April 2025

NEURAL AUDIO EDITING TECHNIQUES

A short literature review on audio editing

Matthieu Cervera
matthieu.cervera@hotmail.com

Introduction

The goal of this report is to conduct a short literature review on recent neural audio editing techniques. I will try to list and describe the most recent convincing methods to edit audio signals. It is interesting to note that editing techniques rely on generative models and if recent large generative models are cited, the focus will be on their editing capabilities. The study will also briefly address techniques used in image editing that to my knowledge have not been implemented for audio yet. As image editing is a more popular and more researched domain, many of audio editing techniques come initially from image editing and therefore latest works on images are interesting to look on. Finally, I will briefly discuss these methods and try to intuit research directions to improve the computational efficiency and editing performance of existing models.

Contents

Trained editing models

"Non edit-specialized" audio generative models

A great portion of models that can achieve audio editing are not really specialized in editing. They are rather large audio generative models that have the capability, given their training, to perform edit tasks. *MusicGen* [1] and *MusicLM* [2] for example are text-to-audio (TTA) autoregressive models leveraging a transformer architecture conditioned on text. Both of these models also allow an optional conditioning on an input audio (such as a melody), allowing to execute text-based editing.

Specialized editing models

AUDIT [3] is a specialized text-based audio editing model. It relies on a diffusion network and on triplets training data. The triplets are constructed specially for audio editing tasks and are composed of an audio input, an audio output and the instruction text. The edits are separated in 5 different tasks (adding, dropping, inpainting, replacement, super-resolution). *InstructME* [4] has a similar architecture and training scheme but also focuses on music editing challenges such as preserving harmony and consistency. To overcome these challenges, it introduces chord progression conditioning and multi-scale aggregation strategies. However, while providing a specialized solution, these techniques are limited in terms of editing tasks and don't generalize well. Moreover, they require the construction of a large triplets dataset from a text/audio dataset before training the model.

Audio Prompt Adapter [5] relies on the training of a light module (the "adapter"). This training is meant to be computationally less demanding (22M parameters instead of 1B+ for a large diffusion model) than training a whole pipeline but still needs a relatively large dataset. Inspired by the *Image Prompt Adapter* [6] pipeline, the adapters feed audio features to the model through attention, in order to perform editing tasks. The goal of the adapters is to add a decoupled audio cross-attention to the existing text cross-attention and therefore to capture both text and audio input features during the sampling process. When trained, the adapter allows the model to perform zero-shot edits. It is an interesting strategy, but still needs relevant training data to optimize the audio cross-attention layers parameters.

Training-free editing models

Training a whole model for editing purposes is demanding in terms of resources and data. Therefore different interesting methodologies leveraging large pre-trained models emerged recently. We can separate these "training-free" methods into two categories. The first category relies on test-time optimization strategies. The second category is built on inversion techniques. Inversion techniques often allow zero-shot editing and therefore seem to be a privileged direction of further research. Note that most of methods from both categories are based on large diffusion TTA models.

Inference-time Optimization

Several methods are exploring the test-time optimization of pre-trained models in order to perform accurate edits. For example, it can be by fine-tuning the model or optimizing the text-embedding to better reconstruct the input. Plitsis et al. [7] explored and evaluated these methods, initially introduced as part of image editing, for music editing. They demonstrated that these methods (namely *Textual Inversion* and *Dreambooth*) could also be effective when applied to audio signals. Another idea based on an image editing method *Imagic* [8] is to combine both methods and use an interpolation between the desired edit text embedding and an optimized version to reconstruct the input audio as demonstrated by Paissan et al. [9]. In

practice, with the diffusion model parameters frozen, the text prompt embedding e_{text} is first optimized to get e_{opt} by minimizing the reconstruction loss between the audio input x_0 and the reconstructed audio : $e_{opt} = argmin_e \mathcal{L}(x_0, Sampler(x_T, e, \epsilon_\theta))$. Then the diffusion model parameters ϵ_θ are fine-tuned to better reconstruct the input (still with a reconstruction loss) at a frozen e_{opt}. Finally, to apply the desired changes to the audio input, the model computes an interpolation between the text embedding and the optimized embedding : $e = \eta e_{opt} + (1 - \eta) e_{text}$. Adjusting the interpolation coefficient η allows to find a balance between preserving the structure and characteristics of the input audio and still applying the desired edits $x_{edited} = Sampler(x_T, e, \epsilon^*_\theta)$.

