Text-Conditioned Audio Editing with Diffusion Model

Junwon Lee
School of Electrical Engineering, KAIST
james39@kaist.ac.kr

Yoonjin Chung
Graduate School of AI, KAIST
yoonjin.chung@kaist.ac.kr

Abstract

Although there are abundant audio samples which is ready to use, it is extremely hard to retrieve or generate the exact one that matches the desired intention. This issue is problematic in not only the audio domain but also the image or video domain. Recently text-conditioned content editing has been actively explored to resolve the problem, in which the task aims to edit the given original content according to the semantics incorporated in the text prompt. The editing approach based on natural language, which got intention due to the rapid growth in text-conditioned generative models, enables intuitive and detailed editing at the same time. We propose a text-conditioned audio editing framework for the first time. Our framework leverages pretrained text-to-audio generative diffusion model *Diffsound*(Yang et al., 2022) while inspired by *Imagic*(Kawar et al., 2022). We assess the edited result in a qualitative manner based on the given text edit prompt while considering the interpolation ratio which is a hyperparameter. Besides, the achievements and limitations are enumerated in terms of audio quality and edited auditorial features for further future works.

1 Introduction

Since finding specific data that satisfy various conditions in the real world, generating new data by editing samples has been a goal to achieve and a challenging task as well. Recent work (Kawar et al., 2022) has been conducted to edit images with text caption conditions, which has shown promising results. However, the same task for the audio domain, which we named text-conditioned audio editing, has not been explored actively. In this study, we address the text-conditioned audio editing problem for the first time, to the best of our knowledge, by fine-tuning and interpolating text embeddings of audio captions. Our task aims to generate target sounds that are slightly modified from existing audio referring to given caption conditions. For instance, let us say there is a pair of audio-caption that the caption annotated as *"Dog barks loudly several times"* to the corresponding audio. As demonstrated in Figure 1, the goal of our task is to generate the edited audio which follows the semantic of the target text prompt, *"cat cries loudly several times'"*, while preserving the structure and composition of the original audio. To achieve our goal, we propose a semantic audio editing method using the pre-trained text-to-audio diffusion model, which can generate audio containing various types of events in the wild. As our method is basically motivated by recent works of image editing, it follows a 3-step process as shown in Figure 2. First, the text embedding for captions is optimized so that the target prompt is mapped to the input audio closely. Then, we fine-tune the diffusion model for the quality of audio generation. After fine-tuning modules, we compute the linear interpolation between the optimized embedding, which is intentionally aligned to the input audio, and the non-optimized target embedding. This enables obtaining the specific embedding which produces the target semantic of the prompt and retaining the structure of input audio at the same time. Finally, the edited audio is generated through the whole model conditioned on the interpolated embedding.

Our contributions can be summarized as:

- We address the text-conditioned audio editing task, which has rarely been studied.
- We optimize the given text embedding and fine-tune the discrete diffusion model to generate semantically edited sound.
- We provide the qualitative analysis on caption prompts and interpolation intensity by

comparing our results from various types of prompts.

2 Related Work

2.1 Text-to-Audio Generation

Text-to-Audio Generation is a task that generates audio in waveform upon a descriptive text condition. The task is recently tackled by a few generative models trained on a large amount of audio-text pair data. The ultimate goal is to generate a high-quality waveform from a phrase or sentence-level text that captions the auditorial feature of sound.

Diffsound(Yang et al., 2022) is the first model that solves the text-to-audio generation task directly. They proposed a discrete-diffusion-based audio generative model conditioned on a text prompt. The model consists of a pretrained CLIP(Radford et al., 2021) text encoder, a discrete diffusion model, a Vector Quantized Variational Autoencoder(VQ-VAE) and a vocoder. After the text prompt is encoded as a feature, the diffusion decoder transfers the text feature to a sequence of quantized tokens which is passed to the pretrained VQ-VAE decoder to reconstruct a mel-spectrogram. Finally, the vocoder generates a waveform from the transferred mel-spectrogram. This approach, unlike traditional CLIP-based models, directly converts text embeddings into audio mel-spectrogram instead of learning a co-embedding space between text and audio domain. *Diffsound* is trained on two datasets: Audioset(Gemmeke et al., 2017) with caption generated with Mask-based text generation method and Audiocaps(Kim et al., 2019) which is an audio captioning dataset.

