T5-Harmony: Fine-Tuning a Large Language Model to Understand and Generate Symbolic Music

Warren Kozak
Carleton College
Northfield, MN
kozakw@carleton.edu

Omar Sobhy
Carleton College
Northfield, MN
sobhyo@carleton.edu

Abstract

Encoder–decoder models, commonly referred to as *seq2seq* models, are highly capable of correlating input sequences with appropriate outputs. We explore this capability in the context of music to determine whether these abilities extend to musical understanding. We fine-tuned two T5-base models on a corpus of more than 30,000 songs encoded in a textual representation from Hooktheory. The first, a melody model, generates melodies from chord progressions, while the second, a chords model, produces harmonizing chord progressions from input melodies. To evaluate these models, we conducted both quantitative and qualitative analysis methods. Our models achieve up to 100% bar-sequence correctness and 92.9% bar-duration accuracy at low temperature, and human listeners judged the outputs to be melodious (4.5/5) and stylistically coherent. Musically, the models predominantly produce diatonic harmonies and coherent melodic contours, and higher sampling temperatures (0.9) enable stylistic variation and extended chordal structures in accordance to genre conditioning. Our results show that the models capture fundamental relationships between harmony and melody and can produce musically coherent outputs, although they still struggle with stylistic consistency. These findings suggest that transformer based encoder-decoder architectures can generalize aspects of musical understanding from large-scale symbolic datasets and offer a novel music generation model.

Keywords: Machine Learning, Artificial Intelligence, Large Language Models, Music, Harmony, Melody, MIDI, T5

1 Introduction

Encoder-decoder transformer models excel at a wide variety of seq2seq tasks, and are especially used for machine translation [1]. The encoder portion of these models takes in an input sequence, and utilizes the traditional layered transformer architecture to output a fixed-length vector representation of the input sequence that captures its important information. The decoder portion of the model then functions as a conditional generation language model. It utilizes a cross-attention layer to attend to the vector representation of the source text to generate an output sequence. By leveraging the power of transformers to encode long-term dependencies in text, and cross-attention to map the input to the output information, encoder-decoder transformer models are highly successful at seq2seq tasks.

In particular, the T5 series of language models developed by Google showed state-of-the-art performance and efficiency on a wide variety of seq2seq tasks upon its release [2]. Although newer models can now outperform T5 on certain tasks, such as decoder-only models, T5 remains a strong contender in the present day for seq2seq tasks. Due to its high performance, open-source availability, and relatively small size, we picked this model to fine-tune for our specific chord and melody generation task.

In this work, we ask whether the same mechanisms that enable T5 to translate natural language can be applied to translating between melodies and harmonizing chord progressions. We make three contributions. First, we introduce a symbolic representation for melody–harmony pairs based entirely on textual tokens to enable transformer-based encoder–decoder models to process musical information in a way that is analogous to natural language. Second, we fine-tune two T5-base models on more than 30,000 lead sheets from the Hooktheory dataset: one model generates melodies conditioned on chord progressions, and the other generates chord progressions conditioned on melodies. Third, we evaluate these models using both quantitative structural metrics (bar alignment, duration correctness, rhythmic correctness, chord validity) and a qualitative listening survey that assesses musicality and perceived creativity. These contributions show that encoder–decoder transformers can generalize meaningful aspects of melodic and harmonic structure within a symbolic way of representing music.

2 Related Work

With transformer-based models proving to be able to form complex representations of written text, many recent approaches to symbolic music understanding have attempted to use transformers to varying degrees of success. We present a few previous researchers' approaches to treating chord and melody generation as a seq2seq task, as well as outline their overall approaches and results.

2.1 Other Music Generation Model Architectures

Music generation models relying on audio generation have been the prevalent music machine learning solutions. One particularly notable example is MusicGen, created by a team

at Facebook, which uses an approach rooted in language modeling to operate over streams of compressed, discrete audio packets [3]. This approach utilizes an autoregressive transformer-based decoder to achieve conditional generation on a text or melody input. Another one is Suno, a commercial, closed source model that shows public interest in generative music AI, though no technical details about its architecture have been published [4].