DITTO's [10] work is based on the assumption that in a diffusion reverse process, the initial noise latent x_T encodes a large proportion of the semantic content in the generated output. Therefore to improve the desired output, the strategy is to optimize this initial noise latent by minimizing a loss between a target feature y, and the feature extracted from the sampled output $x_0 = Sampler(x_T, c, \epsilon_\theta)$. They perform editing by choosing a differentiable feature extractor f and loss $\mathcal{L}$ and then performing gradient descent optimization steps to approximate $x^*_T = argmin_{x_T} \mathcal{L}(f(x_0), y)$. This is a flexible paradigm that can target different key music features (through the feature extractor) for a wide range of editing tasks. However it is too slow for real-time inference and too memory consuming even with gradient checkpointing strategies. Too improve that, *DITTO-2* [11] uses faster sampling techniques leveraging diffusion models distillation, namely Consistency Models Distillation and Consistency Trajectory Models Distillation. Using their distilled model, they first optimize the latent x_T and then sample for M steps to get the edited result.

Inversion techniques

DDPM Inversion

Diffusion Denoising Probabilistic Models (DDPM), also referred as diffusion-models, are generative models that are inspired by thermodynamics. Conceptually, during training the DDPM gradually adds noise to a data point during a forward diffusion process defined by:

$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1 - \alpha_t} \epsilon_t, \quad t \in \{1, ..., T\}$$

where $\{\alpha_t\}_{t=1}^T$ is the noise variance schedule and $\{\epsilon_t\}_{t=1}^T$ are independent normal vectors $\sim \mathcal{N}(0, I)$. With $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$, this process can be rewritten as :

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \tilde{\epsilon}_t, \quad \tilde{\epsilon}_t \sim \mathcal{N}(0, \mathbf{I})$$

where x_0 is a data point sampled from the data distribution and $\tilde{\epsilon}_t$ is an aggregated noise capturing the cumulative effect of all noise steps up to time t. During inference, the DDPM removes the noise iteratively using a neural network ϵ_θ to reconstruct the data points from noisy inputs following the reverse process :

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t, c) \right) + \sigma_t \epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I})$$

The neural network ϵ_θ is trained to predict the noise added at step t, σ_t is the sampling standard deviation. ϵ_θ usually takes as inputs noisy data x_t, the process timestep t and optionally a conditioning information c. The first term, sometimes written $\mu_t(x_t)$ is a linear function of the posterior mean prediction (PMP) $\hat{x}_{0|t}$.To guide the model conditioned on text, recent methods largely adopt classifier-free guidance (CFG) for its efficiency.

For editing tasks, instead of departing from a random noise, we want to alter an input x_0. Inversion involves determining a noise representation that reconstructs the input x_0 upon sampling. To do that, the DDPM inversion computes the noisy data representations (often simply referred to as 'noise vectors') $x_1, ..., x_T$ associated with the input x_0 and uses them during the reverse process. It is worth noting that many diffusion models operate in latent spaces where the input and its noisy representations more commonly appear as $z_0, z_1, ..., z_T$. The basic idea of DDPM inversion, applied for example in *ZETA* [12], is therefore to extract the noise vectors from the input audio with a forward diffusion process while optionally injecting a prompt describing the input audio. Then the model reconstructs the edited audio using the noise vectors to apply the reverse process conditioned on the desired edit prompt. Both directions are conditioned using CFG. However, DDPM processes can fail to reconstruct faithfully the input due to their stochasticity. To improve the results, Manor and Michaeli [12] use an "edit-friendly" DDPM inversion method [13] that computes edit-friendly noise vectors in a different manner than classic DDPM inversion. The edit-friendly noise vectors are different than those used in the original diffusion process but capture better the features of the input. Another interesting contribution of their work is the unsupervised editing capabilities. Instead of using a prompt to guide the edit during the sampling process, *ZEUS* can edit the audio by adding perturbations to the prediction $\hat{x}_{0|t}$ used at each denoising step.