Audiogen(Kreuk et al., 2022) , in contrast with *Diffsound*, generates audio in an auto-regressive manner. Text and audio are encoded by a pretrained T5, and a pretrained auto-encoder-like architecture adapted from (Zeghidour et al., 2021) with quantizer, respectively. Then a GPT-2-like model, as in (Radford et al., 2019), generates quantized audio tokens to form a sequence. At this step, text and audio embeddings are concatenated at each time step before going through self-attention and cross-attention. The audio decoder which is a pair with the previous audio encoder finally generates the resulting waveform. *Audiogen* is trained on two types of datasets to achieve better performance. Firstly, 6 multi-label-annotated audio datasets including Audioset are used. To generate text captions, word-level tags are simply concatenated. Secondly, 4 audio-caption pair datasets including Audiocaps are exploited. To make the caption distribution accord with the previous ones, phrase or sentence-level captions are pre-processed with stop word remover and lemmatizer. Additional data augmentation technique is utilized that mixes two audio in random timestep and concatenates captions.

2.2 Text Prompt Image Editing

Text prompt Image editing is an emerging task as neural text-to-image generation models like Dall-E(Ramesh et al., 2021) achieve high performance. Unlike text-to-image generation which generates an image from scratch upon a natural language caption, text prompt image editing modifies an existing image according to the prompt. It aims to edit the given image according to the complicated natural language description by adapting its semantics while preserving the original contents of the image.

Imagic(Kawar et al., 2022) is the only model, to the best of our knowledge, that tackles the image editing task conditioned on a text prompt. *Imagic* exploits a pretrained text-to-image diffusion mddel called Imagen(Kawar et al., 2022). To infer with *Imagen*, text input goes through a text encoder, a text-conditioned diffusion model which consists of generative diffusion and super-resolution diffusions. Given a single text prompt and image pair, *Imagic* finetunes *Imagen* in several steps. As the given prompt does not describe the original image but the change to be applied, only finetune the text encoder first on the text-image pair while freezing others, to get optimized embedding e_{opt} which is distinct from e_{tgt} acquired from original text encoder in *Imagen*. In other words, the whole model can generate the original image from the editing caption. The subsequent step is to finetune the diffusion part to achieve better image reconstruction quality. To get the edited image, they finally input the embedding which is a linear interpolation of e_{tgt} and e_{opt}. The authors found the optimal range of interpolation ratio in terms of the tradeoff between image editability and fidelity. In addition, they showed that interpolation is semantically valid in a qualitative manner by providing examples corresponding to several editing categories.

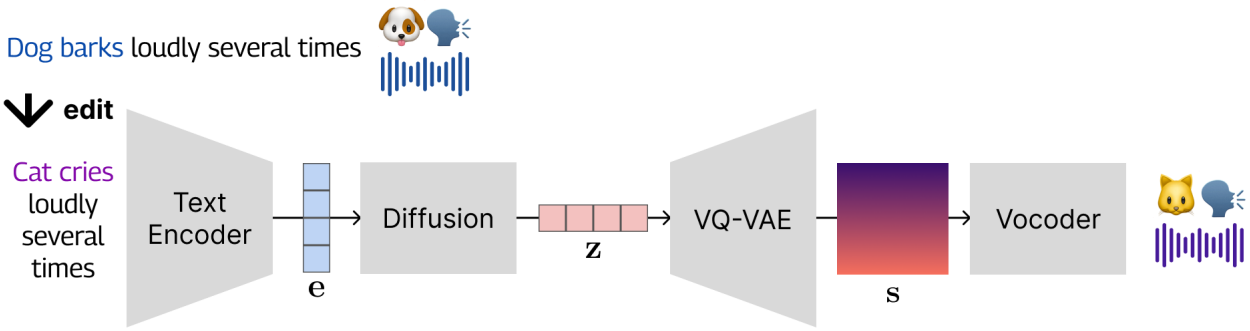

Figure 1: The overall model architecture summarized where $\mathbf{e}$ is text embedding, $\mathbf{s}$ for quantized mel tokens, and $\mathbf{s}$ for mel spectrogram.