2.2 Early Transformer Approaches to Music

A significant early contribution to transformer-based music generation came from Huang et al. with the Music Transformer, which was trained on MIDI representations of music [5]. The Music Transformer utilized the self-attention mechanism in a way that adapts it for musical sequences. This allowed it to handle the necessary long context windows when it comes to music generation in addition to handle repeating phrases (motifs). This was one of the first machine learning models to successfully generate coherent musical performances.

2.3 Transformer-Based Chord Generation Model

One attempt by Li et al. [6] to train an encoder-decoder model for chord generation inspired us to attempt to represent notes and chords symbolically. Many previous models that have been trained on music used MIDI representations, which are undecipherable to the human eye, and significantly differ from how natural language is symbolically represented. In this paper, Shuyu et al. outline their approach to representing music symbolically, pre-training an encoder on their symbolic music corpus, and generating novel chord progressions using a decoder.

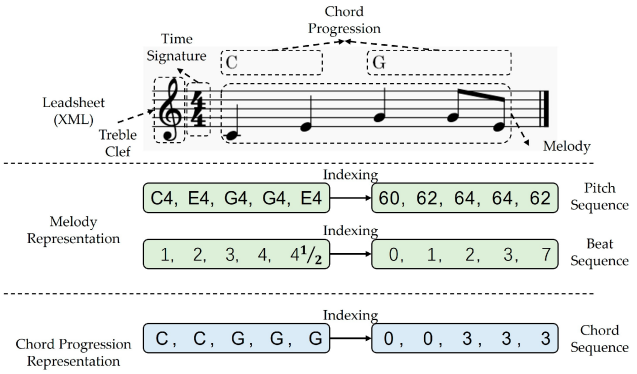

Figure 1. Symbolic representation of music created by Shuyu et al.

To represent their data symbolically, Shuyu et al. came up with a standardized notation for chords and notes. The authors decided to adopt an index-based approach, where melodies, the melodies' beat sequences, and the corresponding chord progressions are denoted as a series of indices that correspond to a symbolic melody, beat, or chord. The melodies are represented as notes followed by their corresponding octave, which corresponds one-to-one-to the beat sequence, indicating at what time each note begins to play. 46 unique beats were identified in their dataset. In their corpus, 428 unique chords were identified, including triads, seventh chords, and ninth chords, and are classified as major, minor, diminished, and augmented. This encoding scheme was greatly influential when we were selecting one of our own.

During the pretraining phase of their work, Shuyu et al. devised several noising algorithms to corrupt their training data. This is similar to the training process of BART, allowing the encoder to learn the process of filling in missing notes in order to gain a deeper understanding of musical context [7].

Shuyu et al. used the *Hits@k* method to evaluate their proposed model, finding that it outperforms BERT, GPT2, and an LSTM-based approach.

3 Dataset

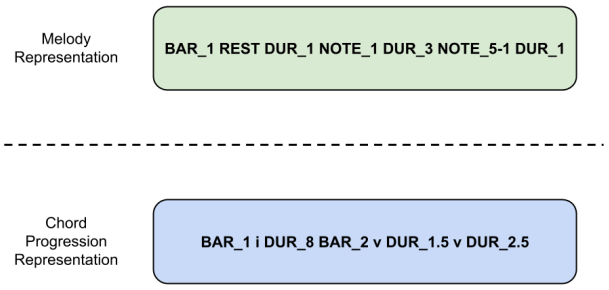

Figure 2. Our symbolic music representation format.

