Probing for Advanced Music Theory Concepts in Generative Music Models

Derek Kwan(✉) and Patrick J. Donnelly

Oregon State University, Corvallis, OR 97330, USA
{kwand,donnellp}@oregonstate.edu

Abstract. Western music theory concepts underlie the compositional processes of much of the music consumed today and consequently serve as the conceptual underpinnings of much of the training data for generative music models. Applying the interpretability technique of probing, Wei et al. [58] studied how well basic musical concepts such as scales, time signatures, and simple triadic harmonic progressions are encoded in the latent spaces of the `Jukebox` and `MusicGen` generative models. Through constructing datasets isolating these concepts, they trained simple classifiers to classify elements of these concepts based on their latent representations. We extend their work by creating datasets focused on five higher-level and more advanced music theory concepts: polyrhythms, dynamics, seventh chords, mode mixture, and secondary dominants. Performing experiments with the same `Jukebox` and `MusicGen` models, we train classifiers on the datasets' latent representations. We find that these advanced concepts are encoded in their latent spaces and that secondary dominant classification is the most difficult of our probing tasks.

Keywords: Generative Music · Music Theory · Probing

1 Introduction

Generative AI has quickly become a prominent aspect across the field of music. In 2025, streaming music platforms reported a large influx of music characterized as AI-generated or spam. Deezer categorized around 30,000 daily uploads to their platform as spam [47] while Spotify labeled around 75 million songs likewise over a year [50]. Furthermore, a growing number of commercial generative AI music tools have become available to consumers, including Suno [52], Udio [22], Stability AI's Stable Audio [20], and Adobe's Generate Soundtrack [57].

Although they may enable the mass generation of what Spotify labels "spammy tracks" [50], generative music models, particularly those which allow text- or metadata-conditioned generation, offer novel ways of interacting with the music composition process. With these models, aspiring composers can dictate their musical ideas through natural language instead of playing an instrument or directly manipulating symbolic music representations. This added layer of abstraction, combined with generators trained on audio rather than symbolic notation, raises questions about the use of Western musical theory concepts that typically guide traditional means of composition.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
P. Machado et al. (Eds.): EvoMUSART 2026, LNCS 16523, pp. 252–270, 2026.
https://doi.org/10.1007/978-3-032-24350-8_17

Wei et al. [58] addressed this line of inquiry by creating the SynTheory dataset featuring the Western music theory concepts of notes, major and minor scales, intervals, triad chords, major and minor chord progressions, tempi, and time signatures. Investigating OpenAI's `Jukebox` [18] and Meta's `MusicGen` [15], they extracted the generative music models' latent representations of the dataset and trained probing classifiers to discriminate types of the music theory concepts. As probing classifiers measure the mutual information between a concept's latent representation and its class [6], high degrees of mutual information between a dataset's latent representations and its concept classes suggest that the concept is encoded within a model's latent space. A model that does not encode these concepts may still be able to generate aesthetically pleasing music. However, as concepts such as harmony can be described in terms of syntax [46], this music may sound like gibberish within the context of Western music.

In this work, we extend the approach of [58] to explore other high-level and advanced music theory concepts through creating datasets[1] and associated classification tasks centered around these concepts. Our expansion explores the music theory concepts of polyrhythms, dynamic expression, seventh chords, mode mixture, and secondary dominants. With polyrhythms, we expand on [58]'s tempo dataset and explore the ability of a model to encode not just the absolute rate of one but the relative rates of two simultaneous rates of constant pulses. We extend the previous work's focus on pitch and rhythmic concepts towards loudness with the dynamics dataset. With seventh chords, we expand on the concept of the previous triad chord dataset to investigate the encoding of four-note chords. While the previous work investigated the ability to classify chord progressions, we investigate the higher-level concept of classifying types rather than identities of chord progressions with the mode mixture and secondary dominants datasets.

We follow [58] by extracting layer outputs for these datasets from `MusicGen`'s audio encoder `EnCodec`, `MusicGen-small`, `MusicGen-medium`, `MusicGen-large`, and `Jukebox`. We train single-hidden-layer MLP probing classifiers on each classification task. Expanding on [58]'s probing techniques, we also train a neural network with no hidden layers to examine linear relationships within each task.

2 Related Works

The technique of training classifier and regression probes on neural network (NN) latent spaces has been applied across a wide variety of network architectures and applications. Examples of this include studying the internal changes in image latent representations within Convolutional Neural Networks by measuring their classification accuracies [3,4] and assessing the ability of various graph neural network and transformer architectures to encode molecular structures [2]. Large Language Models have been a particularly active target of probing research. This includes probing for lower-level linguistic concepts in LSTMs through part-of-speech and semantic tagging tasks [7] as well as high-level linguistic concepts

[1] Available at https://huggingface.co/datasets/derekxkwan/syntheory_plus, last accessed 22 Jan 2026.

including hyperbole [48] and metaphor [1] in the intermediate word presentations of `BERT` [17], `RoBERTa` [36], and `ELECTRA` [14] language models.

The past five years have seen a growing body of research related to probing NN latent spaces with general music information retrieval (MIR) and music theory tasks. In the general MIR domain, Castellon et al. [12] used the MagnaTagATune [33], GTZAN [53,54], Giantsteps-MTG[2], and Giantsteps [30] datasets to probe `Jukebox` embeddings with the tasks of music tagging, music genre classification, key detection, and emotion recognition. For example, the authors probed `Jukebox` with the Giantsteps dataset for key detection and achieved an accuracy score of 66.7% over 24 classes (12 key centers with major or minor modes) [12]. Their focus was to demonstrate the viability of `Jukebox` embeddings as representations useful for other, non-generative MIR tasks such as those generated by the `MERT` [35] music understanding model. However, they also demonstrated the encoding of general music concepts in these embeddings, inspiring others, such as Wei et al. [58], to probe for musical concepts in other generative models. Koo et al. [31] continue this line of research of music concept probing by focusing on the same tasks as [12] with the addition of instrument recognition on `MusicGen` embeddings from MUSDB [45] and MoisesDB [43] datasets. These authors [32] later expanded on this work through using intervention techniques to encourage the presence of certain instruments in generated music, preceding the work on finer-grained music generation control by Facchiano et al. [21] and Panda et al. [41]. In work that seeks to do the opposite of probing by discovering concepts in models' latent spaces, Singh et al. [49] used the MusicSet [39] dataset and sparse autoencoders to discover latent concepts in `MusicGen-large` and `-small` embeddings. They also devised a pipeline to automatically assign labels to latent concepts using Google's `Gemini Flash 1.5`, pretrained audio classifiers from Essentia [8], and the CLAP text-audio embedding model [59].

