Prompt-guided Precise Audio Editing with Diffusion Models

Manjie Xu [1 2] **Chenxing Li** [2] **Duzhen Zhang** [2] **Dan Su** [2] **Wei Liang** [1] **Dong Yu** [3]

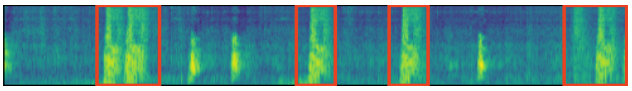
Abstract

Audio editing involves the arbitrary manipulation of audio content through precise control. Although text-guided diffusion models have made significant advancements in text-to-audio generation, they still face challenges in finding a flexible and precise way to modify target events within an audio track. We present a novel approach, referred to as Prompt-guided Precise Audio Editing (PPAE), which serves as a general module for diffusion models and enables precise audio editing. The editing is based on the input textual prompt only and is entirely training-free. We exploit the cross-attention maps of diffusion models to facilitate accurate local editing and employ a hierarchical local-global pipeline to ensure a smoother editing process. Experimental results highlight the effectiveness of our method in various editing tasks.

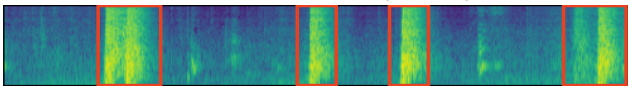
1. Introduction

Recent progress in image synthesis (Ramesh et al., 2021; Rombach et al., 2022; Ramesh et al., 2022) has inspired the application of text-guided diffusion models in text-to-audio (TTA) generation, known for their realism and diversity (Huang et al., 2023b; Ghosal et al., 2023; Liu et al., 2023a;b; Huang et al., 2023a; Yang et al., 2023a). Thanks to large-scale training data and prompt-enhanced methods, these diffusion models have demonstrated great potential in modeling long continuous signal data, and have successfully learned to produce sounds based on given text.

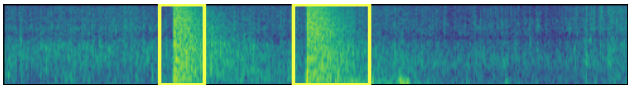
However, these generative models still face challenges in editing tasks, particularly in **precise** editing. Precise audio editing involves modifying the target events within an audio track while preserving the unrelated part unchanged.



(a) Source Audio: A dog barking



(b) Precise Editing: Replace dog barking to gun shooting



(c) Traditional Editing: Replace dog barking to gun shooting

Figure 1: **Precise audio editing.** Such editing requires modifying the target events while preserving the unrelated events and keeping the overall structure unchanged.

As illustrated in fig. 1, precise editing (fig. 1b) replaces "dog barking" with "gun shooting" in the original place. In contrast, traditional editing often overemphasizes the replacement of the content itself, leveraging regeneration steps to ensure the emergence of gun shooting, while often changing the overall structure of the original audio (fig. 1c).

The key to precise editing lies in the ability to accurately differentiate between targeted sections for editing and unrelated parts, ensuring that manipulations are strictly confined to the intended areas. Image editing methods often achieve this by providing spatial localization masks, which can be time-consuming and labor-intensive. Furthermore, such methods cannot be easily generalized to audio editing, as target events mixed in an audio piece are often difficult to identify and separate manually. More recently, researchers mainly utilize pre-trained TTA generation models or concentrate on end-to-end training with human-provided instructions (Wang et al., 2023; Liu et al., 2023a; Huang et al., 2023b; Yang et al., 2023c), while these methods do not guarantee precision in the process of regeneration. They can also be resource-intensive during the end-to-end training process, as they require a large number of editing demonstration pairs as training data.

In this work, we propose Prompt-guided Precise Audio Editing (PPAE), a training-free approach for precise audio editing. Building on the success of image editing techniques that rely on attention map manipulation (Hertz et al.,

Work done while Manjie Xu and Duzhen Zhang were interns at Tencent. [1]Beijing Institute of Technology [2]Tencent AI Lab Beijing [3]Tencent AI Lab Seattle. Correspondence to: Chenxing Li <lichenxing007@gmail.com>, Wei Liang <liangwei@bit.edu.cn>, Dong Yu <dongyu@ieee.org>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

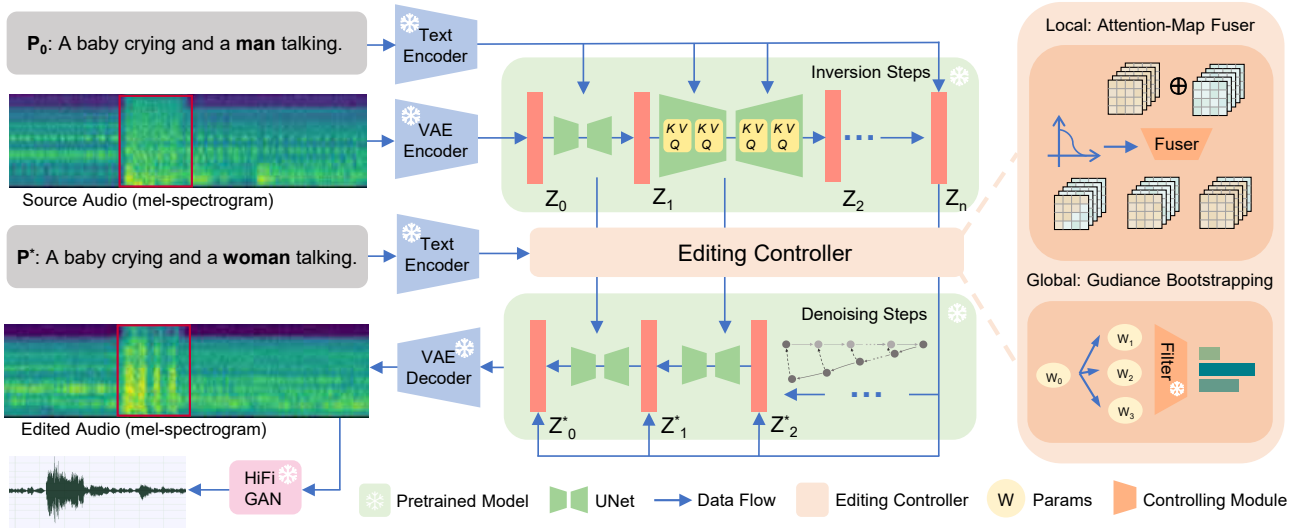


Figure 2: **The overview of the proposed PPAE.** Given the edit instruction, the source audio will first be inverted into the given diffusion model's domain, and then edited on the attention-map level under the guidance of our editing controller. The controller accomplishes precise editing by utilizing hierarchical guidance throughout the diffusion process. The whole editing pipeline is training-free and is adaptable to common diffusion models.

2022; Parmar et al., 2023; Patashnik et al., 2023), PPAE focuses on the cross-attention layer of diffusion models, where text and audio features are interconnected. We demonstrate that diverse types of precise audio editing can be achieved by manipulating the cross-attention map during the denoising process. Our approach serves as a flexible audio editing interface, where users only need to provide edited textual prompts. Additionally, it can function as a plug-in editing module in various diffusion models.

To perform editing, we first convert the original audio into edit-friendly noise spaces and then perform editing by injecting cross-attention maps into the diffusion process. We additionally design a hierarchical pipeline to guarantee the editing effectiveness. Locally, we import a particular Fuser module to integrate different attention maps seamlessly in single diffusion steps. This integration is crucial for mitigating the abrupt transitions often caused by sudden changes in the attention map. Globally, we employ a bootstrapping method to adjust the guidance scale, acknowledging the variability of editing targets across different audio samples. We find such a method allowing for tailored editing, adapting to the unique requirements of each audio piece. Our ablation studies confirm the effectiveness of these innovations. Moreover, we observe that the editing targets in audio differ markedly from those in images, necessitating distinct hyperparameters for optimal performance. This distinction underlines the unique challenges inherent in audio editing, and our method's adaptability in addressing these challenges.

To the best of our knowledge, our work showcases the first attempt at utilizing attention map-level manipulation to achieve precise editing for audio. The proposed PPAE approach, compared with traditional audio editing methods, offers several key benefits: it achieves **precise** editing, which can ensure the manipulations are confined to the intended event; the editing process is **flexible** based on the textual prompts; it is entirely **training-free**; and it is **compatible** with widely-used diffusion models. We hope that this work serves as a step towards addressing the distinct challenges associated with audio editing[1].

2. Related Work

2.1. Diffusion-based TTA Models

Diffusion models have emerged as a promising approach for generating high-quality and diverse samples, from image synthesis (Ramesh et al., 2021; Rombach et al., 2022; Ramesh et al., 2022) to text-to-audio generation (Kreuk et al., 2022; Huang et al., 2023b; Ghosal et al., 2023; Liu et al., 2023a). More recently, methods like Diffsound (Yang et al., 2023b) and AudioLDM (Liu et al., 2023a;b) work based on the diffusion model to produce better test-to-sound (TTS) and TTA generation results; Tango (Ghosal et al., 2023) adopts an instruction-tuned Large Language Models (LLM) FLAN-T5 to utilizing its powerful representational ability in text-to-audio (TTA) generation; Make-An-Audio (Huang et al., 2023b;a) leverages prompt-enhanced methods to train diffusion models

[1]See the project page at `https://sites.google.com/view/icml24-ppae`.

with high-quality text-audio pairs.

2.2. Diffusion-based Image Editing

Text-conditioned editing based on diffusion models has recently garnered significant interest in the image domain (Shi et al., 2023; Kawar et al., 2023; Hertz et al., 2022; Zhang et al., 2023a). When compared to Generative Adversarial Network (GAN)-based methods such as DragGAN (Pan et al., 2023), diffusion-based approaches are considered to have better generality and higher-quality editing effects due to the advantages of diffusion models (Shi et al., 2023). On the one hand, techniques like Glide (Nichol et al., 2021) and Diffusionclip (Kim et al., 2022) excel at global editing or detailed local editing when precise masks of the restricted area are provided. On the other hand, works such as Prompt-to-Prompt (PTP) (Hertz et al., 2022) achieve intuitive image editing by examining the semantic strength in cross-attention maps and performing injection. Various inversion methods (Mokady et al., 2023; Huberman-Spiegelglas et al., 2023) have been introduced to help invert the given image, thereby facilitating further editing.

