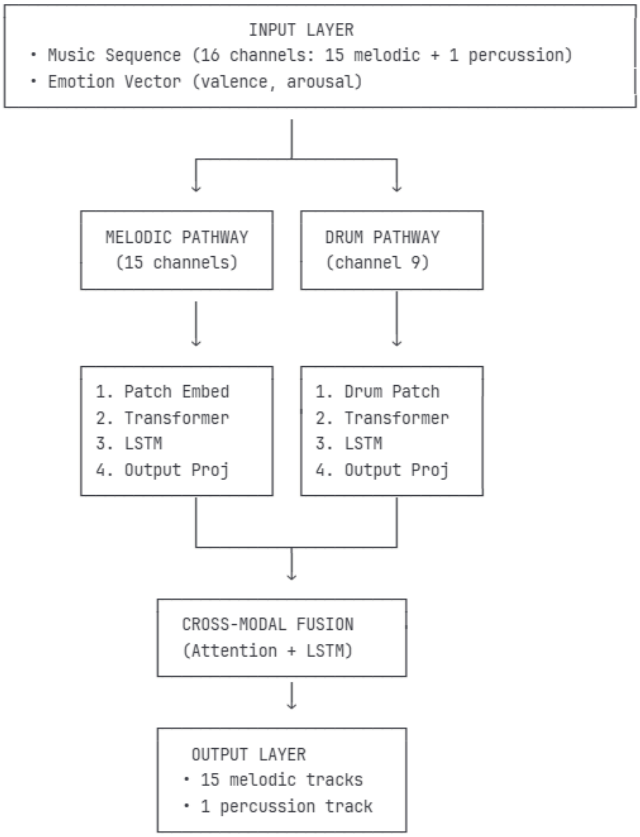


Fig. 2. System Diagram for hybrid PatchTST+LSTM architecture

For each step in the sequence, the feature vector from the structure-generator is concatenated with the corresponding feature vector from the detail generator. This creates a combined 8-dimensional feature vector that serves as the input to the fusion layer. A Rectified Linear Unit (ReLU) activation function and dropout layer is used to enable the model to learn conditional relationships between the structural and detail features. For example, the hybrid fusion layer can learn to prioritize the LSTM's output for melodic passages while emphasizing the PatchTST's output for rhythmic sections.

Through training, this layer discovers the optimal method for blending the global structure with local detail, resulting in a musically coherent output that retains the strengths of both the Transformer and LSTM architectures.

V. Experimental Investigation

To determine the efficacy of our hybrid PatchTST+LSTM model, we compare our hybrid model with a vanilla PatchTST model and a vanilla LSTM model. The PatchTST and LSTM models used for comparison are given in the appendix. All models were trained and evaluated on the YM2413-MDB, a dataset of chiptune music annotated with emotional labels corresponding to Russell's Circumplex Model of Affect This allows us to test each model's ability to generate musical sequences conditioned on specific emotional targets.

A. Experimental Setup

All models were implemented in PyTorch. For training, we utilized the AdamW optimizer. The models were trained for 50 epochs with a batch training size of 8 and validation batch size of 16. Three loss functions were used.

1) Mean square root was used for the melodic loss function, given by

$$L_{melodic} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{1}$$

2) Binary cross entropy was used for percussion instruments, given by

$$L_{drums} = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{2}$$

3) Mean absolute error for temporal loss, given by

$$L_{temporal} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3}$$

B. Evaluation Metrics

For evaluation, we employed a suite of objective metrics designed to assess different facets of musical quality. These metrics analyze the generated musical sequence for coherence, complexity, and structure. The key metrics include:

1) Melody Score: This is a composite score derived from two sub-metrics

a) Pitch Consistency: Measures the standard deviation of pitches among active notes. A lower deviation (higher score) indicates a more stable and less erratic pitch palette.

b) Melodic Contour: Assesses the smoothness of melodic lines by calculating the average interval size between consecutive notes. Smaller average intervals yield a higher score, reflecting more stepwise and vocally-inspired melodic motion.

