

EXPLORING MUSICAL ROOTS: APPLYING AUDIO EMBEDDINGS TO EMPOWER INFLUENCE ATTRIBUTION FOR A GENERATIVE MUSIC MODEL

Julia Barnett
Northwestern University

Hugo Flores García
Northwestern University

Bryan Pardo
Northwestern University

ABSTRACT

Every artist has a creative process that draws inspiration from previous artists and their works. Today, “inspiration” has been automated by generative music models. The black box nature of these models obscures the identity of the works that influence their creative output. As a result, users may inadvertently appropriate or copy existing artists’ works. We establish a replicable methodology to systematically identify similar pieces of music audio in a manner that is useful for understanding training data attribution. We compare the effect of applying CLMR [1] and CLAP [2] embeddings to similarity measurement in a set of 5 million audio clips used to train VampNet [3], a recent open source generative music model. We validate this approach with a human listening study. We also explore the effect that modifications of an audio example (e.g., pitch shifting) have on similarity measurements. This work is foundational to incorporating automated influence attribution into generative modeling, which promises to let model creators and users move from ignorant appropriation to informed creation. Audio samples accompanying this paper are available at tinyurl.com/exploring-musical-roots.

1. INTRODUCTION

For creators and users of generative models to be informed and responsible, there needs to be a mechanism that provides information about works in the model’s training data that were highly influential upon generated outputs. This would enable both citation of existing work and offer the opportunity to learn about the influences of their creation. We assume a model-generated product that is a copy or near-copy of a work in the model’s training set indicates the model was influenced by that work. To develop methods to automatically detect the influences upon model-generated products it is, therefore, essential to develop good measures of similarity between works.

We define a measure of approximate memorization in deep generative audio models by establishing a thresh-

old for high similarity and memorization of training data against a large repertoire of 5,000,000+ song clips. We take inspiration from the “split-product” measure for image similarity from Somepalli et al. [4], which breaks the embedded feature vector of images into smaller chunks to compare inner products of corresponding localized features. In our work, every audio file is split into 3-second segments (a.k.a. clips), each of which is encoded as a feature vector (either a CLMR [1] or CLAP [2] embedding) produced by a machine learning model trained to encode audio for the purpose of measuring similarity. See Section 3.3.2 for details. We measure similarity between generated clips and training data clips to find similarity between subportions of songs (e.g., a single musical phrase), returning the songs with the most similar clips. We also evaluate the extent to which similarity measured in this way agrees with similarity assessments by human listeners (Section 4).

We apply our approach to VampNet [3], an open-source music audio generation model trained on 795k music songs. VampNet is representative of a widely-used class of generative models: language-model-style generation. This approach is used in AudioLM [5], JukeBox [6], MusicLM [7], SoundStorm [8], among others [9, 10]. While we utilize VampNet as a case study, our evaluation framework is both model and training data agnostic.

This paper makes the following **key contributions**. Primarily, it establishes **an easily replicable methodology and framework to perform training data attribution for a generative music model** (Section 3), which has been **validated in a human-listener study** (Section 4). Second, **we systematically explore the robustness of embedding-based similarity measures for music audio (CLMR and CLAP) to audio perturbations such as pitch shift, time stretch, and mixture with different types of noise** (Section 5.1). Generative models, even when creating near-copies of training data, are likely to add some form of variation to the outputs, making it essential to understand how robust this method is to such anticipated perturbations.

Our formal research questions are:

1. Can we measure similarity between generated music and music in the training data in a way that human listeners would agree with?
2. How do different perturbation types and amounts affect the ability of the evaluated similarity measure(s) to quantitatively identify similar pieces of music?



2. RELEVANT LITERATURE

2.1 Memorization in Non-Audio Generative Models

It is well established that large language models (LLMs) applied to text are capable of memorizing part of their training data [11–18]. LLMs like the 6 billion parameter GPT-J model can memorize at least 1% of training data [19]. If access to the training data is available, it is relatively straightforward to detect when language models copy strings of text verbatim due to the ability to check for exact sequences of tokens.

Memorized images created by generative models pose risks similar to memorized training data from text LLMs such as sensitive data leaks and copyright infringement. Detecting memorization and duplication by image models is fundamentally different from detecting duplication from a text-based language model; instead of memorizing and reproducing items verbatim from the training data, image models create images that are not identical to the training data, but are sufficiently similar to warrant being called content replication [4].

Carlini et al. [20] propose an approximation of a distance metric for memorization in the image space. A generated image whose nearest neighbor in the training data falls closer than a determined threshold (δ), when embedded in the appropriate manifold, is labeled as a memorized example even if not a verbatim copy. Somepalli et al. [4] demonstrate that diffusion models replicate images from training data with high fidelity, setting a lower bound for memorization of Stable Diffusion at 1.88% of generations [21]. We extend this methodology [4, 20] into the audio domain for our paper.

