DO MUSIC GENERATION MODELS ENCODE MUSIC THEORY?

Megan Wei[1*] **Michael Freeman**[1*] **Chris Donahue**[2] **Chen Sun**[1]
[1] Brown University [2] Carnegie Mellon University
meganwei@brown.edu, michael_freeman@alumni.brown.edu

ABSTRACT

Music foundation models possess impressive music generation capabilities. When people compose music, they may infuse their understanding of music into their work, by using notes and intervals to craft melodies, chords to build progressions, and tempo to create a rhythmic feel. To what extent is this true of music generation models? More specifically, are fundamental Western music theory concepts observable within the "inner workings" of these models? Recent work proposed leveraging latent audio representations from music generation models towards music information retrieval tasks (e.g. genre classification, emotion recognition), which suggests that high-level musical characteristics are encoded within these models. However, probing individual music theory concepts (e.g. tempo, pitch class, chord quality) remains under-explored. Thus, we introduce **SynTheory**, a synthetic MIDI and audio music theory dataset, consisting of tempos, time signatures, notes, intervals, scales, chords, and chord progressions concepts. We then propose a framework to probe for these music theory concepts in music foundation models (Jukebox and MusicGen) and assess how strongly they encode these concepts within their internal representations. Our findings suggest that music theory concepts are discernible within foundation models and that the degree to which they are detectable varies by model size and layer.

1. INTRODUCTION

State-of-the-art text-to-music generative models [1–3] exhibit impressive generative capabilities. Past work suggests that internal representations of audio extracted from music generative models encode information relating to high-level concepts (e.g. genre, instruments, or emotion) [4–7]. However, it remains unclear if they also capture underlying symbolic music concepts (e.g. tempo or chord progressions) [8].

We aim to investigate if state-of-the-art music generation models encode music theory concepts in their internal representations and to what extent. Confirming this could enable the creative alteration of these concepts, providing artists with new methods towards more detailed and lower-level control [9] (e.g. changing the key of a song or editing a particular chord in a chord progression). Furthermore, by benchmarking these foundation models, we identify potential avenues for improvement towards stronger concept encoding. Our approach is based on work in probing and editing concepts in language models, which have shown promise in identifying emergent representations in autoregressive models and editing factual knowledge [9–12]. For music generative models, the probing approach has been applied to high-level concepts, such as emotion, genre, and tagging [4–7]. Moreover, existing datasets such as HookTheory [13] do contain rich annotations for music theory concepts but are associated with copyrighted music, potentially complicating their use.

Our first contribution is a framework to generate diagnostic datasets for probing music theory concepts, by programmatically specifying which concepts to vary and which to keep constant, while controlling the presence of potential distractor concepts. Our synthetic music theory dataset, **SynTheory**, consists of seven music concepts based on Western music theory: tempo, time signatures, notes, intervals, scales, chords, and chord progressions. SynTheory serves as a customizable, copyright-free, and scalable approach towards generating diagnostic music clips for probing real-world music generative models.

Our second contribution is the analysis of two state-of-the-art music generative models Jukebox [3] and MusicGen [1] with our SynTheory benchmark. We extract representations for the concepts defined in SynTheory from MusicGen and Jukebox and assess whether these models encode meaningful representations of these concepts. To analyze the internal representations of these models for SynTheory, we use a supervised approach to train probing classifiers [14] based on ground truth music theory concept labels. A higher classification accuracy implies that these models learn internal representations that "understand" music theory concepts, which can be decoded by a multi-layer perceptron (MLP) or a linear model.

Our results show that music foundation models encode meaningful representations of music theory concepts. These representations vary across different sections of the model (audio codecs, decoder LMs), different layers within the decoder LMs, and different model sizes. Furthermore, the nature of the concepts, from time-varying (e.g. chord progressions) to stationary (e.g. notes, chords) influence the performance of these models across these tasks. We hope our insights on probing music founda-