2) Rhythm Score: This score combines two metrics to evaluate rhythmic coherence:

a) Duration Consistency: Measures the standard deviation of note durations, where a higher score signifies a more stable rhythmic feel.

b) Rhythm Regularity: Calculated from the standard deviation of inter-onset intervals (the time between consecutive note starts). A higher score indicates a more consistent and predictable beat.

3) Harmony Score: This metric quantifies the harmonic richness by calculating the proportion of timesteps in which multiple notes (i.e., chords) are played simultaneously. A higher score suggests a greater presence of harmonic content.

4) Overall Structure Score: This provides a measure of the composition's overall form and complexity by averaging three distinct features:

a) Note Density: The number of notes per time unit, acting as a proxy for the music's "busyness" or complexity.

b) Dynamic Range: The difference between the highest and lowest MIDI velocities, reflecting the piece's expressive variance.

c) Harmony Score: As described above, included here to contribute to the structural assessment.

5) Emotion Alignment Score: To quantify how well a generated piece reflects the target emotion, we developed a metric that maps musical features to the valence-arousal space. It uses note density as a proxy for arousal (higher density implies higher arousal) and average pitch height as a proxy for valence (higher pitches imply higher valence). The Euclidean distance between the musical features' implied emotion and the target emotion vector is calculated and converted into a similarity score from 0 to 100.

Fig. 3. Sample of music generated with hybrid PatchTST+LSTM

VI. Results

To assess the performance of the three models, we analyzed both their training behavior and their ability to generate emotionally-aligned music.

As shown in Figure 5, the training and validation loss curves reveal significant differences in learning capability. Both the vanilla LSTM and the hybrid PatchTST+LSTM model achieve low validation loss, indicating that they successfully learned the underlying patterns in the chiptune dataset. The pure LSTM model marginally outperforms the hybrid model, achieving the lowest overall validation loss. In contrast, the vanilla PatchTST model's validation loss remains substantially higher, suggesting it struggled to effectively model the sequential nature of the musical data on its own.

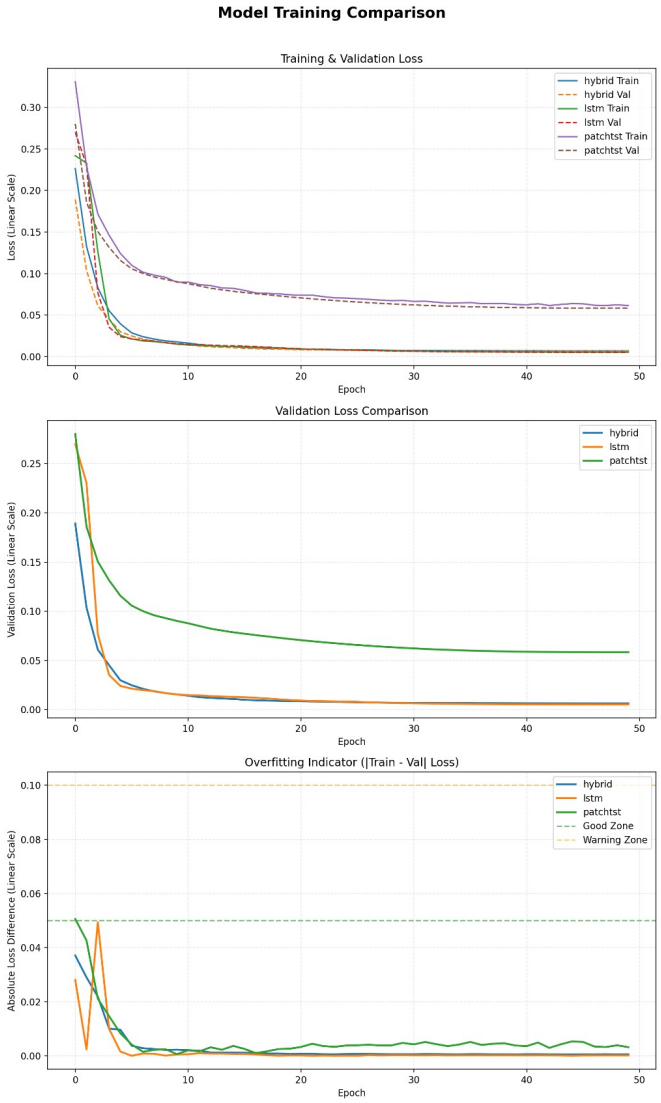
The primary goal of our investigation was to assess how well each model could generate music that aligns with a target emotion. Figure 4 shows the loss components of the three models. The vanilla LSTM and the hybrid model both achieved high alignment scores. The vanilla PatchTST model consistently scores the lowest, indicating its output is less aligned with the intended emotion. Crucially, in this instance, these results show that the addition of a PatchTST component in our hybrid model offered no significant benefit to emotional alignment compared to the vanilla LSTM.