2.2 Audio Retrieval and Music Similarity

2.2.1 Music Similarity in Generative Audio Models

Popularized in the early 2000s, audio fingerprinting [22, 23] aims to detect exact copies of a given piece of audio. In 2006, Shazam popularized this method for the general public with a system utilizing query-by-example for everyday users [24]. Traditional audio fingerprinting (e.g., Shazam [25]) depends on low-level structural details that are not typically regenerated by generative models, so it is not a relevant approach for our methodology.

Most of the limited work examining similarity of audio made by generative models has been in the context of a different purpose, rather than the focus of an in-depth exploration. Examples include creating new strategies for text-to-music generation in order to create more novel songs [26] or brief ad-hoc memorization evaluations at time of release [7, 9]. Perhaps the closest work to our own is by Bralios et al. [27], who examined replication of audio utilizing text-to-audio latent diffusion models for general audio sounds, such as explosions or people cheering. They define replication of training data as “nearly-identical complex spectro-temporal patterns.” They did not perform any subjective evaluation by human listeners to validate their approach to measuring similarity. Our work instead focuses on music, uses a much larger dataset, and is intended

to be easily adoptable by any model creator.

2.2.2 Measuring Audio Similarity with Embeddings

The key to measuring similarity effectively is to have a representation that highlights the task-relevant features. Most popular right now in the age of generative modeling is measuring audio similarity with embeddings. Audio embeddings are continuous vector representations for excerpts of audio that are based on the internal representations of a neural model trained on a proxy task like generative pre-training [28], contrastive learning [1, 2], classification [29], autoencoding [30, 31], and other methods [32, 33].

To use an audio embedding model to measure the similarity of a collection of audio excerpts, we pass the audio signals through the embedding network, which gives us a multi-dimensional vector output for each audio signal: the “audio embedding”. To obtain a list of the most similar audio signals for a given query audio signal, we extract the embeddings for each audio signal using an embedding model of our choice. We then compute a cosine or L1 distance between our query audio signal and the signals in the database, returning a ranked list, where audio signals with higher similarity to the query audio are ranked higher.

The choice of audio embedding model can have a large impact on the results. There are a variety of embeddings capturing different features of audio, such as [1, 2, 28–30, 32, 33]. We focus on CLAP [2] and CLMR [1] embeddings for this work. Both are state-of-the-art (SOTA), produce human-validated similarity in our listening tests, are robust to perturbation, and are able to return relevant top songs.

3. DATA AND METHODOLOGY

3.1 Scope of Analysis

We want to create a system that identifies music both quantitatively similar and subjectively similar to humans. We do not focus on measuring similarity of any individual feature of music (e.g., timber, rhythm, lyrics), but rather use one of two embedding approaches (CLAP or CLMR) to encode audio and examine whether similarity in these embedding spaces aligns with human subjective evaluation (Sec. 4).

3.2 Data and Models Used

Though our approach is model agnostic, we validate our framework on VampNet [3], a generative model trained on 795k songs collected from the internet. VampNet takes a masked acoustic token modeling approach to music audio generation. In the first stage, a Descript Audio Codec (DAC) [31] learns to encode the audio data in a discrete vocabulary of “tokens”, of which it is then trained to model sequences. To create audio files, the token sequence is converted back into the input domain via the DAC decoder. VampNet adopts a masked generative modeling approach with a parallel iterative decoding procedure. Conditioning is done through example audio, either with a prefix (generating a continuation), postfix (generating an introduction) or as infill (masking the middle). We denote musical outputs of VampNet as “vamps.”

We chose VampNet because, at the time of writing, no other model made available both the training data and model weights. We trained another version of VampNet on a smaller training data and found no noticeable differences. While VampNet has a diverse set of music for the training set, our companion website also includes a small analysis of efficacy on various genres in the GTZAN [34] dataset.

3.3 Methodology

3.3.1 Similarity Metric

We define a measure of approximate memorization in generative audio models by establishing a threshold for high similarity and memorization of training data against a large collection of 5 million 3-second song clips drawn from the 795k songs in VampNet’s [3] training data. We take inspiration from the “split-product” measure for image similarity [4, 20], which breaks the images into smaller chunks to compare inner products of corresponding localized features. In our work, every audio file is split into 3-second clips, each of which is encoded as a feature vector. We measure the cosine similarity between generated clips and training data clips to find similarity between sub-portions of songs, returning the most similar songs.

3.3.2 Embeddings

We focus on two main embeddings: (1) contrastive learning of musical representations (CLMR) embeddings [1] and (2) contrastive language-audio pretraining (CLAP) embeddings [2]. We use both CLAP and CLMR embeddings because they can be applied to any dataset of raw music audio without the need for any transformation or fine-tuning, generalize well to out-of-domain datasets, and can be used as a baseline across different models and genres. Utilizing publicly available embeddings that generalize to any dataset is helpful in encouraging adoption.

We put all of the embeddings and their corresponding musical metadata in a vector database (Pinecone) that lets us quickly and efficiently search through millions of embeddings and return the top k similar songs by a chosen similarity metric (e.g., cosine similarity) in milliseconds.

