Musicologist: Charting Music Space with Interpretable Diffusion Trajectories

Coby Mulliken
Department of Computer Science
Brown University
Providence, RI 02912
jacob_mulliken@brown.edu

Daniel Kyte-Zable
Department of Applied Mathematics
Brown University
Providence, RI 02912
daniel_kyte-zable@brown.edu

Shangyang Min
Department of Computer Science
Brown University
Providence, RI 02912
shangyang_min@brown.edu

Justin Chen
Department of Computer Science
Brown University
Providence, RI 02912
shengmai_chen@brown.edu

Kyle Lam
Department of Computer Science
Brown University
Providence, RI 02912
kyle_lam@brown.edu

Abstract

Diffusion models demonstrate remarkable music generation capabilities. However, the intermediate states produced by these models as they iteratively convert Gaussian noise into realistic samples remain poorly understood. This paper presents *Musicologist*, a novel framework for analyzing and understanding how musical concepts emerge during the diffusion process in audio generation models. We use *Stable Audio 1.0*, a leading latent audio diffusion model, to study how musical concepts evolve over time during the reverse diffusion process.[4] Our approach employs two complementary methodologies designed to analyze concept emergence at multiple levels of abstraction. To capture high-level concepts such as genre and mood, we employ a text-guided diffusion pipeline that leverages a diverse caption dataset and truncated sampling to observe concept formation over time. In contrast, for low-level analysis, we combine contrastive audio embeddings with Concept Activation Vectors (CAVs) to identify and quantify fine-grained features like rhythmic drive and timbre.[5] By examining the emergence of high-level musical concepts (such as *genre* and *mood*) and low-level components (like *rhythmic drive*) across diffusion steps, we provide insights into how these concepts develop and interact. This work advances interpretable AI in music generation by offering tools and methodologies to analyze the conceptual building blocks of generated audio. Code is available at `https://github.com/dkytezab/musicologist`.

1 Introduction

Recent advances in diffusion models have dramatically improved the quality and diversity of generative models for music. These models transform Gaussian noise into complex audio signals through a gradual denoising process, which involves systematically reducing noise to reveal high-fidelity audio

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

results across various musical domains. Despite these advancements, the interpretability of these models remain largely opaque, particularly in understanding how semantically meaningful musical concepts, such as rhythm, timbre, or genre, emerge during the reverse diffusion process.

In this work, we introduce *Musicologist*, a novel framework designed to open the "black box" of latent diffusion-based music generators. Rather than concentrating exclusively on output quality, we aim to trace and interpret the trajectories that generated samples follow through the latent space. Building on the opacity of the model's inner workings, our central hypothesis is that musical concepts emerge in a structured and interpretable manner during the denoising process, and that these trajectories can be analyzed to understand the hierarchy and timing of concept formation.

To this end, we utilize *Stable Audio 1.0*, a state-of-the-art text-to-audio diffusion model [4], as our baseline to explore and validate our hypothesis regarding the emergence of musical concepts features like rhythmic drive and timbre with fine-grained precision.

Our research provides both a methodological toolkit for the interpretable analysis of diffusion trajectories and a large dataset of annotated intermediate audio states. Together, these resources facilitate the development of more transparent and controllable generative music models, which have significant implications for AI-assisted music composition and broader model interpretability.

2 Background

Diffusion models Diffusion models enable sampling from arbitrarily complex distributions. In the classic diffusion setup, a *forward process* transforms a distribution p_0 into a standard Gaussian $p_1 = \mathcal{N}(0, \sigma^2 \boldsymbol{I})$, so that for each $t \in [0, 1]$, p_t is an intermediate distribution 'between' the original distribution and the Gaussian. The job of the diffusion model is to learn the *reverse process*, wherein a sample $\boldsymbol{z} \sim p_1$ is transformed into a sample from the target distribution $\boldsymbol{x} \sim p_0$. In practice, this means learning a model q such that, for some set of discretized time steps $t_0, \ldots, t_N \in [0, 1]$, $\boldsymbol{x}_{t_{i+1}} = q(\boldsymbol{x}_{t_i} \mid t = t_i)$.

In diffusion-based audio generation, the target distribution p_0 represents the distribution of audio signals [4]. The forward process gradually adds noise to a clean audio sample until it becomes indistinguishable from random noise, typically Gaussian noise. The reverse process then learns to denoise this random noise step-by-step, conditioned on extra information, such as text prompts or musical scores, to generate novel audio samples that adhere to the characteristics of the training data. For conditioning, audio embeddings, such as those from CLAP [13], are used for rich audio representations with text annotations.

Music modeling In contrast to image generation, which may need to consider local (textures and edges) and global (shape and scene layout) features for cohesiveness, audio generation, especially music, is often considered more complex, requiring extensive context on temporal dependencies (tempo and timbre) and hierarchical structure (composition and tone). Early music generation models used generative adversarial networks (GANs) [2, 14], but have since been supplanted by diffusion-based models operating either on symbolic forms like MIDI [7], directly on waveforms [6, 1], or on a VAE-encoded latent space [4].