3 Proposed Method

3.1 Model Architecture

We exploit the pretrained *Diffsound* model as our base framework as it is the only text-to-audio model which is open-source. As we finetune only some part of *Diffsound* initializing it with provided pretrained weights, the overall architecture is identical. See Fig 1 to see a simply summarized diagram. Among the components, we do not fine-tune VQ-VAE, the decoder part that generates mel spectrogram from a sequence of quantized mel spectrogram tokens, and MelGAN(Kumar et al., 2019), the vocoder, which is both trained on *Audioset* respectively. The two are not involved in processing text captions but in generating the audio sound. In this paper, we focus on CLIP which encodes the text prompt, and the discrete diffusion model which generates a quantized token sequence conditioned on the text embedding.

3.2 Method Details

We follow the proposed method suggested in *Imagic*(Kawar et al., 2022) to achieve our goal. Given the original input audio, we aim to output the edited audio through the semantics included in the text prompt. Note that the prompt is not a caption that describes the original audio but the desirable modified version of it. Let the text representation which encodes this prompt be a target embedding. As there is no way for the text-to-audio model to generate audio similar to the original one, the key idea is to find an optimized embedding that enables it when passed through the generative layers. This approach is inspired by previous research based on GAN such as (Patashnik et al., 2021), (Roich et al., 2022), and (Tov et al., 2021). The brief procedure is the following:

1. We finetune the CLIP text embedding layer to acquire an optimized embedding vector near the target text embedding. The ideal optimized text embedding, as an input of the audio generative model, may nearly reconstruct the original audio.

2. We finetune the discrete diffusion model, with optimized embedding representation as input, to generate the original audio better.

3. Compute the linear interpolation of the target embedding and the optimized embedding and input to the generation part. Proper interpolation enables us to find a representation that maintains the balance between audio fidelity and text prompt alignment.

which is described visually in Fig 2. The following paragraphs explain each step in detail.

Text Embedding Finetune The pretrained CLIP would encode the given text prompt as $\mathbf{e}_{tgt} = TextEncoder(text) \in \mathbf{R}^{L \times d}$, where L is the text token length and d is token embedding dimension. To acquire an optimized embedding $\mathbf{e}_{opt} \in \mathbf{R}^{L \times d}$ which represents the original audio content while being near the $\mathbf{e}_{tgt}$, we finetune the text encoder CLIP with the input image $\mathbf{x}$ and text prompt pair as the other modules are fixed. The optimized em-

