Generating Audio Illusions with Diffusion Models

Aileen Mi, Valli Nachiappan, Manisha Pillai
University of California, Berkeley
{mia1, valli_nachiappan, manishapillai}@berkeley.edu

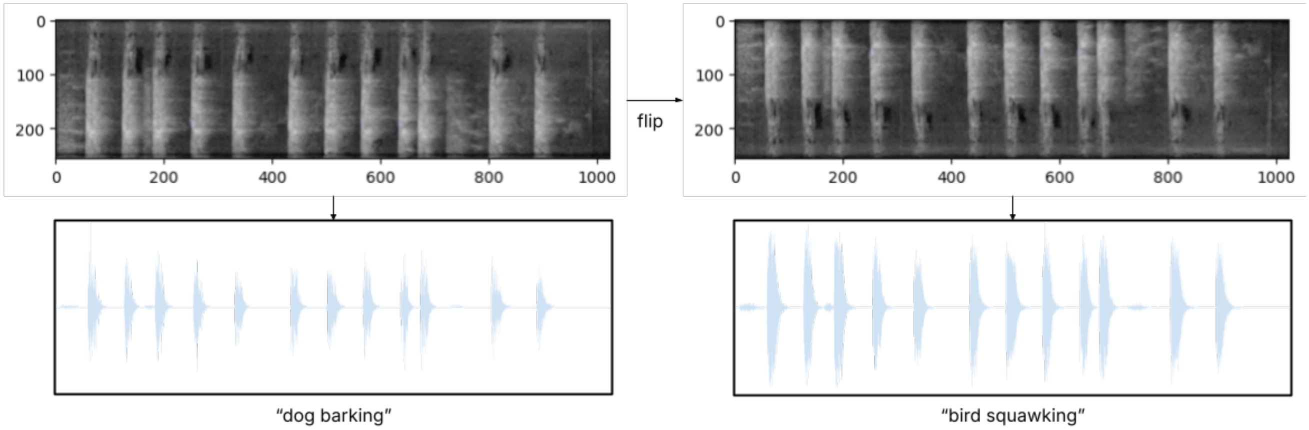


Figure 1. **Visualization of generated spectrograms.** The example spectrogram (top left) is transformed via a Flip. The resulting spectrograms can be converted into audio waveforms (second row) aligning with the input audio prompts.

Abstract

Inspired by the use of diffusion models to create optical illusions, we extend this idea to the auditory domain by introducing a method for synthesizing audio illusions which can be played in various directions. We leverage a pre-trained text-to-spectrogram latent diffusion model to perform zero-shot generation of such illusions. During the reverse diffusion process, we combine noise estimates derived from transformed noisy latent representations to denoise the spectrogram. This enables the generation of audio samples that align with different textual prompts depending on how they are played. Transformations include playing audio samples forward, in reverse, or flipped. We evaluate our approach using quantitative metrics and qualitative analysis. Code for this project is available at https://github.com/beepo34/cs280-final-proj.

1. Introduction

Audio can be visualized using spectrograms, with time on the x-axis, signal frequency on the y-axis, and amplitude as color/shading. Since spectrograms are essentially heatmaps and therefore images, text-to-spectrogram diffusion models can operate similarly to text-to-image diffusion models.

In this project, we extended the compositional generation task to the audio domain. We used a zero-shot method that leverages a pre-trained text-to-spectrogram audio diffusion model to produce spectrograms that adhere to different audio text prompts when transformed. The transformed spectrograms were then converted to waveforms using a pre-trained vocoder.

The main transformations explored are reversing and flipping, such that playing the untransformed and transformed audio results in two audio samples adhering to different text prompts. Additionally, we performed quantitative evaluation using CLAP [3] and qualitative analysis to explore the nuances of text prompt and transformation selection in our method.

1.1. Related Work

Diffusion Models. Diffusion models [8] are a class of generative models that operate by iteratively denoising Gaussian noise to a sample from a learned data distribution. Diffusion models achieve state-of-the-art results in a wide range of domains including image, video, and audio generation. In this project we used Auffusion [15], a latent diffusion model finetuned from Stable Diffusion [12] for the purpose of text-conditioned audio generation, with a scheduler following the DDIM [13] update rule for denoising.

Text and Audio Learning. Recent developments have worked to bridge the gap between text and audio. Audio diffusion models specializing in text-to-audio (TTA) generation utilize latent diffusion techniques to generate audio from text prompts, enabling zero-shot audio synthesis with high fidelity. These models may be text-to-waveform [4, 5] or text-to-spectrogram [15]. Our work builds upon these foundations by exploring the generation of audio illusions through a text-to-spectrogram latent diffusion model.