2.3. Audio Editing

Traditional audio editing methods predominantly focus on global editing tasks, such as audio super-resolution (Birnbaum et al., 2019), audio inpainting (Adler et al., 2012; Moliner & Välimäki, 2023; Wang et al., 2023) and style transfer (Grinstein et al., 2018; Lu et al., 2019; Cífka et al., 2020; Netzorg et al., 2023). Among these, uSee (Yang et al., 2023c) proposes a unified model to perform speech editing given text description and specific arguments; Loop Copilot (Zhang et al., 2023b) and InstructME (Han et al., 2023) enable generating and refining generated music. Recent research has started to explore more fine-grained audio editing techniques, concentrating on tasks such as adding, removing, or replacing particular audio events within a specific audio piece. Audit (Wang et al., 2023) trains a latent diffusion model on editing tasks and supports instruction-guided audio editing. More recently, TTA generation is combined with personalization methods to meet user preferences (Plitsis et al., 2024).

2.4. Comparison

Achieving precise editing in audio is challenging, especially when compared to visual media editing, due to the inherent temporal and spectral intricacies of audio signals. Conventional approaches like Audit and PerMod depend on training a latent diffusion model specifically for audio editing tasks and regeneration. Conversely, PPAE enables a training-free and flexible manipulation of audio content and is not limited to music or speech editing only. Additionally, PPAE offers a higher degree of granularity, empowering users to edit specific audio elements within a track—a level of precision that has been challenging to attain with existing methods. This fine-grained control also parallels the progress seen in text-conditioned image editing using diffusion models, while uniquely adapted to the audio domain in editing tasks.

3. Method

Formally, let A represent a given audio piece, and P denotes the textual description (prompt) of that audio piece; we aim to edit the input audio guided solely by the edited text prompt P^*, resulting in the final edited audio A^*. Researchers are also interested in cases where the original text prompt is absent but only with a command such as "Replace the A with a B." We note that this scenario can be addressed through audio captioning. In this paper, we mainly focus on the former task.

3.1. System Overview

Our method comprises three parts: an inversion module that maps the given audio piece into the pre-trained Latent Diffusion Model (LDM)'s domain, an LDM pre-trained on audio, and a hierarchical editing controller which is plugged into the LDM that facilitates audio editing. LDMs for audio generation often contain an variational autoencoder (VAE) that projects the input mel-spectrograms into the latent space, a textual-prompt encoder that transforms the text prompt into embeddings, and a diffusion network. The system's overview is illustrated in 2.

3.2. LDM

The fundamental concept of diffusion models revolves around the iterative refinement of a randomly sampled noise input $x_t \sim \mathcal{N}(0, I)$, in a controlled manner, to x_0. With a trained perceptual compression model, LDMs focus on the efficient, low-dimensional latent space, where the goal is to derive z_0 from z_t. With a given text description P, to perform sequential denoising, we train a network ϵ_θ predict artificial noise, following the objective:

$$\min_{\theta} \mathbb{E}_{z_0,\epsilon\sim\mathcal{N}(0,I),t\sim\text{Uniform}(1,T)} \|\epsilon - \epsilon_\theta(z_t, t, \psi(P))\|_2^2, \quad (1)$$

where the condition $C = \psi(P)$ is the text embedding of the description. In TTA generation, z is the latent representation of the mel-spectrogram of the audio, where researchers often leverage a pre-trained VAE to help compress the mel-spectrogram into the latent space.

For the conditional generation of LDMs, classifier-free guidance has proven to be an effective method for text-guided generation and editing (Ho & Salimans, 2022). When given a latent and a textual prompt, the generation

is performed both conditionally and unconditionally and then extrapolated according to a given weight. Formally, let $\varnothing = \psi(\text{" "})$ be the null text embedding, the generation can be defined by:

$$\tilde{\epsilon}_\theta = w \cdot \epsilon_\theta(z_t, t, \psi(P)) + (1 - w) \cdot \epsilon_\theta(z_t, t, \varnothing), \quad (2)$$

where w denotes the guidance scale.

3.3. Attention Map Editing

Popular diffusion-based generation models utilize U-Nets (Ronneberger et al., 2015) with the cross-attention mechanism (Vaswani et al., 2017) as the diffusion network. When doing conditional generation, the embeddings of different modalities are often fused in the cross-attention layers. Researchers have shown that injecting the cross-attention maps of the input enables precise editing while maintaining the original composition and structure of the original input (Hertz et al., 2022). As commonly defined, we note attention maps as:

$$M = Softmax(\frac{QK^T}{\sqrt{d}}), \quad (3)$$

where $Q = \ell_Q(\phi(z_t))$ is the query matrix of the deep spatial features of the noisy input, $K = \ell_K(\psi(P))$ and $V = \ell_V(\psi(P))$ are the key matrix and the value matrix of the textual embedding, and ℓ_Q, ℓ_K, ℓ_V are the learned linear projections.

In image editing, researchers perform replacements to get a new attention map M^* when generating the new target z_t^* from $P*$, and override the original attention map M in the computation of a single step t of the diffusion process, noted as $DM(z_t^*, P^*, t, s)\{M \leftarrow \hat{M}_c\}$. Although this approach is also applicable in TTA editing, we observe that a significant abrupt change in the diffusion step may result in the generation of indistinct audio with low quality. An intuitive solution is to incrementally incorporate the editing component into the original attention map, transitioning from a low to a high ratio. We propose a fusion mechanism, utilizing a cosine scheduler to manage the transition of the attention map during the editing process. We denote this as $M_{edit} = Fuser(M_t, M_t^*, t)$. For different editing tasks, we denote our method as follows:

$$Fuser := \begin{cases} S_{ca}(t) \cdot M_t^* + (1 - S_{ca}(t)) \cdot M_t, \\ \qquad \text{for Audio Replace} \\ S_{ca}(t) \cdot (M_t^*)_{i,j} + (1 - S_{ca}(t)) \cdot M_t, \\ \qquad \text{for Audio Refine} \\ c \cdot S_{ca}(t) \cdot (M_t^*)_{i,j} + (1 - S_{ca}(t)) \cdot M_t, \\ \qquad \text{for Audio Reweight} \end{cases} \quad (4)$$

where $S_{ca}(t)$ is the fusion ratio determined by the CosineAnnealing scheduler at step t, $(M_t^*)_{i,j}$ means that we only modify the pixel value i according to the selected text token j, and c denotes the scale extent of the reweighted token in Reweight task. The CosineAnnealing scheduler can be expressed as:

$$S_{ca}(t) = \eta_{\text{min}} + \frac{1}{2}(\eta_{\text{max}} - \eta_{\text{min}}) \left(1 + \cos\left(\pi \cdot \frac{t - t_s}{t_e - t_s}\right)\right), \quad (5)$$

where t_s and t_e represent the starting and ending steps of the transitional phase during diffusion steps respectively, η_{min} and η_{max} are the minimum and maximum values for the ratio, which commonly be 1 and -1.

3.4. Inversion

Text-guided editing with the method mentioned above requires inverting the given audio and textual prompt. Many previous works on text-to-image generation have focused on Denoising Diffusion Implicit Models (DDIM) inversion (Song et al., 2020; Dhariwal & Nichol, 2021), as DDIM sampling is considered as a deterministic sampling process that maps the initial noise to an output.

However, such inversion has been found lacking when classifier-free guidance is applied. To overcome this issue, Null-text Inversion (Mokady et al., 2023) imports pivotal inversion for diffusion models and null-text optimization to achieve high-fidelity editing of natural images. DDPM-based inversion methods have also been developed (Wu & De la Torre, 2022; Huberman-Spiegelglas et al., 2023). We have adapted different inversion modules according to LDMs, see appendix B for details.

3.5. Guidance Bootstrapping

The guidance scale w plays a crucial role in controlling the level of importance assigned to a given prompt during the generation process in diffusion models. Generally, there exists a common scale for w that enables the model to produce creative or precise outputs. For example, Stable Diffusion (Rombach et al., 2022)'s w lies between 5 and 15. However, determining a universally applicable w for audio generation presents a challenge, as the editing components in audio generation can vary significantly, ranging from the resounding nuances of human voices to blurred background sounds. Existing work (Liu et al., 2023a) has also shown the effectiveness of guidance scale on key metrics like Kullback–Leibler (KL) and Fréchet distance (FD). We introduce a bootstrapping approach that aims to avoid the selection of the guidance scale. We initialize a list of $W = [w_1, w_2, w_3, ...w_n]$ before editing, and then computing z_0 for each w_i separately. We employ a Filter module to help get the final editing results. By default, PPAE uses a contrastive language-audio pretraining (CLAP) model (Elizalde et al., 2023) as the naive filter

function f to identify the output with the highest relevance to the target prompt. The process can be fully paralleled.

3.6. Overall Framework

The overall algorithm of PPAE can be described as follows: given an audio piece and its prompt, we firstly set the guidance scale to 0 and perform an inversion on the hidden space with the source prompt; then we do denoising process on the source prompt and target prompt separately, under the guidance of the PPAE editing controller. Locally, we edit the attention map based on the editing instructions under the guidance of Fuser, and decode the final latent variable and filter on the guidance scale to obtain the edited audio. More details can be found in appendix B.

Algorithm 1 PPAE

Input: A piece of audio a and its description as the source prompt P, a target prompt P^*.
[Optional for $Fuser$**]:** η_{min} and η_{max}, t_s and t_e;
[Optional for $Bootstrapping$**]:** $W = \{w_i\}$;
[Optional for $Filter$**]:** $Filter_w = filter_w(a, f)$;
Output: A piece of edited audio a^*.

$w = 0, z_0 = Encoder(a)$;
$\{z_T, z_{T-1}, ..., z_1\} = Inversion(z_0, P, w)$;
$z_T^* \leftarrow z_T$;
for $t = T, T-1, \ldots, 1$ **do**
  $z_{t-1}, M_t \leftarrow DM(z_t, P, t)$;
  $M_t^* \leftarrow DM(z_t^*, P^*, t)$;
  $M_{ct} \leftarrow Fuser(M_t, M_t^*, t)$;
  $z_{t-1}^* \leftarrow DM(z_t^*, P^*, t, W)$ with $M \leftarrow M_{ct}$;
end for
$a^* = Filter_w(Decoder(z_0^*))$;
Return: a^*

4. Experiments

4.1. Editing Task

We mainly focus on the three following audio editing tasks:

Audio Replace: This task aims to replace a specific audio event in a given audio piece with another, while keeping the remaining part unchanged. For example, given an audio piece "A cat meowing and then a baby crying," the task could involve changing "cat meowing" to "dog barking."