VII. Discussion and Further Research

Our experimental results lead to several key conclusions for the task of emotion-conditioned music generation.

First, the vanilla LSTM model proved to be effective, outperforming the more complex PatchTST and performing on par with, or even marginally better than, the hybrid model. This suggests that the sequential patterns and temporal dependencies in chiptune music are sufficiently captured by the recurrent mechanisms of an LSTM. Chiptune music, often characterized by strong melodic lines and arpeggiated chords, may rely more on local, note-to-note relationships than on the very long-range structural dependencies that Transformers are designed to capture.

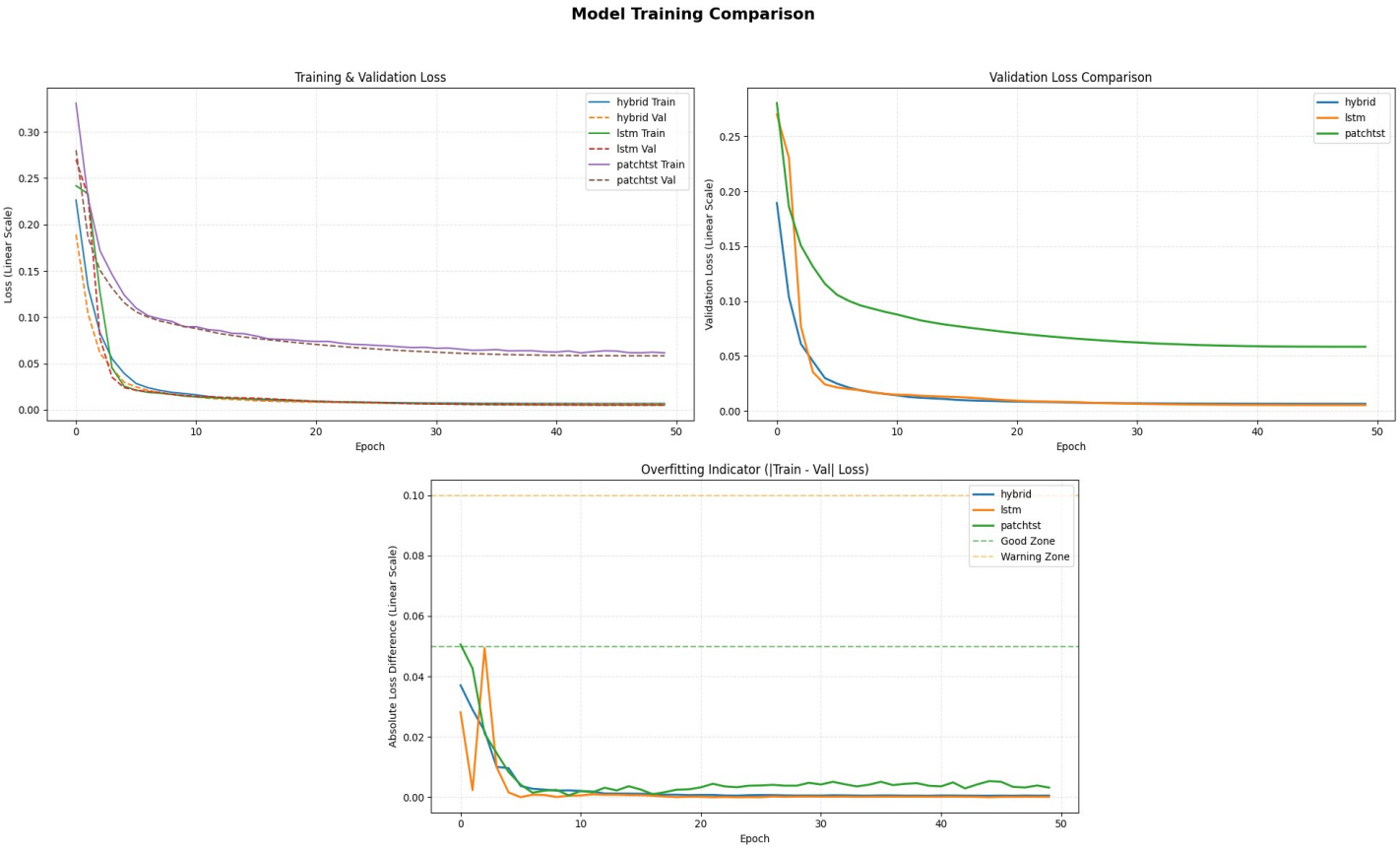
Fig. 4. Melodic, Drums, and Temporal loss components for Hybrid, LSTM, and PatchTST