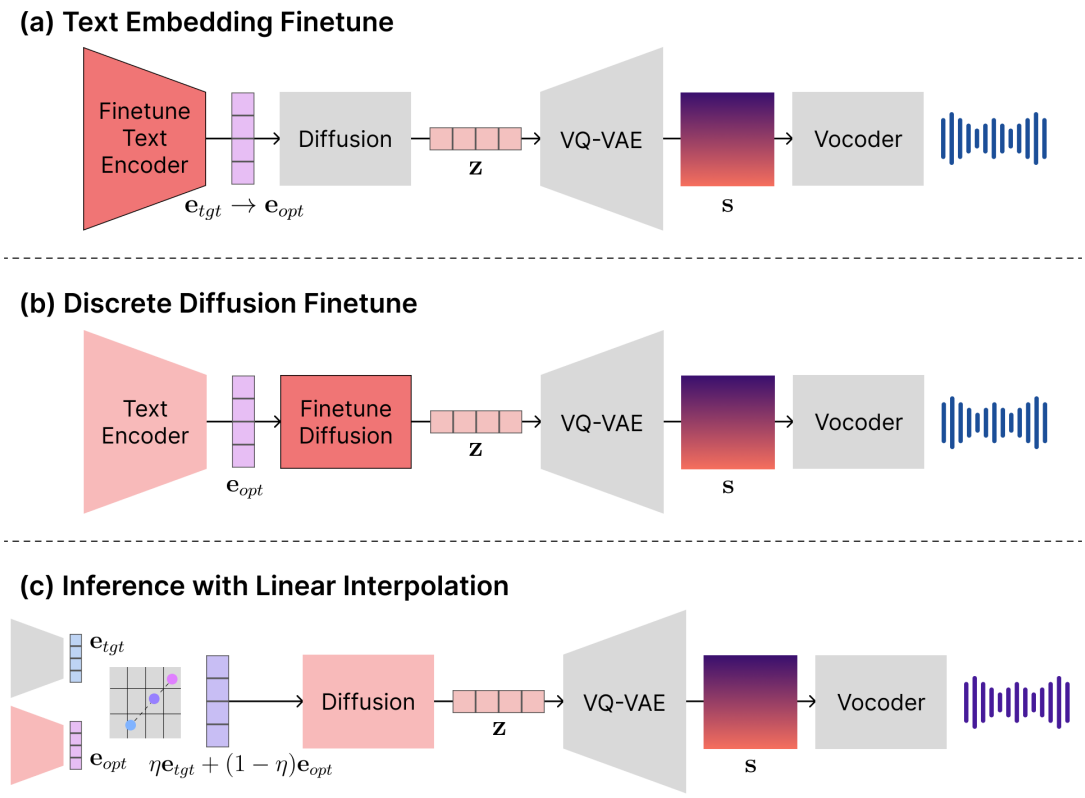

Figure 2: Proposed 3-step method described. For each step, (a) Text Embedding Finetune to acquire optimized text embedding $\mathbf{e}_{opt}$, (b) Discrete Diffusion Finetune to aid higher quality in the audio generation, and (c) Inference with Linear Interpolation of the two text embeddings, $\mathbf{e}_{tgt}$ and $\mathbf{e}_{opt}$, to result in a final edited audio output.

bedding should be close enough to the target embedding to contain the semantics to be edited, and far enough to represent the content in the original image, at the same time.

Discrete Diffusion Finetune Finding $\mathbf{e}_{tgt}$ that would generate the exact original audio is almost impossible and unnecessary as our objective is to edit the given audio. However, the existing discrepancy inevitably lowers the quality of generated audio as it would modify the unintended parts. This is where the finetuning of the discrete diffusion model takes its part. We finetune the parameter θ of the discrete diffusion model $p_\theta(\cdot)$ with the text prompt and image $\mathbf{x}$ pair for better audio reconstruction. Other parts such as finetuned CLIP $TextEncoder_{finetuned}(\cdot)$ are frozen.

Inference with Linear Interpolation After the finetuning stages, the whole model is ready to generate edited audio. We make the final inference by passing a linear-interpolated embedding of $\mathbf{e}_{tgt} = TextEncoder(text)$ and $\mathbf{e}_{opt} = TextEncoder_{finetuned}(text)$ to the discrete diffusion model $p_{\theta,finetuned}$ as a condition. After the interpolation, we get

$$\tilde{\mathbf{e}} = \eta \cdot \mathbf{e}_{tgt} + (1-\eta) \cdot \mathbf{e}_{opt} \tag{1}$$

where $\eta \in (0, 1)$ is a hyperparameter named interpolation interval. The quantized mel spectrogram tokens $\mathbf{z}$ generated by the diffusion model are sent to the VQ-VAE decoder and vocoder subsequently to form a mel spectrogram $\mathbf{s}$ and a waveform respectively. Intuitively, a small η value will result in audio close to the original audio, and a large η value will induce a modified sound in contrast.

Loss for Finetune In terms of the loss function for finetuning, the one which is used to train the generative discrete diffusion model in *Diffsound* is applied. The overall loss is formulated as the following:

$$\mathcal{L} = \lambda \mathcal{L}_{x_0} + \mathcal{L}_{vlb} \tag{2}$$

where λ is a hyperparameter, $\mathcal{L}_{x_0}$ is an auxiliary denoising term, and $\mathcal{L}_{vlb}$ is a variational lower bound

loss(VLB). VLB is formulated as the following:

$$\mathcal{L}_{vlb} = \sum_{t=1}^{T-1}[D_{KL}[q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{e})]] + D_{KL}[q(\mathbf{x}_T|\mathbf{x}_0)||p(\mathbf{x}_T)] \tag{3}$$

where $\mathbf{x}_t$ is the audio data with reverse process time step $t \in \{T, T-1, ..., 1, 0\}$, $\mathbf{e} = TextEncoder(text)$ is the text embedding of given prompt, $p_(\mathbf{x}_T)$ is a stationary distribution, $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{e})$ is a discrete diffusion network that aims to learn posterior transition distribution $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$. The auxiliary denoising term is formulated as the following:

$$\mathcal{L}_{x_0} = -log[p_\theta(\mathbf{x}_0|\mathbf{x}_t, \mathbf{e})] \tag{4}$$

For further details, please refer to the *Diffsound* (Yang et al., 2022) paper.

4 Experimental Setup

4.1 Dataset

Audioset(Gemmeke et al., 2017) provides the 2M of audio clips from Youtube videos with multiple labels. The audios have a duration of 10 seconds and the corresponding labels contain a description of the sound event which covers a wide range of everyday sounds, from human and animal, sounds to natural and environmental sounds, to musical and miscellaneous sounds. In Diffsound, the multiple labels are concatenated with a random number of mask tokens to use them as an audio caption.