3.3.3 Code and Tools Used

To recreate this study, use the following code and tools. To generate audio using VampNet: github.com/hugofloresgarcia/vampnet. To put audio in a format suitable for Pinecone and to add noise to clips (see Section 5.1) use: github.com/julbarnett/exploring-musical-roots.

4. LISTENING TEST: EXPERIMENTAL DESIGN

Presumably, output that is highly similar to a training audio clip was influenced by that clip. Of course, similarity is in the ear of the listener and many similarity measures do not align with human opinions. To build a replicable framework that will not require other audio researchers to conduct costly and cumbersome human listening tests, we conduct an experiment with human listeners to demonstrate the alignment of our quantitative technique with hu-

man listening. We utilize ReSEval, a framework that enables us to build subjective evaluation of audio tasks deployed on crowdworker platforms [35].

4.1 Dataset Preparation

To create the data for our study we take a random sample of 1,000 3-second clips from VampNet’s training data. For each of these 1,000 clips, we rank its top 10,000 closest clips in the training dataset by cosine similarity. For each embedding (CLAP and CLMR), we fit a Gaussian to the distribution of similarity scores of the top 10,000 clips (histograms in Table 1). The further above the mean a similarity score is, the more similar the clips are. We segment the data into 4 meaningful bins: the mean cosine similarity of the top 10,000 (CLAP: 0.815; CLMR: 0.693), $+1\sigma$ (CLAP: 0.885; CLMR: 0.784), $+2\sigma$ (CLAP: 0.955; CLMR: 0.875) and “random” (CLAP: 0.513; CLMR: 0.151). For the random bin, we take two random sets of 1,000 clips from the full 5 million clip dataset and measure pairwise cosine similarity; the mean similarity of this distribution gives the expected similarity score of random song pairs. We use these bin centers to create bins ± 0.02 for these similarity scores.

4.2 ABX Trials

Cartwright et al. [36, 37] overcame the difficulties of deploying time-consuming lab-based listener studies by utilizing pairwise comparison performed over the web, duplicating the findings of a lab-based test. We leverage these findings and employ a pairwise comparison study design, performing the study on Mechanical Turk (MTurk).

In our study, listeners are asked to perform ABX trials. The target audio clip (X) is presented, along with two other clips (A and B). The listener is asked to rate which clip (A or B) is more like the target X. The proportion of listeners that find A more similar than B is an estimate of the probability that people find A more similar to X than B.

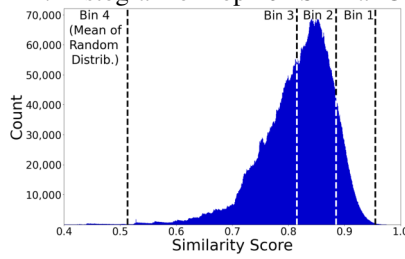
Given a clip X drawn from a random sample of 1,000 clips, one can then create a pair of examples A and B by selecting them randomly from different bins (see Section 4.1). This lets us create ABX trials with known differences in cosine similarity to X between the paired examples A and B. We can then collect statistics on the probability that users will find A more similar to X than B is to X. The greater the difference in cosine similarity, the more skewed we expect the listening results to be. If true, our objective measure’s similarity rankings align with human rankings.

We have 4 bins, resulting in 6 different pair-wise comparisons (bin 1 vs. bin 2, 1v3, 1v4, 2v3, 2v4, and 3v4). To have 150 evaluations per bin (900 evaluations total), we need 90 people to listen to 10 ABX comparisons each. We randomly choose 15 prompt “X” clips from the training data, with their respective 4 clips within the bins chosen as detailed above for the A and B comparison. An example set of clips for an ABX evaluation is at tinyurl.com/exploring-musical-roots.

4.3 Participant Recruitment

We utilized MTurk to recruit 150 participants each to evaluate similarity scores of CLAP and CLMR embeddings.

Human Evaluation Results: ABX Listening Test

		CLAP Embeddings			
		Bin 2	Bin 3	Bin 4	Total
		(0.885 ±0.02)	(0.815 ±0.02)	(0.513 ±0.02)	(All Trials)
CLAP: Histogram of Top 10k Similar Clips 	B				
	A				
	Bin 1	96.2%	98.0%	98.1%	97.4%
	(0.955 ±0.02)	(n = 156)	(n = 150)	(n = 162)	(n = 468)
Bin 2		73.3%	93.6%	83.7%	
(0.885 ±0.02)		(n = 135)	(n = 141)	(n = 276)	
Bin 3			81.5%	81.5%	
(0.815 ±0.02)			(n = 178)	(n = 178)	