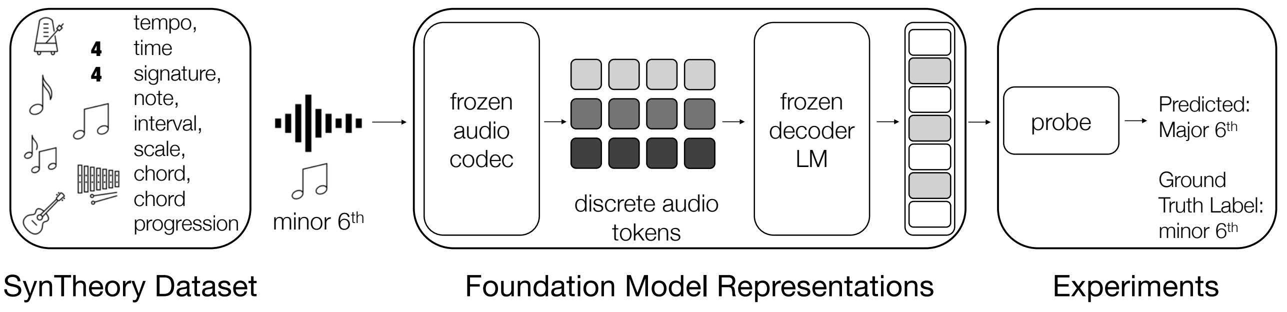

Figure 1. Overview of our SynTheory benchmark and our Jukebox and MusicGen probing setup. Our SynTheory benchmark consists of **Rhythmic** (tempos and time signatures) and **Tonal** (notes, intervals, scales, chords, and chord progressions) concepts. We assess whether music foundation models (Jukebox and MusicGen) encode these music theory concepts within their internal representations. For each task from the SynTheory dataset, we extract representations from the music foundation model. We pass an audio input, embodying the concept (e.g. Perfect 4th), into a pretrained foundation model. The audio codec tokenizes the audio into discrete audio tokens. Then, it passes these tokens into a decoder language model. From there, we extract the representations. We then train a probe classifier (linear and two-layer MLP) on these representations to predict particular classes (e.g. pitch class, intervals, and chords) for each SynTheory concept.

tion models, along with the synthetic music data generation framework, encourage and facilitate future endeavors on symbolic controllability in music generative models.

For reproducibility, we release the code for dataset generation, embedding extraction, probing, and evaluation in our GitHub repository [1] and our website [2].

2. RELATED WORK

The success of large language models (LLMs) [15–18] has sparked new research on probing and editing their internal representations to measure their understanding of linguistic concepts [19, 20] and world knowledge [11, 12, 21] as well as editing the encoded knowledge to make LLMs more faithful to factual knowledge [9, 10]. Studies have shown that large language models can encode grounded representations on color [22], direction [23], and auditory representations [24]. Thus, it is interesting to investigate if large music generative models, which often share similar model architectures and training objectives as LLMs, are able to encode abstract concepts from high-level music information (e.g. genre, emotion) to low-level music theory (e.g. tempo, chords).

Recent work has indeed shown promise in uncovering conceptual representations from probing audio and music generative models, leveraging different music foundation model architectures towards music understanding tasks. Castellon and Donahue et al. [4] propose using representations from language models trained on codified audio towards downstream MIR tasks as a better alternative to conventional tagging models. The authors train probing classifiers on Jukebox representations on the music tagging, genre identification, key identification, and emotion recognition tasks. These results demonstrate the effectiveness of internal model representations in downstream MIR tasks. Koo et al. [7] focus primarily on probing MusicGen's attention heads in instrument recognition tasks, benchmarking against the tasks highlighted in [4] and propose leveraging these representations for inference-time control. Other works [5, 6] focus on the impact of model architecture and self-supervised approaches towards music understanding tasks.

However, prior work primarily uses real-world data, which is often concept-entangled and potentially subject to copyright concerns. For example, some of these works use Giantsteps-MTG and Giantsteps [25], which are datasets of primarily electronic dance music with tempo and key annotations, obtained from Beatport. Won et al. [5] use HookTheory for chord recognition, where they focus on major and minor chord identification for each pitch class. The authors also use Harmonix Set [26] and GTZAN [27] for beat and downbeat detection. In the language modality, the authors of ChatMusician [28] produce a multi-choice question answering dataset, MusicTheoryBench, with expert annotation from a professional college music teacher. MusicTheoryBench aims to assess the music understanding capabilities of LLMs but through natural language alone. To the best of our knowledge, there is a lack of music theory probing benchmarks in the audio domain that are accurately-labeled, copyright-free, and scalable, prior to our proposed SynTheory.

3. SYNTHEORY: SYNTHETIC DATASET OF MUSIC THEORY CONCEPTS

We design seven datasets to capture isolated music theory concepts – similar to synthetic audio for ear training. Musicians may "train their ear" to recognize music concepts like intervals or chord quality in an isolated setting before advancing to the harder, more entangled case that arises in non-pedagogical music. Assessing concept recognition through isolated concepts mitigates the possibility that one intuits or guesses the answer from its context. Literature on instrument-specific absolute pitch in humans corroborates the notion that timbral information may be exploited in identifying a different concept like pitch class [29]. As

[1] https://github.com/brown-palm/syntheory
[2] https://brown-palm.github.io/music-theory

such, our dataset is designed to remove or reduce features that may correlate with a concept, but are not strictly necessary for identifying it. Our intent is a more pointed assessment towards theoretical concepts as abstract ideas rather than as acoustically realized audio. A more practical motivation for this work is that extracting such low-level, isolated concepts from existing datasets may require non-trivial engineering or domain expert labor. It may even be impossible to disentangle all overlapping concepts. Music stem isolation and concept isolation are distinct; an isolated instrument in a multi-track recording may still exhibit several, intricately intertwined theory concepts. It is not clear how to "unmix" such concepts once they are blended.