With respect to probing for music theory concepts, Foscarin et al. [23] created music theory concept-focused MIDI datasets to probe NN layer outputs, like Wei et al. [58], but focused their experiments on the work of classical music composers. Training a `ResNet-50` model to classify an excerpt's piano-roll representation by composer, they extracted its penultimate layer embeddings from their datasets focused on the higher-level music theory concepts of alberti bass, "difficult-to-play music", and contrapuntal texture. Given binary classification tasks, they adapted the Testing with Concept Activation Vectors (TCAV) technique by Kim et al. [28] and calculated vectors corresponding to each concept. They used the CAVs to perturb the embeddings of music used to train the `ResNet-50` model. By calculating the directional derivatives of these perturbations with respect to the model's output, they obtained a measure of how significant a concept is to a composer's music.

Focusing on generative models, Ma and Xia [38] created a dataset of triads to probe `MusicGen` attention heads. Elaborating on [58], they supplemented their training with interventions to supplement the task of classifying chord quality

[2] https://github.com/GiantSteps/giantsteps-mtg-key-dataset, last accessed 22 Jan 2026.

with chord root information. Ma et al. [37] created datasets to probe the intermediate layer outputs of `MusicGen` models as well as `MERT` [35] on the tasks of chord quality and chord root. Additionally, they used the NSynth dataset [19] to probe these representations on the tasks of pitch and timbre classification. Elaborating on the basic probing classification technique, they investigated how modifying `MusicGen-small` layer outputs affected pitch and chord root classification.

Facchiano et al. [21], Panda et al. [41], and Zhao et al. [60] use probing classifiers to derive fine-grained control of music generation. [21] use `GPT-o1` to produce sets of prompts for `MusicGen-melody` to generate music in fast and slow tempi as well as bright and dark timbres. Calculating the embedding means at a given layer for each of these sets, they use the resulting tempo and timbre steering vectors to control their respective aspects in generation. Using the SynTheory [58] and GTZAN [53] datasets, [41] calculate steering vectors using the `DiffMean` technique and linear probes to intervene on generated music genre and instrumentation. [60] used SynTheory to both probe `MusicGen-large` using kernel ridge regression and control generation using recursive feature machines.

3 Datasets

We extend Wei et al.'s SynTheory [58], which focused on tempi, time signatures, notes, scales, intervals, triad chords, and triad chord progressions, by constructing datasets featuring five advanced musical concepts (see Table 1). Two datasets focus on the pitch-independent concepts of **polyrhythms** and **dynamics** expression patterns. The other three datasets focus on the harmony-related concepts of **seventh chords**, **mode mixture**, and **secondary dominants**.

To generate musical examples for our datasets, we emulate [58] and procedurally render 4-second MIDI files using the Python `mido`[3] library. These MIDI files are realized as WAV files using `RustySynth`[4] with the `TimGM6mb`[5] soundfont. The MIDI instruments are chosen from five non-pitched percussive instruments for the non-pitch-centric datasets and from the 92 pitched MIDI instruments used by [58] for the pitch-centric datasets. The five percussion instruments slightly diverge from [58]'s choices in the interest of timbral variety and include "Agogo", "Woodblock", "Taiko Drum", "Melodic Tom", and "Snare Drum 1". Like [58], we construct MIDI files for each non-pitch-centric concept example with reverb levels of 0, 63, and 127 and randomized start-time offsets.

Polyrhythms are two streams of constant pulses occurring at separate coprime rates and are defined by their integer ratio [40]. Given coprime integers $r_i, r_j = 2, \ldots, 11$, the polyrhythms dataset (Fig. 1, top row) consists of 30,600 examples of polyrhythms with ratio $r_i : r_j$ ($r_i < r_j$) realized over one 4/4 bar. We rendered these with instrument pairs i_i, i_j and considered assignments (i_i to r_i, i_j to r_j) and (i_j to r_i, i_i to r_j) as distinct but referring to the same polyrhythm

[3] https://github.com/mido/mido, last accessed 22 Jan 2026.
[4] https://github.com/sinshu/rustysynth, last accessed 22 Jan 2026.
[5] http://sourceforge.net/p/mscore/code/HEAD/tree/trunk/mscore/share/sound/, last accessed 22 Jan 2026.

Table 1. The five datasets consist of 153,876 examples in total.

Dataset	Examples	Classes	Pitch-Centric	Balanced
Polyrhythms	30,600	34	No	Yes
Dynamics	41,580	7	No	No
Seventh Chords	35,328	8	Yes	Yes
Mode Mixture	19,872	2	Yes	Yes
Secondary Dominants	26,496	3	Yes	Yes

$r_i : r_j$. To ensure invariance of the probed concept to overall speed, we rendered each polyrhythm as one bar at 60 BPM, two at 120, and three at 180. Like [58], we introduce randomized offsets in start times of polyrhythm figures within the MIDIs. However, we limit two offset start times to values less than a 16^{th} note at 60 BPM (250 ms) so that each example has at least one complete polyrhythm iteration. We also include the offset of 0 ms as well as two offsets with values up to 1,600 ms to ensure examples with the lowest r_i value of 2 have two sounding pulses. For consistency and to ensure an even distribution throughout the possible range, we set these values from a NumPy random number generator.

The **dynamics** dataset (Fig. 1, bottom row) focuses on classifying the seven dynamic patterns of a hairpin (a crescendo followed by a decrescendo [29]), "reverse hairpin" (a decrescendo followed by a crescendo), subito transition from soft to loud dynamic, subito transition from loud to soft, crescendo, decrescendo, and no dynamic change ("flat"). All examples are realized over one 4/4 bar at 60 BPM and consist of a constant stream of notes of quarter-note subdivisions two through eight. All patterns aside from "flat" are enumerated over all pairs of soft dynamics *pp*, *p*, and *mp* and loud dynamics *mf*, *f*, and *ff* in which the MIDI velocity values for each dynamic are obtained by dividing the maximum velocity value 127 by six, rounding when necessary. We enumerate over all six dynamics for the 1,890 total "flat" examples. For the first four pattern types, we denote the changing point of dynamic direction (the crest of loudness for hairpins and the change in dynamics for subito patterns) inflection points and realize them over the second, third, and fourth beats. These pattern types have 8,505 examples each. Crescendi and decrescendi occur over the full bar and have 2,835 examples each. We use curated offsets of 0 ms and two offsets limited to the maximal duration of an eighth note at 60 BPM (500 ms).

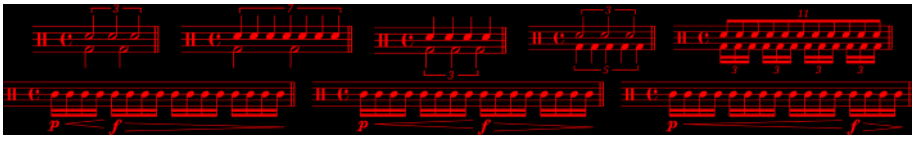


Fig. 1. The **top** row features the **polyrhythm** ratios 2:3, 2:7, 3:4, 3:5, and 11:12. The **bottom** row features the **dynamic** expression of a "hairpin" realized in sixteenth notes with inflection points at beats two, three, and four.