AudioCaps(Kim et al., 2019) consists of 46K audio clips with human-written text pairs collected via crowdsourcing on the AudioSet dataset. It includes 49,256 of training, 494 of validation and 957 of test sets. For the training set, every audio clip contains one annotated caption, while the validation set and test set, contain five captions per audio. In this study, we picked several audio clips from the training set and edited the corresponding captions to our target captions.

4.2 Data Pre-processing

Audio clips are resampled to 22,050 Hz and zero-padded for some samples that total lengths are shorter than 10 seconds. Log mel-spectrograms are extracted with 1024 size of Hanning window, 256 hop size and 80 mel-bins.

4.3 Experimental Details

We arranged our 2 stages of fine-tuning and inferences based on the pre-trained weights from Diffsound. The VQ-VAE module is also pre-trained on Audioset with the 256 dimensions of each token vector and codebook Z. For the other details, we have archived our code at GitHub repository[1].

Text Encoder The 512 embedding size of CLIP text encoder is fine-tuned with the 2e-6 of the learning rate for 400 epochs, and Adam optimizer(betas are (0.5,0.9)) is used.

Discrete Diffusion We set diffusion time steps as 100 and the auxiliary loss weight as 5e-4 for the discrete diffusion model. Adam optimizer is used as same settings as the above. The learning rate is set as 3e-6 for total 800 epochs.

Vocoder Our vocoder implementation is based on the official MelGAN(Kumar et al., 2019) implementation and pre-trained weights from Diffsound. It is pre-trained for 200 epochs using 256 dimensions of Mel-spectrograms on only 40% of the Audioset data, due to the time complexity of the training process.

5 Result Analysis

For effective result analysis, we performed audio editing with the text-audio-paired samples in the *Audiocaps* which are not seen during the training phase of *Diffsound*. As we have the text caption about the original audio, we can compare the edited result.

5.1 Qualitative Assessment

We have verified that substitution in the sound source (i.e. who or what is making sound, which kind of sound has occurred) is successful while preserving the structural feature of the audio. Figure 3 visualizes some sample audio results with their text prompt. To listen to the actual audio rather than mel-spectrogram, refer to the demo link. [2] As you can see from the given samples, it is able to edit the horn to the bell sound while maintaining a structure in which the bell rings once loudly and then fades away as same as the original audio.

[1] https://github.com/YoonjinXD/Text-conditioned-Audio-Editing.git

[2] https://docs.google.com/presentation/d/19JkFHqdXuEgYC5IW4lbi8AG-E2l3y252oVK_I3RRw2s/edit?usp=sharing