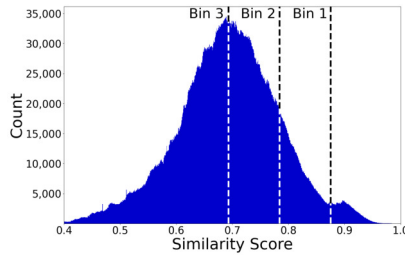
		CLMR Embeddings			
		Bin 2	Bin 3	Bin 4	Total
		(0.784 ±0.02)	(0.693 ±0.02)	(0.151 ±0.02)	(All Trials)
CLMR: Histogram of Top 10k Similar Clips 	B				
	A				
	Bin 1	90.7%	91.0%	98.5%	93.2%
	(0.875 ±0.02)	(n = 150)	(n = 156)	(n = 135)	(n = 441)
Bin 2		71.6%	93.6%	82.7%	
(0.784 ±0.02)		(n = 155)	(n = 157)	(n = 312)	
Bin 3			80.7%	80.7%	
(0.693 ±0.02)			(n = 140)	(n = 140)	

Table 1. Results from the listening experiment. Results show the percent of time listeners rated clip “A” (the clips with higher similarity scores to the prompt “X”) as more similar to the prompt clip “X” than clip “B” (those with lower similarity scores to prompt “X”). Histograms of the top 10k similar songs can be found to the left of the table. Bin regions are shown on these histograms. Bin 3 is centered on the mean of the top 10,000 most similar clips, Bin 2 = +1σ, Bin 1 = +2σ, and Bin 4 is the mean similarity score of a randomly selected clip from the entire training data (not just the top 10k).

We paid each evaluator \$1.50 to annotate 1 set of 10 ABX trials (estimated \$22.50/hour). We recruited US residents with an approval rating of at least 98 and 1,000 approved tasks. We filtered out bots by excluding evaluations that failed a pre-screening listening test. There were no requirements for music expertise beyond passing a listening test.

4.4 Results

Table 1 contains the results of our listening experiment. We found that human evaluations closely aligned with our quantitative metrics. For both CLAP and CLMR evaluations, listeners affirm by a wide margin that clips with higher similarity scores (lower bin numbers) sound more similar to the prompt clip than those with lower scores (higher bin numbers). Clips drawn from the most-similar bin (Bin 1) to the prompt track “X” were rated as more similar to the prompt clip than clips from any other bin 97.4% of the time for CLAP (93.2% for CLMR). For both embeddings, the vast majority of listeners ranked the clips with high similarity to the prompt track (“A”: Bins 1-3) as sounding more similar than the random song (“B”: Bin 4).

5. ANALYSIS OF OBJECTIVE MEASURES

5.1 Robustness to Perturbations

Our second research question focuses on the effect of different perturbations on our methodology’s ability to correctly return similar songs. Any generative music model will add some degree of variation to a training example during the generation process—the aim of these models is

not to replicate the training data exactly. This variation could take many forms (e.g., changing the pitch, speed). Therefore, in this section we evaluate the ability of our methodology to return target songs that have been modified by given perturbations. For varying amounts of each perturbation, we evaluate how frequently the target song (the unmodified clip) is returned as the most similar, within the top 5 similar songs, and within the top 10 most similar songs. The 7 types of perturbations we evaluate are:

- **Pitch shift** (in semitones; range: -12 to 12)
- **Time stretch** (in % of song; range: -20% to +20%)
- **White noise** overlaid on top of music (in dB; range: -30 to 30 dB in relation to original audio clip)
- **“Mash-up”** of two clips from training data (range: 5/95% to 95/5%; e.g., 60/40%)
- **“Mash-up”** of one clip from inside and one outside training data (range: 5/95% to 95/5%; e.g., 60/40%)
- **“Mash-up”** of a prompt clip and the generated vamp (range: 5/95% to 95/5%; e.g., 60/40%)

We selected these because we envision them as common alterations to music that would not render it unrecognizable by a human listener. We are not seeking to evaluate all types of adversarial noise since we are assuming users and creators are working cooperatively with these generative models to create something novel—not acting maliciously.

We evaluate all of the audio perturbations for both CLAP and CLMR embeddings to understand the robustness of our methodology while utilizing different embedding networks. For all perturbations except higher levels