The **seventh chords** dataset (Fig. 2, top row) enables the classification task of discriminating between eight seventh chord qualities: major 7^{th}, minor 7^{th}, major-minor 7^{th}, minor-major 7^{th}, half-diminished 7^{th}, fully diminished 7^{th}, augmented major 7^{th} (major 3^{rd}, major 7^{th}; known as the Lydian augmented chord [34]), and augmented minor 7^{th} (major 3^{rd}, minor 7^{th}; associated with the whole-tone scale [34]). As with the triad dataset in [58], these chords are realized over all twelve pitch classes and all inversions (root, 1^{st}, 2^{nd}, 3^{rd}) in closed voicing and in a register closest to middle C. We note that this process results in identical sounding samples for fully-diminished seventh chords, but leave these in for class balance. The chords are realized as four quarter notes at 60 BPM but, unlike [58], we use a sustain of 90% of a beat to clarify attacks.

The **mode mixture** and **secondary dominant** datasets focus on distinguishing characteristics of chord progressions. The **mode mixture** dataset (Table 2; Fig. 2, middle row) is focused on distinguishing between a major chord progression from SynTheory [58] and a version that substitutes iv for IV and ♭VI (a major chord built on the lowered sixth scale degree) for vi. The **secondary dominant** dataset (Table 3; Fig. 2, bottom row) consists of seventh chord progressions with circle-of-fifths root motion. Our task is to distinguish diatonic versions of progressions from versions with a secondary dominant or a tritone substitution (a substitution of a dominant seventh chord with another with its root a tritone away [34]). We focus on the major and minor progressions in Table 3 where $\flat II^7$ is a major-minor seventh rooted one semitone above the tonicized scale degree. Following [58], both datasets realize each chord in a progression in root position and as a quarter-note at 60 BPM. Each progression is realized over all twelve pitch classes and we transpose chords so that their root is within a tritone of C4. With each dataset, we include metadata such as the identity of a progression at various levels of detail to enable other classification tasks.

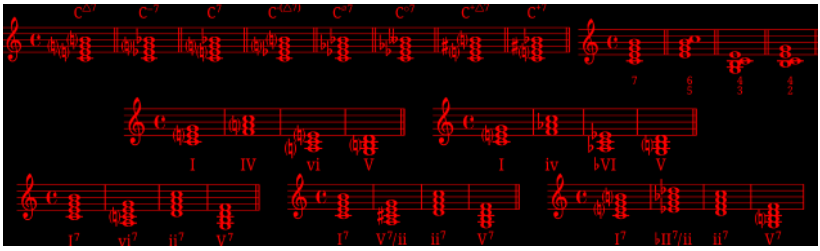


Fig. 2. The **top** row demonstrates the chord qualities in the **seventh chord dataset** as well as a major 7^{th} chord in root position and 1^{st}, 2^{nd}, and 3^{rd} inversions. The **middle** row shows I-IV-vi-V in original and **mode mixture** variants. The **bottom** row shows the I^7-vi^7-ii^7-V^7 major progression from the **secondary dominant** dataset with original, secondary dominant, and tritone substitution variants.

Table 2. We selected progressions from [58] and added **mode mixture** variants.

Original Progression	With Mode Mixture
I – IV – V – I	I – iv – V – I
I – IV – vi – V	I – iv – ♭VI – V
I – V – vi – IV	I – V – ♭VI – iv
I – vi – IV – V	I – ♭VI – iv – V
ii – V – I – vi	ii – V – I – ♭VI
V – IV – I – V	V – iv – I – V
V – vi – IV – I	V – ♭VI – iv – I
vi – IV – I – V	♭VI – iv – I – V
IV – I – V – vi	iv – I – V – ♭VI

Table 3. Major (top) and minor (bottom) chord progressions follow circle-of-fifths root movement in two of the three classes in the **secondary dominant** dataset.

Original Progression	With Secondary Dominant	With Tritone Substitution
$I^{\Delta 7}$-vi^7-ii^7-V^7	$I^{\Delta 7}$-V^7/II-ii^7-V^7	$I^{\Delta 7}$-♭II^7/II-ii^7-V^7
$I^{\Delta 7}$-$I^{\Delta 7}$-$IV^{\Delta 7}$-V^7	$I^{\Delta 7}$-V^7/IV-$IV^{\Delta 7}$-V^7	$I^{\Delta 7}$-♭II^7/IV-$IV^{\Delta 7}$-V^7
$I^{\Delta 7}$-ii^7-V^7-$I^{\Delta 7}$	$I^{\Delta 7}$-V^7/V-V^7-$I^{\Delta 7}$	$I^{\Delta 7}$-♭II^7/V-V^7-$I^{\Delta 7}$
$I^{\Delta 7}$-iii^7-vi^7-V^7	$I^{\Delta 7}$-V^7/VI-vi^7-V^7	$I^{\Delta 7}$-♭II^7/VI-vi^7-V^7
i^7-$VI^{\Delta 7}$-$ii^{\varnothing 7}$-V^7	i^7-V^7/II-$ii^{\varnothing 7}$-V^7	i^7-♭II^7/II-$ii^{\varnothing 7}$-V^7
i^7-i^7-iv^7-V^7	i^7-V^7/IV-iv^7-V^7	i^7-♭II^7/IV-iv^7-V^7
i^7-$ii^{\varnothing 7}$-V^7-i^7	i^7-V^7/V-V^7-i^7	i^7-♭II^7/V-V^7-i^7
i^7-$III^{\Delta 7}$-$VI^{\Delta 7}$-V^7	i^7-V^7/VI-$VI^{\Delta 7}$-V^7	i^7-♭II^7/VI-$VI^{\Delta 7}$-V^7

4 Extracting Latent Representations in Models

We use our five datasets to probe for their corresponding music theory concepts in the latent spaces of OpenAI's `Jukebox` [18], the `small`, `medium`, and `large` variants of Meta's `MusicGen` model [15], and its audio encoder `EnCodec` [16]. The number of layers of each model's encoder (including experiment baselines) and the dimension of their extracted representations are shown in Table 4. We illustrate our entire experiment pipeline in Fig. 3. Following [58], we average over the sequence length dimension of each latent representation before probing.

`Jukebox` is a generative model which conditions on artist and genre with optional lyrics or preexisting audio to output novel audio. Its encoder encodes raw audio at three hierarchical levels of quantization, achieved through training separate VQ-VAE (vector quantized variational autoencoders) using various hop lengths [18]. Like [58], we utilize `jukemirlib`[6] to extract `Jukebox` layer outputs. `jukemirlib` takes as input dataset raw audio sampled at 44.1 kHz, encodes it

[6] https://github.com/rodrigo-castellon/jukemirlib, last accessed 22 Jan 2026.