During the fine-tuning process, we used the Hooktheory Lead Sheet Dataset [8]. Hooktheory-related datasets are often used in machine learning research pertaining to music [9]. Hooktheory is a platform that allows users to create compositions by providing tools and a large database of popular songs. It features an extensive, crowd-sourced library of song analyses with chords and melodies. The combination of both the chords and melodies is commonly referred to as a lead sheet, where both are transcribed by Roman numerals and scale degrees rather than absolute pitches. Hooktheory includes about 30,000 song segments in a variety of genres—though most genres skew towards pop and rock music [10]. These song excerpts vary in length as Hooktheory does not impose length limits; however, the average length per excerpt is eight bars. In 2022, Wang et al. found that including the Hooktheory Lead Sheet Dataset in training can improve state-of-the-art boundary detection for audio segmentation tasks [11].

While Hooktheory does not provide an official API for data access, there is an internal API for accessing lead sheets

formatted as JSON files. As part of previous research that has used the Hooktheory Lead Sheet Dataset, web scraping tools have been implemented in the past to download and sort these files [12]. However, due to recent Hooktheory website updates, these tools no longer function. In order to get our annotated dataset, we modified the script developed by Yeh et al. This modified script has been provided as part of our submitted code files.

After retrieving the JSON files for each song, we had to process these and turn the musical information represented in the JSON into our desired data scheme for model fine-tuning. This scheme is shown in **Figure 2**.

When determining a symbolic way to represent the melodic and chord data, we deliberately opted for a text-based representation over the industry-standard MIDI format. We hypothesized that treating music as a textual language would allow the model to leverage its inherent NLP capabilities more effectively. Additionally, a text-based approach simplifies the input by abstracting away the complex control messages found in MIDI (such as velocity and pitch bends). This allows the model to focus strictly on composition. We also believe this approach presents a novel research direction, as text-based generation is significantly less explored than MIDI generation.

With this approach in mind, we wanted to ensure that our notation was both complete enough to capture some of the complexities of music, but also simple enough so that the encoder would not struggle to learn patterns across melodies and chords. As shown in **Figure 2**, we represent individual notes in the melody as a single token, denoted by the position of the note in the scale, followed by a duration token, which specifies for how many beats the note plays. Similarly, the chords are denoted by their Roman numeral representation, again followed by a duration token specifying how long they last. Throughout both the melody and chord sequences, we include bar tokens specifying the beginning of each sequential bar, giving the encoder an absolute reference for the time between the melody sequence and chords sequence.

Two additional tokens were prepended to the chords model's input (melody): rhythm and genre. The genre was fetched directly from that assigned through Hooktheory and takes the form `GENRE_[genre]`. There are 14 distinct genres in the Hooktheory dataset.

We defined rhythm complexity heuristically as follows: if 90% of the chords in the chord progression are four beats long (following the simple 4/4 time signature), the rhythm is considered simple. If 35% to 90% of the chords are four beats long, the rhythm is considered medium. Below 35%, it is considered complex. This adds extra nuance to chord generation and follows our assumption that a chord progression's quality depends not only on its harmonics but also its accompanying rhythm.

Table 1. T5-Base model specifications

Category	Value
Parameters	220M
Model Dimension	768
Attention Heads	12
Encoder Layers	12
Decoder Layers	12

We did not prepend the inputs for the melody model with these tokens. We hypothesize that melodies do not intrinsically depend on genre or rhythm the way chords do. Additionally, quantifying a melody's rhythmic character would require an analysis of its complexity, which is computationally more intensive and deemed out of scope for this research [13].

Once our entire dataset was converted into the above format and each chord-melody pair was written to a single JSONL file, we were ready to move on to the fine-tuning stage. We had a roughly 80/10/10 split for the training, validation, and test splits, respectively.

4 Methods

We initially considered training an encoder-decoder model from scratch, but due to the time and computational limitations, as well as large amount of training data required, we decided to instead fine-tune a pretrained encoder-decoder model. As mentioned in the introduction, we decided on Google's pretrained T5 model for this task. Specifically, we picked the T5-Base model available on Huggingface, which has 220 million parameters. **Table 1** outlines some of the important specifications of the T5-Base model. We also considered T5-Small (60 million parameters) and T5-Large (770 million parameters); however, we decided that T5-Base presented the right trade off between training time and number of parameters.