Compositional generation and Illusions with Diffusion Models. Diffusion models have demonstrated the ability to compose concepts by combining noise estimates. Noise estimates may be interpreted as gradients of a conditional data distribution [14]. Thus, summing or averaging these gradients can maximize the likelihood of multiple conditions, allowing for the composition of different attributes into generated samples [11]. Recent work [1, 2, 6] uses compositional generation to generate illusions using diffusion models. Namely, Geng et al. [6] combine noise estimates with orthogonal transformations from multiple conditional distributions to generate multi-view optical illusions, and Chen et al. [2] uses a similar approach but with the added novelty of sampling from the joint distribution of both images and spectrograms, creating cross-modal audio-image illusions. Building off of these works, our project studies the technique of compositional generation in the audio domain by composing text prompts and transformations (e.g. reversing or flipping samples) during the diffusion process to create audio illusions.

2. Method

Our goal is to generate transformable audio illusions using a pre-trained diffusion model. When the spectrogram is converted into a waveform and played normally in the forward direction, the sound matches an audio prompt. When the spectrogram is transformed via a flip or reversal then converted into a waveform, the sound matches a different audio prompt. To do this, we guided the reverse diffusion process toward a composite of two text-conditioned distributions to produce a spectrogram that simultaneously satisfies both conditions under their respective transformations.

2.1. Diffusion Models

Diffusion models iteratively denoise standard Gaussian noise $x_T \sim \mathcal{N}(0, I)$ to produce a clean sample x_0 from some data distribution. At each timestep t in the denoising process, the model uses a U-Net to estimate the noise $\epsilon_\theta(x_t, y, t)$ from an intermediate noisy sample x_t and a condition y, such as a text prompt embedding. The estimated noise is then used in an update rule such as DDIM [13] to obtain the next sample x_{t-1} from x_t.

We may also apply classifier-free guidance (CFG) [7] to improve the quality of generated samples by interpolating between conditional and unconditional diffusion processes. The noise estimate is then given by:

$$\hat{\epsilon_\theta}(x_t, y, t) = \epsilon_\theta(x_t, \varnothing, t) + \gamma(\epsilon_\theta(x_t, y, t) - \epsilon_\theta(x_t, \varnothing, t)) \tag{1}$$

where γ is the guidance strength and $\varnothing$ is the unconditional embedding of either the empty string or a negative prompt.

2.2. Parallel Denoising

We produced transformable audio illusions by using a diffusion model to simultaneously denoise two transformations of a spectrogram. We took two text prompts y_1 and y_2, associated with transformation functions v_1 and v_2 respectively. At each step in the denoising process, we first applied the transformations to the partially denoised latent x_t separately. Next, we computed noise estimates from the transformed latents with the corresponding text-conditioning embeddings. Finally, inverse transformations were applied to the estimated noises and the resulting estimates are averaged. This process is computed as:

$$\tilde{\epsilon_\theta} = \frac{1}{2}(v_1^{-1}(\hat{\epsilon_\theta}(v_1(x_t), y_1, t)) + v_2^{-1}(\hat{\epsilon_\theta}(v_2(x_t), y_2, t))) \tag{2}$$

where $\hat{\epsilon_\theta}$ is the CFG noise estimate and the transformations v_1 and v_2 have corresponding inverse transformations v_1^{-1} and v_2^{-1} to transform the estimated noise back to the original position.

The combined noise estimate $\tilde{\epsilon_\theta}$ was then used to perform an update step of DDIM to obtain the latents at the next time step. At the end of the process, the clean latent x_0 was decoded using the pre-trained Variational Autoencoder (VAE) from the model to reconstruct the output spectrogram, which was further converted into an audio waveform using a pre-trained vocoder.

2.3. Transformations

We considered three main transformations: Identity, Reverse, and Flip. The Identity transformation aligns the untransformed spectrogram with a chosen prompt. The Reverse transformation flips the spectrogram along the y-axis, equivalent to playing the audio backwards. The Flip transformation flips the spectrogram along the x-axis, such that high and low frequencies are swapped.