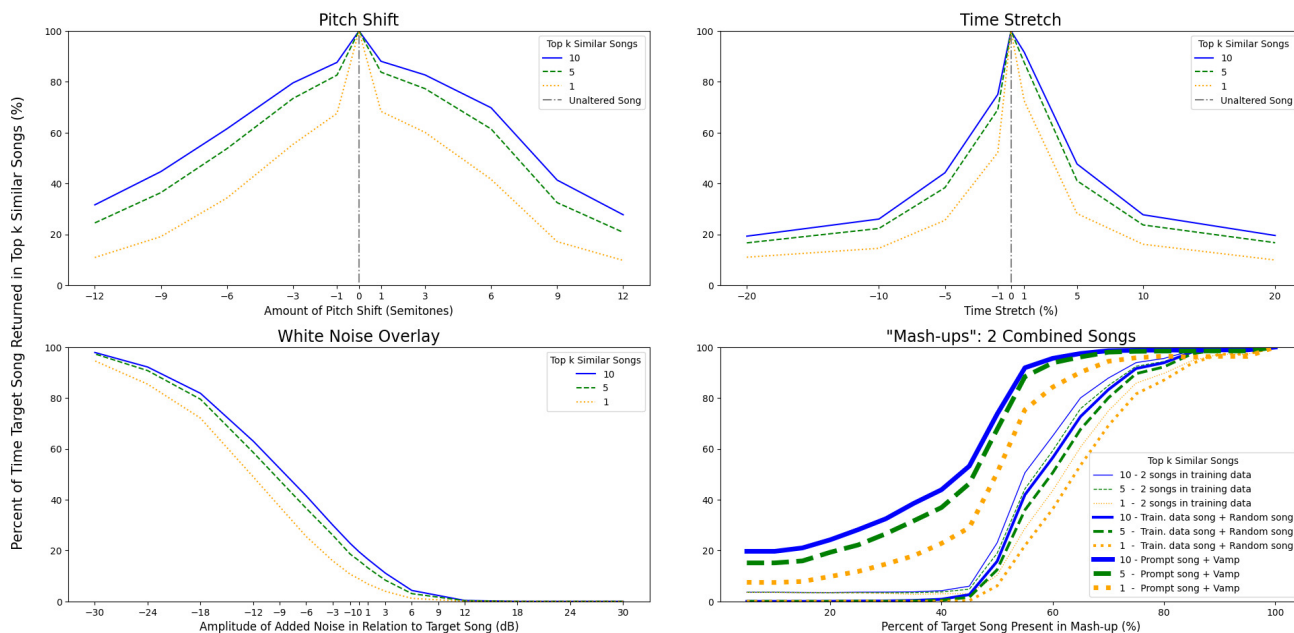


Figure 1. Plots of various amounts of noise perturbations to clips and the percent of the time they were returned in the top $k = 10$, $k = 5$, and $k = 1$ song using our methodology for CLMR embeddings. Displays pitch shift in semitones, time stretch as percent shortened/elongated, white noise overlay in decibels to target clip, and mash-ups of 2 songs in training data, 1 song in training data and one random, and a prompt song and its generated vamp.

of time stretch, CLMR embeddings are more robust than CLAP embeddings; all of the results using CLMR are presented in Figure 1. Example perturbations available at tinyurl.com/exploring-musical-roots.

Pitch shift is a common perturbation to audio that involves raising or lowering the original pitch of an audio clip without adjusting the length of the clip. Notably, human perception is extremely robust to pitch shift. Both embedding types were robust to small pitch shifts; for changes of ± 3 semitones the target song was returned the vast majority of the time. Both embedding types had a lower recall of the target song for larger pitch shifts.

Time stretching audio clips involves speeding up or slowing down audio while keeping the pitch constant. For this perturbation, we evaluate stretching the clip from 20% slower to 20% faster. Both embeddings consistently returned the target song for small amounts of time stretch, but were impacted by larger amounts ($> \pm 10\%$).

White noise overlay involves adding randomly generated white noise to audio clips. We evaluate the noise level in relation to the amplitude of the original clip in decibels, ranging from -30 to 30dB (-30dB being the quietest). Though we were only able to consistently return the target song at quiet levels of white noise overlay (≤ -18 dB) barely perceptible to the human ear, this perturbation has the largest impact on our method’s ability to identify the target track. Luckily, this is not an anticipated type of noise; generative models will add more “musical” variation to songs rather than white noise.

“Mash-ups” of two combined songs are defined here as splicing two clips together at different percentage levels (e.g., for 75/25% the first 2.25 seconds are the target song

and the last 0.75 seconds are some other song). We evaluate three types of “mash-ups”: combining (1) two clips from the training data, (2) one clip from the training data and one outside of the training data, and (3) a prompt track and its generated vamp from VampNet. For each mash-up, we seek to identify the percent of time the target (or prompt) track is returned in the top similar songs. CLMR embeddings only need 50-60% of the target song present in the mash-up to consistently return it in the top similar songs (CLAP need $\geq 80\%$). At each mash-up proportion the model returned the target song (prompt song) for mash-ups with vamps more consistently than for combining two different songs, indicating the vamp is more similar to the prompt song than two randomly selected songs are to each other. When the majority of the song analyzed is the vamp (i.e., $x\text{-axis} \leq 50\%$), it does not return the target (prompt) but rather other songs in the training data.