Instead of attempting to disentangle several concepts from existing audio, SynTheory implements this "ear training" quiz setting by explicitly producing individual concepts. Each of the seven datasets ablates a single musical feature while fixing all others, thereby isolating it to a degree not typically found in recorded music. These ablated concepts consist of tempo, time signatures, notes, intervals, scales, chords, and chord progressions. We adopt isolation as a design choice to mitigate context that may be exploited in deep learning models as "shortcuts", i.e. heuristics that correlate with concepts most of the time but do not truly encode the concept.

Using this music theory concept-driven synthesis design, we construct label-balanced and copyright-free data. The synthetic approach avoids annotation errors present in other contemporary MIR datasets. For example, the HookTheory data processing step for SheetSage [13] required ad-hoc time-alignment of the expert annotations. In the released SheetSage dataset, $17,980/26,175$ (68.7%) samples required more precise time alignment. While our synthetic data is no substitute for real music data, to our knowledge, no other dataset so strictly isolates each concept.

SynTheory contains two categories: tonal and rhythmic. We make this distinction for stronger concept isolation; we wish to keep the rhythm samples tonally consistent and the tonal samples rhythmically consistent. For each **tonal** dataset, we voice the same MIDI data through 92 distinct instruments. The selection of instrument voices is fixed, making the distribution of timbres sufficiently diverse but also class-balanced. Each instrument corresponds to one of the 128 canonical MIDI program codes and is voiced through the `TimGM6mb.sf2` [30] soundfont. A MIDI "program" is a specific instrument preset. The canonical program set includes many named instruments, e.g. "Acoustic Grand Piano", "Flute", etc. We exclude programs that are polyphonic, sound effects (e.g. "Bird Tweet", "Gun Shot"), and highly articulate. A highly articulate program has some unchangeable characteristic (e.g. pitch bending) that destabilizes its pitch. For each **rhythmic** dataset, we define five metronome-like timbral settings. Each setting uses one of the distinct instruments: "Woodblock Light", "Woodblock Dark", "Taiko", "Synth Drum", and the MIDI drum-kit, following the voicing done in Sheetsage [13]. Each setting produces a distinct sound on the upbeat and the downbeats, which defines the time signature concept.

Concept	Total Samples
Tempo	4,025
Time Signatures	1,200
Notes	9,936 [3]
Intervals	39,744
Scales	15,456
Chords	13,248
Chord Progressions	20,976

Table 1. **SynTheory** contains seven synthetic datasets, each of which captures an isolated music theory concept. We present an overview of these datasets and their sizes.

3.1 SynTheory-Rhythmic

3.1.1 *Tempo*

We voice integer tempi within 50 to 210 BPM (beats per minute) inclusive in $\frac{4}{4}$ time. To ensure diverse start times, we produce five random offsets per sample. There are $(5 \text{ CLICK SETTING} \cdot 161 \text{ TEMPO} \cdot 5 \text{ OFFSET}) = 4,025$ samples in total.

3.1.2 *Time Signature*

We voice the following time signatures: $\frac{2}{2}, \frac{2}{4}, \frac{3}{4}, \frac{3}{8}, \frac{4}{4}, \frac{6}{8}, \frac{9}{8},$ and $\frac{12}{8}$. The tempo is fixed at 120 BPM. To add acoustic variation, we add three levels of reverb from completely dry to spacious. We find empirically that this acoustic perturbation increases the difficulty of the probing task. Like the Tempo dataset, we produce ten random offsets for each sample. There are $(8 \text{ TIME SIGNATURE} \cdot 3 \text{ REVERB LEVEL} \cdot 5 \text{ CLICK SETTING} \cdot 10 \text{ OFFSET}) = 1,200$ samples.

3.2 SynTheory-Tonal

3.2.1 *Notes*

We voice all twelve Western temperament pitch classes, in nine octaves, using 92 instruments. The note is played in quarter notes at a tempo of 120 BPM, with no distinction between the upbeat or downbeat. There are $(12 \text{ PITCH CLASS} \cdot 9 \text{ OCTAVE} \cdot 92 \text{ INSTRUMENT}) = 9,936$ configurations. However, there are only $9,900$ distinct *samples* because 36 configurations at extreme registers are unvoiceable in our soundfont. These silent samples are listed for completeness in our GitHub repository.