Second, the PatchTST model, while powerful for other time-series tasks, was not well-suited for this generative music application in its vanilla form. Its struggle, evidenced by the high validation loss and low emotional alignment, indicates that simply treating music as a standard time-series fails to capture the nuances required for coherent generation.

The hybrid model performed marginally worse than the pure LSTM in terms of validation loss and showed no improvement in emotional alignment. This implies that the high-level structural representation generated by the PatchTST did not provide a meaningful advantage for the subsequent LSTM generation step. It is possible that for this specific domain, the structural guidance from the PatchTST was either redundant to what the LSTM could learn on its own or introduced a level of abstraction that was not beneficial for fine-grained note generation.

Fig. 5. Training and Validation Loss for the Vanilla LSTM, Vanilla PatchTST, and Hybrid models.



It is important, however, to consider the limitations of this study and point toward promising directions for future research.

A. Dataset and Emotion Encoding

In encoding the 19 annotated emotions present in the dataset to points in emotion space, we considered three approaches.

1) Dominant: Using the first, dominant emotion label mapping to a particular point in emotion space

2) Linearly Weighted: Using a linearly weighted average of the coordinates for the n emotion labels for each

3) Exponential Weighted: Using an exponentially weighted average of the coordinates for the n emotional labels for each dataset

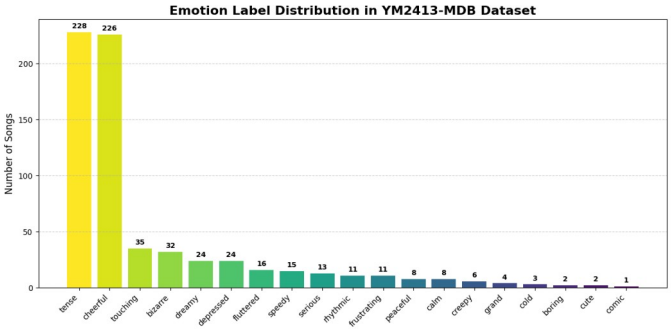
In this study, we simply utilised the first option. However, using a weighted average of the coordinates with the dominant emotion label weighted exponentially more than the secondary labels may provide better precision by considering the secondary emotion annotations.

As shown in figure 6, by only considering the dominant emotion, the annotations in YM2413-MDB were heavily weighted towards two emotions: tense and cheerful. A larger training dataset could produce better emotional alignment for pieces not categorized as cheerful or tense.