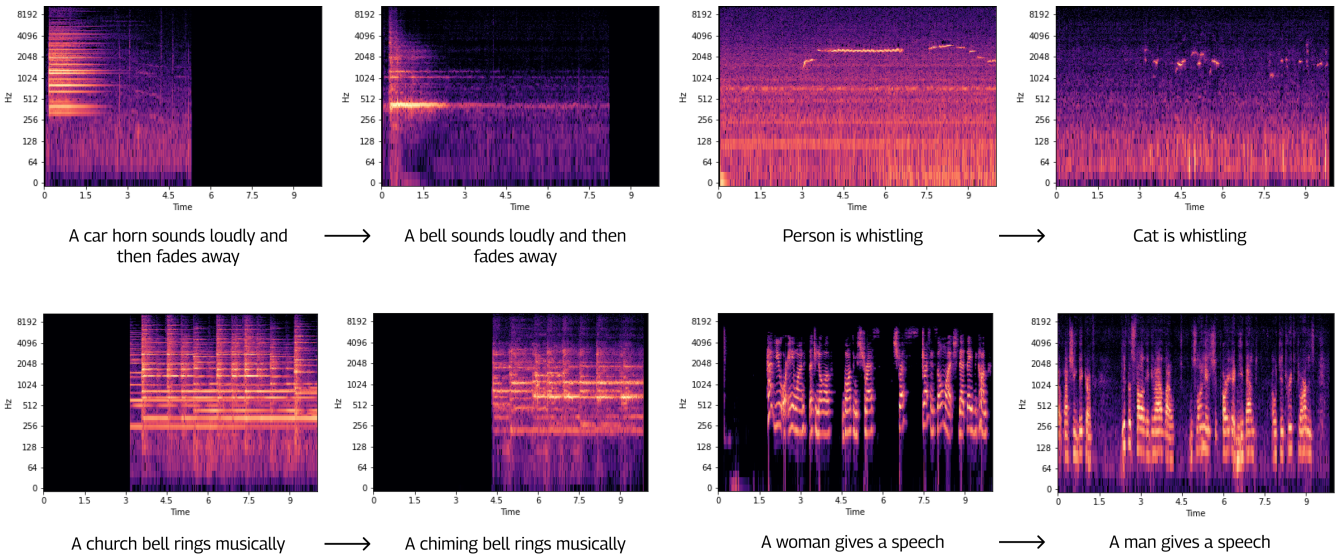

Figure 3: Sample edited results. The left mel-spectrograms visualize the original input audios in *Audiocaps*. Their corresponding captions are written below. The edited audio result is shown in the right mel-spectrograms with their respective text prompts below. Focus on the fact that the overall structure is preserved while the intended timber substitution is done successfully.

5.2 Interpolation

Choosing a proper value for $\eta \in (0, 1)$ in equation (1) is critical for high-quality audio editing as it controls the trade-off between audio fidelity and text prompt alignment. A small η value may lead to a result close to the original audio while a large value results in a generated sound aligned with the text condition.

To show that the proposed linear interpolation is valid to capture the semantics between the original audio content and the text prompt, we visualize a series of edited results with varying η values of a single audio-text instance in Figure 4. Original audio in *Audiocaps* captioned as "Someone snores in the background" is edited with a conditioning text "Someone groans in the background." In between reconstructing the original audio (i.e. $\eta = 0.0$) and generating a whole new audio conditioned on the text prompt (i.e. $\eta = 1.0$), η values between 0.0 and 1.0 retains some inputted original auditorial features while blending in newly generated components. This figure visually shows the trade-off between audio fidelity and text prompt alignment when linearly interpolating two text embeddings: target embedding $\mathbf{e}_{tgt}$ and optimized embedding $\mathbf{e}_{opt}$.

Note that η is a hyperparameter which is an interpolation ratio. An appropriate value of η varies on each audio-text input as the model-finetune process in our proposed method is done for a single instance. Unfortunately, it is hard to find the optimized η value without a manual exhaustive search as there is no exact quantitative metric that can measure audio fidelity or text prompt alignment. Moreover, it was hard to guarantee whether there exists a meaningful interval in the interpolation ratio because the value chosen differed from each editing instance.

6 Limitations

Since the audio generative diffusion model that we used is only trained with word-level ontological tags from Audioset, it seems that cannot capture complex semantics such as temporal information such as relative time position (e.g. after, then) or duration, spatial information(e.g. foreground, background) in the audio domain. To edit such semantics, audio dataset with detailed audio captions such as *Audiocaps* must be used to train the whole model. With our current resources, it was almost impossible to train the model on *Audiocaps* as it took more than weeks to learn on one epoch. We plan to adapt the *Audiocaps*-trained version of the model as soon as the *Diffsound* author releases the trained weight.