5.2 Systematic Evaluation of Generative Music Model

As a case study, we systematically evaluate VampNet [3] to demonstrate how to employ this technique to understand training data attribution on both individual songs and an entire model. To evaluate VampNet, we generate 10,000 vamps from 1,000 10-second prompt clips (10 different vamps per clip), and evaluate the most similar clips in the training data to the vamps. We embed each of the 10,000 vamps as a feature vector using both CLMR and CLAP embeddings and analyze the most similar 50 clips by cosine similarity (out of the five million+ clips from VampNet’s training data in our vector store). For each of the 10,000 vamps, the prompt that generated the vamp was rarely among the top similar clips returned by our methodology. Thus, we seek to understand the attribution of the

Systematic Evaluation of Generated Music (Vamps)			
Similarity Score		Vamp & Prompt	Vamp & #1 Similar Track
CLAP	Mean	0.393	0.795
	Median	0.402	0.815
	St. Dev.	0.151	0.084
CLMR	Mean	0.166	0.846
	Median	0.153	0.850
	St. Dev.	0.189	0.054

Table 2. Systematic evaluation of VampNet’s generations. Generated pieces of music (vamps) are less similar to the prompt song provided to the model at generation time than they are to other music from the training data.

rest of the training data on generations. For CLAP embeddings, the average cosine similarity between a prompt clip and generated vamp was 0.393, ($\sigma = 0.151$), whereas on average, the closest clip had a similarity score of 0.795 ($\sigma = 0.084$). CLMR had a similar disparity; full descriptive results are in Table 2. As noted in the above analysis on robustness to perturbations in Section 5.1, our methodology utilizing CLMR (rather than CLAP) embeddings is more robust to perturbations combining elements of new clips with clips present in the training data. Thus for the remainder of this section we will focus on CLMR embeddings for this case study using VampNet.

Leveraging insight from our listening study (Section 4), human evaluation affirms that within CLMR embeddings, music clips with a similarity score of ≥ 0.875 sound significantly more similar than clips with lower similarity scores. For this analysis, we utilize that same top bin as a benchmark and evaluate how often the most similar clips have similarity scores ≥ 0.875 . Findings are presented in Table 3. Over 30% of the vamps generated had at least one song with a similarity score ≥ 0.875 . Looking at scores in 0.02 increments above this benchmark similarity score, almost 20% of vamps had at least one song in the training data with a similarity score ≥ 0.895 , 9% ≥ 0.915 , 3% ≥ 0.935 , and almost 1% ≥ 0.955 . Evaluating more broadly among the top 10 songs, songs with these high similarity scores were concentrated among most similar couple clips, as opposed to having the entirety of the top 10 most similar clips have extremely high similarity scores. This indicates that at least 30% of the time, small sets of songs from the training data were highly influential on generated vamps.

6. DISCUSSION

These findings establish that the framework we propose is an effective means to systematically evaluate the training data attribution on any generative music model. This method is replicable and should be employed by model creators so they are able to have a greater understanding of their outputs. If exposed to end users, this framework also enables anyone to verify if they are copying music and learn about influences of their “novel” generations.

The authors first acknowledge the limitations of this ap-

Vamps with Highly Similar Songs in Training Data				
		Count of Songs in Top k		
		$k = 1$		$k = 10$
Similarity Score	k -clips ($n = 10,000$)	% Total		k -clips ($n = 100,000$)
		$k = 1$	$k = 10$	
≥ 0.955	89	0.89%	254	0.25%
≥ 0.935	317	3.17%	1,223	1.22%
≥ 0.915	924	9.24%	3,139	3.14%
≥ 0.895	1,929	19.29%	8,786	8.79%
≥ 0.875	3,201	32.01%	17,291	17.29%

Table 3. For 10,000 vamps, displays how many top k most similar training data songs were at or above given similarity scores for CLMR embeddings. The lowest similarity score in this table (0.875) corresponds to the highest benchmark (Bin 1) from the human listening test (Sec 4).

proach. First, the scope is intentionally limited to exclude lyrics. As generative music models continue to progress this can become an important area of memorization and copyright infringement, and we encourage future research to examine lyric memorization in tandem with our approach. Our scope also did not include any individualized feature levers for similarity (e.g., timbre or rhythm). We did this to both focus on a low-burden implementation for model creators who would follow this methodology as well as to identify encompassing interacting similarities without isolating any musical feature. However, these could be useful for both model creators and users.

Two potential harms of generative audio models are cultural appropriation and copyright infringement [38]. Our work aims to combat these issues both at the time of output generation and prior to model release. Our method can prevent cultural appropriation by giving users the opportunity to engage with the influences of the music, and prevent copyright infringement if the user realizes the generated piece of music is too similar to the identified influences.

7. CONCLUSION

We have proposed an easily-implementable framework for creators of generative music models to evaluate training data attribution. It can be used to prevent appropriation, copyright infringement, and otherwise uninformed creations, enabling model creators and users to understand the influences on their generated outputs by identifying similar songs in the training data. We evaluated a measure of cosine similarity for two embeddings and verified that they align with human perception with a subjective listening test. We also evaluated how robust our framework is to various forms of perturbations we anticipate models adding to training data during the transformation to “novel” output. We perform a case study on VampNet [3] in order to validate the efficacy of our framework. This work is a step towards transforming a generative model from a crutch replacing artistic knowledge to a tool creators and users alike can use to become better and more informed artists.