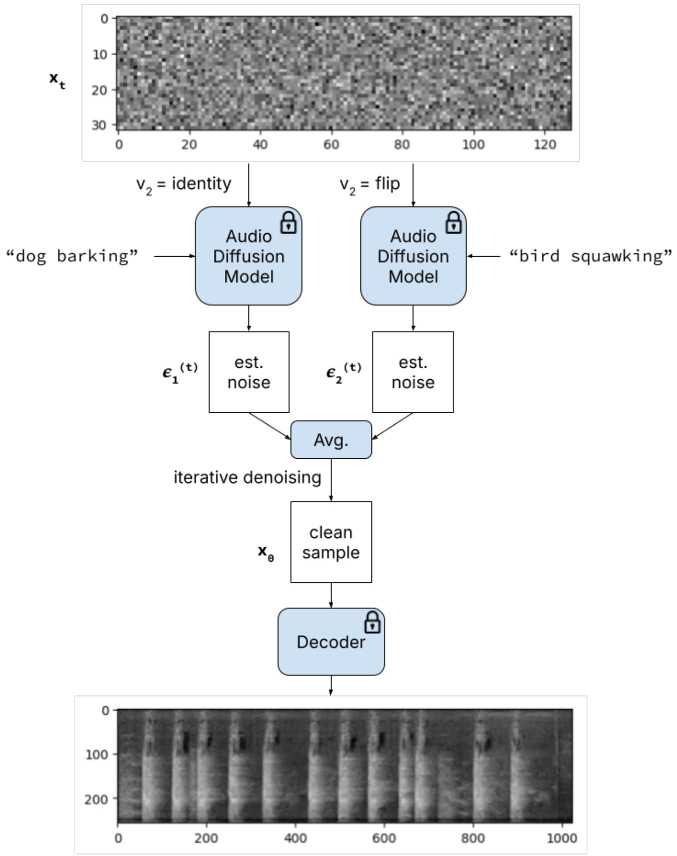


Figure 2. **Algorithm Overview.** Given a noisy latent x_t, we composed the text-conditioned noise estimates $\epsilon_1^{(t)}$ and $\epsilon_2^{(t)}$ from two audio prompts, then performed iterative denoising to obtain the clean latent x_0. The clean latent is decoded into a spectrogram, which then may be converted into a waveform using a pre-trained vocoder.

3. Results

We produced a variety of audio samples with varying degrees of success. We evaluated our samples quantitatively using Contrastive Language-Audio Pretraining (CLAP) [3] to measure alignment between text and audio.

3.1. Sample Results

We show visualizations of sample qualitative results in Fig. 3. Additional samples and evaluation metrics can be found at this link.

3.2. CLAP Evaluation

To quantitatively evaluate the semantic alignment between the generated audio and their corresponding text prompts, we used CLAP (Contrastive Language–Audio Pretraining), a pretrained model that embeds audio and text into a joint multi modal space. CLAP is capable of measuring cross-modal similarity, making it suitable for evaluating whether generated audio corresponds to the intended text descriptions.

Our evaluation pipeline proceeds as follows: we begin with generated audio stored as NumPy arrays (e.g., `forward_audio.npy` and `flipped_audio.npy`), representing the outputs of our auditory illusion system. These arrays are converted into WAV files using a 16kHz sample rate after normalizing and flattening the audio signal.

We define a pair of text prompts describing the intended audio percepts: `"a dog barking"` (for the forward spectrogram) and `"a bird squawking"` (for the flipped version). Using CLAP, we encode both the audio and text into L2-normalized embeddings and compute the similarity between every audio-text pair. This results in a similarity matrix containing the strengths of each audio waveform alignment with each textual prompt. This includes match scores (audio with its intended prompt) and mismatch scores (audio with the opposite prompt).

CLAP produces similarity scores ranging from -1 to 1, although in practice they tend to fall within a narrower band. For most audio-text pairs, scores between 0.2 and 0.4 we consider to have good perceptual alignment. Scores above 0.4 are rare and indicate very strong alignment. Conversely, scores below 0.2 suggest weaker alignment. We determined these comparison metrics based on prior research papers evaluating text to audio alignment [9, 10]. A summary of the results from this evaluation is shown in Tab.1.

The corresponding results for samples in Fig.3 are plotted in Fig.4 as bar plots for ease of evaluation. For example, when analyzing the cat-cricket sample, we observed that forward audio had very strong alignment with its intended text prompt ("a cat purring") with a score of ~ 0.4. The flipped 'cricket' audio had less alignment with its intended prompt ("a cricket chirping") with a score of ~ 0.16 which fell below our ideal score range. However, we still concluded these to be very good results, considering that the mismatch scores of the audio similarity with opposite prompts were very low (< 0.05). Using this method of analysis, we observed that the majority of our samples had match scores higher than their mismatch scores. In other words, samples matched more with their intended text prompts and less with their opposite prompts for both the transformed and untransformed spectrograms.