DDIM Inversion

One drawback of DDPM is the large number of unskippable iteration steps ($T \sim 1000$) due to their markovian formulation. Indeed the processes used to alter and reconstruct x_0 require computing sequentially the noisy data representations x_t or the noise $\epsilon_{\theta,t}$ at each timestep. Therefore, during inference DDPM inversion requires two passes through all the timesteps and can be computationally heavy. Denoising Diffusion Implicit Models (DDIM) reformulate the diffusion process in a non-markovian way, by setting the noise schedule σ_t to vanish for example, allowing them to skip timesteps while maintaining quality results.

$$x_{t+1} = \sqrt{\bar{\alpha}_{t+1}} f_\theta(x_t, t, c) + \sqrt{1 - \bar{\alpha}_{t+1}} \epsilon_\theta(x_t, t, c)$$

DDIM usually requires between 50 and 100 steps, significantly reducing inference time. DDIM inversion is conceptually the same as DDPM inversion but using DDIM formulation instead. DDIM proved to be efficient for unconditional generation and with classifier-guidance, however it gets suboptimal when using classifier-free guidance, which is almost necessary for editing purposes. It is therefore used generally in combination with manipulations over the attention or the latent space. Models using DDIM inversion are for example *MusicMagus* [14] and *AudioEditor* [15]. Strategies to improve the DDIM inversion used in these papers will be further described in the next paragraph.

Attention control and latent space manipulations

As stated earlier, DDIM inversion used with CFG causes some issues. Therefore, recent works combine these methods with attention manipulation methods. Even when using DDPM inversion, controlling the attention is a good practice to ensure a better alignment between the audio input and the expected edit.

A strategy found in *AudioEditor* [15] is to optimize the null-text prompts that are used in the CFG reconstruction process : instead of using a constant $\varnothing = TextEncoder(" ")$ the process optimizes null-text prompt embeddings $\{\varnothing_t\}_{t=1}^T$ at each step. To do that, the model first computes a latent trajectory $\{z_t^*\}_{t=1}^T$ using CFG DDIM inversion with $\omega = 1$. Then, at each step from the reverse process, $\varnothing_t$ is optimized several times to minimize the distance between the reversion latent state z_{t-1}^* and the computed latent state $z_{t-1}(z_t, t, e_{prompt}, \varnothing_t)$ at guidance scale

$\omega = 7.5$. This technique, also referred as Null-Text inversion, helps high-fidelity reconstruction of audio but the additional optimization steps adds computational weight. *AudioEditor* also incorporated End-of-Tokens (EOT) suppression. As the model takes as input a target caption that specifies the text edit region, the prompt embedding e_{prompt} can be separated between negative embedding (parts of prompt that needs to be edited) and positive embedding (other parts). Then, using Singular Value Decomposition on the negative and EOT embedding, they apply a regularization technique to suppress or enhance the singular values according to the operation needed. They reconstruct therefore a regularized version of the prompt embedding $\hat{e}_{prompt}$ On top of that, they also introduce an attention loss that compares, at each step, the cross-attention maps between audio embedding and regularized $\hat{e}_{prompt}$ with those between the audio embedding and e_{prompt}. This attention loss is used to further refine $\hat{e}_{prompt}$.