8. RESEARCH ETHICS AND SOCIAL IMPACT

The authors of this paper took the ethical considerations and social impact of this work seriously. A recent exhaustive study of the ethical implications of generative audio models [38] found that less than 10% of research on generative audio models published discussed any sort of potential negative impact of their work. We took that as inspiration to center our work around the ethical concerns and attempt to build a bridge between ethicists and generative audio engineers.

As mentioned in the discussion (Section 6), among the negative impacts uncovered for generative music models were the potential for cultural appropriation, copyright infringement, and loss of agency and authorship of the creators. This work aims to combat these issues at the time of generation, on a track by track level. By uncovering the roots of a given piece of generated music, we can empower the user of the model to understand where the music came from and learn about the influences.

A primary concern the authors have for this work is that future model creators will simply use this framework as a checkbox to complete their ethical evaluations. They may use this framework and assume since they did so, there are no other potential societal impacts or ethical harms to consider in regard to generative music models. This work only tackles a portion of the issues, and is only a first step in doing so. Though our method can highlight instances of copyright infringement and cultural appropriation, it by no means will catch everything. Though this can assist with educating users about the influences of their work, it will not solve the potential loss of agency and authorship users and musicians could feel when using these models. It does nothing to address creativity stifling, predominance of western bias, overuse of publicly available data, non-consensual use of training data, or job displacement and unemployment. It also requires energy consumption to generate the embeddings and perform searches, so it contributes to the issue of energy consumption of generative models rather than combating it.

In regard to the experiment utilizing human evaluators to subjectively analyze similar pieces of music, we ensured that our study was in line with institutional review board standards (our study was determined to be exempt). We had a thorough consent form for the crowdworkers and ensured they knew they could quit at anytime without any sort of penalty. We timed ourselves taking the survey and attempted to pay them a fair wage (estimated \$22.50 per hour, higher than any minimum wage in the United States). We even paid users who failed the listening pre-screening test for their time and thus were not able to take our survey, even though they did not contribute data to our study. However, we acknowledge that ethical crowdsourcing goes beyond fair pay [39, 40], and tested the listening test thoroughly prior to launch to be certain there would be no burden to crowdworkers beyond potential boredom. The most sensitive data we had access to were the Mechanical Turk IDs of users, but we held these on secure servers.

The authors determined that the positive impact of this

work outweighed these potential harms, especially since the primary motivation of this work is to address a few existing ethical issues in generative audio. However, it is essential to acknowledge these potential risks and where our method falls short.

9. ACKNOWLEDGMENTS

The authors would like to thank the reviewers for their helpful feedback on this work, as well as Max Morrison for his help with the use of Reproducible Subjective Evaluation (ReSEval) [35]. This research is partially supported by USA National Science Foundation award 2222369.

10. REFERENCES

- [1] J. Spijkervet and J. A. Burgoyne, “Contrastive learning of musical representations,” *arXiv preprint arXiv:2103.09410*, 2021.
- [2] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, “Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation,” in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.
- [3] H. F. Garcia, P. Seetharaman, R. Kumar, and B. Pardo, “Vampnet: Music generation via masked acoustic token modeling,” *arXiv preprint arXiv:2307.04686*, 2023.
- [4] G. Somepalli, V. Singla, M. Goldblum, J. Geiping, and T. Goldstein, “Diffusion art or digital forgery? investigating data replication in diffusion models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6048–6058.
- [5] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi *et al.*, “Audiolm: a language modeling approach to audio generation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.
- [6] 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.
- [7] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi *et al.*, “Musiclm: Generating music from text,” *arXiv preprint arXiv:2301.11325*, 2023.
- [8] Z. Borsos, M. Sharifi, D. Vincent, E. Kharitonov, N. Zeghidour, and M. Tagliasacchi, “Soundstorm: Efficient parallel audio generation,” *arXiv preprint arXiv:2305.09636*, 2023.

