Coherent Audio-Visual Editing via Conditional Audio Generation Following Video Edits

Anonymous authors
Paper under double-blind review

Abstract

We introduce a novel pipeline for joint audio-visual editing that enhances the coherence between edited video and its accompanying audio. Our approach first applies state-of-the-art video editing techniques to produce the target video, then performs audio editing to align with the visual changes. To achieve this, we present a new video-to-audio generation model that conditions on the source audio, target video, and a text prompt. We extend the model architecture to incorporate conditional audio input and propose a data augmentation strategy that improves training efficiency. Furthermore, our model dynamically adjusts the influence of the source audio based on the complexity of the edits, preserving the original audio structure where possible. Experimental results demonstrate that our method outperforms existing approaches in maintaining audio-visual alignment and content integrity.

1 Introduction

Recent advancements in video editing have enabled sophisticated manipulation of visual content (Jiang et al., 2025; Kara et al., 2024). However, these techniques typically address only the visual modality, overlooking the crucial role of audio in most real-world videos. As a result, there is a growing demand for editing solutions that maintain strong audio-visual coherence. The emergence of audio-visual generation models, such as Veo 3 (DeepMind, 2025) and Sora 2 (OpenAI, 2025), underscores the increasing importance of synchronized audio and video in content creation.

Existing approaches to joint audio-video editing face significant limitations. Editing audio and video modalities independently often results in poor alignment, as changes in one stream may not correspond to changes in the other. Alternatively, applying video-to-audio models to edited videos can disrupt the integrity of the original audio. To address these issues, joint editing methods have been studied (Liang et al., 2024; Lin et al., 2025), but they are restricted to low frame-rate videos (e.g., 4 fps (Lin et al., 2025)), which is impractical for most applications. In contrast, modern video editing methods can process much higher frame rates (e.g., 16 fps (Jiang et al., 2025)), but lack integrated audio editing capabilities. More recent methods (Zheng et al., 2026a; Ben-Yosef et al., 2026) enable high-fps editing by introducing sophisticated model architectures or leveraging powerful pre-trained models. However, these methods require additional inputs such as the target object mask (Zheng et al., 2026a) or other control information (Ben-Yosef et al., 2026).

To address these challenges, we decouple joint audio-video editing into two sequential sub-tasks as shown in Fig. 1. We first apply an off-the-shelf video editing method to transform the source video into the target video, leveraging recent progress in high frame-rate video editing without modification. We then introduce a new task setting, **audio editing following video edits**, where the goal is to edit the source audio to align with the edited video while preserving the original audio structure as much as possible (e.g., the timing and number of sound events, and the continuity of background ambience), guided by a text prompt. This task is particularly challenging because the model must jointly satisfy text fidelity, audio-visual alignment to the edited visual events, and audio structure preservation.

For this new task, we propose a conditional video-to-audio editing model built by minimally extending a state-of-the-art video-to-audio generation model (Cheng et al., 2025). Specifically, we modify the model to accept the source audio as an additional condition alongside the target video and text, enabling it to preserve

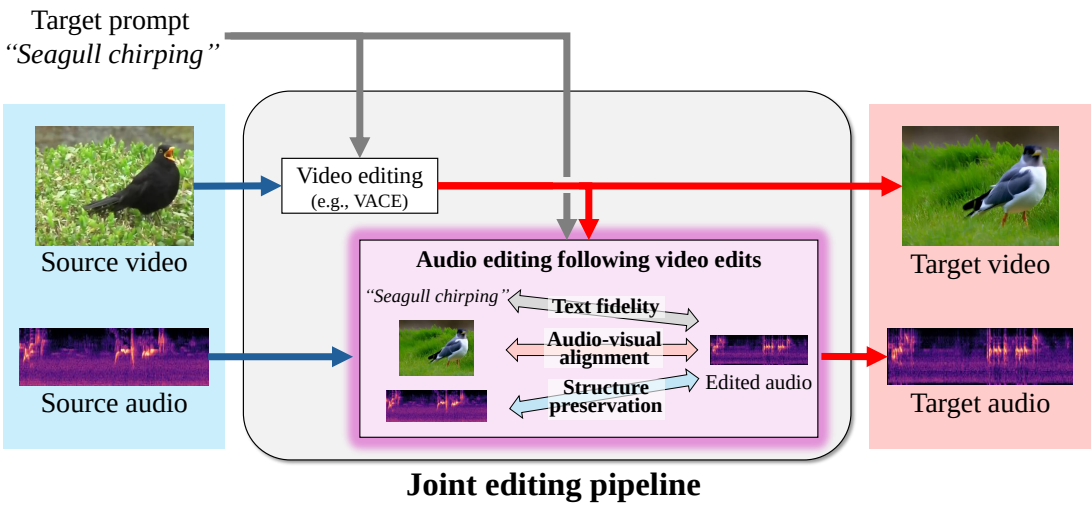


Figure 1: An overview of the proposed pipeline.