MusicMagus [14] is first thought for neural synthetized audio editing, but shows to be capable of editing real audio using DDIM inversion. We will quickly review the text embedding editing technique and then focus on strategies used to improve the diffusion process. Given an initial text prompt p and its associated generated audio x, *MusicMagus* aims to create a revised audio x' with the specific changes specified by the new prompt p^* (for style and timbre transfer tasks mainly). Instead of directly swapping words in the new text prompt, the model computes an editing direction (Δ) that will then be re-injected to the initial embedding : $e_{edit} = e_{prompt} + \Delta$. The direction Δ is defined as the difference between the means of two sets of embeddings generated from captions related to y and y' created by an instruct language model. As in *AudioEditor*, *MusicMagus* computes at each step an attention loss between cross-attention maps of the edited and the base embedding. However, instead of refining the embedding, they perform a single step optimization on the predicted noise $\epsilon_{\theta,t}$ by adjusting the noisy data representation z_t to minimize the attention loss.

Prompt-guided Precise Audio Editing (PPAE) [16] aims to edit precisely an audio input with a initial descriptive text prompt p and a target prompt p^*. Their strategy relies on the edit-friendly DDPM inversion addressed earlier combined with attention map editing. Their work states that even if some image editing techniques compute a new attention map with the new text prompt embedding and replace the original attention map directly in the diffusion process, it can be suboptimal in audio domain. Therefore they introduce a fuser that computes an edited attention map $M_{edit} = Fuser(M_t, M_t^*, t)$ which behaviour changes depending on the editing task (M_t is the original attention map, M_t^* is the attention map with the new prompt p^*). Their work also adopt a guidance bootstrapping method to meet the challenge of determining an universally applicable guidance scale ω for quality audio generation. They reconstruct z_0 for each guidance scale in a list $W = \{\omega_1, ..., \omega_n\}$ and then, using a CLAP model as a filter function they identify the best output (with the highest relevance to the prompt).

Disentangled Inversion Control

MEDIC [17] introduces a Disentangled Inversion Technique to achieve a more accurate editing using DDIM inversion with CFG. In practice the idea is to separate the "branches" - latents computed by the reverse process - into three : the source, the harmonic and the target branch. The target branch can take advantage of the decoupled harmonic branch through cross-attention manipulations and can perform better editing. In other words, at each step the model reconstructs z_t^{source}, $z_t^{harmonic}$ and z_t^{target} from their respective previous latents. The harmonic branch is the contributing novelty and is the one that hosts composition and structural information from both the source and the target. This is ensured by mutual self-attention control (allowing better non-rigid edits). The self-attention mechanism for the harmonic branch takes the Query from the target while preserving the source information with the Key and Value from the source ($Attention(Q_{target}, K_{source}, V_{source})$). The cross-attention for source and harmonic branches is the same. The melodic and harmonic structure contained in the harmonic branch is passed to the target branch through cross-attention control (allowing also rigid edits) : the cross-attention

map M_t^{har} is used to refine the initial target map M_t^{target} globally and locally. To mitigate even more the errors accumulated because of DDIM inversion and CFG, the source latent are corrected at each step using the distance between the computed z_t^{source} and the original inverted latent z_t^*.

Flow-Matching Inversion

Recently, some diffusion models emerged using rather a Flow-Matching (FM) formulation instead of a standard diffusion process. FM aims to construct optimal paths between data point x and noise ϵ.

$$z_t = tx + (1-t)\epsilon, t \in [0,1]$$

The neural network learns to predict the velocity of the flow (t is usually sampled from a logit distribution) :

$$v_\theta(z_t, t, c) = \frac{dz}{dt} = x - \epsilon$$

Then using an Ordinary Differential Equation (ODE) solver, the model can estimate data x $(= z_1)$ from noise ϵ $(= z_0)$: $x = \text{ODE}_{0\to1}(\epsilon, c)$. This formulation improves the training of diffusion models and speeds up inference.

MelodyFlow [18] introduces a FM formulation model for audio generation and editing. FM has a bijective nature so it can be used in latent inversion techniques. Indeed it shares lots of similarity with the DDIM model that relies also on an ODE. To edit x with a guiding condition c_{edit}, the model will first compute an intermediary noisy representation at t_{edit} $z_{t_{edit}} = \text{ODE}_{t_{edit}\leftarrow1}(x, \varnothing)$ by running the solver in the backward direction. Then it will reconstruct the edited $x_{edit} = \text{ODE}_{t_{edit}\to1}(z_{t_{edit}}, c_{edit})$. However as in DDIM inversion, FM inversion has divergence issues when used with CFG. *MelodyFlow* therefore proposes a regularization method using Kullback-Leibler regularization on velocity predictions, making it suitable for the FM formulation. That way, the inversion method is able to build an almost fully reversible trajectory.