We implemented the fine-tuning pipeline on Google Colab, through which we got access to an NVIDIA A100 GPU, which significantly sped up the fine-tuning process. To implement the fine-tuning pipeline, we relied on the `transformers` Python library provided by Huggingface, giving us easy access to the model and also providing the `Seq2SeqTrainer` class for easy fine-tuning. We first tokenized each melody-chord pair in the dataset, utilizing the built-in tokenizer provided by the T5 model. Next, we added all unique tokens that appear across the dataset to the models vocabulary, so that the decoder can successfully generate the desired target tokens. Additionally, we resized the model's embedding matrix to account for the new tokens, so that the model can use these new note and chord tokens.

Table 2 outlines the hyperparameters that we defined in `Seq2SeqTrainingArguments` during the fine-tuning of the

Table 2. Fine-tuning hyperparameters

Parameter	T5-Chords	T5-Melody
Epochs	8	8
Batch Size	16	16
Learning Rate	5e-4	5e-4
Weight Decay	0.01	0.01
Optimizer	AdamW	AdamW

T5-base model. We configured our training procedure following established best practices for fine-tuning T5 models on domain-specific tasks. We used the AdamW optimizer [14] with a learning rate of 5×10^{-4}. We applied a weight decay of 0.01 to prevent overfitting, following recommendations from the original BERT paper [15]. We trained for 8 epochs, monitoring validation loss to ensure convergence without overfitting. The loss function used was cross-entropy loss.

Finally, we began the fine-tuning process using our melody-chord pairs. We ran the training twice, one time fine-tuning the model to generate a chord sequence given an input melody sequence, and the second time fine-tuning the model to generate a melody sequence given an input chord sequence. **Figure 3** shows the loss of the T5-Chords model throughout the fine-tuning process, and **Figure 4** shows the loss of the T5-Melody model throughout the fine-tuning process. During the fine-tuning of both of these models, the validation loss stayed consistent with the training loss, showing that the model was not overfitting on the training data, but rather effectively learning the relationships between melodies and chord progressions.

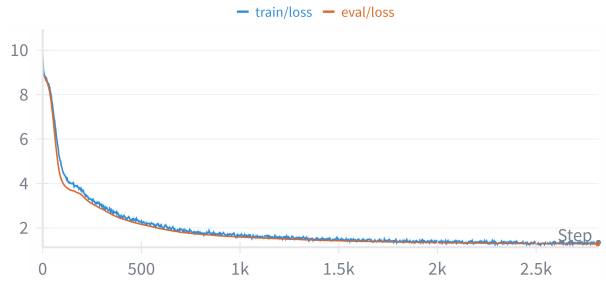

Figure 3. Loss graph of T5-Chords during fine-tuning. The graph displays both the training loss (blue) and the validation loss (orange).

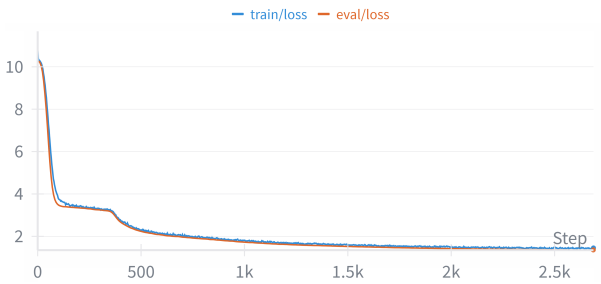

Figure 4. Loss graph of T5-Melody during fine-tuning. The graph displays both the training loss (blue) and the validation loss (orange).

5 Experiments

5.1 Quantitative Evaluation Process

Designing a quantitative metric for evaluating creative tasks like music generation is inherently challenging. Music, by nature, is subjective, and while it can be measured in terms of technical accuracy, such as timing or scale agreement, it is difficult to capture the nuance that comprises good composition. Nevertheless, to assess the performance of our model, we defined two general metrics: bar duration (BD) and correct bar sequence (CBS). Bar duration measures, across all generated outputs from the test set, what percentage of the note or chord durations add up to the correct four beats per bar. Correct bar sequence similarly measure what percentage of bars are in the correct order (i.e. whether BAR_2 follows BAR_1, etc. in the generated output).