3.2.2 *Intervals*

We vary the root note, number of half-steps, instrument, and play style (unison, up, and down). To retain consistent rhythm, the up and down styles repeat four times throughout the sample while the unison play style repeats

[3] There are 9,936 distinct note configurations, but our dataset contains 9,900 non-silent samples. With a more complete soundfont, all 9,936 configurations are realizable to audio.

eight times. There are $(12 \text{ PITCH CLASS } \cdot 12 \text{ HALF-STEP } \cdot 92 \text{ INSTRUMENT } \cdot 3 \text{ PLAY STYLE }) = 39,744$ samples.

3.2.3 *Scales*

We voice seven Western music modes (Ionian, Dorian, Phrygian, Lydian, Mixolydian, Aeolian, and Locrian) in all root notes, in 92 instruments, and in two play styles (ascending or descending). The register is constant; we select root notes close to middle C. There are $(7 \text{ MODE } \cdot 12 \text{ ROOT NOTE } \cdot 92 \text{ INSTRUMENT } \cdot 2 \text{ PLAY STYLE }) = 15,456$ samples.

3.2.4 *Chords*

We voice triads of all twelve root notes, four chord qualities (major, minor, augmented, and diminished), 92 instruments, and three inversions (root position, first inversion, and second inversion). The chord is struck at each quarter note at 120 BPM. Like in the *Scales* dataset, we fix the register close to middle C. There are $(12 \text{ ROOT NOTE } \cdot 4 \text{ CHORD QUALITY } \cdot 92 \text{ INSTRUMENT } \cdot 3 \text{ INVERSION }) = 13,248$ samples.

3.2.5 *Chord Progressions*

We select 19 four-chord progressions, with ten in the major mode and nine in the natural minor mode. The progressions are:

- Major: (I–IV–V–I), (I–IV–vi–V), (I–V–vi–IV), (I–vi–IV–V), (ii–V–I–Vi), (IV–I–V–Vi), (IV–V–iii–Vi), (V–IV–I–V), (V–vi–IV–I), (vi–IV–I–V)
- Natural Minor: (i–ii°–v–i), (i–III–iv–i), (i–iv–v–i), (i–VI–III–VII), (i–VI–VII–i), (i–VI–VII–III), (i–VII–VI–IV), (iv–VII–i–i), (VII–vi–VII–i)

We vary only the root note of the key and instrument. Each chord is played in quarter notes at 120 BPM. There are $(19 \text{ PROGRESSION } \cdot 12 \text{ KEY ROOT } \cdot 92 \text{ INSTRUMENT }) = 20,976$ samples.

One can extend or alter the above configurations using the SynTheory codebase. We provide a framework that enables declarative and programmatic MIDI construction in musical semantics, audio export in any soundfont, and dataset construction for use in our framework.

4. EXPERIMENTS

We describe the evaluation protocols used to analyze the internal representations of music generative models (MusicGen and Jukebox) and handcrafted audio features (mel spectrograms, MFCC, and chroma) for music theory concept encoding.

4.1 Evaluation

A "probe" is a simple or shallow classifier, often a linear model, trained on the activations of a neural network [14]. Accurate performance of such classifiers suggests that information relevant to the class exists in the latent representation within the network. As such, probes may be used as a proxy for measuring a model's "understanding" or encoding of abstract concepts. Motivated by the use of probes to discover linguistic structure and semantics in NLP [31] and more recently in MIR [4], we use probes to assess whether music theory concepts are discernable in foundation models.

We adopt the same probing paradigm as [4] and frame concept understanding as multiclass classification for discrete concepts (notes, intervals, scales, chords, chord progressions, and time signatures) and regression for continuous concepts (tempo). We train linear and two-layer MLP probes on the embeddings of the internal representations of Jukebox and MusicGen and the handcrafted features. We measure the classification accuracy of our trained probes on the SynTheory tasks using the following classes:

- Notes (12): C, C#, D, D#, E, F, F#, G, G#, A, A#, and B
- Intervals (12): minor 2nd, Major 2nd, minor 3rd, Major 3rd, Perfect 4th, Tritone, Perfect 5th, minor 6th, Major 6th, minor 7th, Major 7th, and Perfect octave
- Scales (7): Ionian, Dorian, Phrygian, Lydian, Mixolydian, Aeolian, and Locrian
- Chords (4): Major, Minor, Diminished, and Augmented
- Chord Progressions (19): (I–IV–V–I), (I–IV–vi–V), (I–V–vi–IV), (I–vi–IV–V), (ii–V–I–Vi), (IV–I–V–Vi), (IV–V–iii–Vi), (V–IV–I–V), (V–vi–IV–I), (vi–IV–I–V), (i–ii°–v–i), (i–III–iv–i), (i–iv–v–i), (i–VI–III–VII), (i–VI–VII–i), (i–VI–VII–III), (i–VII–VI–IV), (iv–VII–i–i), and (VII–vi–VII–i)
- Time Signatures (8): $\frac{2}{2}, \frac{2}{4}, \frac{3}{4}, \frac{3}{8}, \frac{4}{4}, \frac{6}{8}, \frac{9}{8},$ and $\frac{12}{8}$

These tasks are trained on a 70% train, 15% test, and 15% validation split, using the Adam optimizer and Cross Entropy loss.

For the Tempos dataset, we train a regression probe, over the 161 tempo values. To increase complexity in the probing task and test generalization to unseen BPMs, the training set consists of the middle 70% of the BPMs. The test and validation sets consist of the top 15% BPMs and the bottom 15% BPMs, randomly shuffled and split in half. We use MSE loss and report the R^2 score.

Each probe is trained independently for its corresponding concept task. That is, the probe trained to identify notes from Jukebox embeddings will not be used to identify intervals, for example.

To select the best performing probe for each concept using the MusicGen audio codec, mel spectrogram, MFCC, chroma, and aggregate handcrafted features, we perform a grid search across various hyperparameters for each task, following those defined in [4]:

- Data Normalization: {True, False}
- Model Type: {Linear, two-layer MLP with 512 hidden units and ReLU activation}
- Batch Size: {64, 256}
- Learning Rate: $\{10^{-5}, 10^{-4}, 10^{-3}\}$
- Dropout: {0.25, 0.5, 0.75}
- L2 Weight Decay: $\{\text{off}, 10^{-4}, 10^{-3}\}$

For the decoder LMs (MusicGen small, medium, and large and Jukebox) as detailed in Section 4.2, we use a fixed set of hyperparameters and select the probe with the best performing layer for each concept, in the interest of computational efficiency:

- Data Normalization: True
- Model Type: two-layer MLP with 512 hidden units and ReLU activation
- Batch Size: 64
- Learning Rate: 10^{-3}
- Dropout: 0.5
- L2 Weight Decay: off

We selected these hyperparameters from the best overall performing probe by fixing a layer in the decoder LMs and performing a hyperparameter search, following the sweep approach outlined in [4].

4.2 Model representations

We extract representations from two text-to-music generative foundation models, Jukebox [3] and MusicGen [1]. We benchmark the probing classifier performance of these representations against that of three handcrafted, spectral features following [4]: mel spectrograms, mel-frequency cepstral coefficients (MFCC), and constant-Q chromagrams (chroma). These handcrafted features are common in traditional methods of MIR and are a more interpretable baseline against the embeddings of the pre-trained music generative models. We additionally report probing classifier performance on the concatenation of all the aforementioned handcrafted features.

Jukebox consists of a VQ-VAE model that codifies audio waveforms into discrete codes at a lower sample rate and a language model that generates codified audio with a transformer decoder. We trim all audio to four seconds and ensure it is mono. We utilize Jukemirlib [4] to pass this audio through the frozen audio encoder and through the decoder language model. We downsample the activation to a target rate of half that in [4], due to resource constraints, using the Librosa FFT algorithm [32]. Then, we meanpool the representations across time to reduce the dimensionality of the embeddings, resulting a dimension of $(72, 4800)$ per sample, where 72 is the number of layers and 4800 is the dimension of the activations. We reduce the dimensionality of these representations by defining a layer selection process similar to [4]; that is, each probing classifier trains on only one of the 72 layers. We train the probe classifiers with fixed hyperparameters on the music concept tasks as described in Section 4.1. For each concept, we select the layer that results in the highest probing score. The final dimension of the Jukebox representation is 4800.

MusicGen consists of a pretrained convolutional autoencoder (EnCodec) [33], a pretrained T5 text encoder, and an acoustic transformer decoder. We resample the audio to 32 kHz (the sampling rate used in the EnCodec model) trim to four seconds, convert to mono, and pass the audio through the frozen EnCodec audio codec. We do not pass text through the text encoder, as we focus on audio representations. We then extract representations from several regions of the model: the final layer of the audio codec before residual vector quantization and the hidden states of the decoder language model. The number of decoder hidden states vary based on the model size: small (24 layers), medium (48 layers), and large (48 layers).