Novel image editing methods

Improving existing methods for faster inference and better quality

To reduce the computation time induced by null-text inversion, Miyake et al. [19] propose a method called negative-prompt inversion. The idea is to replace the optimized null-text embeddings $\{\varnothing_t\}_{t=1}^T$ used in the unconditional prediction by the text prompt embeddings c_{prompt} in null-text inversion. The CFG in this case is newly defined as : $\tilde{\epsilon}_\theta(z_t, t, e_{prompt}, e_{edit}) = w \cdot \epsilon_\theta(z_t, t, e_{edit}) + (1-w) \cdot \epsilon_\theta(z_t, t, e_{prompt})$. By combining their method with for example some cross-attention control, they can perform quicker image editing while maintaining most of the quality of the output image.

DiT4Edit [20] uses a transformer backbone for their diffusion model instead of the classic U-Net. Diffusion transformers, by design, are built upon native cross-attention and self-attention and are therefore better suited for applying unified attention control. This architecture seems better at capturing long-range object information and performing large-scale edits. To improve the sampling speed, *DiT4Edit* also uses a high-order DPM-Solver to invert the input image in less steps than DDIM inversion. Finally, for mutual self-attention control, they reduce the large number of patches processed by the attention mechanism by merging the most similar ones.

Consistency "inversion"

InfEdit [21] presents a Denoising Diffusion Consistency Model (DDCM) as a sampling strategy that eliminates the need of an inversion process and can work with Consistency sampling. By taking $\sqrt{1-\alpha_{t-1}}$ as the noise schedule in the classic diffusion reverse process at step t, they remove the term representing the direction to z_t. The reverse process therefore can be expressed like consistency sampling : $z_{t-1} = \sqrt{\alpha_{t-1}} f(z_t, t; z_0) + \sqrt{1-\alpha_{t-1}}\epsilon_t$. The self-consistency of f is ensured by solving $f(z_t, t; z_0) = z_0$ and computing ϵ_{cons} without parametrization. Obviously this formulation could not work for a simple generative model, but as the aim is to edit an image, z_0 is known. This so called Virtual inversion can reconstruct perfectly z_0 because of the self-consistency. For editing tasks, using a consistency model ϵ_θ, the technique is to estimate the difference $\Delta\epsilon$ between the noise from the target and the source. Then this difference is added to the constrained noise ϵ_{cons} to compute the edited image z_0^{target}. Combined with attention control (creating a new branch using the same framework for attention as in *MEDIC*), their work shows strong performances on image editing.

Localized editing

LOCO [22] method performs localized unsupervised editing that can also be seamlessly plugged into to text-to-image model for supervised editing. First they perform DDIM inversion to get the inverted input image x_T. The goal of *LOCO* is to find the direction vector v_p that will guide the diffusion model during the reverse process to perform the edit. As in *ZEUS* [12], the authors suggest that the most semantically meaningful editing directions correspond to the top eigenvalues (or singular values) of the jacobian of the posterior mean prediction (PMP). Unlike in *ZEUS* where the output of the reverse process is perturbed at each step however, *LOCO* perturbs only once the input x_T of the reverse process : $x_{edit} = DDIM(x_T + \lambda v_p)$ justified by the low-rank and linearity of the PMP. To perform localized edits, they use a mask Ω and compute both jacobians of the masked prediction and the unmasked prediction. Then, they project the singular value of the masked prediction jacobian $v \in range(J_{\Omega,\theta,x_t})$ to the null space of the unmasked prediction jacobian $v_p = proj_{null(J_{\hat{\Omega},\theta,x_t})}(v)$. For supervised editing, the initial editing direction v is based on the difference between the original prompt and the edit prompt conditioned estimated posterior means.