Additionally, we defined two metrics specific to our chord generation model: correct rhythm (CR) and valid chords (VC). Correct rhythm measures the percentage of the generated chord sequences that match the rhythm token included in the melody sequence to ensure the rhythmic conditioning token is used, while valid chords assesses the percentage of generated chords that exist within the model's vocabulary; that is, the model does not hallucinate nonexistent chords.

We attempted to find a baseline comparison point for our experiment by passing in the same prompts and test split to the non-fine-tuned T5-Base model and computing the metrics similarly. However, the non-fine-tuned model failed to recognize the structure of our tokens entirely, and it failed to produce any measurable outputs despite multiple attempts and prompt variations—effectively making it have a 0% accuracy for any of our metrics. Consequently, we were not able to meaningfully quantify the performance of the T5-Base model. This serves as evidence that fine-tuning is necessary. The T5-Base model's' failure functions as our baseline comparison.

It is important to note that while these metrics provide insight into structure, they do not measure musicality in any way beyond adherence to rigid patterns within the training split. Additionally, metrics like CR may be biased by uneven dataset distributions.

5.2 Qualitative Evaluation Process

As mentioned in the previous subsection, music is inherently subjective. Therefore, we hypothesize that its primary evaluation technique must be qualitative. Music created by

machine learning solutions specifically requires a more distinctly subjective evaluation approach that employs human listening tests [16]. We created a survey based on some of the recommendations outlined by Chu et al. Additionally, Lerch et al. propose the idea of preference testing—that is, whether or not respondents prefer excerpts generated by machine learning solutions to those created by humans in a blind test [17].

Additionally, we sought to evaluate the creativity of the models' outputs which we hypothesize directly correlates to the models' understanding of the subtlety of music composition. Jordanous proposes what they dub the Standardized Procedure for Evaluating Creative Systems (SPECS) [18]. This provides some techniques for testing creativity that we employ in our survey.

To prepare the audio for these listening tests, we reconstructed MIDI files from the model's text output and synthesized them into piano sound using a script we designed that depends on the python module `music21`. Each excerpt was standardized to 30 seconds in length, with two examples included for each model category.

6 Results

6.1 Example Output

Table 3. Example output of the melody-to-chords model.

Melody: GENRE_POP RHYTHM_COMPLEX BAR_1 REST DUR_1 NOTE_3 DUR_0.5 REST DUR_1.5 NOTE_3 DUR_1 REST DUR_1 BAR_2 NOTE_2 DUR_0.5 NOTE_2 DUR_0.25 NOTE_2 DUR_0.5 NOTE_2 DUR_0.25 NOTE_1 DUR_0.5 NOTE_1 DUR_1 REST DUR_0.5 NOTE_5-1 DUR_0.5
Chords: SCALE_MAJOR BAR_1 I DUR_2 vi7 DUR_2 BAR_2 I DUR_4 BAR_3 IV DUR_2 V DUR_2

Table 4. Example output of the chords-to-melody model.

Input: BAR_1 iii DUR_1.75 IV DUR_3.25 BAR_2 V DUR_0.75 vi DUR_2.25 BAR_3 iii DUR_1.75 IV DUR_3.25
Output: SCALE_MAJOR BAR_1 REST DUR_1 NOTE_3 DUR_2 NOTE_2 DUR_1 NOTE_1 DUR_0.5 NOTE_2 DUR_0.5 BAR_2 NOTE_2 DUR_0.25 NOTE_1 DUR_0.25 NOTE_1 DUR_1.5 NOTE_1 DUR_0.5 NOTE_1 DUR_1 NOTE_2 DUR_1 BAR_3 NOTE_2 DUR_0.25 NOTE_1 DUR_0.25 NOTE_1 DUR_1.5 NOTE_1 DUR_0.5 NOTE_6-1 DUR_0.5 NOTE_2 DUR_0.5

Table 3 shows an example of a chord sequence generated by the model, given the melody sequence as an input. Similarly, **Table 4** shows an example of a melody sequence generated by the model, given the melody sequence as an input.