Interpreting diffusion trajectories The question of when and how high-level concepts emerge in the outputs of diffusion models is relatively new. Wang and Vastola [11] proposed that image diffusion models generate images "like painter", deciding early in the trajectory on the outlines of objects and their identities, and determining the finer, higher-frequency details toward the end. More recently, Tinaz et al. [10] showed that the large-scale composition of a diffusion-generated image was predictable from the model's internal activations as early as the first diffusion step. Notably, these same questions have yet to answer.

More recently, Tinaz et al. [10] introduced sparse autoencoders (SAE) trained on U-Net activations to identify interpretable features that emerge and evolve as the diffusion process progresses. Their work showed that meaningful concepts such as the artistic style do not appear randomly. Instead, they show up at specific stages during the reverse diffusion process. This means the model builds up its ideas step by step, and we can look at each stage to see when different types of information are added.

VAE models with rich audio representations like *Stable Audio 1.0* [4] provide a unique angle for interpreting diffusion trajectories in audio generation. By first encoding audio into a lower-dimensional latent space, the diffusion process operates within a more semantically organized representation compared to raw waveforms. This latent space can disentangle various audio attributes, making it easier to understand how the diffusion model manipulates these features during the reverse process.

Furthermore, the usage of embeddings like CLAP [13], which contains semantic features from text-annotated audio samples, offers more range for interpretability. These embeddings can capture high-level concepts and characteristics of music with human-level understanding. When a diffusion model is conditioned on text-guided embeddings, the generation processed can be guided by these semantic features. Therefore, analyzing the diffusion trajectory in the latent space, especially those conditioned on rich audio embeddings, can reveal which latent dimensions are most influenced by specific semantic features, how the model progressively incorporates semantic features, and insights on how the model understands the relationship between text and audio.

3 Method

3.1 Prompt generation and annotation

To obtain a sufficiently diverse set of diffusion trajectories, we employed an LLM-based prompt generation pipeline. OpenAI's GPT-4.1 mini was chosen for its low per-token cost and strong performance on large context sizes [8]. The LLM was tasked with generating a diverse set of prompts for a music diffusion model, and given several example prompts as a guideline; 1,000 prompts were then generated in batches of 250. The same LLM was then used to *annotate* each of the prompts it had generated with concepts from NSynth's concept set. (See Appendix [FILL IN] for both system prompts.)

3.2 Diffusion trajectory generation

We chose to use Stable Audio 1.0 as our diffusion model, as it is state of the art open source text-to-audio diffusion model as of May 2025 [4]. Due to limited computational resources, we chose to use 50 step diffusion trajectories, and to collect samples at 10 timesteps, $t = 5, 10, \ldots, 45, 50$. For each of the 1,000 prompts, 7 trajectories were sampled. Due to the constraints of CLAP, our embedding model (discussed later), the sample length was chosen to be 10 seconds. (Sample length is a controllable conditioning parameter of Stable Audio, and can be as high as 47 seconds.)

As Stable Audio is a latent diffusion model, each trajectory began with sampling Gaussian noise in the latent dimension. The sample was then iteratively passed through the pretrained diffusion transformer (DiT), conditioned on the timestep and selected prompt. For 'collectible' timesteps—that is, those in $\{5, 10, \ldots, 50\}$—the sample was then passed through the model's pretrained VAE (variational autoencoder) decoder to obtain a sample in music space. Audio after 10 seconds was removed, the audio was normalized, and the clip was saved. In sum, this method produced a dataset of 70,000 10-second samples.

3.3 Embedding pipeline

Generated clips, as well as concept clips, were fed through LAION-CLAP (Contrastive Language-Audio Pretraining, an open-source audio embedding model) to create 512-dimensional embedding vectors. CLAP was chosen for its ability to generate rich, lower-dimensional representations of high-dimension audio. CLAP is contrastively pretrained on a dataset of 630k audio-text pairs [13].

3.4 Concept classifiers

We chose to use the NSynth dataset to train our concept classifiers due to its diversity and accurate labeling scheme [3]. The NSynth dataset comprises 305,979 clips of single notes, annotated with pitch and timbre. For each timbre type (i.e. `brass`, `warm`, etc.), a corresponding concept dataset was generated [12]. Each dataset consisted of a positive set (i.e. for `brass`, all samples annotated with `brass`), and a negative set (i.e. a random subset of samples *not* labeled as `brass`). Higher-level concept datasets were assembled by compiling sub-concepts datasets.

For each concept dataset, a corresponding classifier was trained on the embeddings to separate the positive and negative sets. The two classifier architectures chosen were a logistic classifier and a support vector machine (SVM). The former was implemented in `torch` and optimized with gradient descent, and the latter was implemented using `scikit-learn`.

4 Experiments & Results

For each concept and corresponding classifier, true positive rate (TPR), true negative rate (TNR), and accuracy were calculated for every timestep. True positive rate was calculated as the percentage of samples classified as having a concept that were annotated with that concept, and true negative rate was calculated analogously. Accuracy was calculated as the percentage of samples classified in agreement with their annotation.