Table 4. Model layer dimension (Dim.) and number of layers (N.L.) MG(A/S/M/L) = `MusicGen`-(Audio Codec/`Small`/`Medium`/`Large`), Concat. = concatenated mel, chroma, and MFCC features. (+1) refers the outputs of the decoder embedding layer [56].

	Mel	MFCC	Chroma	Concat.	MGA	MGS	MGM	MGL	Jukebox
N.L.	1	1	1	1	1	24 (+1)	48 (+1)	48 (+1)	72
Dim.	768	120	72	960	128	1024	1536	2048	4800

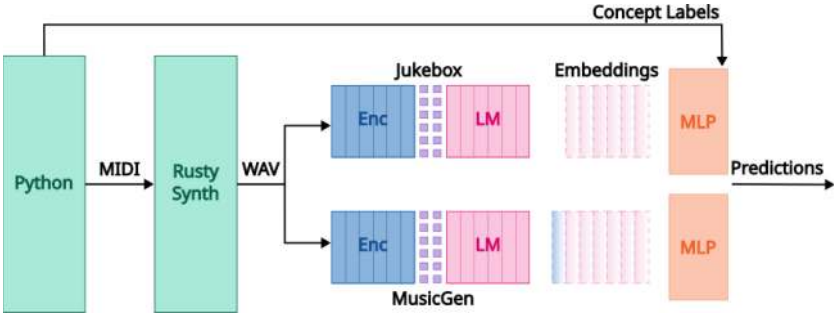


Fig. 3. This diagram illustrates our experimental process from MIDI creation to probing. Parallel pathways show the usage of different models for layer output extraction.

through its VQ-VAE, and extracts the layer outputs from the sparse transformer [13] corresponding to the topmost hierarchical quantization level.

`MusicGen` processes raw audio through the `EnCodec` audio encoder [16], which generates K codebook vectors per audio time step. These vectors are the result of residual vector quantization (RVQ) in which an input is quantized to entries in the first codebook and each successive codebook is learned from the previous quantization error. Given conditioning on T5-encoded text [44], these K codebook vectors are predicted using a "delay" interleaving approach [15]. Following [58], we use raw audio resampled to 32 kHz and processed through `EnCodec` alongside an empty input string for `MusicgenForConditionalGeneration`'s[7] forward pass. We extract the resulting decoder hidden states (outputs of each decoder transformer layer including residual connection) as well as initial decoder embeddings. We apply this process to the `MusicGen-small`, `-medium`, and `-large` variants. Like [58], we also input our audio into `MusicGen-large`'s encoder and extract the output before RVQ.

5 Experimental Design

We follow Wei et al.'s [58] splitting methodology and allocate 70%, 15%, and 15% of each dataset for training, validation, and testing, respectively. To aid reproducibility and consistency across different computational resources, we divide

[7] https://huggingface.co/docs/transformers/v4.57.0/en/model_doc/musicgen, last accessed 22 Jan 2026.

each dataset into 20 folds where the first 14 are used for training while the next three are used for validation and the last three for testing. The folds were determined using `scikit-learn`'s [11,42] `StratifiedKFold` and we stratify based on task-defined classes for polyrhythms, dynamics, and seventh chords. For the mode mixture and secondary dominant datasets, we stratify based on specific chord progressions, which ensures both an even distribution among folds for chord progressions but also the mode mixture type present and the type of secondary dominant used.

Following [58], our classifier probe is a single-hidden-layer MLP in PyTorch with a hidden dimension size of 512 and ReLU activation. The MLP has an initial dropout operating on its inputs and dropout after the first layer's ReLU. We optimize the probe's weight parameters using cross-entropy loss and use normalized inverse-class-proportional weighting for the dynamics dataset experiments as its class proportions are unbalanced. In experiments involving data normalization, we reimplement `scikit-learn`'s `StandardScaler` [11] in PyTorch, as our experiments run utilizing CUDA. The parameters of the scaler, which divides its input by the running element-wise mean and divides by the running element-wise standard deviation, are found through calculations over each training batch over the 100 training epochs. These parameters are frozen during validation and evaluation. Over the 100 epochs, we track accuracy over the validation set and stop training if the best score is not improved upon over the last 10 epochs. For each experiment, we save the model with the highest validation accuracy. For our unbalanced dynamics dataset, we instead track balanced accuracy.

We adopt [58]'s method of hyperparameter grid search for probes, making adjustments when necessary. When models have only one layer (all baselines and `MusicGen`'s audio encoder), we search over learning rates $10^{-3}, 10^{-4}, 10^{-5}$; dropouts $0.25, 0.5, 0.75$; using or not using data normalization; batch sizes 64 and 256; and weight decay values of $0, 10^{-3}, 10^{-4}$ for the ADAM optimizer. For multi-layered model experiments (`Jukebox` and `MusicGen-small`, `-medium`, and `-large`), we fix all hyperparameters (a 10^{-3} learning rate, using data normalization, a batch size of 64, and no weight decay), aside from dropout, in order to reduce the number of trials needed. In these experiments, we search for the transformer layer (including decoder embedding layer [56]) whose output yields the best accuracy score. We search over dropout values, unlike [58] who fixed it to 0.5. We found fixing all hyperparameters led to `MusicGen` audio codec producing better validation accuracy scores over some datasets (particularly for polyrhythms) than multi-layer `MusicGen` models (particularly `-small`).

We perform experiments using the same baselines as [58] and duplicate their baseline extraction pipeline[8]. They extracted the MFCC, Constant-Q Chroma, and Mel Spectrogram features of the input using `librosa` and concatenated the mean and standard deviations of the zeroth-, first-, and second-order differences across frames. A fourth baseline is derived from concatenating all three baselines.

To explore the strength of linear relationships between our concepts in generative model latent space and their classes, we go beyond the experiment method-

[8] https://github.com/brown-palm/syntheory, last accessed 22 Jan 2026.

ologies of [58] and repeat our experiments using a neural network with no hidden layer. As these probes lack a hidden layer, we remove the ReLU activation function after the first and only layer of neurons but retain the initial dropout.

6 Results and Discussion

In this section, we provide the results from the single-hidden-layer MLP probe and linear probe experiments conducted over all models, baselines, and datasets. For comparison, we provide the results from Wei et al. [58] as Table 5.

6.1 MLP Probe Results

In our MLP results (Table 6), there is a marked difference in performance on the baseline features between our pitch-centric and non-pitch-centric datasets. The baseline experiments on the pitch-centric datasets achieved relatively high accuracies comparable to the generative model experiments. Chroma experiments showed the strongest performance, which can be attributed to the feature directly representing twelve semitones of equal temperament [10]. Although not directly representing musical semitones, mel spectrograms measure spectral power with bins also aligned to human pitch perception [51] and achieve the third strongest baseline performance. The weaker performance of MFCCs can perhaps be attributed to smaller latent dimension, as we follow [58] and collect 20 coefficients per frame compared to the mel's 128. Furthermore, MFCCs are focused on representing a mel spectrogram's shape [26,55] and thus may be less suited for chord-related tasks (particularly when generalizing over inversions). As the concatenated feature contains both mels and chroma, its experiments exhibit the strongest performance across all baselines over the pitch-centric datasets. For our non-pitch-centric datasets, MFCC experiments instead exhibit the strongest performance out of the non-concatenated baselines. As pitch is irrelevant in distinguishing polyrhythms or dynamic patterns, MFCCs rather than mels and chroma are better suited for their classification tasks as the features represent a frame's spectral energy with a smaller-dimension vector.