6.2 Quantitative Evaluation

Table 5 shows the evaluation of the chords to melody model on our test data at four different temperature values. Interestingly, at the lower temperature values of 0.1 and 0.5, the model generated the correct bar sequence 100.0% of the time, with this metric decreasing as the temperature increased. Similarly, the percentage of correct bar durations generated from the test dataset also decreased with the temperature. This makes sense, as increasing the temperature shifts more probability mass to less likely next tokens, which lowers the scores on our defined metrics.

Similarly, **Table 6** and **Figure 5** show the evaluation of the melody to chords model on our test data at the same four temperature values. This table includes the correct rhythm and valid chords metrics, as these were specific to generated chord sequences only. Generally, as the temperature of the model increases, the scores across the board drop, although at varying rates. One notable exception is the correct rhythms metric. As the temperature increases, the percentage of correct rhythms generated increases. This is likely due to the fact that our test dataset contained a disproportionate amount of melodies with the `RHYTHM_COMPLEX` prepended token. As we increased the temperature value, the generated chords and chord durations became more random, which as a byproduct meant that the rhythms of the generated chord sequences became artificially more complex.

Quantitatively, both models demonstrated a strong ability to preserve bar structure and harmonic continuity, especially at low temperatures. For the chords-to-melody model, bar duration accuracy remained above 70% at temperatures 0.1 and 0.5, with a perfect 100% correct bar sequence in both cases. The melody-to-chords model showed similarly strong performance, achieving a 92.9% bar-duration score at temperature 0.1 and maintaining perfect chord validity across all but the highest temperature. As expected, increasing the sampling temperature led to systematic reduction across all structure-based metrics for both models. This happens because a higher temperature encourages the model to generate more unlikely tokens.

6.3 Qualitative Evaluation

6.3.1 Survey Outcomes. We surveyed a small but diverse group of participants. Given time constraints, the number of respondents was limited, and expanding both the sample size and the depth of our questions would provide a more robust qualitative assessment. The survey began with a short questionnaire about each participant's musical background and included questions such as:

Q1: The audio samples were melodious (the musical phrases were pleasant and flowed smoothly). [Rate 1-5]
Q2: The improvisations demonstrate domain competence. [Rate 1-5]
Q3: The performance felt natural and human-like. [Rate 1-5]
Q4: For the following series of examples, denote which example you prefer: excerpt 1 or excerpt 2.
Q5: Does the generated melody or chord progression fit the provided context?
Q6: Open ended response: leave any thoughts below either on perceived errors in model-generated music and/or any observations about what the model did particularly well .

Based on the responses to the survey, we found that there was generally a 50/50 split as to which type of music is preferred: human-synthesized excerpts versus those generated by our model. Ratings for melodiousness averaged 4.5/5 across examples, and participants judged the outputs to demonstrate moderate musical competence (3/5), and moderately sounding "human-like" (3.5/5). Participants unequivocally denoted in the open-ended section that the results are both harmonically and melodiously pleasing.

We believe that due to the simplicity of our model, it was able to capture a significant amount of what makes a melody and/or harmony sound "human" properly. Because it was trained largely on pop music, which employs relatively simple harmonic and melodic structures, the model was forced to focus on harmonies and melodies in conjunction rather than other musical complexities.

Due to our small participant size of four (one musician, three non-musicians), we cannot conclusively generalize these results, but they nonetheless offer preliminary insight into how listeners perceive the model's musical outputs. We believe that expanding our qualitative evaluation in future work would yield deeper insight into the characteristics of music generated through machine learning.