Audio Refine: Audio Refine here involves modifying an existing audio piece to meet additional requirements or preferences. The task aims to transform the original audio piece according to various extra adjective descriptions, such as altering the style or incorporating new features, while maintaining the essence of the music.

Audio Reweight: Audio Reweight is to alter the audio balance to emphasize or de-emphasize some aspects without compromising the audio's overall clarity. This could involve amplifying the sound of raindrops in a track where rain and thunder are present, or reducing the volume of background music in a conversation.

We only focused on these three local editing tasks here, as these tasks allow for the addition, drop, or replacement of certain audio elements or adjusting the balance of different sounds. These tasks can often be predictable and repetitive across different audio pieces and files.

4.2. Test Set

We construct our test set utilizing a cleaned subset of the Fsd50k dataset (Fonseca et al., 2021; Li et al., 2023). A pivotal aspect of precise audio editing is implementing precise modifications while maintaining the other elements of the audio unchanged. In each task, we select two distinct audio clips, treating one as the target for editing. For each task, we randomly sample 100 editing tasks as the test set. See appendix F for the detailed construction process. We also select preliminary editing tasks as case studies.

4.3. Metrics

For objective metrics, we leverage commonly used metrics to evaluate the editing effects. We leverage Fréchet distance (FD), Fréchet audio distance (FAD), Spectral distance (SD), and Kullback–Leibler (KL) divergence to measure the distance between the edited audio and the ground truth. Specifically, we also use CLAP Score in some cases as an extra metric to calculate how well the target prompt aligns with the edited audio, as for tasks like Refine and Reweight, it is challenging to construct a corresponding target audio that can serve as ground truth for comparison. For subjective metrics, we primarily employed two metrics: Relevance, which measures how well the output audio matches the input editing prompt, and Consistency, which assesses the extent to which the original audio is edited in accordance with the editing goal. Details of these metrics can be found in appendix C and appendix D.

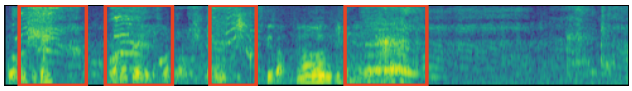
4.4. Experimental Settings

In this work, we primarily utilize Tango (Ghosal et al., 2023) as our TTA backbone model due to its success in TTA generation, while it's worth mentioning that our methods can be applied to a wide range of popular diffusion models. We run our experiments with 100 inference steps and retain the original hyperparameters from Tango. For editing, we run the denoising steps with 0.8 cross-replace steps, 0.0 self-replace steps, and 50 skip steps. The bootstrapping num n is set to 5. We reset our Fuser configs to fit these settings, mainly η_{min} and η_{max}, t_s, and t_e. We also leverage our reproduced PTP for audio as an editing

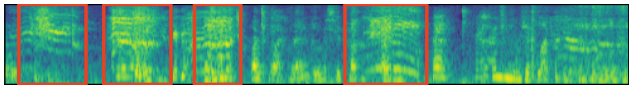
baseline. See appendix B for detailed discussion about the backbone model choice and baseline implementation.

5. Results

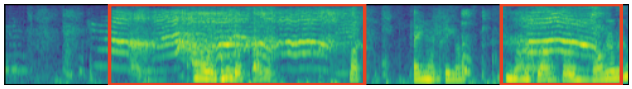
5.1. Audio Replace



(a) A baby crying while a man talking.wav (Source)



(b) A baby crying while a woman talking.wav (Edited)



(c) A baby crying while a woman talking.wav (Regenerated)

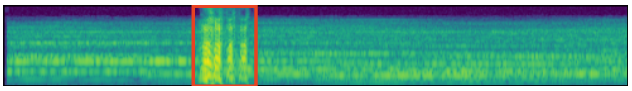
Figure 3: **Case Study (Audio Replace)**

The PPAE efficiently performs replacement within a given audio piece by manipulating the attention map, and exhibits sufficient precision to preserve the overall audio structure. As demonstrated in fig. 3, the PPAE replaces the "man talking" event with "woman talking" while maintaining the original "baby crying" content and even the talking component in the initial audio piece, in contrast to the regenerated result. We quantitatively evaluate the replacement editing outcomes as presented in table 1. The results reveal that the PPAE achieves substantial editing enhancements across the majority of metrics.

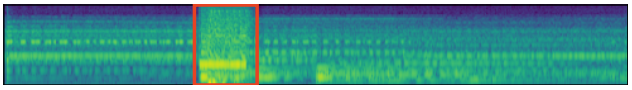
Table 1: **Replace Editing Results**

Replace	FAD ↓	LSD ↓	FD ↓	KL ↓	CLAP↑
PPAE	2.15	**1.51**	**27.53**	**1.30**	0.62
Regenerated	4.93	1.74	32.94	1.69	0.63
Unedited	1.86	5.98	45.99	3.28	0.12
PTP	2.95	2.83	45.91	4.42	0.57

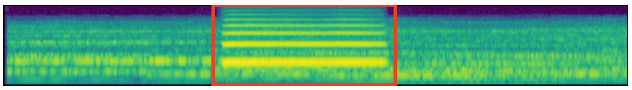
Despite the scarcity of competitive open-source baselines for audio editing tasks, we attempt to compare the PPAE with Audit. This alternative audio editing technique attains state-of-the-art performance in analogous tasks. The editing approach employed by the PPAE diverges significantly from Audit's, as the latter trains a specialized end-to-end diffusion model on editing instructions and an extensive set of data pairs. Conversely, the PPAE conducts training-free edits on attention map layers, rendering it compatible with a diverse range of diffusion models. Since Audit is not open-source, we compare these two audio editing techniques using Audit's publicly accessible demos, as depicted in fig. 4. For the task of "Replace laughter to trumpet,"



(a) Task: Replace laughter to trumpet (Source Audio)



(b) Edited by PPAE

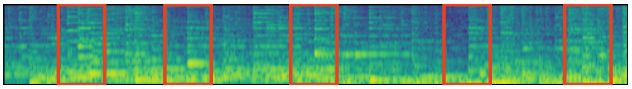


(c) Edited by Audit

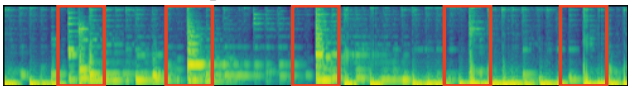
Figure 4: **Case Study (PPAE compared with Audit)**

Audit regenerates the audio based on the given instruction, resulting in a change in the audio structure. On the other hand, PPAE only performs replacement on attention maps related to "laughter," thus preserving the original structure. The source and target prompts are generated through audio captioning and human relabeling. We want to additionally note that the editing quality of PPAE can be affected by the data bias between the input audio and the training set of the utilized diffusion model.

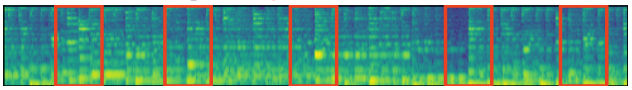
5.2. Audio Refine



(a) A piece of music.wav (Source)



(b) A piece of jazz music.wav (Edited)



(c) A piece of shrill music.wav (Edited)

Figure 5: **Case Study (Audio Refine)**

PPAE demonstrates that injecting into the attention map can aid in enhancing or modifying a given audio clip to meet supplementary requirements or preferences. As illustrated in fig. 5, when provided with the source audio "a piece of music," PPAE successfully refines it according to different adjective descriptions. It transforms the original audio into a different style, such as "jazz," or infuses it with a new characteristic, like "shrill," while striving to preserve the original musical content.

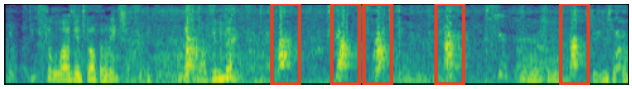
We present the Refine editing effects in table 2. For comparison purposes, we report the corresponding metrics of the PPAE editing results alongside the unedited audio. It is

Table 2: **Refine Editing Results**

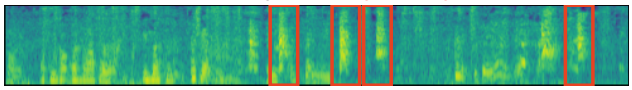
Refine	FAD ↓	LSD ↓	FD ↓	KL ↓	CLAP↑
PPAE	**6.86**	**1.55**	**43.31**	1.92	**0.63**
Unedited	8.19	1.70	49.91	1.85	0.25
PTP	9.35	1.94	45.88	1.17	0.32

worth noting that obtaining an audio with the same structure that also satisfies the prompt is challenging. Therefore, we regenerate the audio according to the target prompt as the ground truth, which results in worse performance in terms of distance metrics. Nevertheless, the PPAE results still outperform the unedited audio, demonstrating significant editing effects. For the CLAP score, we employ CLAP to calculate the similarity between the edited audio and the target prompts. The CLAP scores are softmaxed. Generally, the results demonstrate that the refined editing effects significantly improve the audio quality and better align with the target prompts.

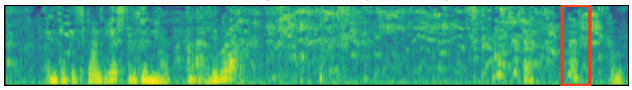
5.3. Audio Reweight



(a) A woman talking and a dog barking.wav, c = 2



(b) A woman talking and a dog barking.wav, c = 0



(c) A woman talking and a dog barking.wav, c = -2

Figure 6: **Case Study (Audio Reweight), controlled token = *"dog barking"***

Table 3: **Reweight Editing Results**

Reweight	Original	2	1	0	-1	-2
Reweight ↑↓	0.74	**0.83**	0.73	**0.35**	**0.10**	**0.12**
Reweight(PTP) ↑↓	0.74	0.79	0.74	0.49	0.32	0.25
Unrelated →	0.91	0.93	0.91	0.89	0.82	0.85

Our methods demonstrate effective control in strengthening or weakening a specific audio event based on the textual token. fig. 6 illustrates edited audio with prompts "A woman talking and a dog barking," where we reweight on the "barking" effect. Results show that the barking component in the edited audio is controlled according to the specified controlling parameter c.