the original audio structure. We further introduce a data augmentation strategy for this new conditional input to improve training efficiency, and we also adopt an adaptive conditioning mechanism to adjust the influence of the source audio based on the complexity of the edits, maximizing audio structure preservation. With these changes, our model can dynamically balance text fidelity, audio-visual alignment, and audio structure preservation. Experimental results demonstrate that our pipeline achieves superior performance compared to existing methods, particularly in maintaining audio-visual alignment and content integrity.

2 Preliminaries and related work

2.1 Flow matching

Flow matching (Lipman et al., 2023) is a framework for generative modeling that has been widely adopted across various domains due to its strong performance and high scalability. It employs a strategy essentially similar to diffusion models (Ho et al., 2020) but offers greater flexibility in model design (Gao et al., 2025). In this framework, the model learns a time-dependent velocity field. Let $\mathbf{x}_0$ and $\mathbf{x}_1$ denote samples from the prior and target distributions, respectively, and define $\mathbf{x}_t = t\mathbf{x}_0 + (1-t)\mathbf{x}_1$ as the linear interpolation between them at time $t \in [0, 1]$. The model $u_\theta(\mathbf{x}, t)$, parameterized by θ, predicts the velocity at each time t.

The model is trained by minimizing the conditional flow-matching objective:

$$\min_\theta \mathbb{E}_{p(\mathbf{x}_0),p(\mathbf{x}_1),\mathcal{U}(t)} \|u_\theta(\mathbf{x}_t, t) - u(\mathbf{x}_t|\mathbf{x}_0, \mathbf{x}_1)\|^2, \tag{1}$$

where $u(\mathbf{x}_t|\mathbf{x}_0, \mathbf{x}_1)$ is the conditional velocity. In the current setting, where $\mathbf{x}_t$ is defined as a linear interpolation between $\mathbf{x}_0$ and $\mathbf{x}_1$, the conditional velocity simplifies to $u(\mathbf{x}_t|\mathbf{x}_0, \mathbf{x}_1) = \mathbf{x}_1 - \mathbf{x}_0$.

Given a trained flow-matching model, we can define the following ordinary differential equation (ODE):

$$\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = u_\theta(\mathbf{x}_t, t). \tag{2}$$

By solving this ODE from $t = 0$ (with initial sample $\mathbf{x}_0 \sim p(\mathbf{x}_0)$) to $t = 1$, we can transform a sample from the prior distribution into a sample from the target distribution $p(\mathbf{x}_1)$. By choosing a tractable distribution (e.g., Gaussian) for $p(\mathbf{x}_0)$ and the target data distribution for $p(\mathbf{x}_1)$, this model can serve as a generative model.

When modeling conditional generation, the model is extended to accept a conditional input $\mathbf{C}$, resulting in $u_\theta(\mathbf{x}_t, t, \mathbf{C})$, and the training objective in Eq. (1) is modified to sample from the joint distribution $p(\mathbf{x}_1, \mathbf{C})$

instead of $p(\mathbf{x}_1)$. During inference, classifier-free guidance (Ho & Salimans, 2021) can be applied to the velocity prediction to improve fidelity to the given condition:

$$\tilde{u}_\theta(\mathbf{x}_t, t, \mathbf{C}) = u_\theta(\mathbf{x}_t, t, \varnothing) + w(u_\theta(\mathbf{x}_t, t, \mathbf{C}) - u_\theta(\mathbf{x}_t, t, \varnothing)), \tag{3}$$

where $\varnothing$ denotes the null condition, and w is a coefficient to control the strength of the guidance. To enable the model to handle the null condition, the conditional input is randomly dropped with a small probability (e.g., 0.1) during training.

2.2 Single-modal data editing

Editing refers to the process of creating target data by modifying specific content in given source data, as intended by the user, while preserving unrelated elements. This section briefly reviews editing methods based on flow matching or diffusion models in both audio and video domains.