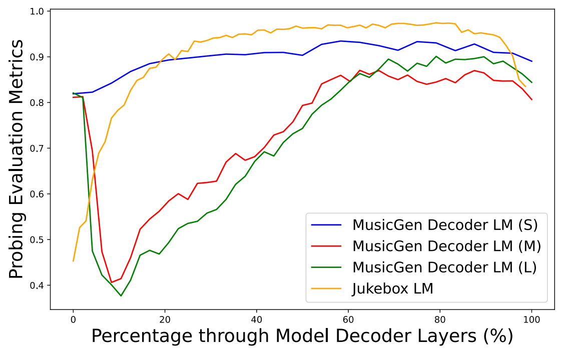

Figure 2. Probing evaluation metrics averaged across all SynTheory concepts over the model layers of Jukebox and MusicGen decoder models. The probing evaluation metric is R^2 for tempos and accuracy for the rest of the SynTheory concepts (notes, intervals, scales, chords, chord progressions, and time signatures). Features extracted from deeper layers generally perform better, with a slight drop-off near the final layers.

For our four second audio clips, the audio codec representations are of dimension $(128, 200)$, where 128 is the dimension of the activation after the final layer of the audio codec and 200 is the sequence length. We meanpool the values of the representations across time, resulting in a final dimension of 128 for the MusicGen audio codec.

The decoder hidden states for the small, medium, and large MusicGen models have dimensions $(24, 200, 1024)$, $(48, 200, 1536)$, $(48, 200, 2048)$ respectively, where the first axis corresponds to the number of layers, second corresponds to sequence length, and third corresponds to hidden size. To reduce the dimensionality of these representations, similar to what was done with Jukebox, we select the most optimal layer for each decoder model size based on probing scores. We visualize results from probing across layers per model (MusicGen and Jukebox) averaged across concepts in Figure 2. After selecting the best performing layer per concept and model size, the dimensions of the representations are $(200, 1024)$ for MusicGen small, $(200, 1536)$ for MusicGen medium, and $(200, 2048)$ for MusicGen large. To further reduce the dimensions, we also meanpool across time as done in Jukebox representations, resulting in dimensions of 1024 for MusicGen small decoder, 1536 for MusicGen medium decoder, and 2048 for MusicGen large decoder.

We extract the handcrafted features (mel spectrograms, mel-frequency cepstral coefficients, and constant-Q chromagrams) with librosa [32]. Similar to [4], we concatenate the mean and standard deviation across time of these features along with their first- and second-order discrete differences. Furthermore, we concatenate the mel spectrogram, chroma, and MFCC features and obtain their mean and standard deviation across time and their first- and second-order differences to obtain an aggregate representation of the handcrafted features.

	Notes	**Intervals**	**Scales**	**Chords**	**Chord Progressions**	**Tempos**	**Time Signatures**	**Average**
Jukebox LM	0.951	0.995	0.978	0.997	0.971	0.993	1.000	0.984
MusicGen LM (S)	0.897	0.995	0.949	0.990	0.942	0.969	0.911	0.950
MusicGen LM (M)	0.851	0.983	0.863	0.989	0.870	0.956	0.883	0.914
MusicGen LM (L)	0.866	0.972	0.905	0.989	0.901	0.965	0.905	0.929
MusicGen Audio Codec	0.729	0.965	0.383	0.879	0.330	0.947	0.677	0.701
Mel Spectrogram	0.712	0.995	0.897	0.988	0.723	0.785	0.827	0.847
MFCC	0.467	0.822	0.370	0.863	0.872	0.923	0.688	0.715
Chroma	0.954	0.820	0.989	0.994	0.869	0.847	0.672	0.878
Aggregate Handcrafted	0.941	0.997	0.972	0.992	0.868	0.947	0.833	0.936

Table 2. We report probing results on the SynTheory dataset for the Jukebox LM, MusicGen Decoder LM (Small, Medium, and Large), MusicGen Audio Codec models as well as handcrafted features (Mel Spectrogram, MFCC, Chroma, and Aggregate Handcrafted). For the tempos dataset, we report the R^2 score from the regression probe. For all other concepts (notes, intervals, scales, chords, chord progressions, and time signatures), we report the probing classifier accuracy. For MusicGen Audio Codec, Mel Spectrogram, MFCC, Chroma, and Aggregate Handcrafted, we report the metrics of the best performing probe for each task using the best validation performance from our hyperparameter search. For MusicGen Decoder LM (Small, Medium, and Large) and Jukebox models, we report the metrics of the best performing probe for each task using layer selection. We also report an average performance across all concepts for each model/feature.