Ultimately, our findings are specific to the domain of relatively simple chiptunes from the YM2314-MDB dataset. This genre is often characterized by limited polyphony and straightforward melodic structures. It is plausible that the architectural benefits of the hybrid model would only become apparent when applied to more harmonically and structurally complex music. Further research should therefore explore applying these models to different training datasets to reveal whether the conclusions drawn here are generalizable or specific to the chiptune domain, and under what conditions of musical complexity a hybrid approach is warranted.

Fig. 6. Dominant emotion distribution for YM2413-MDB dataset

B. Emotional Alignment Evaluation



A key limitation lies in our Emotion Alignment Score, which relies on simplified heuristics like note density and pitch to assess emotion. While these are common proxies, they do not capture the full complexity of musical emotion, which is also conveyed through harmony, rhythm, timbre, and mode. This simplification may have prevented us from observing more subtle emotional differences in the models' outputs. Therefore, further research is needed to develop and integrate more sophisticated evaluation methods. Employing evaluation models trained on human perceptual data or conducting subjective listening tests with human participants would provide a more nuanced and accurate assessment of emotional alignment.

C. Length of Compositions

The theoretical advantage of Transformer-based models like PatchTST typically emerges when modeling very long-range dependencies. The musical sequences used in our experiments may not have been sufficiently long to necessitate the global structural planning at which Transformers excel. For shorter pieces, an LSTM's strong ability to model local coherence can be sufficient. A clear avenue for future work is to test these architectures on tasks requiring the generation of significantly longer and more structured compositions, such as multi-movement pieces with recurring motifs.

V. Conclusion

This work investigated emotion-conditioned chiptune music generation by comparing three deep learning architectures: a vanilla LSTM, a vanilla PatchTST, and a novel hybrid PatchTST-LSTM model. Our experiments on the YM2413-MDB dataset showed that the vanilla LSTM achieved the best performance, with the lowest validation loss and highest emotion alignment scores. These results suggest that for short musical sequences like chiptunes, simpler recurrent architectures can outperform more complex Transformer-based models. The strong melodic patterns and local dependencies in chiptune music are effectively captured by LSTMs, while the Transformer's advantages in long-range modeling were not necessary for this task. While the hybrid architecture successfully combined both approaches, it did not improve upon the vanilla LSTM. This highlights an important lesson in machine learning: more complex models are not always better, and the architecture should match the specific requirements of the task.

Our study has several limitations. The YM2413-MDB dataset showed significant imbalance, with 77% of samples in just two emotion categories. The emotion alignment metric, while objective, uses simplified heuristics. Additionally, the short length of chiptune pieces may not showcase the full capabilities of Transformer models.

Future work should test these architectures on longer, more complex musical compositions to determine when Transformer-based approaches become advantageous. More sophisticated emotion evaluation methods, including human listening studies, would provide better assessment of emotional alignment.

Finally, exploring different emotion encoding schemes could improve the precision of emotion conditioning. This research demonstrates that LSTM-based models remain highly effective for real-time music generation in applications like video game soundtracks, where computational efficiency and emotional responsiveness are crucial.

As AI-generated music continues to evolve, understanding which architectures work best for different musical tasks will be essential for creating practical systems that can adapt music to user preferences and context.

References

[1] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are Transformers Effective for Time Series Forecasting?," Aug. 17, 2022, *arXiv*: arXiv:2205.13504. doi: 10.48550/arXiv.2205.13504.

[2] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A Time Series is Worth 64 Words: Long-term Forecasting with Transformers," Mar. 05, 2023, arXiv: arXiv:2211.14730. doi: 10.48550/arXiv.2211.14730.

[3] R. Mitra and I. Zualkernan, "Music Generation Using Deep Learning and Generative AI: A Systematic Review," IEEE Access, vol. 13, pp. 18079–18106, 2025, doi: 10.1109/ACCESS.2025.3531798.

[4] G. Hadjeres, F. Pachet, and F. Nielsen, "DeepBach: a Steerable Model for Bach Chorales Generation," June 17, 2017, *arXiv*: arXiv:1612.01010. doi: 10.48550/arXiv.1612.01010.