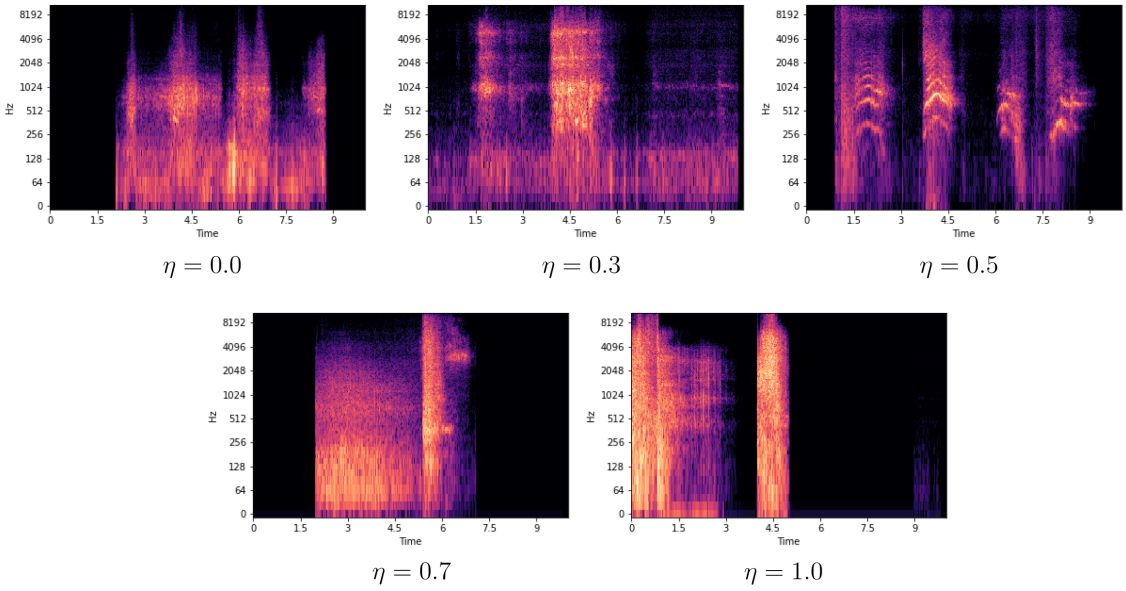

Figure 4: Linear interpolation of text embedding. The generated audios in the form of mel-spectrogam are listed with their corresponding interpolation ratio (i.e. η value). $\eta = 0.0$ is identical to using optimized embedding $\mathbf{e}_{opt}$ only to reconstruct the original audio, as $\eta = 1.0$ gives a generated audio conditioned on target text embedding $\mathbf{e}_{tgt}$. The caption tagged on the original audio is "Someone snores in the background", and the editing text prompt is "Someone groans in the background."

Another limitation comes up due to the audio quality of the dataset. *Audioset* and *Audiocaps* which were exploited during the pretraining and editing, have poor audio quality with background noise and uneven volume. This results in poor edited audio quality. To relieve this problem, we can consider utilizing other datasets with better audio quality or introducing post-processing such as noise gates or volume normalization.

This work lacks quantitative evaluation in the edited result. To assess if the result retains the overall structure or the timber feature of the original audio, we may introduce quantitative metrics or human evaluation. In addition, to prove our hypothesis that the model reconstructs the original audio better as the interpolation ratio, i.e. η, gets larger, various reconstruction losses such as L2-norm can be adopted. If there are some metrics which can measure the audio fidelity or text prompt alignment of the edited result, it may be helpful finding an ideal η value automatically or specifying a certain interval for effective editing.

7 Conclusion and Future Work

This study proposes the first text-conditioned audio editing method based on the pre-trained generative diffusion model. With our method, an existing audio and caption pair describing the desired editing result optimizes the model to generate the edited sounds while preserving the remaining factors in the original audio. In the analysis of the results, we found that finding proper interpolation intervals and also fine-tuning the diffusion model plays an essential role in our task. We know that detailed audio captions must be trained in order to capture the temporal and spatial semantics from prompts. However, the size of the model and dataset was too large to train on time with our resources. Thus, our future work may focus on training detailed audio captions properly, so that the model can also edit upon complex prompts while enhancing the fidelity of input audio and structure preservation. Further, as a quantitative metric, we can compare the reconstruction error between the generated samples of each interpolation interval and the original audio input.

8 Author Contribution

All authors are involved in all parts flexibly and equally. We participated in all parts which contain the idea proposal, searching references and resources, implementing the referenced codes, data pre-processing, experiments, analyzing the results

and writing the report in an equal manner.

References

Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. 2017. Audio set: An ontology and human-labeled dataset for audio events. In *Proc. IEEE ICASSP 2017*, New Orleans, LA.

Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. 2022. Imagic: Text-based real image editing with diffusion models. *arXiv preprint arXiv:2210.09276*.

Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. 2019. AudioCaps: Generating captions for audios in the wild. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 119–132, Minneapolis, Minnesota. Association for Computational Linguistics.

Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi. 2022. Audiogen: Textually guided audio generation. *arXiv preprint arXiv:2209.15352*.

Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C Courville. 2019. Melgan: Generative adversarial networks for conditional waveform synthesis. *Advances in neural information processing systems*, 32.

Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. 2021. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2085–2094.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR.

Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. 2022. Pivotal tuning for latent-based editing of real images. *ACM Transactions on Graphics (TOG)*, 42(1):1–13.

Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. 2021. Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics (TOG)*, 40(4):1–14.

Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu. 2022. Diffsound: Discrete diffusion model for text-to-sound generation.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. 2021. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507.