However, not all samples had the same amounts of success. While the majority of our flipped audios succeeded in achieving higher match scores, achieving this measure of success with reversed audios was far more difficult. The first plot in Fig.4 represents the reverse transform sample in Fig.3, and we see that both audios align more with "cat meowing", including the forward audio that was conditioned on "dog whining". This was a consistent issue for most of the reversed samples, which led us to focus more on generating good flipped samples due to its increased flexibility. More of these ”failed” samples, including failed flipped samples, are discussed in B and **Limitations**.

4. Discussion

We invite discussion on the nuances of working within the audio domain. Unlike images, which represent spatial relationships in two dimensions, audio spectrograms encode time and frequency on their axes, with amplitude represented through intensity. While both are technically two-dimensional, the semantics of their axes differ significantly. In images, spatial structures can be interpreted flexibly across both axes, enabling a wide range of illusions or compositional manipulations. In contrast, audio spectrograms are constrained by the temporal progression of sound and the harmonic structure imposed by frequency, limiting the model’s freedom to reinterpret or recompose elements without disrupting perceptual coherence.

4.1. Limitations

One limitation of our method is that generating reversible audio samples performs poorly when the two audio prompts differ significantly in pitch or timbre. For example, high-pitched sounds such as a cat meowing or a bird chirping do not compose well with low-pitched sounds like thunder or the hum of an engine. This makes sense intuitively, since reversing an audio signal does not alter its frequency values. Therefore, attempting to compose the latent representations of two acoustically different audio samples using Reverse will yield a poor result.

We found that the Flip transformation allowed for more flexibility in generating audio illusions, yielding more successful results. However, we also observed challenges when transforming the audio by flipping its high and low frequencies. In these cases, sounds that were monotonous/continuous paired poorly with sounds that were discrete. Continuous audio samples, such as the sound of a river flowing or a drone, feature relatively uniform waveforms. On the other hand, discrete audio samples contain sharp peaks with periods of silence in between, such as a dog barking or the sound of a typewriter. These structural differences make it difficult for the model to satisfy both conditions simultaneously within a single spectrogram representation. Examples of results from such poorly paired sounds and their corresponding CLAP evaluation scores are presented in Fig.5 and Tab.2 where we experiment with matching continuous sounds such as rain falling to punctuated, staccato sounds like a dog barking.

These findings highlight that choosing good prompts is crucial to generating good audio illusions. To produce convincing audio illusions, we recommend choosing prompt pairs that avoid these pitfalls and minimize conflicting constraints during generation.

4.2. Future Work

Although this preliminary work was mostly done with simple animal sounds and ambient noises, we think it can be further expanded to a large set of sounds. Right now we are using Auffusion as our pretrained backbone model to generate our spectrograms. Auffusion doesn’t support more audio components such as human speech and instrumentals. For this purpose, we can look into other models such as Stable Audio to generate audio from text to accommodate a wider range of capabilities. Stable Audio [4] is a latent text-to-waveform diffusion model that can generate up to three minutes of musical audio content, which can help experiment with audio diffusion with various instruments and their pitches. Stability AI’s smaller tool, called Stable Audio Open [5] is targeted towards shorter audio samples for sound effects. This can help gather a larger amount of data to expand this project further. With these capabilities to generate audio samples, we can experiment more with human speech audio illusions as well as illusions using musical instrument audio.

4.3. Conclusion

In this project, we create audio illusions by leveraging the ability of diffusion models to perform compositional generation in latent space. As audio generation models become progressively more powerful, we see the application of this method to newer and better-quality audio diffusion models as future work. We hope this project supports possibilities for further exploration in the intersection of audio, generative modeling, and multi-modal learning.

References

[1] Ryan Burgert, Xiang Li, Abe Leite, Kanchana Ranasinghe, and Michael S. Ryoo. Diffusion illusions: Hiding images in plain sight, 2023. 2

[2] Ziyang Chen, Daniel Geng, and Andrew Owens. Images that sound: Composing images and sounds on a single canvas, 2025. 2