In our MLP experiments over four datasets and over all models (excluding `MusicGen`'s audio codec), we achieve accuracies over 94%. This is notable as although our evaluated datasets focus on more advanced and higher-level concepts than [58], we are still able to train MLP probes with comparable performance to probes trained on more fundamental concepts. Aside from the `MusicGen-medium` mode mixture results, our accuracies monotonically decrease from deeper models with larger-dimension latent spaces to shallower models with smaller-dimension spaces. This follows intuition as larger-dimension latent spaces and deeper models have more expressive capability and are better able to model a diverse set of concepts. Although [58] found that `MusicGen-small` representations yielded the highest `MusicGen` model accuracies on their fundamental concepts, our results align with Copet et al. [15] who found that the larger `MusicGen` models generally performed better on metrics such as Frechet Audio Distance [27] and the KL-Divergence between ground-truth and generated music tags.

Although our secondary dominant experiments produced relatively weaker results, we achieved scores that are nearly double that of chance and higher (≥ 0.667) over `Jukebox` and `MusicGen` models. `Jukebox`, the largest and deepest model, was the sole model to outperform baseline representations. We attribute these results to the difficulty of the dataset's task. Whereas the mode mixture experiments focus on detecting if major chord progressions borrow chords from their relative minor, the secondary dominant experiments involve classification of chord borrowing from outside a progression's major or minor mode. This consequently requires generalization over both major and minor chord progressions. Another potential source of difficulty could also be a prevalence of irregularly-resolving secondary dominants [25] in the generative models' training data.

Consistent with [58], we found that the `MusicGen` audio codec returned the lowest non-baseline accuracy scores. We suspect that the representation's small dimension (128) compared to the other models ($\geq$1024) may contribute to these results. Furthermore, `EnCodec` is a general purpose audio codec trained on non-music datasets (Common Voice [5], Audio Set [16,24]) in addition to a music dataset (MTG-Jamendo [9]). Thus, its representative capability is not specifically catered towards music concepts like pitch. It is also likely that averaging over the sequence length of its output (for model comparison) is not beneficial for classification tasks. Although `EnCodec` is the audio encoder for the `MusicGen`, `MusicGen` models benefit from training solely on music [15] as well as from calculating their latent representations from unaveraged, discretized `EnCodec` output.

Given the high triad chord probing accuracies achieved in [58], the high accuracies we achieved in seventh chords experiments are not surprising. The accuracies achieved in the mode mixture experiments are higher than [58]'s reported chord progression results over all models and nearly all baselines. Although our classification task operates at a higher level of abstraction, it is also a binary classification class compared to the 19 classes in [58]'s chord progression task. Although the tempo regression task is not directly comparable to our polyrhythm classification task, tempo is the most closely related concept in [58] to polyrhythms and can provide rough expectations of polyrhythm results. Given that polyrhythms characterize relative rather than absolute rates, it is not surprising that we achieved weaker performance in our metric (accuracy) and steeper performance decline in smaller models than [58]'s tempi R^2 scores.

The representations with the highest accuracies largely correspond to the middle or last half of a model's decoder layers but never the final layers (Table 7). This echoes [58]'s results probing for more fundamental music concepts. This also somewhat aligns with the findings in Facchiano et al. [21] for which `MusicGen-melody` layers 10 through 18 (0-indexed, 48 layers total) provided the best representations for tempo and brightness steering vectors. Although this range precedes any best `MusicGen-medium` and `-large` layers, our concepts involve higher-level musical aspects than tempo or timbre brightness. We also observe that the best `Jukebox` and `MusicGen-large` layers occur at equal or greater depths in the pitch-centric datasets than the non-pitch-centric

Table 5. These accuracy results are from [58]'s publication.

Model	Notes	Intervals	Scales	Chords	Chord Prog.	Tempi (R^2)	Time Sig.
Jukebox	0.951	0.995	0.978	0.997	0.971	0.993	1.000
MGL	0.866	0.972	0.905	0.989	0.901	0.965	0.905
MGM	0.851	0.983	0.863	0.989	0.870	0.956	0.883
MGS	0.897	0.995	0.949	0.990	0.942	0.969	0.911
MGA	0.729	0.965	0.383	0.879	0.330	0.947	0.677
Mel	0.712	0.995	0.897	0.988	0.723	0.785	0.827
MFCC	0.467	0.822	0.370	0.863	0.872	0.923	0.688
Chroma	0.954	0.820	0.989	0.994	0.869	0.847	0.672
Concat.	0.941	0.997	0.972	0.992	0.868	0.947	0.833

Table 6. These MLP probe micro-averaged accuracy scores result from evaluating on the test set given the best performing hyperparameters found in grid search.

Model	Polyrhythms	Dynamics	Seventh Chords	Mode Mixture	Secondary Dominants
Jukebox	1.0000	0.9990	0.9958	0.9980	0.9814
MGL	0.9998	0.9684	0.9764	0.9919	0.7560
MGM	0.9976	0.9527	0.9751	0.9802	0.6574
MGS	0.9935	0.9432	0.9724	0.9936	0.7137
MGA	0.7155	0.5376	0.6738	0.8916	0.5279
Mel.	0.4871	0.6572	0.8399	0.9671	0.6203
MFCC	0.7240	0.7720	0.5040	0.8100	0.5783
Chroma	0.3695	0.3933	0.9560	0.9973	0.8708
Concat.	0.7608	0.8600	0.9322	0.9879	0.7825

datasets but cannot make any justifying conclusions from these experiments alone.

6.2 Linear Probe Results

Aside from the dynamics dataset, we observe accuracy decreases over all models and baselines when substituting the MLP with the linear probe (Table 8). This decrease is more prominent with pitch-centric concepts than non-pitch-centric concepts, suggesting that the relationships between chord qualities and progression types and their latent representations are less linear than the relationships between specific rhythms and dynamics and their latent representations. The accuracy decrease from larger to smaller models (including the `MusicGen` audio codec) is more pronounced in the linear probe experiments than in the MLP experiments. As before, this decrease is more prominent in the pitch-centric concept experiments. This suggests that increased number of layers and larger-dimension latent spaces enable generative models to learn relationships between

Table 7. These layer indices (1-indexed) for single-hidden-layer MLP probe experiments correspond to the evaluation accuracy scores.