We mainly employ the CLAP Score to quantitatively evaluate the reweight degree of the target event in the given audio. For the results presented in table 3, we show the CLAP scores for the reweighted event and unrelated event in reweight editing. See the error bars in table 7. Take "A woman talking and a dog barking" as an example. If we want to reweight the "dog barking" component, we compute the CLAP score between the edited audio and "dog barking" to obtain the reweight score, and "a woman talking" to get the unrelated score. Our edited results demonstrate that a positive controlling parameter effectively strengthens the reweighted component in the audio, enabling the CLAP model to recognize it more accurately. Conversely, a negative controlling parameter assists in weakening this component. Also, the results indicate that as the controlling parameter decreases from 2 to -2, the reweighted component diminishes. The unrelated components remain relatively stable, showing that the edit will not change the other attributes of the audio.

5.4. Subjective Evaluation

Table 4: **Subjective Evaluation Results**

Metric	Replace		Refine		Reweight	
	PPAE	*Comp*	*PPAE*	*Comp*	*PPAE*	*Comp*
Relavence ↑	**95.71**	89.28	81.42	81.42	**99.28**	92.14
Consistency ↑	**95.0**	81.42	**85.71**	81.42	**94.28**	82.85

We assessed the editing effects of the PPAE through a Subjective Evaluation. The evaluation involves judging the relevance and consistency of each edited audio file with its file name and the original audio file. The metric **Relevance** refers to the match between the audio and the input textual prompt, whether the text content appears in the audio, and whether the audio corresponds to the text's semantics. **Consistency** assesses the degree of similarity between the current audio and the original audio.

For each task, we engaged 14 participants to evaluate the editing effect and precision on 30 randomly sampled edited audio pairs. We leverage regenerated audio based on P^* as the comparison. The results in table 4 show that PPAE significantly improves the precision of the editing while maintaining a good editing effect, whereas the regenerating baseline failed in terms of consistency. See the error bars in table 8. More details of our Subjective Evaluation can be found in appendix D.

5.5. Ablation Study

5.5.1. Generation Configurations

Specific generation configurations significantly impact the output quality. In particular, configurations derived from

Table 5: **Ablation study on the generation configuration.** We show editing results with different generation configurations, especially the guidance scale w, the cross-replace steps $Cross$, and the self-replace steps $Self$.

Parameters	FAD↓	LSD↓	FD↓	KL↓	CLAP↑
7/0.8/0	5.97	2.01	40.90	1.88	0.51
25/0.8/0	**3.34**	**1.62**	**25.74**	**0.62**	**0.68**
75/0.8/0	3.67	1.98	43.24	1.83	**0.69**
75/0.8/0.2	12.92	1.73	48.90	1.62	0.57
75/0.6/0.2	6.79	2.11	35.63	1.37	0.41

Table 6: **Ablation study on the Fuser with different schedulers.** Editing results with different schedulers show that a scheduler can help generate better-edited audios.

Fuser Scheduler	FAD↓	LSD↓	FD↓	KL↓
PTP(w/o)	5.36	2.09	31.32	0.72
Exponential	3.65	**1.51**	35.02	1.21
Linear	3.47	1.61	**25.74**	**0.62**
CosineAnnealing	**3.15**	1.73	**25.75**	**0.63**

image generation and editing sometimes translate poorly when applied to audio. Among these configurations, guidance scales and replace steps are vital for successful generation and editing effects.

We conduct several groups of editing studies, as shown in table 5. Generally, generation with the cross-replace steps around 0.8 and self-replace steps around 0 improves the metrics. While w around 25 achieves the best performance, the generation can also potentially work with a larger or smaller guidance scale. These results also demonstrate the importance of guidance bootstrapping, as we cannot find a universally applicable w. A more extensive guidance scale results can lead to a more significant editing effect, as evidenced by the CLAP score, although it may affect the quality of the generated audio. An alternative solution is to increase the diffusion steps while keeping the guidance scale small. We have concluded that $steps = 1000/w$ generally works.

5.5.2. Attention-Map Fuser Scheduler

The Fuser scheduler controls when and how the attention map of the source audio and the edited audio are mixed, thus being the critical module in the proposed editing pipeline here. In PPAE, we use the CosineAnnealing scheduler, while here we compare the editing effects of different schedulers like linear and exponential schedulers without bootstrapping. Details of these schedulers can be found in appendix E. The results are shown in table 6.

We have observed that the Fuser module enhances the editing effects compared to the original PTP model. Additionally, editing results slightly vary when different Fuser schedulers are used, with the CosineAnnealing scheduler proving slightly superior to the linear and exponential schedulers. Interestingly, the PTP model without a Fuser exhibits the most drastic changes (fig. 26), while the Cosine function is the most subtle, which corresponds to the editing effects. A plausible explanation could be that it facilitates a smoother combination of noisy latents, thereby aiding in merging sources from different audios. This conclusion has been similarly tested on images, where researchers linearly interpolate the noisy latent from various sources (Dong et al., 2023). While they observed cluttered content in images due to the spatial (rather than semantic) mix of the source object and the edited object, we find such a problem is quite different in audio editing.

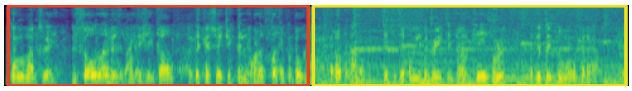
5.5.3. Discussion about the Hyperparameters

There are a number of hyperparameters introduced by this approach such as the number of inference steps, the guidance scale, and the parameters of the cosine scheduler. While the method is training-free in that the diffusion model need not be re-trained, results indicate that selection of these parameters is important for performance. We consider this aspect not as a drawback, but as a strategic design choice. Firstly, for the majority of editing tasks, there exist common selections for these hyperparameters that have been empirically found to perform well across a wide range of scenarios. This standard configuration serves as a solid starting point for users, ensuring that effective editing can be achieved without the need for extensive parameter tuning in general cases. For certain parameters, such as the guidance scale, we have developed a bootstrapping strategy that simplifies the selection process. This strategy assists users in automatically identifying appropriate parameter values based on the characteristics of their specific audio editing tasks, reducing the complexity involved in manual tuning. Secondly, the inclusion of these hyperparameters was a deliberate choice to provide users and researchers with the flexibility needed to fine-tune the editing process according to specific needs and constraints. This flexibility is paramount in the diverse field of audio editing, where the optimal settings for hyperparameters can significantly vary depending on the task at hand and the specific characteristics of the audio data being processed.

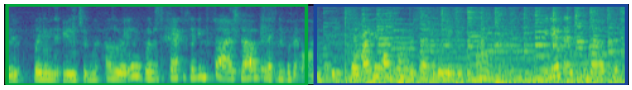
More details and further ablation studies on other configurations can be found in appendix G.

6. Audio Refusion

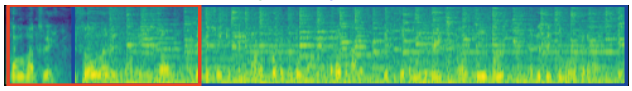
The previous success of audio editing based on attention maps sets the stage for us to tackle a more challenging task



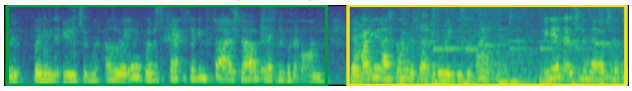
(a) A man talking and a soft music.wav (Source)



(b) A woman talking and a jazz music.wav (Source)



(c) A man talking and a jazz music.wav (Refused)



(d) A woman talking and a soft music.wav (Refused)

Figure 7: **Case Study (Audio Refuse)**

called Audio Refusion: given two audio pieces, $a_1 + a_2$ and $a_3 + a_4$, the goal is to create a fusion, such as $a_1 + a_4$. We introduce Audio Refusion here to further demonstrate PPAE's editing capability.

We perform attention map fusion as follows:

$$Fuser := S_{ca}(t) \cdot (M_{1,t})_{i,j} + (1 - S_{ca}(t)) \cdot (M_{2,t})_{k,l}, \quad (6)$$

where M_1 and M_2 represent the attention maps from the two given audio sources, $(M_1)_{i,j}$ and $(M_2)_{k,l}$ denote different components of these audio sources, such as the selected text token j from the first audio and the selected text token l from the second audio. We fuse their attention maps M_i and M_k under the guidance of the fuser scheduler. It is important to note that the scheduler should aim for a more balanced fusion, as we are combining the attention maps rather than replacing them. We recommend setting η_{min} and η_{max} to 0.4 and 0.6, respectively, for optimal results.

Audio Fusion offers significant creative opportunities in audio production while nevertheless presenting considerable challenges. We demonstrate the fusion editing in fig. 7. As shown, we are given two distinct audio pieces featuring human speech and different music backgrounds, and attempt to arbitrarily combine the speech with different music backgrounds through one-step editing. The PPAE successfully fuses audio components from different sources by utilizing inversion and fusion techniques in the attention map. Although the original audio content of the event is preserved, as shown in fig. 7, we observe that such fusion does not guarantee precise editing at the structural level. This can be attributed to the fact that attention map fusion does not inherit the structure from the source audio, unlike injection methods used in previous tasks such as replacement.

7. Conclusion

In this paper, we introduce a novel approach PPAE for precise audio editing. By adeptly manipulating the attention map of a pre-trained diffusion model, we have demonstrated a training-free and adaptive approach for precise audio editing within diffusion models. Our approach facilitates a wide range of audio editing tasks, including content replacement and recombination, but does so while preserving the semantic essence and overall structure of the original audio. The experiments conducted showcase its potential as a highly effective editing tool. We hope our work can offer a new horizon in audio processing that is both precise and flexible to a myriad of audio editing needs.

future work We see several potential avenues for enhancing the capabilities and applications of our PPAE framework. Briefly, 1) while PPAE demonstrates significant advancements in editing efficiency and flexibility, further research is needed to enhance the overall audio quality; 2) PPAE and similar frameworks require processing time that limits their use in real-time applications, as the diffusion process is relatively slow. 3) it's crucial to continue exploring mechanisms for ensuring the ethical use of PPAE, including safeguards against misuse for creating misleading or harmful content.