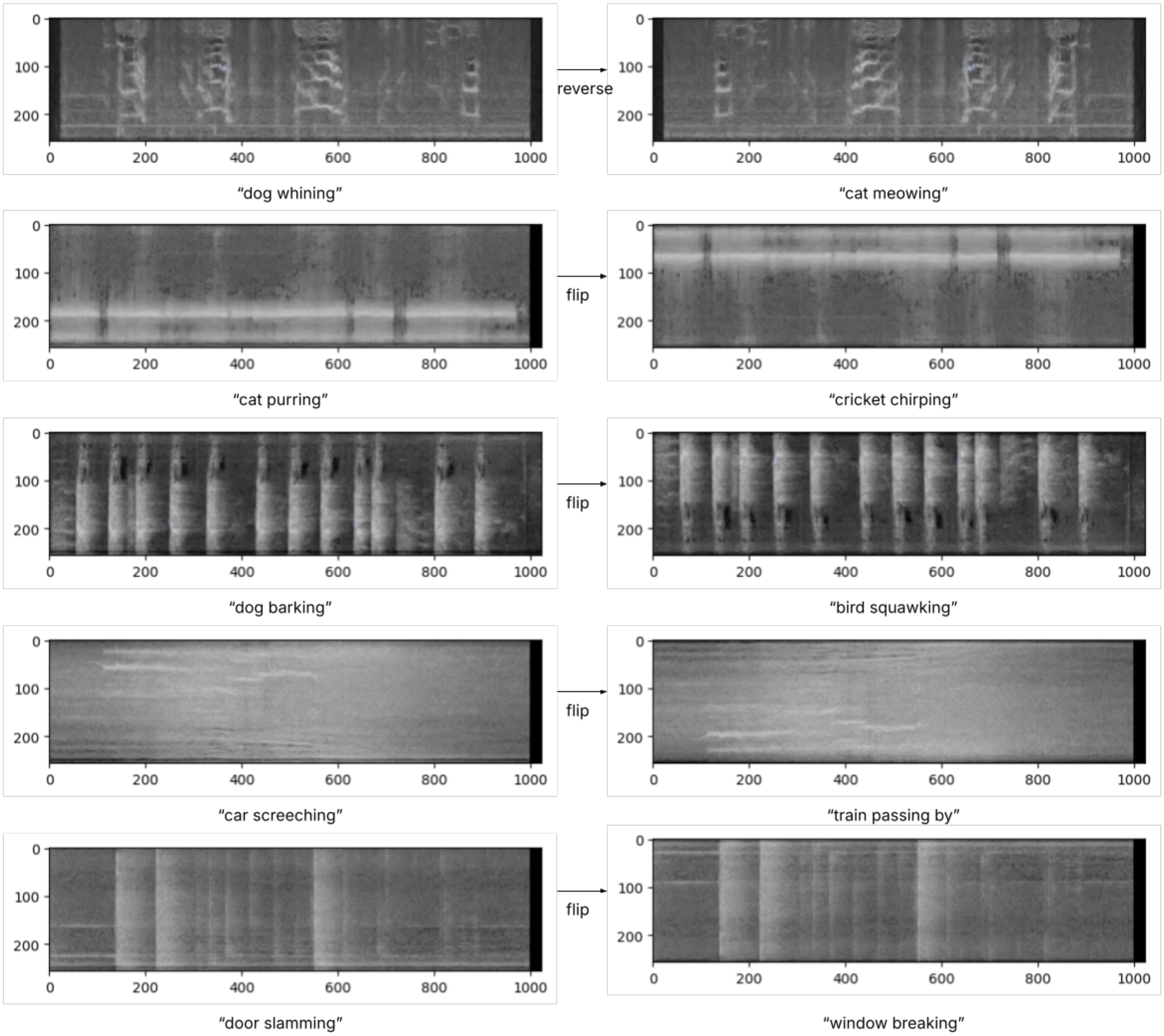
[3] Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. Clap: Learning audio concepts from natural language supervision, 2022. 1, 3

[4] Zach Evans, CJ Carr, Josiah Taylor, Scott H. Hawley, and Jordi Pons. Fast timing-conditioned latent audio diffusion, 2024. 2, 4

[5] Zach Evans, Julian D. Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Stable audio open, 2024. 2, 4

[6] Daniel Geng, Inbum Park, and Andrew Owens. Visual anagrams: Generating multi-view optical illusions with diffusion models, 2024. 2, 6

[7] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. 2



"dog whining" "cat meowing"

"cat purring" "cricket chirping"

"dog barking" "bird squawking"

"car screeching" "train passing by"

"door slamming" "window breaking"

Figure 3. **Visualizations of qualitative results.** Spectrograms in the first column are generated using Identity (no transformation). Captions below each spectrogram indicate the text prompt used for conditioning. Arrows between spectrograms represent the transformations applied to the generated samples.

	cat–dog		cat–cricket		dog–bird		car–train		door–window	
	Original	Reversed	Original	Flipped	Original	Flipped	Original	Flipped	Original	Flipped
Forward Prompt	0.1403	0.3044	0.4048	0.0381	0.3679	-0.3679	0.3886	0.1302	0.3566	0.0333
Flipped Prompt	0.0668	0.3443	0.0123	0.1583	0.1871	0.2714	0.2377	0.3020	0.2319	0.3066

Table 1. **Summary of Quantitative Results.** Comparison of CLAP scores for alignment between original prompts and flipped prompts corresponding to the 5 qualitative samples shown in Fig. 3. Audio samples generally score higher with their corresponding prompts and score lower with the the flipped prompts. Score < 0.2 suggests weaker alignment. Score $0.2 \sim 0.4$ suggest good alignment. Score > 0.4 indicate very good alignment.