[5] S. Oore, I. Simon, S. Dieleman, D. Eck, and K. Simonyan, "This time with feeling: learning expressive musical performance," *Neural Comput & Applic*, vol. 32, no. 4, pp. 955–967, Feb. 2020, doi: 10.1007/s00521-018-3758-9.

[6] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music," Nov. 11, 2019, *arXiv*: arXiv:1803.05428. doi: 10.48550/arXiv.1803.05428.

[7] G. Brunner, A. Konrad, Y. Wang, and R. Wattenhofer, "MIDI-VAE: Modeling Dynamics and Instrumentation of Music with Applications to Style Transfer," Sept. 20, 2018, *arXiv*: arXiv:1809.07600. doi: 10.48550/arXiv.1809.07600.

[8] A. Vaswani *et al.*, "Attention Is All You Need," Aug. 02, 2023, *arXiv*: arXiv:1706.03762. doi: 10.48550/arXiv.1706.03762.

[9] C.-Z. A. Huang *et al.*, "Music Transformer," Dec. 12, 2018, *arXiv*: arXiv:1809.04281. doi: 10.48550/arXiv.1809.04281.

[10] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context," June 02, 2019, *arXiv*: arXiv:1901.02860. doi: 10.48550/arXiv.1901.02860.

[11] J. Copet *et al.*, "Simple and Controllable Music Generation," Jan. 30, 2024, *arXiv*: arXiv:2306.05284. doi: 10.48550/arXiv.2306.05284.

[12] S. Engels, T. Tong, and F. Chan, "Automatic Real-Time Music Generation for Games," *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 11, no. 1, pp. 220–222, 2015, doi: 10.1609/aiide.v11i1.12775.

[13] P. E. Hutchings and J. McCormack, "Adaptive Music Composition for Games," *IEEE Transactions on Games*, vol. 12, no. 3, pp. 270–280, Sept. 2020, doi: 10.1109/TG.2019.2921979.

[14] V. E. Qi Leo, S. H. Bte Haniffah, X. Teo, B. Y. Xiao Tian, and N. H. Loong Wong, "Dynamic Automatic Chiptune Generation for Game Music," in *2024 4th International Conference on Robotics, Automation and Artificial Intelligence (RAAI)*, Dec. 2024, pp. 231–235. doi: 10.1109/RAAI64504.2024.10949540.

[15] G. Amaral, A. Baffa, J.-P. Briot, B. Feijó, and A. Furtado, "An adaptive music generation architecture for

games based on the deep learning Transformer model," in *2022 21st Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*, Natal, Brazil: IEEE, Oct. 2022, pp. 1–6. doi: 10.1109/SBGAMES56371.2022.9961081.

[16] F. Zumerle, L. Comanducci, M. Zanoni, A. Bernardini, F. Antonacci, and A. Sarti, "Procedural music generation for videogames conditioned through video emotion recognition," in *2023 4th International Symposium on the Internet of Sounds*, Oct. 2023, pp. 1–8. doi: 10.1109/IEEECONF59510.2023.10335439.

[17] F. Marra and L. N. Ferreira, "Long-Form Text-to-Music Generation with Adaptive Prompts: A Case Study in Tabletop Role-Playing Games Soundtracks," May 21, 2025. doi: 10.5281/zenodo.14908040.svg.

[18] E. Choi, Y. Chung, S. Lee, J. Jeon, T. Kwon, and J. Nam, "YM2413-MDB: A Multi-Instrumental FM Video Game Music Dataset with Emotion Annotations," Nov. 14, 2022, *arXiv*: arXiv:2211.07131. doi: 10.48550/arXiv.2211.07131.

[19] J. A. Russell, "A circumplex model of affect.," *Journal of Personality and Social Psychology*, vol. 39, no. 6, pp. 1161–1178, Dec. 1980, doi: 10.1037/h0077714.

[20] K. Zheng *et al.*, "EmotionBox: A music-element-driven emotional music generation system based on music psychology," *Front. Psychol.*, vol. 13, Aug. 2022, doi: 10.3389/fpsyg.2022.841926.