6.3.2 Musical Analysis. Musically, we found that the model generally tends to follow diatonic harmony. There were some generated cases (which we included in our survey) that did employ chromatic harmony, such as certain transitions from *iv* to *IV* within the same generated song. The high prevalence of diatonic harmony is logical, given that diatonic structures are most common within the largely pop-music corpus on which the model was fine-tuned.

While the model occasionally struggled to strictly adhere to the established chord progression norms for the genres in its training data, we found that setting the sampling temperature to 0.9 significantly improved its output. At this temperature, the model reliably provided chord progressions properly reminiscent of the conditioned genre. For example, when conditioned with the `GENRE_JAZZ` token, the model would more often than not generate chords featuring extended harmonies (e.g., added chords or diminished chords).

Table 5. Chords to melody experimental results.

Temperature	BD	CBS
0.1	78.0	100.0
0.5	72.76	100.0
0.9	42.4	82.0
5	1.5	7.7

Table 6. Melody to chords experimental results.

Temperature	BD	CBS	CR	VC
0.1	92.9	83.0	29.0	100.0
0.5	88.2	79.0	30.0	100.0
0.9	61.4	50.0	31.0	99.8
5	2.8	7.3	45.8	97.1

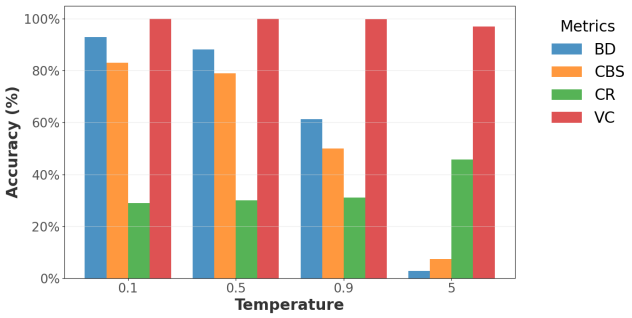

Figure 5. Bar chart representing the accuracy metrics (BD, CBS, CR, and VC) at various temperature values.

7 Ethics and Limitations

We recognize that the creation of music is inherently a creative process; following defined style and convention guidelines, or trying to fully mimic existing songs, does not always result in a creative output. However, music creation is an iterative process, and many songs are built upon the conventions of songs created before them. In the fine-tuning of this model, we do not intend to take away from the creativity that is a necessity of music, but rather seek to identify the trends that are present in modern music in order to better understand and explore the relationships between melody and chords. In making this, we also aim to provide a songwriting tool for artists rather than a music generator—it is merely one piece of the creative process.

Additionally, it is important to note that the majority of songs contained in the Hooktheory Lead Sheet Dataset are contemporary pop songs written in English. Because of this,

our fine-tuned model does not accurately represent the musical diversity present in the world. Many cultures have musical traditions that are not easily transcribed into our representational format, such as polyrhythms and microtones. Since these songs were not present in our data, our model will not generalize well to such music.

Beyond dataset diversity, our symbolic representation itself introduces limitations. By design, our notation prioritizes simplicity in lieu of expression. Features such as non-diatonic embellishments, tuplets, swing feel, tempo changes, secondary dominants, and modal interchange are either flattened or omitted entirely. This choice was deliberate as it made training feasible for a project of this scope; however, it prevented the model from capturing many of the subtleties that characterize real musical performance and composition. As a result, even when the model produces outputs that are structurally correct, they may lack nuance and/or stylistic depth. More generally speaking, this is a challenge that is hard to overcome for any sort of textual or symbolic music training task as it is incredibly difficult to create nuanced and detailed datasets of notated music.

Evaluating music quantitatively additionally remains a complicated process. Bar-duration and bar-sequence metrics measure structural correctness, but they do not meaningfully quantify musicality. Creativity and phrasing, for example, are difficult to evaluate numerically. Our qualitative evaluation partially compensates for this, but it is still constrained by the musical background and biases of our survey participants. A larger or more diverse evaluation pool may lead to different conclusions.