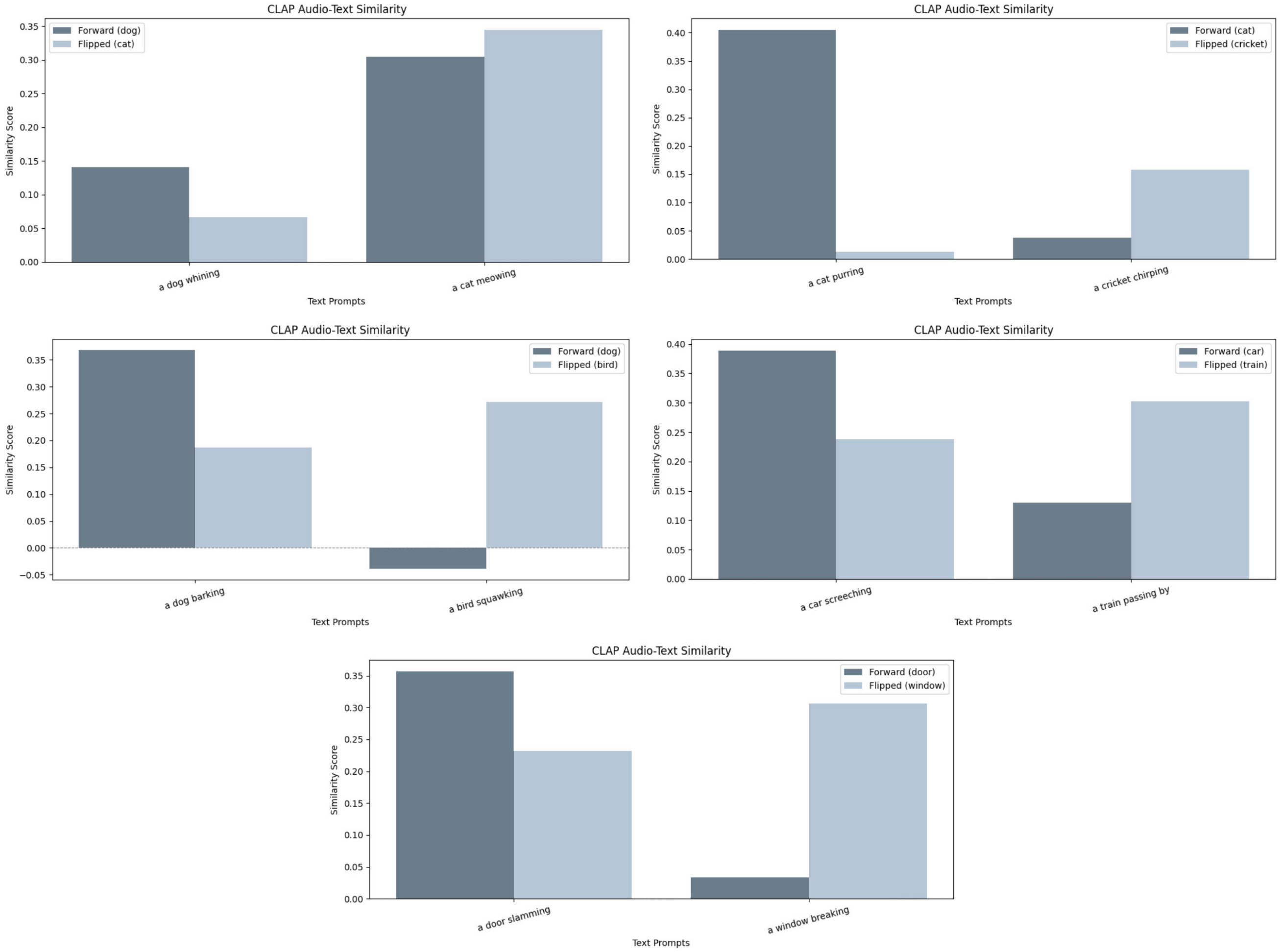


Figure 4. **Visualizations of quantitative results.** Bar plots of CLAP scores for spectrograms in Figure 3. Text prompts below graphs correspond to text prompts used for conditioning for each spectrogram.