Model (n layers)	Polyrhythms	Dynamics	Seventh Chords	Mode Mixture	Secondary Dominants
Jukebox (72)	54	56	58	56	58
MGL (48)	32	33	42	43	40
MGM (48)	33	32	45	32	32
MGS (24)	9	16	13	10	12

latent representations and classes that are increasingly linear. It also suggests that pitch-centric concepts benefit most from this added dimension and depth.

Although we intuitively expect that tasks with a higher-level of abstraction with respect to their raw musical material may have relationships with their classes that are more complicated than linear, the largest overall drop is exhibited in the seventh chord dataset accuracies. It is unclear why this dataset's experiments demonstrate the largest declines. We observe that the secondary dominant dataset's results, which also involve seventh chords, exhibit declines in accuracy, but more experiments are needed to explore this potential connection.

The lack of decrease (and in some models, improvement) of accuracy scores from probing the dynamics dataset suggest a strongly linear relationship between the latent representations of the dataset examples and their respective classes. The poorer performance over several of the generative models in the MLP probe experiments compared to the linear probe experiments may suggest that the MLP probe is overparameterized for the classification task and overfit to the training data, despite efforts towards regularization and overfitting prevention through dropout and early stopping. Surprisingly, the score from probing the `MusicGen` audio codec decreased compared to the MLP probe. Again, this may be due to the process of averaging over the sequence length. It is noteworthy that out of the baseline experiments, the mel representation as well as the concatenated baseline (perhaps due to its inclusion of the mel representation) proved resilient to the probe substitution while the MFCC representation did not. This

Table 8. These micro-averaged accuracy scores for linear probes result from evaluating on the test set given the highest performing hyperparameters found in grid search.

Model	Polyrhythms	Dynamics	Seventh Chords	Mode Mixture	Secondary Dominants
Jukebox	1.0000	0.9992	0.9564	0.9970	0.9227
MGL	0.9847	0.9936	0.6863	0.8916	0.6684
MGM	0.9503	0.9825	0.5649	0.8701	0.6735
MGS	0.8481	0.9530	0.5915	0.7909	0.6153
MGA	0.2617	0.3850	0.2114	0.5532	0.3784
Mel.	0.2928	0.6484	0.2914	0.6600	0.4031
MFCC	0.1776	0.5964	0.2510	0.5955	0.4230
Chroma	0.1251	0.2259	0.1665	0.6945	0.5846
Concat.	0.4257	0.7879	0.3656	0.7966	0.6365

suggests that patterns in the mel frequency bins (as in the mel representation) have stronger linear relationships with dynamics classes than patterns in the coefficients describing their overall shape (as in MFCCs).

7 Limitations and Future Work

Our experiments were limited to one partition of the 20 folds for splits and one setting of RNG seeds due to limited computational resources. Although we performed class-based stratified sampling during fold creation and expect similar latent space data distributions in each fold, experiments with different partitions and seeds can provide insight into the statistical significance of our results. Ablation study experiments can provide further insights into generalization from our synthetic data to non-synthetic, real-world music examples. Although we have attempted to account for performance variability through generalizing over polyrhythm tempi and dynamic inflection points, we seek to further these efforts through systematically introducing unevenness in realizing polyrhythm as well as crescendi and decrescendi. For mode mixture and secondary dominants, we seek to investigate voicings in other inversions or systematic approaches to smooth voice leading between chord voices. Additionally, we anticipate that shuffling the chord order in these datasets and not averaging over embedding sequences could provide insights into how identities and types of chord progressions are encoded.

Although we have found that our dataset concepts are encoded in the latent spaces of these generative models, it is beyond the scope of this work to investigate if these models will generate distinct classes in these concepts. It is also beyond the scope of this work to investigate which text prompts or sequence of text prompts can be used to generate distinct concept classes, as we do not vary our text input nor artist or genre information when generating latent representations. From their encoding in latent space, we can only conclude that is possible to generate distinct concept classes and leave their generation for future work.

8 Conclusion

We have shown through the use of single-hidden-layer MLP probing classifiers that the concepts of polyrhythms, dynamics, seventh chords, mode mixture, and secondary dominants are encoded in the latent spaces of the generative music models `Jukebox` and `MusicGen-small`, `-medium`, and `-large`. Through linear probing classifiers, we have shown that there exist strong linear relationships between the dataset generative model encodings and their classes, particularly in the larger models. As these concepts are encoded in the latent spaces of these generative models, we have demonstrated that these models have the capacity to potentially generate distinct classes of these concepts. We have demonstrated through our polyrhythms and dynamics probing experiments that non-pitch-centric concepts beyond tempi and time signatures are encoded by these generative models. Through probing for mode mixture and secondary dominant concepts, we expanded on the work of [58] to demonstrate that higher-level music theoretical concepts are encoded by these models as well.

References

1. Aghazadeh, E., Fayyaz, M., Yaghoobzadeh, Y.: Metaphors in pre-trained language models: probing and generalization across datasets and languages. In: Muresan, S., Nakov, P., Villavicencio, A. (eds.) Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics. vol. 1, pp. 2037–2050. Association for Computational Linguistics (2022). https://doi.org/10.18653/v1/2022.acl-long.144
2. Akhondzadeh, M.S., Lingam, V., Bojchevski, A.: Probing graph representations. In: Ruiz, F., Dy, J., van de Meent, J.W. (eds.) Proceedings of The 26th International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 206, pp. 11630–11649. PMLR (2023). https://proceedings.mlr.press/v206/akhondzadeh23a.html, Accessed 26 Oct 2025
3. Alain, G., Bengio, Y.: Understanding intermediate layers using linear classifier probes. https://doi.org/10.48550/arXiv.1610.01644, Accessed 29 Oct 2025
4. Alain, G., Bengio, Y.: Understanding intermediate layers using linear classifier probes. In: International Conference on Learning Representations (2017). https://openreview.net/forum?id=HJ4-rAVtl, Accessed 28 Oct 2025
5. Ardila, R., et al.: Common voice: a massively-multilingual speech corpus. In: Calzolari, N., et al., (eds.) Proceedings of the Twelfth Language Resources and Evaluation Conference, pp. 4218–4222. European Language Resources Association (2020). https://aclanthology.org/2020.lrec-1.520/, Accessed 30 Oct 2025
6. Belinkov, Y.: Probing classifiers: promises, shortcomings, and advances. Comput. Linguist. **48**(1), 207–219 (2022). https://doi.org/10.1162/coli_a_00422
7. Belinkov, Y., Màrquez, L., Sajjad, H., Durrani, N., Dalvi, F., Glass, J.: Evaluating layers of representation in neural machine translation on part-of-speech and semantic tagging tasks. In: Kondrak, G., Watanabe, T. (eds.) Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 1–10. Asian Federation of Natural Language Processing (2017). https://aclanthology.org/I17-1001/, Accessed 26 Oct 2025
8. Bogdanov, D., et al.: ESSENTIA: an open-source library for sound and music analysis. In: Proceedings of the 21st ACM International Conference on Multimedia, New York, NY, USA, pp. 855–858. MM '13, Association for Computing Machinery (2013). https://doi.org/10.1145/2502081.2502229
9. Bogdanov, D., Won, M., Tovstogan, P., Porter, A., Serra, X.: The MTG-Jamendo dataset for automatic music tagging. In: Machine Learning for Music Discovery Workshop, International Conference on Machine Learning (ICML 2019). Long Beach, CA, United States (2019). http://hdl.handle.net/10230/42015, Accessed 30 Oct 2025
10. Brown, J.C.: Fundamental frequency tracking and applications to musical signal analysis. In: Beauchamp, J.W. (ed.) Analysis, Synthesis, and Perception of Musical Sounds: The Sound of Music, pp. 90–121. Springer (2007). https://doi.org/10.1007/978-0-387-32576-7_2
11. Buitinck, L., et al.: API design for machine learning software: experiences from the scikit-learn project. In: ECML PKDD Workshop: Languages for Data Mining and Machine Learning, pp. 108–122 (2013). https://hdl.handle.net/2268/154357, Accessed 28 Jan 2026
12. Castellon, R., Donahue, C., Liang, P.: Codified audio language modeling learns useful representations for music information retrieval. In: Proceedings of the 22nd International Society for Music Information Retrieval Conference, pp. 88–96. ISMIR (2021). https://doi.org/10.5281/zenodo.5624605, Accessed 31 Oct 2025