- [9] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, “Simple and controllable music generation,” *arXiv preprint arXiv:2306.05284*, 2023.
- [10] J. D. Parker, J. Spijkervet, K. Kosta, F. Yesiler, B. Kuznetsov, J.-C. Wang, M. Avent, J. Chen, and D. Le, “Stemgen: A music generation model that listens,” 2023.
- [11] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song, “The secret sharer: Evaluating and testing unintended memorization in neural networks,” in *28th USENIX Security Symposium (USENIX Security 19)*, 2019, pp. 267–284.
- [12] V. Feldman and C. Zhang, “What neural networks memorize and why: Discovering the long tail via influence estimation,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 2881–2891, 2020.
- [13] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson *et al.*, “Extracting training data from large language models,” in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 2633–2650.
- [14] H. Hu, Z. Salcic, L. Sun, G. Dobbie, P. S. Yu, and X. Zhang, “Membership inference attacks on machine learning: A survey,” *ACM Computing Surveys (CSUR)*, vol. 54, no. 11s, pp. 1–37, 2022.
- [15] Z. Peng, Z. Wang, and D. Deng, “Near-duplicate sequence search at scale for large language model memorization evaluation,” *Proceedings of the ACM on Management of Data*, vol. 1, no. 2, pp. 1–18, 2023.
- [16] A. de Wynter, X. Wang, A. Sokolov, Q. Gu, and S.-Q. Chen, “An evaluation on large language model outputs: Discourse and memorization,” *arXiv preprint arXiv:2304.08637*, 2023.
- [17] K. Tirumala, A. Markosyan, L. Zettlemoyer, and A. Aghajanyan, “Memorization without overfitting: Analyzing the training dynamics of large language models,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 38 274–38 290, 2022.
- [18] S. Biderman, U. S. Prashanth, L. Sutawika, H. Schoelkopf, Q. Anthony, S. Purohit, and E. Raf, “Emergent and predictable memorization in large language models,” *arXiv preprint arXiv:2304.11158*, 2023.
- [19] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer, and C. Zhang, “Quantifying memorization across neural language models,” *arXiv preprint arXiv:2202.07646*, 2022.
- [20] N. Carlini, J. Hayes, M. Nasr, M. Jagielski, V. Shwag, F. Tramer, B. Balle, D. Ippolito, and E. Wallace, “Extracting training data from diffusion models,” *arXiv preprint arXiv:2301.13188*, 2023.
- [21] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.
- [22] P. Cano, E. Batlle, T. Kalker, and J. Haitsma, “A review of audio fingerprinting,” *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 41, pp. 271–284, 2005.
- [23] J. Haitsma and T. Kalker, “A highly robust audio fingerprinting system,” in *Ismir*, vol. 2002, 2002, pp. 107–115.
- [24] A. Wang, “The shazam music recognition service,” *Communications of the ACM*, vol. 49, no. 8, pp. 44–48, 2006.
- [25] A. Wang *et al.*, “An industrial strength audio search algorithm,” in *Ismir*, vol. 2003. Washington, DC, 2003, pp. 7–13.
- [26] K. Chen, Y. Wu, H. Liu, M. Nezhurina, T. Berg-Kirkpatrick, and S. Dubnov, “Musicldm: Enhancing novelty in text-to-music generation using beat-synchronous mixup strategies,” *arXiv preprint arXiv:2308.01546*, 2023.
- [27] D. Bralios, G. Wichern, F. G. Germain, Z. Pan, S. Khurana, C. Hori, and J. L. Roux, “Generation or replication: Auscultating audio latent diffusion models,” *arXiv preprint arXiv:2310.10604*, 2023.
- [28] R. Castellon, C. Donahue, and P. Liang, “Codified audio language modeling learns useful representations for music information retrieval,” *arXiv preprint arXiv:2107.05677*, 2021.
- [29] B. Kim and B. Pardo, “Improving content-based audio retrieval by vocal imitation feedback,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 4100–4104.
- [30] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “High fidelity neural audio compression,” *arXiv preprint arXiv:2210.13438*, 2022.
- [31] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, “High-fidelity audio compression with improved rvqgan,” *arXiv preprint arXiv:2306.06546*, 2023.
- [32] Y. Li, R. Yuan, G. Zhang, Y. Ma, C. Lin, X. Chen, A. Ragni, H. Yin, Z. Hu, H. He *et al.*, “Map-music2vec: A simple and effective baseline for self-supervised music audio representation learning,” *arXiv preprint arXiv:2212.02508*, 2022.
- [33] A. L. Cramer, H.-H. Wu, J. Salamon, and J. P. Bello, “Look, listen, and learn more: Design choices for deep

- audio embeddings,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3852–3856.
- [34] B. L. Sturm, “The gtzan dataset: Its contents, its faults, their effects on evaluation, and its future use,” *arXiv preprint arXiv:1306.1461*, 2013.
- [35] M. Morrison, B. Tang, G. Tan, and B. Pardo, “Reproducible subjective evaluation,” in *ICLR Workshop on ML Evaluation Standards*, April 2022.
- [36] M. Cartwright, B. Pardo, and G. J. Mysore, “Crowdsourced pairwise-comparison for source separation evaluation,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 606–610.
- [37] M. Cartwright, B. Pardo, G. J. Mysore, and M. Hoffman, “Fast and easy crowdsourced perceptual audio evaluation,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 619–623.
- [38] J. Barnett, “The ethical implications of generative audio models: A systematic literature review,” in *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, 2023, pp. 146–161.
- [39] B. Shmueli, J. Fell, S. Ray, and L.-W. Ku, “Beyond fair pay: Ethical implications of nlp crowdsourcing,” *arXiv preprint arXiv:2104.10097*, 2021.
- [40] D. Schlagwein, D. Cecez-Kecmanovic, and B. Hanckel, “Ethical norms and issues in crowdsourcing practices: A habermasian analysis,” *Information Systems Journal*, vol. 29, no. 4, pp. 811–837, 2019.