Early emergence of concepts For the concepts which the classifier learned to identify correctly, nearly all gains in TPR (representing the movement of samples into the desired 'concept region') occurred by the 20th diffusion step. (See Figures 3 and 10 for strong examples.) The median diffusion step at which TPR peaked was 20 for both SVM and logistic classifiers, and the means were 23.8 and 18.6 respectively. This suggests that key musical qualities are largely set in stone by the 20th diffusion step — which is to say, more generally, about 40% of the way through the diffusion process.

Good concepts, bad concepts On average, improvements in performance on instrument family-linked classification tasks emerged significantly between steps 10 and 20 and less so between 0 and 10. Interestingly, improvements in performance on instrument source and note-quality linked tasks emerged slightly earlier, between 0–15 diffusion steps. Notably, some concepts—namely vocals (see Figure 8)—despite being highly distinguishable in the NSynth set, were indistinguishable in the diffusion set. This is likely attributable to Stable Audio's poor performance on vocal-related tasks.

Nonlinear trajectories in concept space We examined the distances in concept (i.e. CLAP embedding) space and in audio space between successive diffusion steps. Under both choices of metric (L^1 and L^2), step sizes decreased rapidly until the 20th diffusion step, after which the step size plateaued. A similar phenomenon is observable in the cosine similarities of the embeddings produced by successive steps, where cosine similarity increases rapidly until the 20th step and then plateaus.

5 Conclusion

5.1 Limitations

Use of LLMs Our interpretability work is predicated on being able to generate a synthetic dataset of prompts that approximates the true distribution of music descriptions, and being able to generate sufficient data from this set of prompts. Our limited access to storage and GPUs, however, limited our prompt bank size to roughly one thousand—we imagine that a larger bank would yield greater diversity of samples and in turn more meaningful results from our classifiers. Furthermore, LLMs—as a substitution for human experts—were employed in the annotation of prompts. In most cases, this works well enough; for some concepts, however (`light` and `dark` in particular), the annotations were not particularly accurate, and thus hindered the classifiers' ability to separate the samples.

Depth of probing This work studies only the *outputs* of the diffusion model, rather than the *internal activations* of said model. While this focus is intentional—our central question is when in the diffusion process *perceptual* features emerge, which our framework is well-poised to address—our method does not explain *how* those features emerge. Studying only the former leaves a large blind spot. For example, Tinaz et al. [10] recently showed that the ultimate composition of an image produced by a diffusion model can be accurately predicted from cached model activations as early as the *first* timestep, even though the outputs produced at this stage contain little perceptual information. This suggests that even if—as our main result indicates—perceptual features emerge around 40% of the way through the diffusion process, the model may have 'made its mind up' far earlier; importantly, this phenomenon is impossible to demonstrate through examining outputs alone.

NSynth Dataset While our study remains insightful in interpreting the latent representation in a music generation space, the reliance on the NSynth dataset for training concept classifiers may introduce biases. As NSynth primarily consists of single-note samples with timbres that are fairly faithful to real instruments, but timbre is a fundamentally low-level concept [9]. It may not fully capture the complexity of multi-instrument music or complex audio like in the real world. In multi-instrumental compositions, features such as harmonies, rhythmic interplay, and the interactions among multiple instruments create a rich, highly correlated soundscape. To mitigate this, future work could be enhanced through a supplied polyphonic dataset in addition to the current NSynth dataset, employ data augmentation to simulate real-world musical complexity, and enhance the robustness of our results.

5.2 Future directions

In future work, we aim to explore old and new avenues of diffusion interpretability, namely the usage of concept activation vectors introduced by Kim et al. [5] and sparse autoencoders introduced by Tinaz et al. [10]. We believe that these methods may be able to probe the latent space of diffusion models to retrieve more accurate diffusion trajectories of concepts.

References

[1] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation. *arXiv preprint arXiv:2009.00713*, 2020. URL https://arxiv.org/pdf/2009.00713.

[2] Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment, 2017. URL https://arxiv.org/abs/1709.06298.

[3] Jesse H. Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Douglas Eck, Karen Simonyan, and Mohammad Norouzi. Neural audio synthesis of musical notes with WaveNet autoencoders. *CoRR*, abs/1704.01279, 2017. URL http://arxiv.org/abs/1704.01279.

[4] Zach Evans, Julian D. Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Stable audio open, 2024. URL https://arxiv.org/abs/2407.14358.

[5] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV), 2018. URL https://arxiv.org/abs/1711.11279.

[6] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. DiffWave: A versatile diffusion model for audio synthesis, 2021. URL https://arxiv.org/abs/2009.09761.

[7] Gautam Mittal, Jesse Engel, Curtis Hawthorne, and Ian Simon. Symbolic music generation with diffusion models, 2021. URL https://arxiv.org/abs/2103.16091.

[8] OpenAI. Introducing GPT-4.1 in the API, 2025. URL https://openai.com/index/gpt-4-1/.

[9] Goran Paulin and Marina Ivasic-Kos. Review and analysis of synthetic dataset generation methods and techniques for application in computer vision. *Artificial intelligence review*, 56(9): 9221–9265, 2023.

[10] Berk Tinaz, Zalan Fabian, and Mahdi Soltanolkotabi. Emergence and evolution of interpretable concepts in diffusion models, 2025. URL https://arxiv.org/abs/2504.15473.