Editing with Visual Autoregressive Models

AR-Edit [23] is a novel training-free image editing method leveraging an autoregressive (AR) model instead of a diffusion model. Recent advances in AR modelling introduced a new paradigm called Visual Autoregressive Modelling (VAR) that seems to surpass diffusion transformers models for image generation. Shortly, instead of formulating the sequence x likelihood as a next-token prediction (vanilla AR models), VAR reformulates it as a "next-scale prediction". In other words, rather than being a single token, the AR units in VAR are entire token maps. *AR-Edit* is based on an improved VAR. To perform editing, it stores the token maps (R_k), the attention maps weight (W_k) and the probability distributions (P_k) of the input image and prompt by performing a single forward pass. Then during the editing inference (guided with the target text-prompt), they compare the cached probability P_k with the target P_k^{target} computed at step k and store the difference in masks M_k that indicate the most likely maps to change. They introduce γ that will balance the fidelity to the input image and to the desired edit. For steps k before γ, the cached token maps from the input are used. After $k > \gamma$, they use the mask M_k to guide the sampling of new target token maps. They also refine the attention map using the cached and the newly computed attention maps. Their method seem to show high-fidelity performances and fast inference speed.

Research directions ideas

Between all the different methods and strategies listed in this report, focusing on training-free techniques seems to be a privileged research direction. Indeed even if the huge audio generative models can achieve excellent results, the amount of data and the computation needs to train such models could prevent us to even match the state-of-the-art results. The same point can be made for specialized models that need to process an enormous amount of data for training but still are limited to their learned editing tasks and don't generalize well.

The inference-time optimization techniques bring interesting training-free alternatives, but the several optimizing steps needed for each different editing task can rapidly be too computationally demanding. However, it is important to keep these strategies in mind, because some inversion techniques also include optimizing steps that slow the inference. For example the improvements made by *DITTO-2* to achieve faster sampling using Consistency Models might be worth exploring. For already existing inversion techniques applied to audio editing, *MEDIC* seems to show the most recent and remarkable results by combining disentangled DDIM inversion and unified attention control. Masking strategies can also allow a better control over the desired edit.

Focusing on novel image editing methods, I think the virtual consistency "inversion" is particularly interesting and is worthy of further exploration for audio editing. Consistency Models allow for either one-step or multi-step sampling, with the number of steps adjustable based on the desired level of refinement, and it seems to be a good direction to look at for a fast inference. Also *InfEdit* uses a similar unified attention control as *MEDIC* that could help improving the quality of the results. These thoughts are also motivated by the impressive audio fidelity rendered by recent generative audio consistency models. Finally the virtual inversion and the fast inference could give us more room to add some optimizing steps without compromising too much a real-time use of the editing method.

References

[1] Jade Copet et al. *Simple and Controllable Music Generation*. 2024. arXiv: 2306.05284 [cs.SD]. URL: https://arxiv.org/abs/2306.05284.

[2] Andrea Agostinelli et al. *MusicLM: Generating Music From Text*. 2023. arXiv: 2301.11325 [cs.SD]. URL: https://arxiv.org/abs/2301.11325.

[3] Yuancheng Wang et al. *AUDIT: Audio Editing by Following Instructions with Latent Diffusion Models*. 2023. arXiv: 2304.00830 [cs.SD]. URL: https://arxiv.org/abs/2304.00830.

[4] Bing Han et al. *InstructME: An Instruction Guided Music Edit And Remix Framework with Latent Diffusion Models*. 2023. arXiv: 2308.14360 [cs.SD]. URL: https://arxiv.org/abs/2308.14360.

[5] Fang-Duo Tsai et al. *Audio Prompt Adapter: Unleashing Music Editing Abilities for Text-to-Music with Lightweight Finetuning*. 2024. arXiv: 2407.16564 [cs.SD]. URL: https://arxiv.org/abs/2407.16564.

[6] Hu Ye et al. *IP-Adapter: Text Compatible Image Prompt Adapter for Text-to-Image Diffusion Models*. 2023. arXiv: 2308.06721 [cs.CV]. URL: https://arxiv.org/abs/2308.06721.