Impact Statement

The proposed PPAE is thought beneficial to audio generation and productions related to audio, as it provides a more precise and flexible way to edit audio content automatically. It can potentially enhance the efficiency and effectiveness of audio editing processes in various industries, and make audio editing more accessible to a broader range of users. The potential negative social impact of our methods mainly lines in the fact that precise audio editing technologies can give the evolving landscape of digital content creation and the potential for misuse in generating fake or misleading content. We advocate for increased focus on addressing challenges related to the authenticity of content and its ethical utilization.

Acknowledgements

We would like to thank Dr. Chi Zhang (BIGAI) and Prof. Yixin Zhu (PKU) for helpful discussions. This paper is supported by the Tencent AI Lab, the National Key R&D Program of China (2022ZD0114900), and the NSFC (62172043).

References

Adler, A., Emiya, V., Jafari, M. G., Elad, M., Gribonval, R., and Plumbley, M. D. Audio inpainting. *IEEE Transactions on Audio, Speech, and Language Processing*, 20 (3):922–932, 2012. doi: 10.1109/TASL.2011.2168211.

Birnbaum, S., Kuleshov, V., Enam, Z., Koh, P. W. W., and Ermon, S. Temporal film: Capturing long-range sequence dependencies with feature-wise modulations. *Advances in Neural Information Processing Systems*, 32, 2019.

Cífka, O., Şimşekli, U., and Richard, G. Groove2groove: One-shot music style transfer with supervision from synthetic data. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2638–2650, 2020.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Dong, W., Xue, S., Duan, X., and Han, S. Prompt tuning inversion for text-driven image editing using diffusion models. *arXiv preprint arXiv:2305.04441*, 2023.

Elizalde, B., Deshmukh, S., Al Ismail, M., and Wang, H. Clap learning audio concepts from natural language supervision. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Fonseca, E., Favory, X., Pons, J., Font, F., and Serra, X. Fsd50k: an open dataset of human-labeled sound events. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:829–852, 2021.

Ghosal, D., Majumder, N., Mehrish, A., and Poria, S. Text-to-audio generation using instruction guided latent diffusion model. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 3590–3598, 2023.

Grinstein, E., Duong, N. Q. K., Ozerov, A., and Pérez, P. Audio style transfer. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 586–590, 2018. doi: 10.1109/ICASSP.2018.8461711.

Han, B., Dai, J., Song, X., Hao, W., He, X., Guo, D., Chen, J., Wang, Y., and Qian, Y. Instructme: An instruction guided music edit and remix framework with latent diffusion models. *arXiv preprint arXiv:2308.14360*, 2023.

Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., and Cohen-Or, D. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Huang, J., Ren, Y., Huang, R., Yang, D., Ye, Z., Zhang, C., Liu, J., Yin, X., Ma, Z., and Zhao, Z. Make-an-audio 2: Temporal-enhanced text-to-audio generation. *arXiv preprint arXiv:2305.18474*, 2023a.

Huang, R., Huang, J., Yang, D., Ren, Y., Liu, L., Li, M., Ye, Z., Liu, J., Yin, X., and Zhao, Z. Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models. *arXiv preprint arXiv:2301.12661*, 2023b.

Huberman-Spiegelglas, I., Kulikov, V., and Michaeli, T. An edit friendly ddpm noise space: Inversion and manipulations. *arXiv preprint arXiv:2304.06140*, 2023.

Kawar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., and Irani, M. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6007–6017, 2023.

Kim, G., Kwon, T., and Ye, J. C. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2426–2435, 2022.

Kreuk, F., Synnaeve, G., Polyak, A., Singer, U., Défossez, A., Copet, J., Parikh, D., Taigman, Y., and Adi, Y. Audiogen: Textually guided audio generation. *arXiv preprint arXiv:2209.15352*, 2022.

Li, C., Bai, Y., Wang, Y., Deng, F., Zhao, Y., Zhang, Z., and Wang, X. Image-driven audio-visual universal source separation. 2023.

Liu, H., Chen, Z., Yuan, Y., Mei, X., Liu, X., Mandic, D., Wang, W., and Plumbley, M. D. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023a.

Liu, H., Tian, Q., Yuan, Y., Liu, X., Mei, X., Kong, Q., Wang, Y., Wang, W., Wang, Y., and Plumbley, M. D. Audioldm 2: Learning holistic audio generation with self-supervised pretraining. *arXiv preprint arXiv:2308.05734*, 2023b.

Lu, C.-Y., Xue, M.-X., Chang, C.-C., Lee, C.-R., and Su, L. Play as you like: Timbre-enhanced multi-modal music style transfer. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pp. 1061–1068, 2019.

Mokady, R., Hertz, A., Aberman, K., Pritch, Y., and Cohen-Or, D. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6038–6047, 2023.

Moliner, E. and Välimäki, V. Diffusion-based audio inpainting. *arXiv preprint arXiv:2305.15266*, 2023.

Netzorg, R., Jalal, A., McNulty, L., and Anumanchipalli, G. K. Permod: Perceptually grounded voice modification with latent diffusion models. *arXiv preprint arXiv:2312.08494*, 2023.

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

OpenAI, R. Gpt-4 technical report. *arXiv*, pp. 2303–08774, 2023.

Pan, X., Tewari, A., Leimkühler, T., Liu, L., Meka, A., and Theobalt, C. Drag your gan: Interactive point-based manipulation on the generative image manifold. In *ACM SIGGRAPH 2023 Conference Proceedings*, pp. 1–11, 2023.

Parmar, G., Kumar Singh, K., Zhang, R., Li, Y., Lu, J., and Zhu, J.-Y. Zero-shot image-to-image translation. In *ACM SIGGRAPH 2023 Conference Proceedings*, pp. 1–11, 2023.

Patashnik, O., Garibi, D., Azuri, I., Averbuch-Elor, H., and Cohen-Or, D. Localizing object-level shape variations with text-to-image diffusion models. *arXiv preprint arXiv:2303.11306*, 2023.

Plitsis, M., Kouzelis, T., Paraskevopoulos, G., Katsouros, V., and Panagakis, Y. Investigating personalization methods in text to music generation. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1081–1085. IEEE, 2024.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pp. 8821–8831. PMLR, 2021.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241. Springer, 2015.

Shi, Y., Xue, C., Pan, J., Zhang, W., Tan, V. Y., and Bai, S. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. *arXiv preprint arXiv:2306.14435*, 2023.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Vyas, A., Shi, B., Le, M., Tjandra, A., Wu, Y.-C., Guo, B., Zhang, J., Zhang, X., Adkins, R., Ngan, W., et al. Audiobox: Unified audio generation with natural language prompts. *arXiv preprint arXiv:2312.15821*, 2023.

Wang, Y., Ju, Z., Tan, X., He, L., Wu, Z., Bian, J., and Zhao, S. Audit: Audio editing by following instructions with latent diffusion models. *arXiv preprint arXiv:2304.00830*, 2023.

Wu, C. H. and De la Torre, F. Unifying diffusion models' latent space, with applications to cyclediffusion and guidance. *arXiv preprint arXiv:2210.05559*, 2022.

Yang, D., Tian, J., Tan, X., Huang, R., Liu, S., Chang, X., Shi, J., Zhao, S., Bian, J., Wu, X., et al. Uniaudio: An audio foundation model toward universal audio generation. *arXiv preprint arXiv:2310.00704*, 2023a.

Yang, D., Yu, J., Wang, H., Wang, W., Weng, C., Zou, Y., and Yu, D. Diffsound: Discrete diffusion model for text-to-sound generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023b.

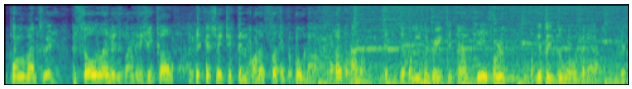
Yang, M., Zhang, C., Xu, Y., Xu, Z., Wang, H., Raj, B., and Yu, D. usee: Unified speech enhancement and editing with conditional diffusion models. *arXiv preprint arXiv:2310.00900*, 2023c.

Zhang, L., Rao, A., and Agrawala, M. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3836–3847, 2023a.

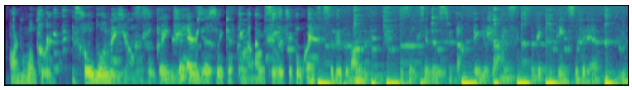
Zhang, Y., Maezawa, A., Xia, G., Yamamoto, K., and Dixon, S. Loop copilot: Conducting ai ensembles for music generation and iterative editing. *arXiv preprint arXiv:2310.12404*, 2023b.

A. More editing results

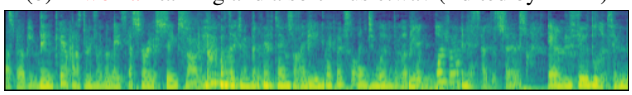
A.1. Audio Replace



(a) A man talking and a soft music.wav (Source)

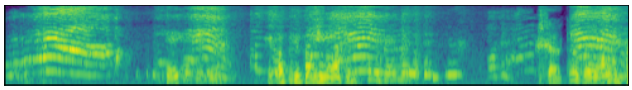


(b) A woman talking and a soft music.wav (Edited by PPAE)

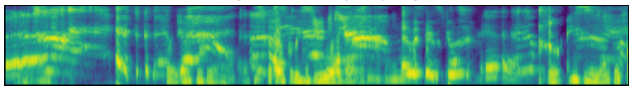


(c) A woman talking and a soft music.wav (Regenerated)

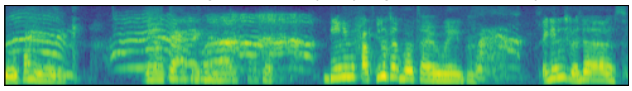
Figure 8: **Case Study (Audio Replace)**



(a) A man talking and a baby crying.wav (Source)

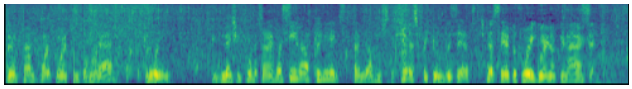


(b) A woman talking and a baby crying.wav (Edited by PPAE)

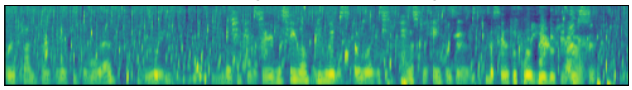


(c) A woman talking and a baby crying.wav (Regenerated)