13. Child, R., Gray, S., Radford, A., Sutskever, I.: Generating long sequences with sparse transformers (2019). https://doi.org/10.48550/arXiv.1904.10509, Accessed 02 Nov 2025
14. Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: ELECTRA: pre-training text encoders as discriminators rather than generators. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=r1xMH1BtvB, Accessed 01 Nov 2025
15. Copet, J., et al.: Simple and controllable music generation. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems. vol. 36, pp. 47704–47720. Curran Associates, Inc. (2023). https://proceedings.neurips.cc/paper_files/paper/2023/hash/94b472a1842cd7c56dcb125fb2765fbd-Abstract-Conference.html, Accessed 21 Oct 2025
16. Défossez, A., Copet, J., Synnaeve, G., Adi, Y.: High fidelity neural audio compression. Trans. Mach. Learn. Res. (2023). https://openreview.net/forum?id=ivCd8z8zR2, Accessed 21 Oct 2025
17. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Long and Short Papers). vol. 1, pp. 4171–4186. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/N19-1423
18. Dhariwal, P., Jun, H., Payne, C., Kim, J.W., Radford, A., Sutskever, I.: Jukebox: a generative model for music (2020). https://doi.org/10.48550/arXiv.2005.00341, Accessed 21 Oct 2025
19. Engel, J., et al.: Neural audio synthesis of musical notes with WaveNet autoencoders. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 1068–1077. PMLR (2017). https://proceedings.mlr.press/v70/engel17a.html, Accessed 02 Nov 2025
20. Evans, Z., Carr, C., Taylor, J., Hawley, S.H., Pons, J.: Fast timing-conditioned latent audio diffusion. In: Salakhutdinov, R., et al., (eds.) Proceedings of the 41st International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 235, pp. 12652–12665. PMLR (2024). https://proceedings.mlr.press/v235/evans24a.html, Accessed 27 Oct 2025
21. Facchiano, S., et al.: Activation patching for interpretable steering in music generation (2025). https://doi.org/10.48550/arXiv.2504.04479, Accessed 05 Oct 2025
22. Fink, C.: Udio AI music raises $10 million, $6.5 million for Spines AI, more Sora, cinematic AI. Forbes (2024). https://www.forbes.com/sites/charliefink/2024/04/11/udio-ai-music-raises-10-million-65-million-for-spines-ai-more-sora-cinematic-ai/, Accessed 28 Oct 2025
23. Foscarin, F., Hoedt, K., Praher, V., Flexer, A., Widmer, G.: Concept-based techniques for "musicologist- friendly" explanations in deep music classifiers. In: Proceedings of the 23rd International Society for Music Information Retrieval Conference, pp. 876–883. ISMIR (2022). https://doi.org/10.5281/zenodo.7316804, Accessed 29 May 2025
24. Gemmeke, J.F., et al.: Audio set: an ontology and human-labeled dataset for audio events. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 776–780 (2017). https://doi.org/10.1109/ICASSP.2017.7952261, ISSN: 2379-190X

25. Hutchinson, R.: Music Theory for the 21st-Century Classroom. University of Puget Sound (2025). https://musictheory.pugetsound.edu/mt21c/MusicTheory.html, Accessed 30 Oct 2025
26. Jensen, J.H., Christensen, M.G., Murthi, M.N., Jensen, S.H.: Evaluation of MFCC estimation techniques for music similarity. In: 2006 14th European Signal Processing Conference, pp. 1–5 (2006). https://ieeexplore.ieee.org/abstract/document/7071659, Accessed 02 Nov 2025
27. Kilgour, K., Zuluaga, M., Roblek, D., Sharifi, M.: Fréchet audio distance: a reference-free metric for evaluating music enhancement algorithms. In: Interspeech 2019, pp. 2350–2354 (2019). https://doi.org/10.21437/Interspeech.2019-2219
28. Kim, B., et al.: Interpretability beyond feature attribution: quantitative testing with concept activation vectors (TCAV). In: Proceedings of the 35th International Conference on Machine Learning. vol. 80, pp. 2668–2677. PMLR (2018). https://proceedings.mlr.press/v80/kim18d.html, Accessed 15 Oct 2025
29. Kim, D.H.S.: The Brahmsian hairpin. 19th-Century Music **36**(1), 46–57 (2012). https://doi.org/10.1525/ncm.2012.36.1.046
30. Knees, P., et al.: Two data sets for tempo estimation and key detection in electronic dance music annotated from user corrections. In: Proceedings of the 16th International Society for Music Information Retrieval Conference, pp. 364–370. ISMIR (2015). https://doi.org/10.5281/zenodo.1414996, Accessed 28 Oct 2025
31. Koo, J., Wichern, G., Germain, F.G., Khurana, S., Le Roux, J.: Understanding and controlling generative music transformers by probing individual attention heads. In: IEEE ICASSP Satellite Workshop on Explainable Machine Learning for Speech and Audio (XAI-SA) (2024). https://www.merl.com/publications/TR2024-032, Accessed 25 Oct 2025
32. Koo, J., Wichern, G., Germain, F.G., Khurana, S., Le Roux, J.: SMITIN: self-monitored inference-time intervention for generative music transformers. IEEE Open J. Sign. Process. **6**, 266–275 (2025). https://doi.org/10.1109/OJSP.2025.3534686
33. Law, E., West, K., Mandel, M.I., Bay, M., Downie, J.S.: Evaluation of algorithms using games: the case of music tagging. In: Proceedings of the 10th International Society for Music Information Retrieval Conference, pp. 387–392. ISMIR (Oct 2009). https://doi.org/10.5281/zenodo.1417647, Accessed 28 Oct 2025
34. Levine, M.: The Jazz Theory Book. Sher Music (1995)
35. Li, Y., et al.: MERT: acoustic music understanding model with large-scale self-supervised training. In: Kim, B., Yue, Y., Chaudhuri, S., Fragkiadaki, K., Khan, M., Sun, Y. (eds.) International Conference on Representation Learning. vol. 2024, pp. 12181–12204 (2024). https://proceedings.iclr.cc/paper_files/paper/2024/hash/33dffa2e3d2ab74a783d1a8c292f66d9-Abstract-Conference.html, Accessed 28 Oct 2025
36. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach (2019). https://doi.org/10.48550/arXiv.1907.11692, Accessed 01 Nov 2025
37. Ma, W., Li, X., Xia, G.: Do music LLMs learn symbolic concepts? a pilot study using probing and intervention. In: Audio Imagination: NeurIPS 2024 Workshop AI-Driven Speech, Music, and Sound Generation (2024). https://openreview.net/forum?id=uvzw0gS0Nn, Accessed 29 May 2025
38. Ma, W., Xia, G.: Exploring the internal mechanisms of music LLMs: a study of root and quality via probing and intervention techniques. In: International Conference on Machine Learning (ICML) Workshop on Mechanistic Interpretability (2024). https://openreview.net/forum?id=Kr6nkNa4TQ, Accessed 29 May 2025