5. RESULTS AND DISCUSSION

We observe that Jukebox performs consistently well across our SynTheory benchmark. All MusicGen Decoder models also exhibit competitive performance across concepts. While [1] claims that larger MusicGen models produce better quantitative and subjective scores and that larger models better "understand" text prompts, our MusicGen Decoder LM (Small) result seems to contrast with traditional discussions on scaling laws. Figure 2 displays the consistent probing score of MusicGen Decoder LM (Small) across all layers and highlights its higher performance compared to that of its larger counterparts. Meanwhile, the larger MusicGen models exhibit a steep drop in probing performance in initial layers, followed by a gradual increase in performance, with the performance tapering off in the final layers.

MusicGen slightly underperforms on the notes dataset. We hypothesize this is because isolated notes in real-world music are not as prominent as intervals, scales, and chords. This reveals how the lowest-level building blocks of music are even harder to distinguish.

In general, the probing results from the pretrained music decoder LMs yield better probing performance compared to the MusicGen Audio Codec representations and the individual handcrafted features. MusicGen Audio Codec exhibits overall poorer performance on these tasks, since these codecs were trained to reconstruct fine-grained, low-level details localized by time.

Because chroma features encode pitch class information, chroma features perform comparably well on tonal tasks. However, they slightly underperform on rhythmic tasks. Chroma features outperform MusicGen Decoder LMs on stationary harmonic tasks (notes, scales, and chords) but are worse for dynamic harmonic tasks (chord progressions and intervals).

The aggregate handcrafted features perform comparably to MusicGen Decoder LMs. This suggests that harder music concept understanding benchmarks should address concepts latent in foundation models but not easily encoded in handcrafted features. These harder benchmarks may include entangled concepts, such as probing for both chord progression type and tempo in a tempo-varying chord progression sample. Probing for more compositional tasks could further our understanding of more realistic concept encoding in both model representations and handcrafted features.

6. CONCLUSION

In this work, we introduce SynTheory, a synthetic dataset of music theory concepts, that is concept-isolated, annotated, and copyright-free. Further, we use this dataset to evaluate the degree to which music theory concepts are encoded in existing state-of-the-art music generative models. Our experiments suggest that music theory concepts are indeed discernible within the latent representations of these generative models. We believe this is a prerequisite to further understand how to isolate and manipulate such concepts, which advances towards low-level controllable generation and music theory evaluation metrics. We encourage the community to build more challenging probing datasets with our framework to further understand the relationship between symbolic and audio-based music generation.

7. ETHICS STATEMENT

Our work aims to understand if music generation models encode music theory concepts in their internal representations. Our dataset may be used to assess music generation models and may be applied towards fine-grained, music-theory based controllable generation.

Our custom dataset, SynTheory, is based on elementary Western music theory concepts and is generated programmatically. The data does not infringe copyright of musical writers or performers. We envision no negative societal impacts from the publication of our report or the release of our dataset.

8. ACKNOWLEDGEMENTS

We would like to thank Professor Cheng-Zhi Anna Huang, Professor Daniel Ritchie, Professor David Bau, Professor Jacob Andreas, Tian Yun, Nate Gillman, and Calvin Luo for their fruitful discussions and feedback towards this work. This project is partially supported by Samsung. We would also like to thank the Center for Computation and Visualization at Brown University for their computational resources towards this project. Finally, we greatly appreciate the insightful questions and thoughtful feedback from the reviewers.

9. REFERENCES

[1] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[2] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. Frank, "MusicLM: Generating music from text," *arXiv preprint arXiv:2301.11325*, 2023.

[3] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," *arXiv preprint arXiv:2005.00341*, 2020.

[4] R. Castellon, C. Donahue, and P. Liang, "Codified audio language modeling learns useful representations for music information retrieval," in *International Society for Music Information Retrieval*, 2021.

[5] M. Won, Y.-N. Hung, and D. Le, "A foundation model for music informatics," *arXiv preprint arXiv:2311.03318*, 2023.

[6] Y. Li, R. Yuan, G. Zhang, Y. Ma, X. Chen, H. Yin, C. Lin, A. Ragni, E. Benetos, N. Gyenge, R. Dannenberg, R. Liu, W. Chen, G. Xia, Y. Shi, W. Huang, Y. Guo, and J. Fu, "Mert: Acoustic music understanding model with large-scale self-supervised training," *arXiv preprint arXiv:2306.00107*, 2023.

[7] J. Koo, G. Wichern, F. G. Germain, S. Khurana, and J. Le Roux, "Understanding and controlling generative music transformers by probing individual attention heads," *IEEE ICASSP Satellite Workshop on Explainable Machine Learning for Speech and Audio (XAI-SA)*, 2024.