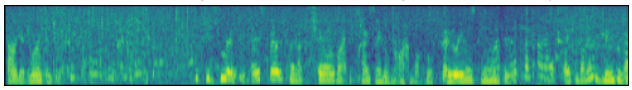
Figure 9: **Case Study (Audio Replace)**



(a) The water flowing and a man talking.wav (Source)

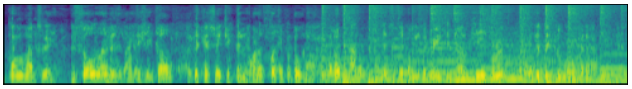


(b) The water flowing and a woman talking.wav (Edited by PPAE)

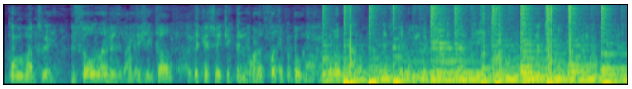


(c) The water flowing and a woman talking.wav (Regenerated)

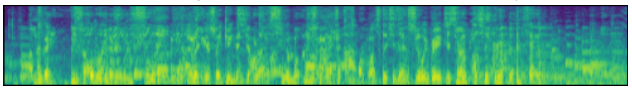
Figure 10: **Case Study (Audio Replace)**



(a) A man talking and a jazz music.wav (Source)

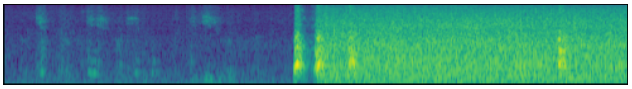


(b) A man talking and a rock music.wav (Edited by PPAE)

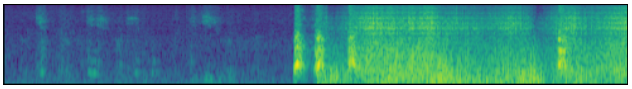


(c) A man talking and a rock music.wav (Regenerated)

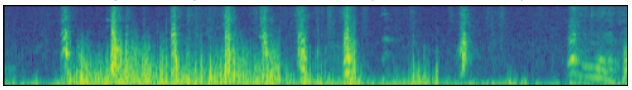
Figure 11: **Case Study (Audio Replace)**



(a) A dog barking and water flowing.wav (Source)

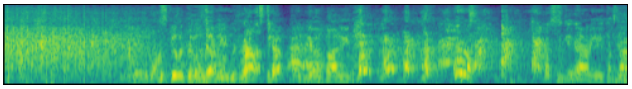


(b) A dog barking and wind blowing.wav (Edited by PPAE)

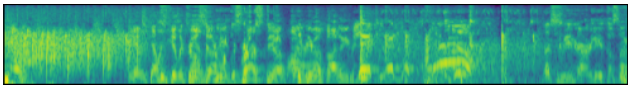


(c) A dog barking and wind blowing.wav (Regenerated)

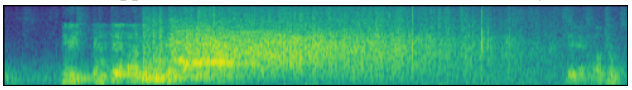
Figure 12: **Case Study (Audio Replace)**



(a) a man applause and a dog barking.wav (Source)



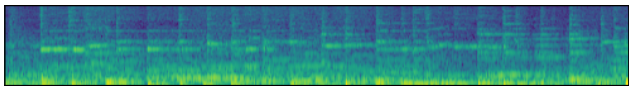
(b) a man applause and a metal collision.wav (Edited by PPAE)



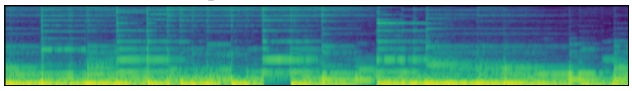
(c) a man applause and a metal collision.wav (Regenerated)

Figure 13: **Case Study (Audio Replace)**

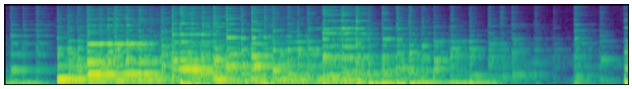
A.2. Audio Refine



(a) A piece of music.wav (Source)

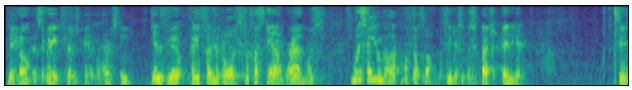


(b) A piece of rock music.wav (Edited by PPAE)

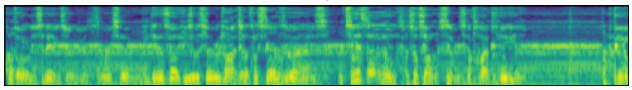


(c) A piece of rock music.wav (Regenerated)

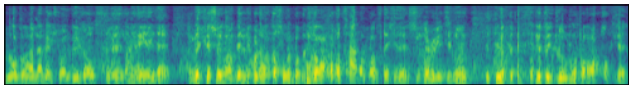
Figure 14: **Case Study (Audio Refine)**



(a) A man talking.wav (Source)

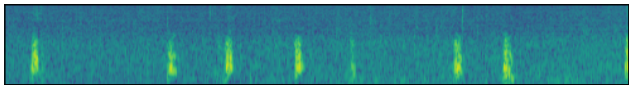


(b) A man talking in a large room.wav (Edited by PPAE)

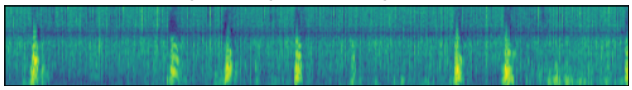


(c) A man talking in a large room.wav (Regenerated)

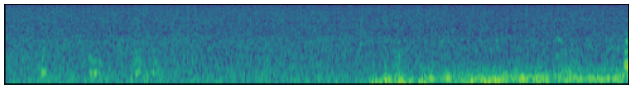
Figure 15: **Case Study (Audio Refine)**



(a) A dog barking and raining.wav (Source)

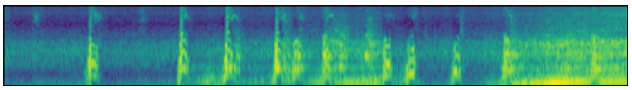


(b) A dog barking and raining heavily.wav (Edited by PPAE)

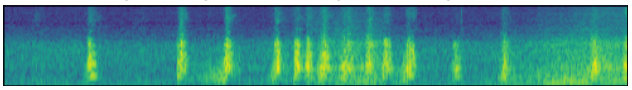


(c) A dog barking and raining heavily.wav (Regenerated)

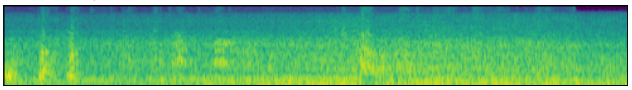
Figure 16: **Case Study (Audio Refine)**



(a) A dog barking and a car engine running.wav (Source)

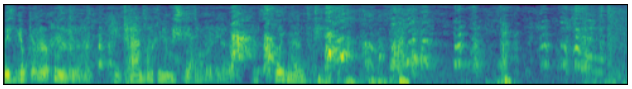


(b) A dog barking and a car engine running in a far place.wav (Edited by PPAE)

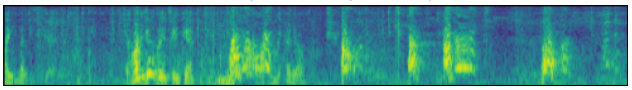


(c) A dog barking and a car engine running in a far place.wav (Regenerated)

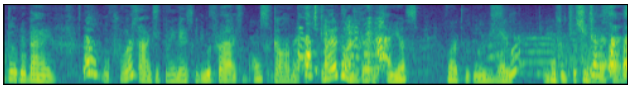
Figure 17: **Case Study (Audio Refine)**



(a) A woman talking and a dog barking.wav (Source)

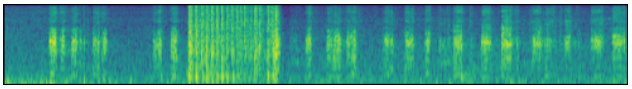


(b) A woman talking and a dog barking loudly.wav (Edited by PPAE)

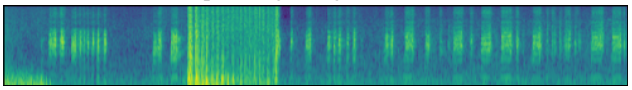


(c) A woman talking and a dog barking loudly.wav (Regenerated)

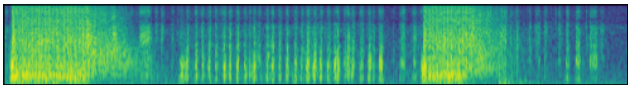
Figure 18: **Case Study (Audio Refine)**



(a) A duck quacking and gunshot.wav (Source)



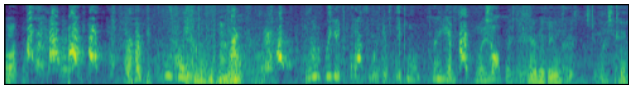
(b) A duck quacking and loud gunshot.wav (Edited by PPAE)



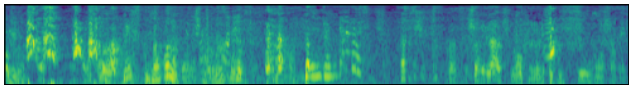
(c) A duck quacking and loud gunshot.wav (Regenerated)

Figure 19: **Case Study (Audio Refine)**

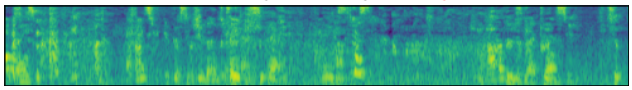
A.3. Audio Reweight



(a) Someone talking and a dog barking.wav, $c = 2$

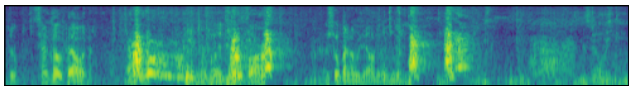


(b) Someone talking and a dog barking.wav, $c = 0$

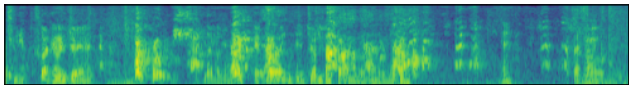


(c) Someone talking and a dog barking.wav, $c = -2$

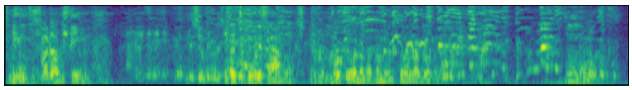
Figure 20: **Case Study (Audio Reweight)**