8 Conclusion and Future Work

Much as natural language can be broken down into sentences, phrases, and words, music can be broken down into constituent parts as well. By leveraging the power of pre-trained encoder-decoder models, fine-tuned on our specific task, we were able to treat chord and melody generation as a machine translation problem. Given an input melody or chord progression, we were able to successfully fine-tune a model to generate a harmonizing counterpart.

Our quantitative evaluation suggests that both models demonstrate a strong ability to preserve the sequential structure of bars as well as harmonic continuity, especially at lower temperature values.

Complementing these results, our qualitative evaluation found that participants rated the generated excerpts highly for melodiousness, and moderately for competence and human-likeness. Preference tests additionally produced an even split between human and model-generated excerpts.

Taking both the quantitative and qualitative results into account, we conclude that the model reliably captures the structural and harmonic conventions present in the training data while producing outputs that listeners perceive as coherent and musically meaningful.

From these results, we can infer that encoder–decoder transformers can generalize meaningful aspects of musical structure without relying on specialized music-specific architectures. At the same time, the limitations of our dataset (which is dominated by Western pop music) mean that the models do not capture the full diversity of global musical practices.

Future work could broaden the dataset to include more varied musical traditions, extend our symbolic representation to support more complex rhythmic and tonal systems, add conditioning variables such as meter or phrase boundaries, or explore larger transformer variants to improve long-range musical coherence. Ultimately, our findings show that encoder-decoder transformer models offer a promising and accessible approach to symbolic music generation and provide a foundation for tools that support human creativity rather than replace it.

Acknowledgments

Thank you to Professor Anna Rafferty for her help and guidance throughout this project.

References

[1] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, with Language Models.* 3rd edition, 2025. Online manuscript released August 24, 2025.

[2] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683, 2019.

[3] Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation, 2024.

[4] Suno, Inc. Suno ai. https://www.suno.ai, 2023. Accessed: 2025-11-24.

[5] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer, 2018.

[6] Shuyu Li and Yunsick Sung. Transformer-based seq2seq model for chord progression generation. *Mathematics*, 11(5), 2023.

[7] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019.

[8] Chris Anderson, Dave Carlton, Ryan Miyakawa, and Dennis Schwachhofer. Hooktheory TheoryTab Database, 2012. Accessed: 2025-11-22.

[9] Seungyeon Rhyu, Hyeonseok Choi, Sarah Kim, and Kyogu Lee. Translating melody to chord: Structured and flexible harmonization of melody with transformer. *IEEE Access*, 10:28261–28273, 02 2022.

[10] Chris Donahue, John Thickstun, and Percy Liang. Melody transcription via generative pre-training, 2022.

[11] Ju-Chiang Wang, Jordan B. L. Smith, and Yun-Ning Hung. Musfa: Improving music structural function analysis with partially labeled data, 2022.

[12] Yin-Cheng Yeh, Wen-Yi Hsiao, Satoru Fukayama, Tetsuro Kitahara, Benjamin Genchel, Hao-Min Liu, Hao-Wen Dong, Yian Chen, Terence Leong, and Yi-Hsuan Yang. Automatic melody harmonization with triad chords: A comparative study. *Journal of New Music Research (JNMR)*, 50(1):37–51, 2021.

[13] Tuomas Eerola and A. North. *Expectancy-based model of melodic complexity*, pages 1177–1183. 01 2000.

[14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[16] Hyeshin Chu, Joohee Kim, Seongouk Kim, Hongkyu Lim, Hyunwook Lee, Seungmin Jin, Jongeun Lee, Taehwan Kim, and Sungahn Ko. An empirical study on how people perceive ai-generated music. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 304–314, New York, NY, USA, 2022. Association for Computing Machinery.

[17] Alexander Lerch, Claire Arthur, Nick Bryan-Kinns, Corey Ford, Qianyi Sun, and Ashvala Vinay. Survey on the evaluation of generative models in music, 2025.

[18] Anna Jordanous. A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computation*, 4(3):246–279, 2012.