[8] G. Brunner, Y. Wang, R. Wattenhofer, and J. Wiesendanger, "Jambot: Music theory aware chord based generation of polyphonic music with lstms," in *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2017, pp. 519–526.

[9] K. Li, O. Patel, F. Viégas, H. Pfister, and M. Wattenberg, "Inference-time intervention: Eliciting truthful answers from a language model," in *Advances in Neural Information Processing Systems*, 2024.

[10] K. Meng, D. Bau, A. Andonian, and Y. Belinkov, "Locating and editing factual associations in GPT," in *Advances in Neural Information Processing Systems*, 2022.

[11] K. Li, A. K. Hopkins, D. Bau, F. Viégas, H. Pfister, and M. Wattenberg, "Emergent world representations: Exploring a sequence model trained on a synthetic task," in *The Eleventh International Conference on Learning Representations*, 2023.

[12] T. Yun, Z. Zeng, K. Handa, A. V. Thapliyal, B. Pang, E. Pavlick, and C. Sun, "Emergence of abstract state representations in embodied sequence modeling," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2023.

[13] C. Donahue, J. Thickstun, and P. Liang, "Melody transcription via generative pre-training," in *International Society for Music Information Retrieval*, 2022.

[14] G. Alain and Y. Bengio, "Understanding intermediate layers using linear classifier probes," in *International Conference of Learning Representations*, 2016.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Association for Computational Linguistics*, 2019.

[16] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[17] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, 2020.

[18] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[19] I. Tenney, D. Das, and E. Pavlick, "Bert rediscovers the classical nlp pipeline," in *Association for Computational Linguistics*, 2019.

[20] I. Tenney, P. Xia, B. Chen, A. Wang, A. Poliak, R. T. McCoy, N. Kim, B. Van Durme, S. R. Bowman, D. Das *et al.*, "What do you learn from context? probing for sentence structure in contextualized word representations," in *Proceedings of the 7th International Conference on Learning Representations*, 2019.

[21] T. Yun, C. Sun, and E. Pavlick, "Does vision-and-language pretraining improve lexical grounding?" in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021.

[22] M. Abdou, A. Kulmizev, D. Hershcovich, S. Frank, E. Pavlick, and A. Søgaard, "Can language models encode perceptual structure without grounding? a case study in color," in *Proceedings of the 25th Conference on Computational Natural Language Learning*, 2021.

[23] R. Patel and E. Pavlick, "Mapping language models to grounded conceptual spaces," in *International Conference on Learning Representations*, 2022.

[24] J. Ngo and Y. Kim, "What do language models hear? probing for auditory representations in language models," in *Association for Computational Linguistics*, 2024.

[25] P. Knees, Á. Faraldo, P. Herrera, R. Vogl, S. Böck, F. Hörschläger, and M. L. Goff, "Two data sets for tempo estimation and key detection in electronic dance music annotated from user corrections," in *International Society for Music Information Retrieval*, 2015.

[26] O. Nieto, M. C. McCallum, M. Davies, A. Robertson, A. M. Stark, and E. Egozy, "The harmonix set: Beats, downbeats, and functional segment annotations of western popular music," in *International Society for Music Information Retrieval*, 2019.

[27] U. Marchand, Q. Fresnel, and G. Peeters, "GTZAN-rhythm: Extending the GTZAN test-set with beat, downbeat and swing annotations," in *ISMIR 2015 Late-Breaking Session*, 2015.

[28] R. Yuan, H. Lin, Y. Wang, Z. Tian, S. Wu, T. Shen, G. Zhang, Y. Wu, C. Liu, Z. Zhou, Z. Ma, L. Xue, Z. Wang, Q. Liu, T. Zheng, Y. Li, Y. Ma, Y. Liang, X. Chi, R. Liu, Z. Wang, P. Li, J. Wu, C. Lin, Q. Liu, T. Jiang, W. Huang, W. Chen, E. Benetos, J. Fu, G. Xia, R. Dannenberg, W. Xue, S. Kang, and Y. Guo, "Chatmusician: Understanding and generating music intrinsically with llm," *arXiv preprint arXiv:2402.16153*, 2024.

[29] L. Reymore and N. C. Hansen, "A theory of instrument-specific absolute pitch," *Frontiers in Psychology*, vol. 11, 2020. [Online]. Available: https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2020.560877

[30] T. Brechbill, "Timidity++," 2004. [Online]. Available: https://timbrechbill.com/saxguru/Timidity.php

[31] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018.

[32] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python." in *SciPy*, 2015, pp. 18–24.

[33] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *arXiv preprint arXiv:2210.13438*, 2022.