39. McCallum, M.C., Korzeniowski, F., Oramas, S., Gouyon, F., Ehmann, A.F.: Supervised and unsupervised learning of audio representations for music understanding. In: Proceedings of the 23rd International Society for Music Information Retrieval Conference, pp. 256–263. ISMIR (2022). https://doi.org/10.5281/zenodo.7316644, Accessed 28 Oct 2025
40. Møller, C., Stupacher, J., Celma-Miralles, A., Vuust, P.: Beat perception in polyrhythms: time is structured in binary units. PLoS ONE **16**(8), e0252174 (2021). https://doi.org/10.1371/journal.pone.0252174
41. Panda, D., et al.: Fine-grained control over music generation with activation steering (2025). https://doi.org/10.48550/arXiv.2506.10225, Accessed 05 Oct 2025
42. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**(85), 2825–2830 (2011). http://jmlr.org/papers/v12/pedregosa11a.html, Accessed 22 Jan 2026
43. Pereira, I., Araújo, F., Korzeniowski, F., Vogl, R.: MoisesDB: a dataset for source separation beyond 4-stems. In: Proceedings of the 24th International Society for Music Information Retrieval Conference, pp. 619–626. ISMIR (2023). https://doi.org/10.5281/zenodo.10265363, Accessed 22 Jan 2026
44. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**(140), 1–67 (2020). http://jmlr.org/papers/v21/20-074.html, Accessed 31 Oct 2025
45. Rafii, Z., Liutkus, A., Stöter, F.R., Mimilakis, S.I., Bittner, R.: MUSDB18 - a corpus for music separation (2017). https://doi.org/10.5281/zenodo.1117372, Accessed 22 Jan 2026
46. Rohrmeier, M., Pearce, M.: Musical syntax I: theoretical perspectives. In: Bader, R. (ed.) Springer Handbook of Systematic Musicology, pp. 473–486. Springer (2018). https://doi.org/10.1007/978-3-662-55004-5_25
47. Sarmiento, I.G.: AI-generated music is here to stay. will streaming services like Spotify label it? National Public Radio (2025). https://www.npr.org/2025/08/08/nx-s1-5492314/ai-music-streaming-services-spotify, Accessed 30 Oct 2025
48. Schneidermann, N., Hershcovich, D., Pedersen, B.: Probing for hyperbole in pretrained language models. In: Padmakumar, V., Vallejo, G., Fu, Y. (eds.) Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics. vol. 4, pp. 200–211. Association for Computational Linguistics (2023). https://doi.org/10.18653/v1/2023.acl-srw.30
49. Singh, N., Cherep, M., Maes, P.: Discovering and steering interpretable concepts in large generative music models. In: NeurIPS 2025 Workshop AI4Music (2025). https://openreview.net/forum?id=jVSlJk5qNA, Accessed 22 Jan 2026
50. Spotify: Spotify strengthens AI protections for artists, songwriters, and producers (2025). https://newsroom.spotify.com/2025-09-25/spotify-strengthens-ai-protections/, Accessed 30 Oct 2025
51. Stevens, S.S., Volkmann, J.: The relation of pitch to frequency: a revised scale. Am. J. Psychol. **53**(3), 329–353 (1940). https://doi.org/10.2307/1417526
52. Suno Inc.: Suno model timeline & information (2025). https://help.suno.com/en/articles/5782721, Accessed 28 Oct 2025
53. Tzanetakis, G.: Manipulation, analysis and retrieval systems for audio signals. Ph.D thesis, Princeton University (2002)
54. Tzanetakis, G., Cook, P.: Musical genre classification of audio signals. IEEE Trans. Speech Audio Process. **10**(5), 293–302 (2002). https://doi.org/10.1109/TSA.2002.800560

55. Urbano, J., Bogdanov, D., Herrera, P., Gómez, E., Serra, X.: What is the effect of audio quality on the robustness of MFCCs and chroma features? In: Wang, H.M., Yang, Y.H., Lee, J.H. (eds.) Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014), Taipei, Taiwan, pp. 573–578. ISMIR (2014). https://doi.org/10.5281/zenodo.1416276, Accessed 19 Oct 2025
56. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017). https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html, Accessed 31 Oct 2025
57. Weatherbed, J.: Adobe's new AI audio tools can add soundtracks and voice-overs to videos. The Verge (2025). https://www.theverge.com/news/807809/adobe-firefly-ai-audio-generate-soundtrack-speech, Accessed 30 Oct 2025
58. Wei, M., Freeman, M., Donahue, C., Sun, C.: Do music generation models encode music theory? In: Proceedings of the 25th International Society for Music Information Retrieval Conference, pp. 680–687. ISMIR (2024). https://doi.org/10.5281/zenodo.14877427, Accessed 19 Aug 2025
59. Wu, Y., Chen, K., Zhang, T., Hui, Y., Berg-Kirkpatrick, T., Dubnov, S.: Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In: International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5. IEEE (2023). https://doi.org/10.1109/ICASSP49357.2023.10095969
60. Zhao, D., Beaglehole, D., Berg-Kirkpatrick, T., McAuley, J., Novack, Z.: Steering autoregressive music generation with recursive feature machines. https://doi.org/10.48550/arXiv.2510.19127, Accessed 23 Oct 2025