[7] Manos Plitsis et al. *Investigating Personalization Methods in Text to Music Generation*. 2023. arXiv: 2309.11140 [cs.SD]. URL: https://arxiv.org/abs/2309.11140.

[8] Bahjat Kawar et al. *Imagic: Text-Based Real Image Editing with Diffusion Models*. 2023. arXiv: 2210.09276 [cs.CV]. URL: https://arxiv.org/abs/2210.09276.

[9] Francesco Paissan et al. "Audio Editing with Non-Rigid Text Prompts". In: *Interspeech 2024*. ISCA, 2024. DOI: 10.21437/interspeech.2024-636. URL: http://dx.doi.org/10.21437/Interspeech.2024-636.

[10] Zachary Novack et al. *DITTO: Diffusion Inference-Time T-Optimization for Music Generation*. 2024. arXiv: 2401.12179 [cs.SD]. URL: https://arxiv.org/abs/2401.12179.

[11] Zachary Novack et al. *DITTO-2: Distilled Diffusion Inference-Time T-Optimization for Music Generation*. 2024. arXiv: 2405.20289 [cs.SD]. URL: https://arxiv.org/abs/2405.20289.

[12] Hila Manor and Tomer Michaeli. *Zero-Shot Unsupervised and Text-Based Audio Editing Using DDPM Inversion*. 2024. arXiv: 2402.10009 [cs.SD]. URL: https://arxiv.org/abs/2402.10009.

[13] Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. *An Edit Friendly DDPM Noise Space: Inversion and Manipulations*. 2024. arXiv: 2304.06140 [cs.CV]. URL: https://arxiv.org/abs/2304.06140.

[14] Yixiao Zhang et al. *MusicMagus: Zero-Shot Text-to-Music Editing via Diffusion Models*. 2024. arXiv: 2402.06178 [cs.SD]. URL: https://arxiv.org/abs/2402.06178.

[15] Yuhang Jia et al. *AudioEditor: A Training-Free Diffusion-Based Audio Editing Framework*. 2024. arXiv: 2409.12466 [cs.SD]. URL: https://arxiv.org/abs/2409.12466.

[16] Manjie Xu et al. *Prompt-guided Precise Audio Editing with Diffusion Models*. 2024. arXiv: 2406.04350 [cs.SD]. URL: https://arxiv.org/abs/2406.04350.

[17] Huadai Liu et al. *MEDIC: Zero-shot Music Editing with Disentangled Inversion Control*. 2024. arXiv: 2407.13220 [eess.AS]. URL: https://arxiv.org/abs/2407.13220.

[18] Gael Le Lan et al. *High Fidelity Text-Guided Music Editing via Single-Stage Flow Matching*. 2024. arXiv: 2407.03648 [eess.AS]. URL: https://arxiv.org/abs/2407.03648.

[19] Daiki Miyake et al. *Negative-prompt Inversion: Fast Image Inversion for Editing with Text-guided Diffusion Models*. 2024. arXiv: 2305.16807 [cs.CV]. URL: https://arxiv.org/abs/2305.16807.

[20] Kunyu Feng et al. *DiT4Edit: Diffusion Transformer for Image Editing*. 2024. arXiv: 2411.03286 [cs.CV]. URL: https://arxiv.org/abs/2411.03286.

[21] Sihan Xu et al. *Inversion-Free Image Editing with Natural Language*. 2023. arXiv: 2312.04965 [cs.CV]. URL: https://arxiv.org/abs/2312.04965.

[22] Siyi Chen et al. *Exploring Low-Dimensional Subspaces in Diffusion Models for Controllable Image Editing*. 2024. arXiv: 2409.02374 [cs.CV]. URL: https://arxiv.org/abs/2409.02374.

[23] Yufei Wang et al. *Training-Free Text-Guided Image Editing with Visual Autoregressive Model*. 2025. arXiv: 2503.23897 [cs.CV]. URL: https://arxiv.org/abs/2503.23897.