[11] Binxu Wang and John J. Vastola. Diffusion models generate images like painters: an analytical theory of outline first, details later, 2024. URL https://arxiv.org/abs/2303.02490.

[12] Megan Wei, Michael Freeman, Chris Donahue, and Chen Sun. Do music generation models encode music theory?, 2024. URL https://arxiv.org/abs/2410.00872.

[13] Yusong Wu*, Ke Chen*, Tianyu Zhang*, Yuchen Hui*, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2023.

[14] Li-Chia Yang, Szu-Yu Chou, and Yi-Hsuan Yang. MidiNet: A convolutional generative adversarial network for symbolic-domain music generation, 2017. URL `https://arxiv.org/abs/1703.10847`.

A Additional Figures

We present additional figures for several selected concepts.

A.1 Pipeline

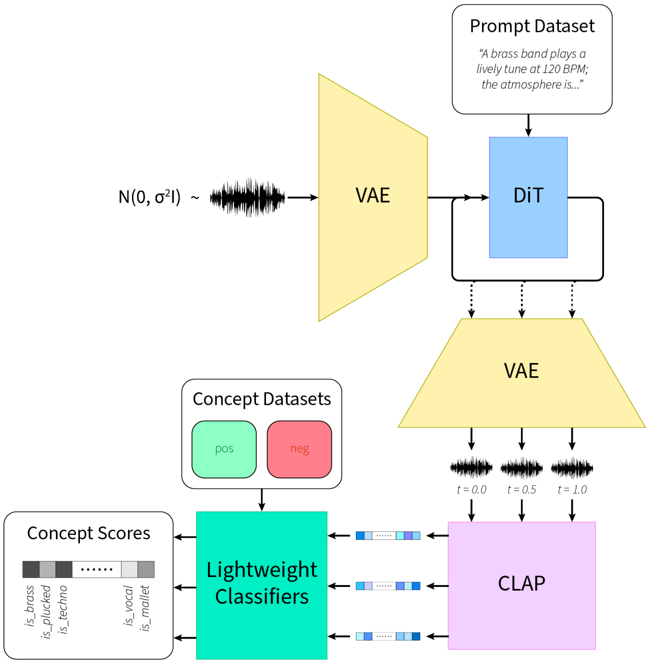

Figure 1: Pipeline of *Musicologist*

A.2 Acoustic

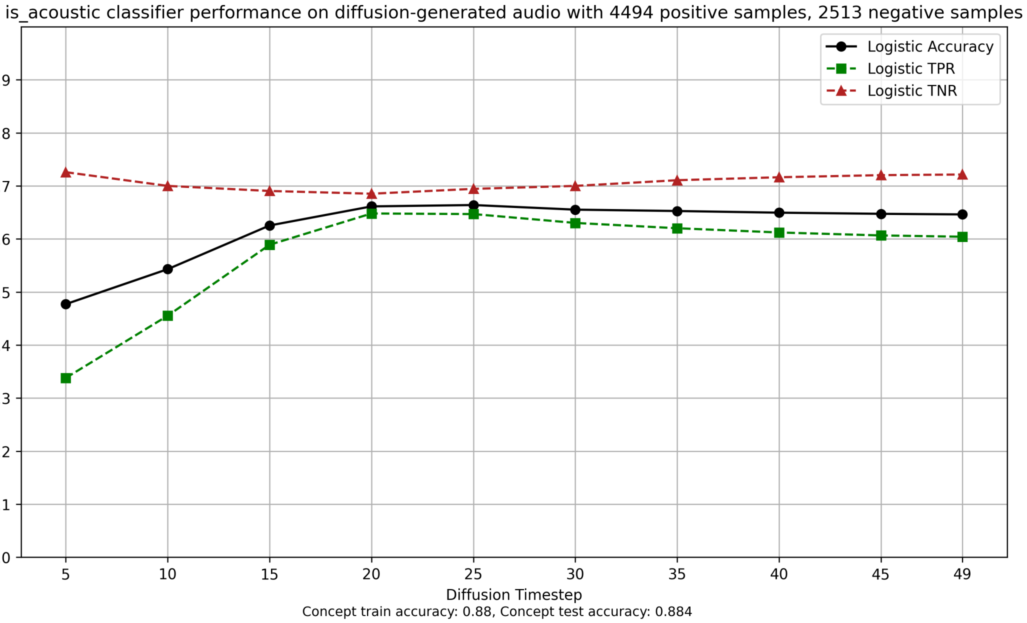

Figure 2

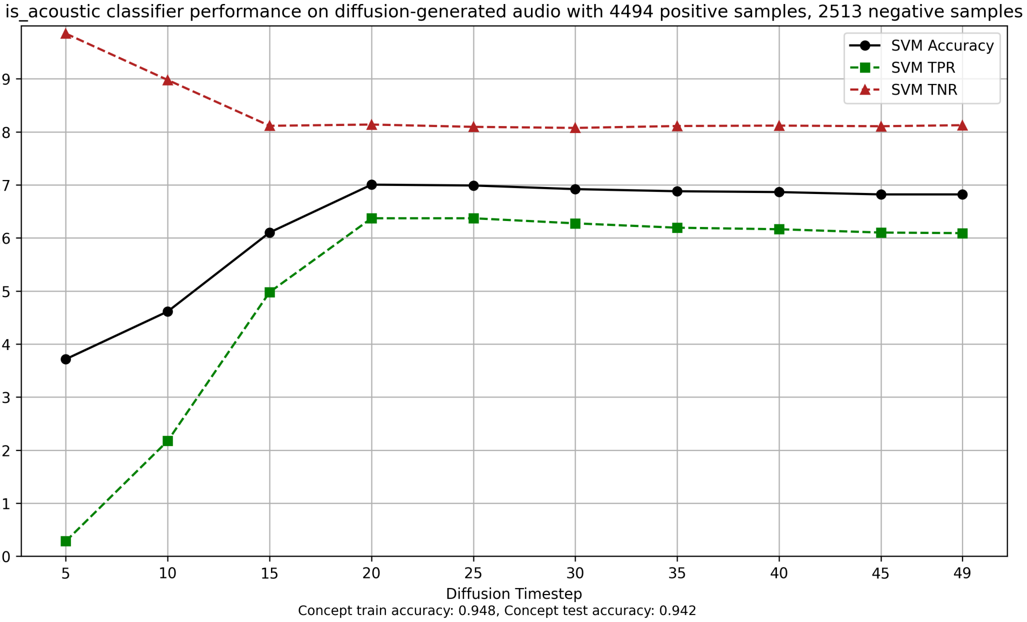

Figure 3

A.3 Distorted

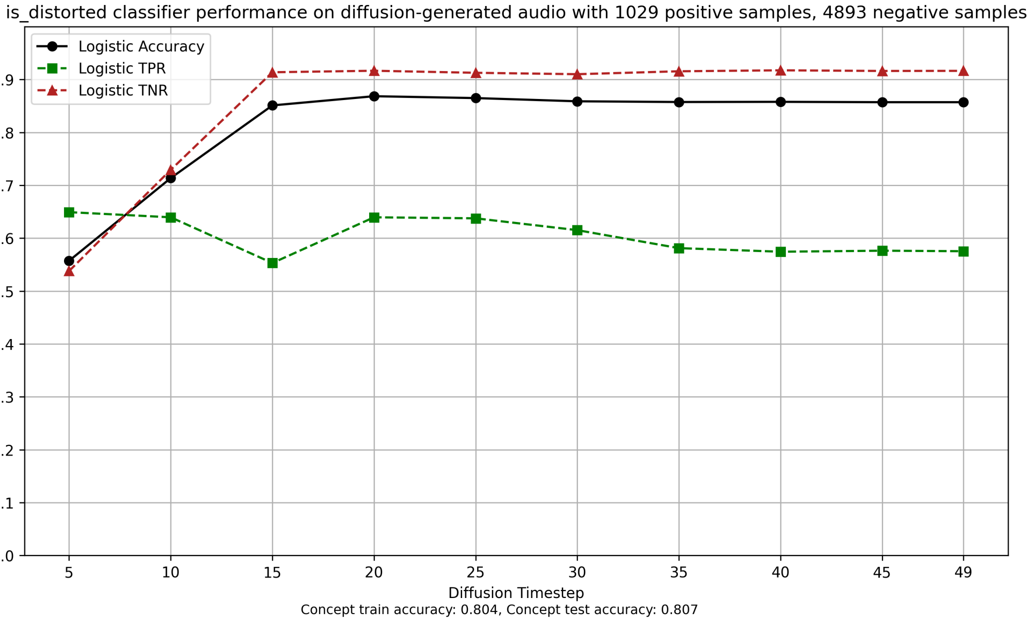

Figure 4

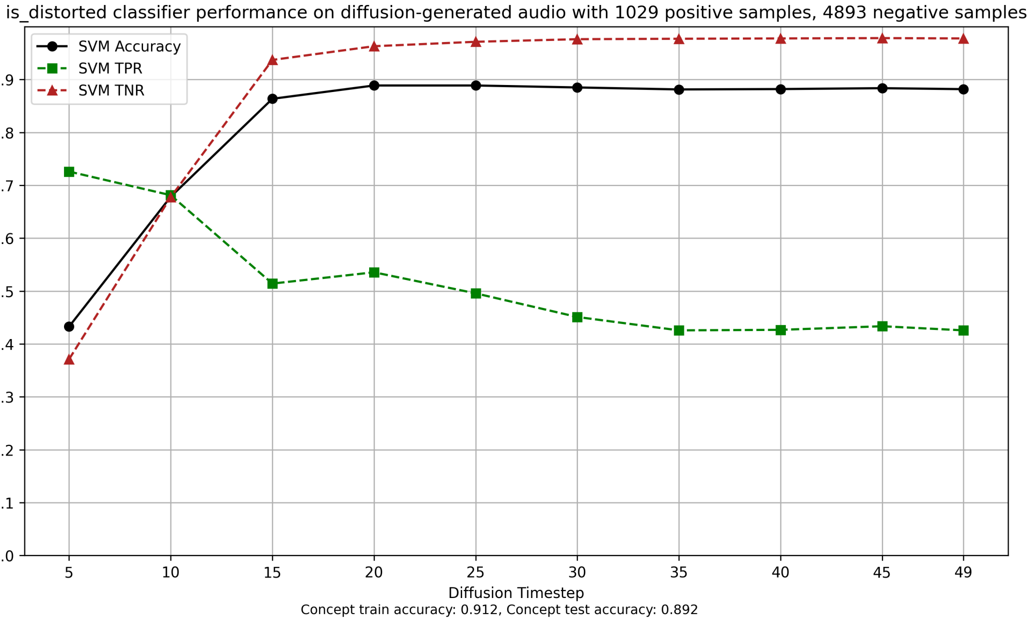

Figure 5

A.4 Electronic

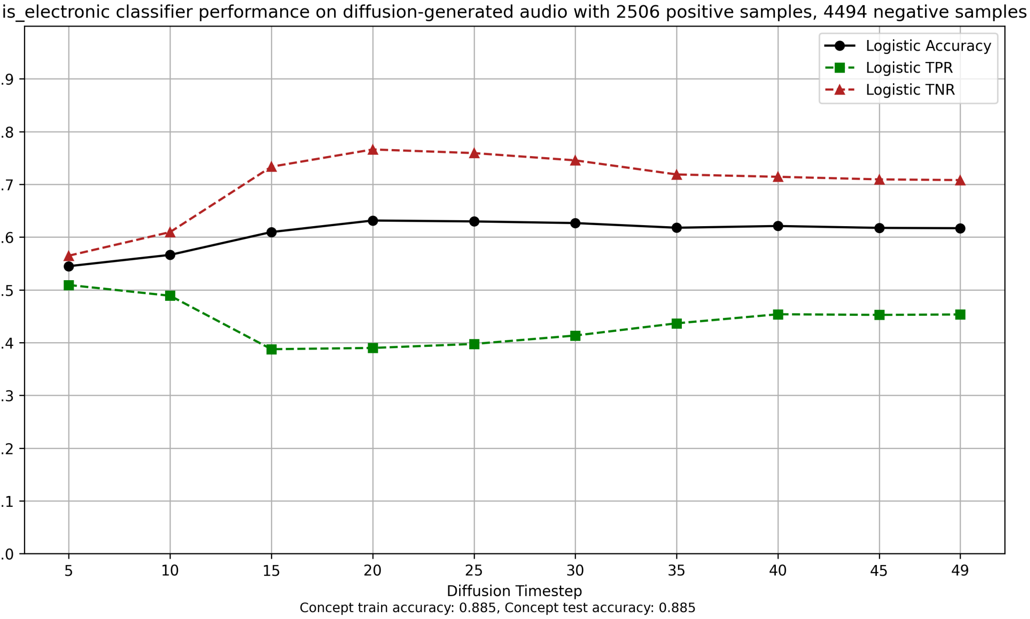

Figure 6

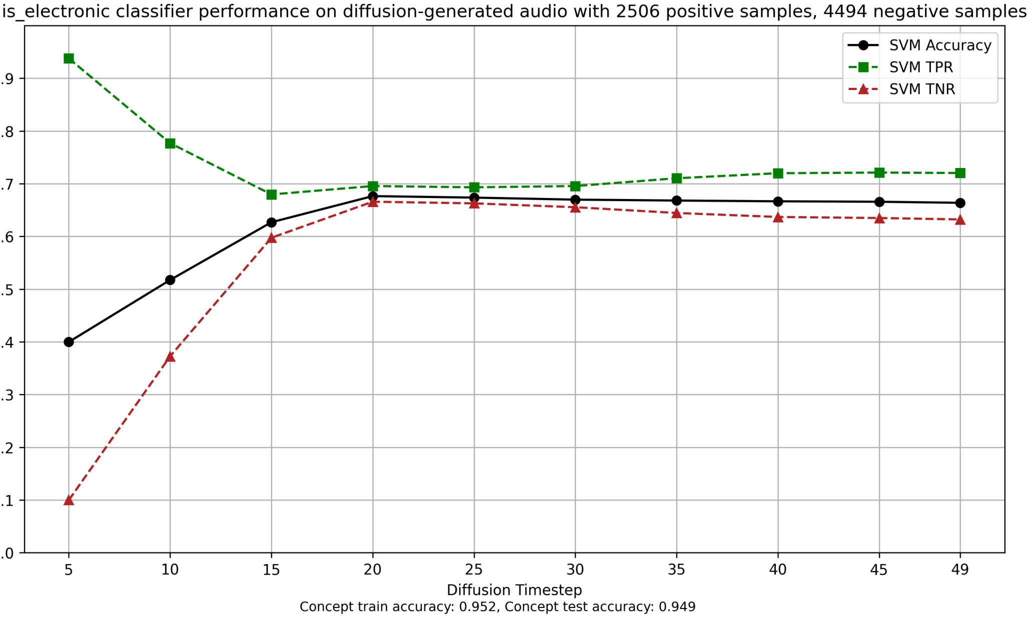

Figure 7

A.5 Vocal

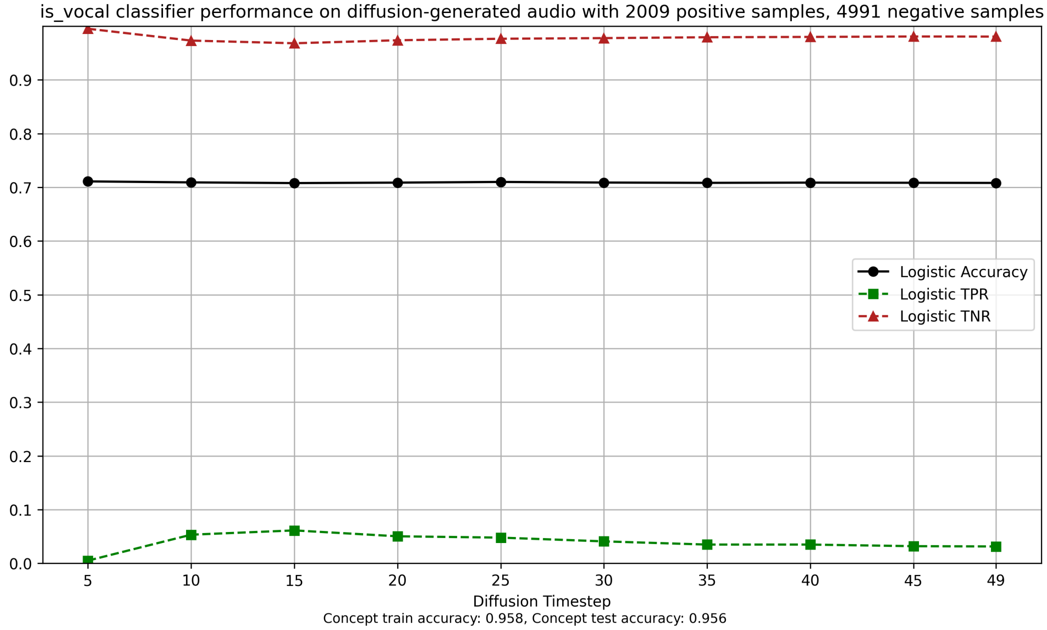

Figure 8