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. 1

[9] Chia-Yu Hung, Navonil Majumder, Zhifeng Kong, Ambuj Mehrish, Amir Ali Bagherzadeh, Chuan Li, Rafael Valle, Bryan Catanzaro, and Soujanya Poria. Tangoflux: Super fast and faithful text to audio generation with flow matching and clap-ranked preference optimization, 2025. 3

[10] Huan Liao, Haonan Han, Kai Yang, Tianjiao Du, Rui Yang, Zunnan Xu, Qinmei Xu, Jingquan Liu, Jiasheng Lu, and Xiu Li. Baton: Aligning text-to-audio model with human preference feedback, 2024. 3

[11] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B. Tenenbaum. Compositional visual generation with composable diffusion models, 2023. 2

[12] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. 2

[13] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. 2

[14] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution, 2020. 2

[15] Jinlong Xue, Yayue Deng, Yingming Gao, and Ya Li. Auffusion: Leveraging the power of diffusion and large language models for text-to-audio generation, 2024. 2

A. Additional Transformations

Besides Identity, Reverse, and Flip, we attempted the Invert and Patch transformations, which worked well for generating optical illusions as seen in [6].

The Invert transformation inverts the spectrogram values, effectively subtracting each amplitude component from the maximum amplitude. Though this operation produced interesting visual results, it yielded incomprehensible audio, often resulting in distorted waveforms with extremely high amplitude, leading to audio samples that were deafening and harsh in volume.

	river–dog		car–door		bird–car	
	Original	Flipped	Original	Flipped	Original	Flipped
Forward Prompt	0.1120	-0.0608	0.4455	0.0268	0.1189	-0.0482
Flipped Prompt	0.0521	0.0035	0.3051	0.0826	-0.1556	0.0066

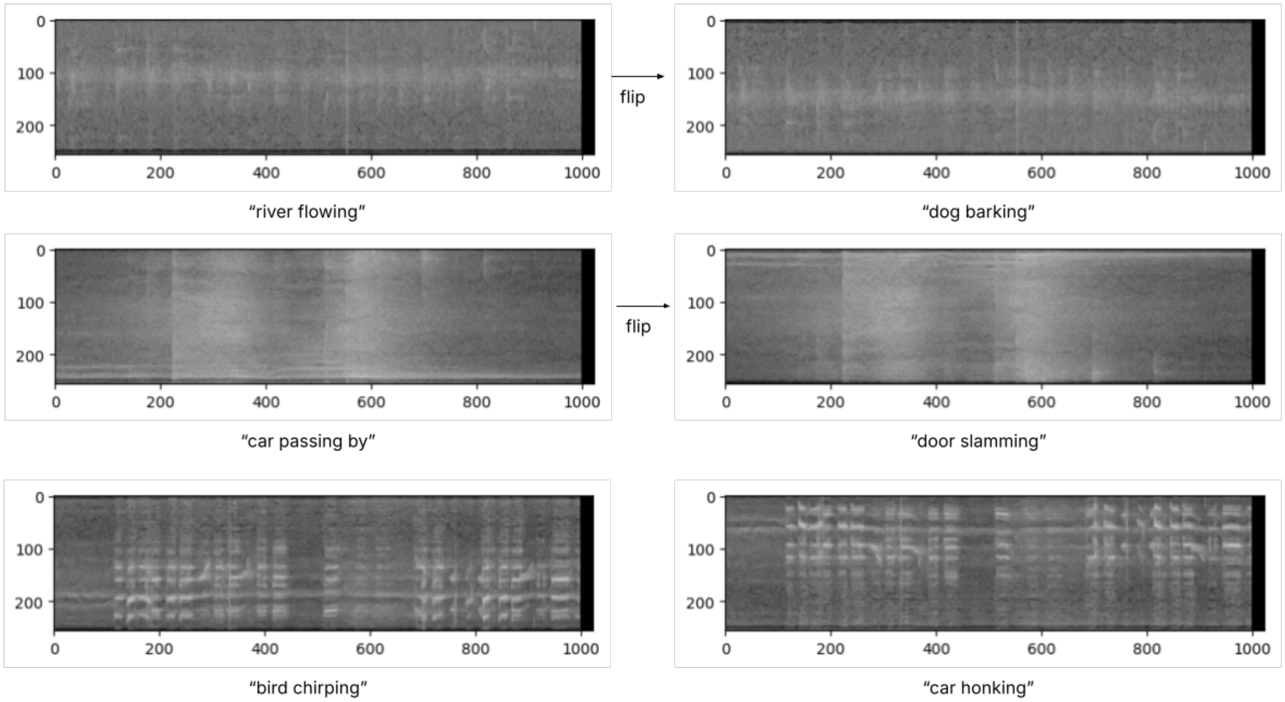
Table 2. **Bad quantitative results.** These audio samples showed weak alignment with their corresponding input text prompts.