(a) A man talking and a dog barking.wav, $c = 2$

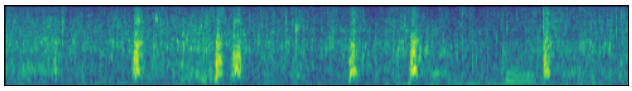


(b) A man talking and a dog barking.wav, $c = 0$

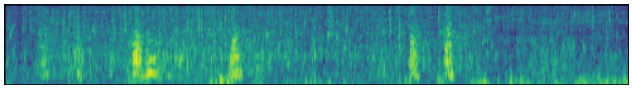


(c) A man talking and a dog barking.wav, $c = -2$

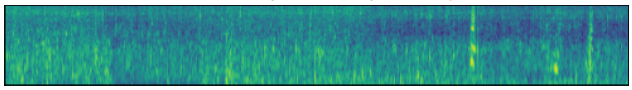
Figure 21: **Case Study (Audio Reweight)**



(a) The water flowing and a dog barking.wav, $c = 2$

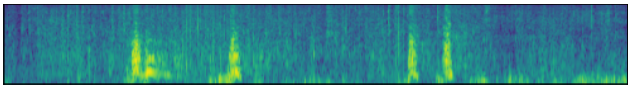


(b) The water flowing and a dog barking.wav, $c = 0$

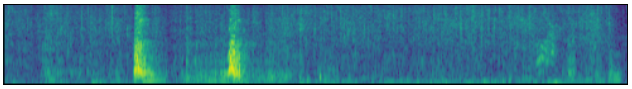


(c) The water flowing and a dog barking.wav, $c = -2$

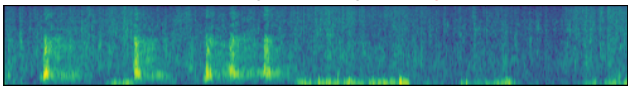
Figure 22: **Case Study (Audio Reweight)**



(a) The rain falling and a dog barking.wav, $c = 2$

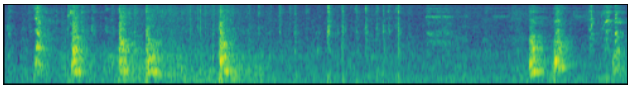


(b) The rain falling and a dog barking.wav, $c = 0$

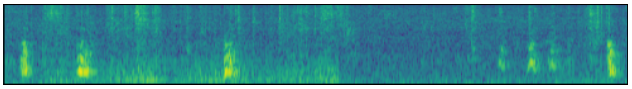


(c) The rain falling and a dog barking.wav, $c = -2$

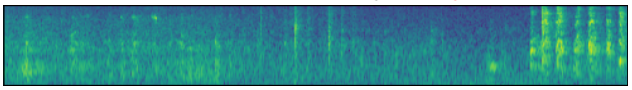
Figure 23: **Case Study (Audio Reweight)**



(a) The machine clicks and a dog barking.wav, $c = 2$

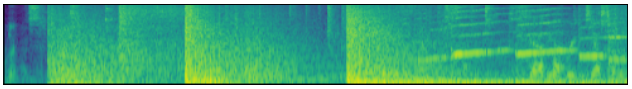


(b) The machine clicks and a dog barking.wav, $c = 0$

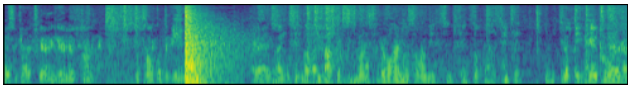


(c) The machine clicks and a dog barking.wav, $c = -2$

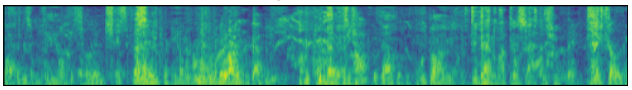
Figure 24: **Case Study (Audio Reweight)**



(a) A man talking and firework.wav, $c = 2$



(b) A man talking and firework.wav, $c = 0$



(c) A man talking and firework.wav, $c = -2$

Figure 25: **Case Study (Audio Reweight)**

B. TTA Models and Preliminaries

B.1. TTA Models

We leverage Tango (Ghosal et al., 2023) as our baseline model in this work for its ability to understand complex concepts in the textual description. We note that our proposed method is also compatible with widely-used diffusion structures, such as Audioldm (Liu et al., 2023a;b) and Make-An-Audio (Huang et al., 2023b;a). In general, these models consist of three primary components: i) a textual-prompt encoder, ii) a LDM, and iii) a mel-spectrogram/audio VAE. The textual-prompt encoder processes the input audio description, which is then utilized to create a latent audio representation or audio prior to standard Gaussian noise through reverse diffusion. Following this, the mel-spectrogram VAE's decoder generates a mel-spectrogram from the latent audio representation. Finally, a vocoder receives the mel-spectrogram as input to produce the resulting audio. Our PPAE method only influences the LDM part.

B.2. Inversion

We elaborate on the inversion function in section 3.6, which involves extracting a sequence of noise vectors that can reconstruct the given source content (image or audio) when used in the reverse diffusion process. Generally, there are two main categories of inversion studied: DDIM inversion and Denoising Diffusion Probabilistic Models (DDPM) inversion. The DDIM scheme employs a deterministic sampling process that maps a single initial noise vector to a generated image, making DDIM inversion relatively simpler. We implement DDIM inversion following the well-known Null-text Inversion algorithm (Mokady et al., 2023), as depicted in algorithm 2:

Algorithm 2 Null-text inversion (DDIM Inversion)

Input: A source prompt embedding $\mathcal{C} = \psi(\mathcal{P})$ and input image $\mathcal{I}$.
Output: Noise vector z_T and optimized embeddings.$\{\varnothing_t\}_{t=1}^{T}$.

Set guidance scale $w = 1$;
Compute the intermediate results $z_T^*, \ldots, z_0^*$ using DDIM inversion over $\mathcal{I}$;
Set guidance scale $w = 7.5$;
Initialize $\bar{z_T} \leftarrow z_T^*$, $\varnothing_T \leftarrow \psi(\text{""})$;
for $t = T, T-1, \ldots, 1$ **do**
  for $j = 0, \ldots, N-1$ **do**
    $\varnothing_t \leftarrow \varnothing_t - \eta \nabla_{\varnothing} \left\| z_{t-1}^* - z_{t-1}(\bar{z}_t, \varnothing_t, \mathcal{C}) \right\|_2^2$;
  end for
  Set $\bar{z}_{t-1} \leftarrow z_{t-1}(\bar{z}_t, \varnothing_t, \mathcal{C})$, $\varnothing_{t-1} \leftarrow \varnothing_t$;
end for
Return $\bar{z_T}$, $\{\varnothing_t\}_{t=1}^{T}$

However, it has been observed that such a DDIM inversion method only becomes effective when a large number of diffusion timesteps are used. Even then, it often results in less than optimal outcomes in text-guided editing. Although the native DDPM noise space is not conducive to editing, researchers have attempted to employ alternative inversion methods to fit better and achieve a more controllable editing space. We adopt the Edit-friendly Inversion (Huberman-Spiegelglas et al., 2023) in DDPM scenarios, as demonstrated in algorithm 3:

Algorithm 3 Edit-friendly DDPM inversion

Input: real image x_0.
Output: $\{x_T, z_T, \ldots, z_1\}$.

for $t = 1$ **to** T **do**
  $\tilde{\epsilon} \sim \mathcal{N}(0,\, 1)$;
  $x_t \leftarrow \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$;
end for
for $t = T$ **to** 1 **do**
  $z_t \leftarrow (x_{t-1} - \hat{\mu}_t(x_t))/\sigma_t$;
  $x_{t-1} \leftarrow \hat{\mu}_t(x_t) + \sigma_t z_t$; // to avoid error accumulation
end for
Return: $\{x_T, z_T, \ldots, z_1\}$

B.3. Baselines

We reimplement the PTP method (Hertz et al., 2022) for audio editing as one of our baselines, primarily because it is closely related to the editing techniques we are exploring. Originally, the PTP method was designed for image editing, where attention maps from the original image are injected into the edited image's attention maps during the diffusion process. To adapt this method for audio input, we migrate PTP to work with TTA LDMs, focusing on the latent representation of the mel-spectrogram for the input audio. It is worth noting that most existing editing works utilizing PTP are based on DDIM inversion. However, for a fair comparison, we adopt the DDPM inversion component from PPAE as the inversion mechanism for PTP.

While both PPAE and PTP have migrated their editing pipelines from image to audio, they differ in two main aspects. Firstly, the core function in editing attention maps is distinct. Audio editing tasks present unique challenges, and a simple injection on attention map layers may not be effective. Secondly, TTA LDMs perform differently during the generation process. Given the complexity involved in the generation and editing process, PPAE requires a global configuration scheduler to boost the quality of the generation results. In the main paper, we primarily discuss the influence of the guidance scale. However, it's worth noting that configurations like generation steps and attention replacement steps also play a crucial role. We demonstrated this in various conducted ablation studies.

C. Objective Evaluation

Fréchet distance FD is a mathematical metric used to measure the similarity or dissimilarity between two curves or sequences in a metric space. In the context of audio, Frechet Distance can be used to compare generated audio samples with target samples.

Fréchet audio distance Inspired by the Fréchet Inception Distance used in image processing, FAD measures the similarity between the distribution of features in the source audio and the edited audio. A lower FAD score indicates that the edited audio is closer to the original in terms of the overall distribution of its features, implying a higher fidelity of the editing process.

Spectral distance This metric evaluates the difference in the spectral characteristics between the original and edited audio. The SD gives a quantitative measure of how much the frequency content has been altered during the editing process.

Kullback–Leibler Divergence KL divergence measures how one probability distribution diverges from the expected one. In the context of audio editing, it is used to compare the distribution of certain audio features between the original and edited audio.