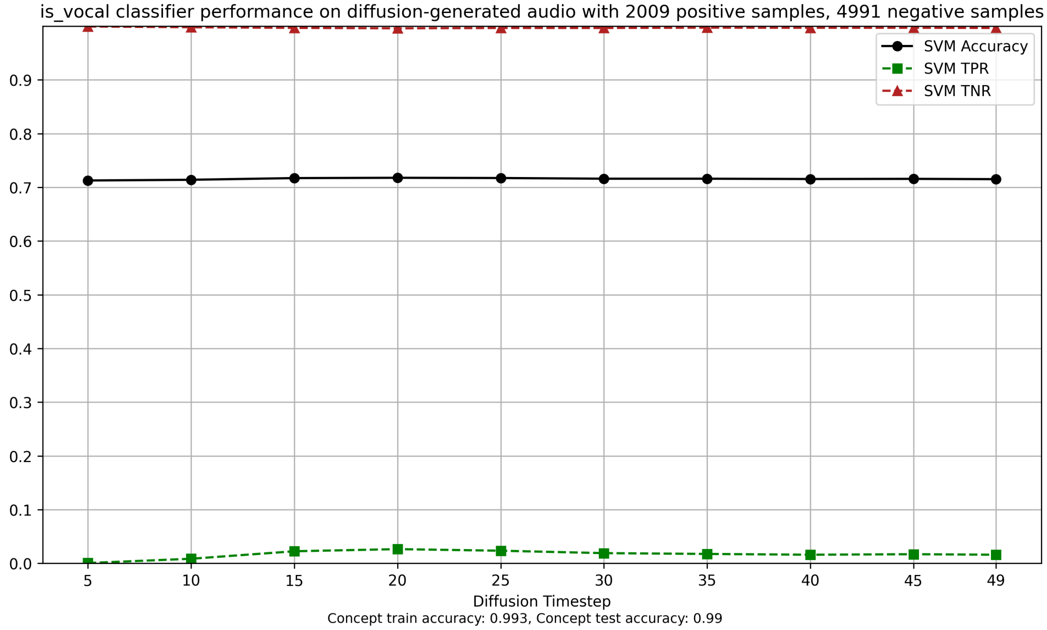

Figure 9

A.6 Guitar

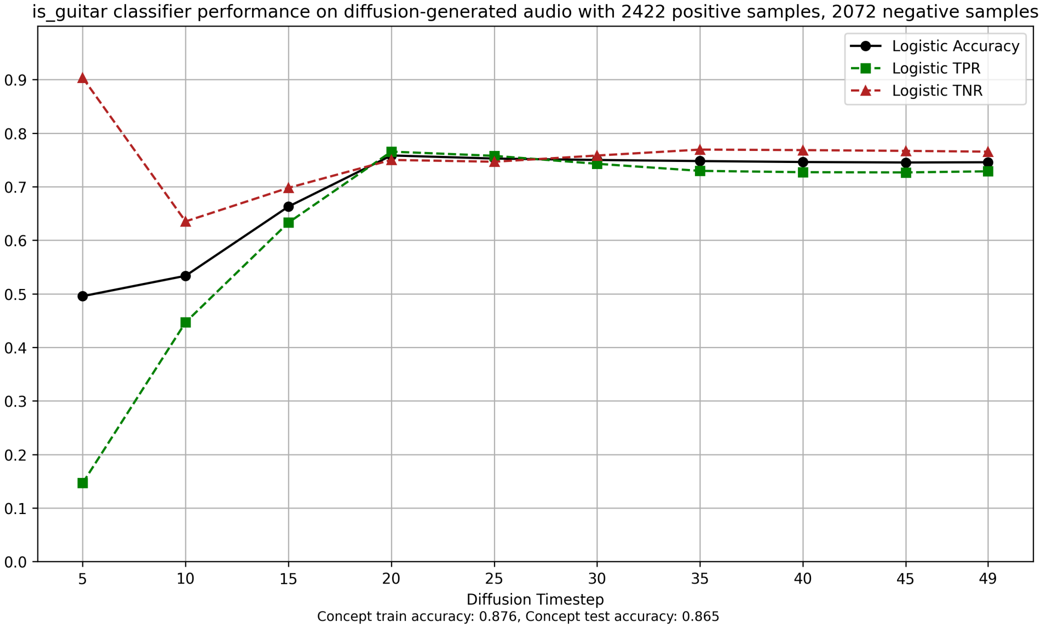

Figure 10

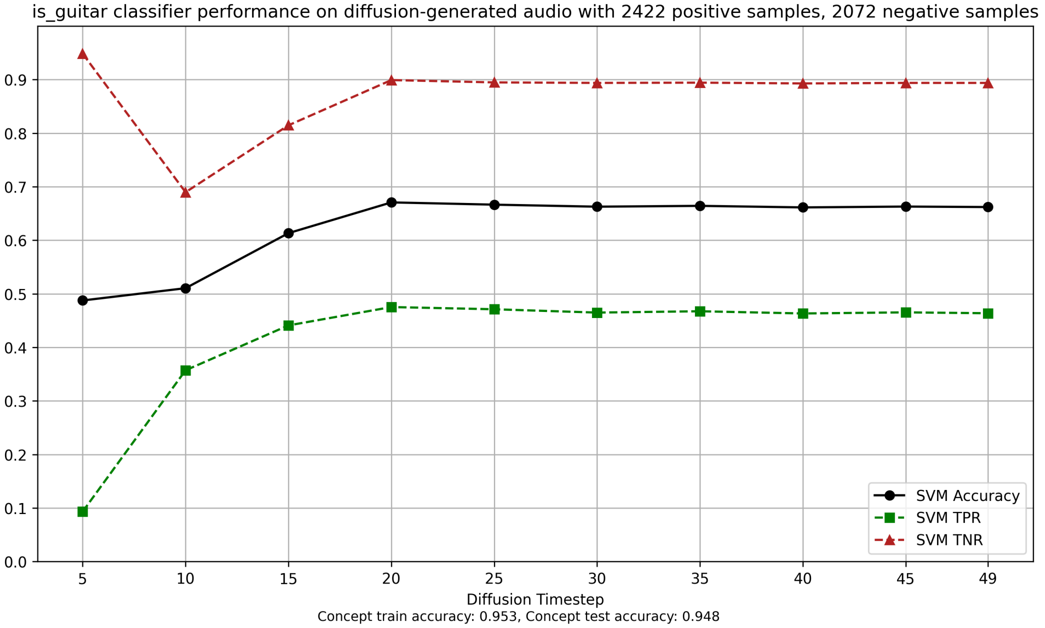

Figure 11

B Full Accuracies

C System Prompts

C.1 Generation prompt

You are a music expert. You have been tasked with generating a list of {N_PROMPTS} prompts for a music generation model. Each prompt should be a short description of a musical piece, including the genre, mood, tempo, and any specific instruments or elements that should be included. The prompts should be diverse and cover a wide range of styles and themes. You should return nothing but the prompts, separated by newlines. Here are some examples:
1. Several ukuleles are playing the same strumming melody together with an acoustic guitar. Someone is playing a shaker slightly offbeat. The song is in 4/4 time and has a tempo of 120 BPM. The mood is happy and upbeat.
2. A solo piano piece with a slow tempo of 60 BPM. The mood is melancholic and introspective, with a focus on minor chords and arpeggios.
3. A fast-paced electronic track with a tempo of 140 BPM. The mood is energetic and uplifting, featuring synthesizers, drum machines, and vocal samples.

C.2 Annotation prompt

You are a music expert. You have been tasked with annotating the following prompts with the following aspects: aspects. For each of the following prompts, please include which aspects are present in the prompt, and the tempo in BPM if it is provided; otherwise, set the BPM field to 0. They should be returned in the following format:\n
{json.dumps({
"prompt": "<prompt text>",
"aspects": [],
"bpm": 0
}
, indent=2)} \n
Output only the JSON objects, one per line, with no additional text or explanation. Here are the prompts:\n
{prompts}

D Division of Labor

Code(Main): Coby Mulliken, Daniel Kyte-Zable
Paper (Main): Shangyang Min, Kyle Lam, Shengmai Chen