The Patch transformation randomly permutes vertical slices of the spectrogram, effectively rearranging segments of the audio in time. We allow up to 128 slices, corresponding to the width of the latent spectrogram representation. While this transformation is appealing in that it introduces variation along the time dimension, its use in combination with our method resulted in low-fidelity outputs. The generated audio samples sounded jumbled and noisy, suggesting that random temporal shuffling disrupts the structure in ways that are difficult for the model to fix during generation.

The above trials highlight the sensitivity of audio reconstruction to certain manipulations due to their limited perceptual flexibility as compared to images.

B. Additional Results

We show additional qualitative and and quantitative results of bad samples in Fig.5 and Tab.2. We notice that text prompts such as "a dog barking" which typically produce audio signals with sharp peaks of amplitude, instead yield flattened signals and low-contrast spectrograms. This indicates that the latent features corresponding to distinct frequency events are being averaged or smoothed out by the influence of the flipped prompt, leading to a loss of temporal structure in the generated spectrogram. We also notice that the plotted CLAP scores in Fig.6 show mismatch scores that beat match scores. Specifically, we see that one prompt dominates, with both forward and flipped audios having high alignment with that one prompt, with lower or even negative scores for the opposite prompts, showcasing the incompatibility of these prompts.



“river flowing” “dog barking”

“car passing by” “door slamming”

“bird chirping” “car honking”

Figure 5. **Bad qualitative results.**

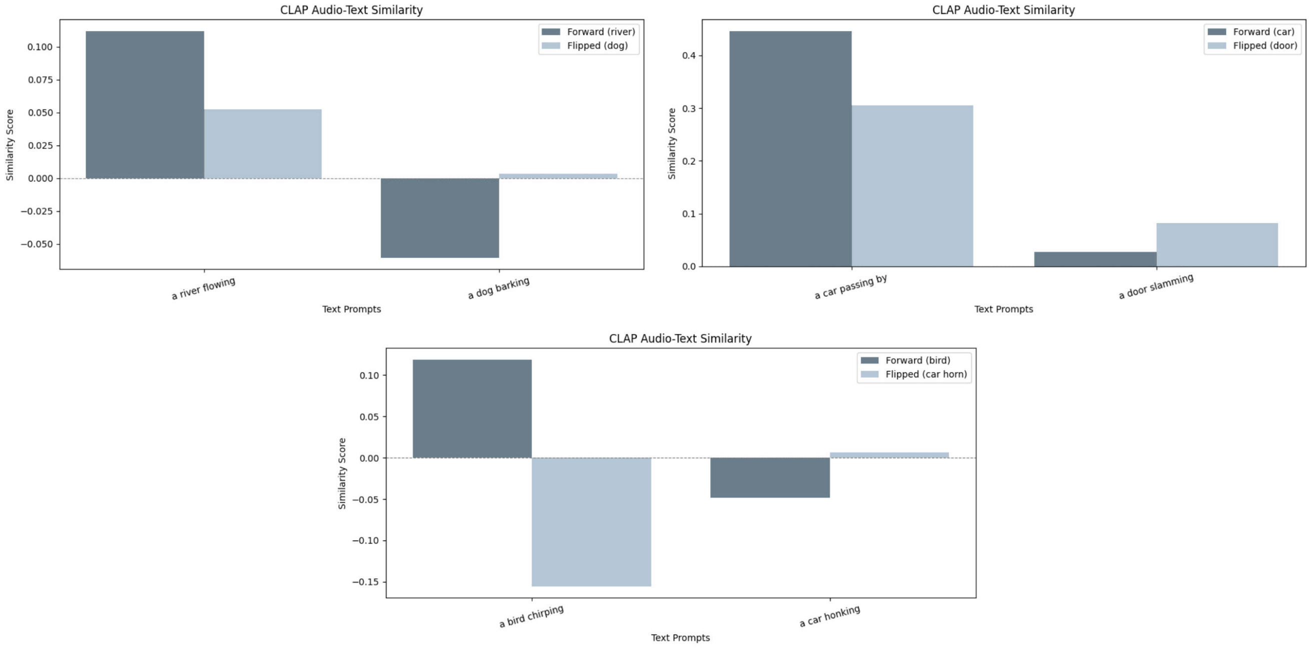


Figure 6. **Visualization of Bad quantitative results.**