CLAP Score For tasks like refine and reweight, it is challenging to construct a corresponding target audio that can serve as ground truth for comparison. Therefore, we leverage the CLAP as an extra metric to calculate how well the target prompt aligns with the edited audio. The pre-trained CLAP model extracts a latent representation of the given audio and text and returns the audio-text similarity score.

FAD, FD, and KL are well-established and widely accepted metrics in text-to-audio generation tasks. Previous works, such as AudioLDM (Liu et al., 2023a;b), Tango (Ghosal et al., 2023), and AudioGen (Kreuk et al., 2022), have similarly employed these metrics. Our intention is not to claim novelty in using these metrics but to adhere to common practices within the domain, ensuring that our evaluation is convincing and reliable. To compute these metrics, we follow the same evaluation pipeline as AudioLDM, utilizing code from the official repository (`https://github.com/haoheliu/audioldm_eval`).

D. Subjective Evaluation

D.1. Evaluation Metrics

The evaluation metrics involve Relevance and Consistency. Relevance refers to the match between the audio and the input textual prompt, whether the text content appears in the audio, and whether the audio corresponds to the text's semantics. Relevance has scores ranging from 1 (lowest) to 100 (highest). Consistency assesses the degree of similarity between the current audio and the original audio. Note that the current audio and the original audio have different descriptions; for example, the current audio is "a man speaking and a dog barking," while the original audio is "a woman speaking and a dog barking." The consistency assessment focuses on whether the content and rhythm of the man's and woman's speech are consistent, as well as whether the dog's barking is consistent, without paying attention to the difference between "man" and "woman."

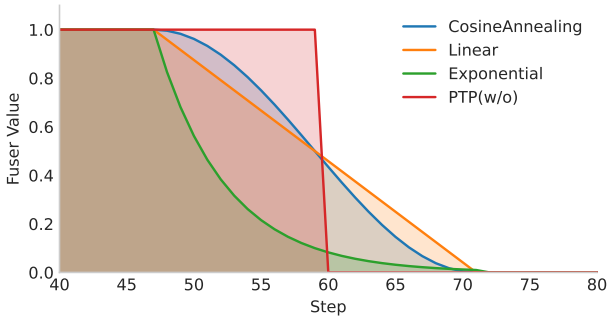


Figure 26: **Decay lines of Fuser with different schedulers.** We assume the cross replace steps is 0.6 and the decay window is 12.

D.2. Test Data

The test set consists of fifteen different audio sets for three editing tasks mentioned in the paper, each containing one original audio file (xxx-0.wav) and two edited audio files (xxx-1.wav and xxx-2.wav). The evaluation process involves scoring each edited audio file based on the criteria mentioned above, with a detailed scoring breakdown provided for both relevance and consistency. The aim is to ensure a comprehensive understanding of the audio editing quality and its effectiveness in achieving the desired editing goals.

D.3. Results with Error Bars

E. Fuser Scheduler

We utilize schedulers to help mix the attention maps of the source audio and the edited audio during the diffusion steps. Specifically, through ablation studies, we find that the CosineAnnealing scheduler is particularly effective in aiding the editing process. Below, we list all the baseline schedulers considered. All of the listed schedulers start from a higher value α_{start} and decay to a lower value α_{end} here, from a start timestep t_0 in a given window size t_w:

Table 7: **Error bars of Reweight CLAP score**

	Original	**2**	**1**	**0**	-1	-2
Reweight	0.74 ± 0.13	0.83 ± 0.09	0.73 ± 0.14	0.35 ± 0.18	0.10 ± 0.07	0.12 ± 0.06
Reweight(PTP)	0.74 ± 0.14	0.79 ± 0.18	0.74 ± 0.25	0.49 ± 0.23	0.32 ± 0.06	0.25 ± 0.09
Unrelated	0.91 ± 0.08	0.93 ± 0.11	0.91 ± 0.07	0.89 ± 0.13	0.82 ± 0.08	0.847 ± 0.13

Table 8: **Error bars of Subjective Evaluation Results**

Metric	**Replace**		**Refine**		**Reweight**	
	PPAE	*Comp*	*PPAE*	*Comp*	*PPAE*	*Comp*
Relevance	95.71 ± 6.22	89.28 ± 12.79	81.42 ± 14.06	81.42 ± 12.45	99.28 ± 2.57	92.14 ± 11.45
Consistency	95.0 ± 5.00	81.42 ± 10.59	85.71 ± 9.03	81.42 ± 11.24	94.28 ± 4.94	82.85 ± 8.80

E.1. Binaray

The scheduler starts with a high value α_{start} for the first t_0 steps. After that, it returns α_{end}. Leveraging this scheduler converts the *Fuser* into a similar way in PTP when mixing attention maps.

$$f(t) = \begin{cases} \alpha_{start} & \text{if } t < t_0 \\ \alpha_{end} & \text{if } t \geq t_0 \end{cases} \tag{7}$$

E.2. Linear

Linear scheduler implements a linear decay of the value from α_{start} to α_{end} over a specified number of steps t_w.

$$f(t) = \begin{cases} \alpha_{start} & \\ & \text{if } t < t_0 \\ \alpha_{start} - \frac{(t-t_0)}{t_w} \cdot (\alpha_{start} - \alpha_{end}) & \\ & \text{if } t_0 \leq t \leq t_0 + t_w \\ \alpha_{end} & \\ & \text{if } t \geq t_0 + t_w \end{cases} \tag{8}$$

E.3. Exponential

The exponential scheduler implements an exponential decay of the value from α_{start} to α_{end} over a specified number of steps t_w, with the decay rate r_{decay}:

$$r_{decay} = (\frac{\alpha_{end}}{\alpha_{start}})^{(\frac{1}{t_w})} \tag{9}$$

$$f(t) = \begin{cases} \alpha_{start} & \text{if } t < t_0 \\ \alpha_{start} \times r_{decay}^{t-t_0} & \text{if } t_0 \leq t \leq t_0 + t_w \\ \alpha_{end} & \text{if } t \geq t_0 + t_w \end{cases} \tag{10}$$

E.4. CosineAnnealing

The implementation of our CosineAnnealing scheduler mainly uses PyTorch's Cosine Annealing learning rate scheduler, in which the decay starts from t_0 to t_w.

$$f(t) = \begin{cases} \alpha_{start} & \\ & \text{if } t < t_0 \\ \frac{\alpha_{start}}{2} \times \left(\cos\left(\frac{\pi \times (t-t_0)}{t_w}\right) + 1\right) & \\ & \text{if } t_0 \leq t \leq t_0 + t_w \\ \alpha_{end} & \\ & \text{if } t \geq t_0 + t_w \end{cases} \tag{11}$$

F. Test Set Construction

For instance, we choose an audio clip a_1 with the prompt p_1 = "a cat meowing" and another a_2 with p_2 = "a baby crying". We combine these as the source audio, generating its description using GPT (OpenAI, 2023). For Audio Editing, we select a third audio clip a_3 with p_3 = "a dog barking". a_3 is randomly replaced with one of the earlier audio clips and merged with the remaining clip to form the target audio for assessment. A corresponding target description is also generated. The data format for this scenario is denoted as $a_1 + a_2 \rightarrow a_1 + a_3$. For the audio refinement task, we consider adding adjective descriptions to one of the original audio pieces and regenerate to get the refined audio $\widetilde{a}$, thus $a_1 + a_2 \rightarrow a_1 + \widetilde{a}_2$. In the audio reweighting task, the format is $a_1 + a_2 \rightarrow a_1 + \alpha \cdot a_2$, wherein the chosen edit target is reweighted prior to merging with another audio clip.

Table 9: **The influence of CLAP selection.**

Select Groups	FAD↓	LSD↓	FD↓	KL↓	CLAP↑
1 (w/ bootstrap)	3.68	1.50	28.32	1.42	0.63
3 (w/ bootstrap)	3.40	1.47	23.91	1.44	0.68
10 (w/ bootstrap)	3.38	1.47	24.24	1.32	0.70
10 (w/o bootstrap)	5.52	2.24	52.41	1.89	0.68

Table 10: **The influence of re-diffusion.**

Extra Steps	FAD↓	LSD↓	FD↓	KL↓	CLAP↑
0	3.72	1.68	29.12	1.51	0.76
10	12.43	3.39	84.90	1.96	0.82
20	14.95	3.35	77.55	1.62	0.74
50	12.80	3.35	71.40	2.03	0.78

G. More Ablation Studies

G.1. Generation Bootstrapping

Bootstrapping generation configurations plays a crucial role in ensuring effective editing, as different editing tasks involve varying components of the original audio, making it challenging to determine the extent of guidance required for the editing process. While our work is the first to explore this issue in editing tasks, similar selection methods using CLAP have been employed in generation tasks to enhance effectiveness. Generally, existing methods create batches of audio and employ CLAP to select the best one. For instance, AudioLDM2 (Liu et al., 2023b) use CLAP to filter generated audios, which they call "clap filtering." Similarly, Audiobox (Vyas et al., 2023) utilizes CLAP reranking with $N = 8$ or even 16 samples using the sound clap model.

We conducted further studies, as illustrated in table 9. First, we attempted to combine batch sampling methods with sample numbers ranging from 1 to 10. The results indicate that this approach only yields a slight improvement in metrics. Next, we removed the bootstrapping module, allowing our model to generate batches of edited audio with a fixed guidance scale of 3 (consistent with the original Tango model). This scenario resulted in a significant performance decline, highlighting the importance of guidance bootstrapping within the entire editing pipeline.

G.2. Regeneration Steps

We observe that regenerating the audio according to the inversion guidance aids in the recovery of the original audio content and structure. However, it negatively affects the audio quality and fidelity, as a more extensive mixture of the attention map results in increased fusion during the intermediate step. This could explain some of the low metrics observed for the PPAE in section 5. To address this issue, intuitively we can incorporate extra diffusion steps, as previously discussed in many previous paper. However, it is crucial to acknowledge the inherent trade-off: while additional diffusion steps lack inversion guidance, increasing their number may cause the generated audio to lose more information from the source audio, as shown in table 10.

H. Limitations

Our proposed audio editing method relies on accurate inversion of the given audio. If the audio content falls outside the model's trained domain, precise editing becomes challenging. Furthermore, our approach primarily focuses on modifications at the attention map level, which inherently restricts the extent of the edits. It may not be suitable for more substantial structural alterations.