Audio Editing. Audio editing methods can be broadly categorized into training-based and training-free approaches. In the training-based approach (Wang et al., 2023; Lan et al., 2025), models are trained as conditional generators that take source audio as input and produce edited outputs. This approach is conceptually straightforward, but it requires large-scale datasets of paired source and target audio samples, which is challenging in practice. In contrast, training-free approaches (Manor & Michaeli, 2024; Xu et al., 2024; Jia et al., 2025) leverage pre-trained text-to-audio generation models (Liu et al., 2023; 2024a; Ghosal et al., 2023) to generate the desired audio edits. Although they eliminate the need for paired training datasets, they require techniques to inject source audio information into the generation process, which typically incurs significantly higher computational costs during generation. Our method is also training-based one, but we mitigate the dataset issue by adopting acoustic feature extraction as described in Section 3.2.

Video Editing. Training video generation models from scratch is considerably more resource-intensive than audio models. As a result, most video editing methods rely on pre-trained generative models. There are two main types of pre-trained models used: text-to-image and text-to-video models.

Early work leveraged text-to-image models such as Stable Diffusion (Rombach et al., 2022) due to their accessibility and ease of use. Since these models are only capable of generating single frames, specialized techniques (Cong et al., 2024; Li et al., 2024; Cohen et al., 2024; Geyer et al., 2024; Kara et al., 2024) and fine-tuning strategies (Wu et al., 2023; Liu et al., 2024b) have been developed to improve temporal consistency across video frames. Recently, the emergence of advanced text-to-video generation models (Blattmann et al., 2023; Yang et al., 2025; Wan et al., 2025) has enabled more effective video editing. Studies utilizing these models (Fan et al., 2024; Shi et al., 2024; Zhang et al., 2025; Jiang et al., 2025) demonstrate excellent temporal consistency.

2.3 Joint audio-video editing

Joint audio-video editing presents unique challenges compared to single-modal editing. Editing audio and video independently often results in poor alignment between the generated modalities, degrading the overall audio-visual coherence of the edited output. Therefore, ensuring strong alignment between audio and video is a central focus in this task, and it has been addressed in several recent studies (Liang et al., 2024; Lin et al., 2025; Zheng et al., 2026a; Ben-Yosef et al., 2026).

Liang et al. (2024) introduced a lightweight module that embeds the source audio-video pair into a specialized text token. This token is then fed into pre-trained text-to-image and text-to-audio models to generate the corresponding video frames and audio. In contrast, Lin et al. (2025) extended score distillation sampling to the cross-modal setting by jointly leveraging pre-trained text-to-image and text-to-audio models. They further incorporated a contrastive loss on relevant and irrelevant regions within the audio-video pair to improve consistency between audio and video edits. However, since both approaches rely on text-to-image models for video editing and lack mechanisms for precise temporal synchronization between the edited audio and video, their applicability is essentially limited to low frame-rate videos (e.g., 1 fps (Liang et al., 2024) or 4 fps (Lin et al., 2025)).

To address the low-frame-rate limitation, more recent methods leverage stronger video or joint audio-video generation models. AVI-Edit (Zheng et al., 2026a) performs joint audio-video editing by integrating three components: a state-of-the-art video generation model extended for audio synchronization, an audio editing agent, and a module that refines user-provided masks to localize the editing target in the video. Meanwhile, AVControl (Ben-Yosef et al., 2026) is built by applying parameter-efficient fine-tuning to a pre-trained joint audio-video generation model, enabling it to handle control signals provided at editing time. While these approaches improve the capability of joint audio-video editing, they assume access to additional user-provided information, such as object-region masks or other control signals. In contrast, this paper tackles a more challenging setting in which only a target caption describing the edited video is given. As concurrent work to this paper, InstructAV2AV (Zheng et al., 2026b) tackles a similar task; however, it requires paired audio-video data from before and after editing for training. Our work, on the other hand, proposes a training framework that does not require such data, which is difficult to collect at scale.

3 Proposed method

In this section, we first show an overview of our method and briefly explain how it works. Then, we describe the details of the newly introduced mechanisms designed for the video-guided audio editing task.

3.1 Overview

Our goal is to build a model that generates edited audio aligned with the target edited video while preserving the acoustic structure of the source audio. To achieve this, we first extract hierarchical acoustic features from the source audio to capture its structural information at multiple levels of detail (Section 3.2). These features, together with conditional features from the target video and a target prompt, are fed into the model (Section 3.3), which is trained to generate the edited audio using a flow matching framework. To improve training, we introduce a new data-augmentation strategy for the acoustic features called detail-temporal masking (Section 3.4). During generation, the model automatically selects the appropriate level of detail for the acoustic features based on the estimated editability score, which is determined by measuring the semantic similarity between the source audio and the target video (Section 3.5).

3.2 Hierarchical acoustic feature extraction

Motivation. To preserve the acoustic structure of the source audio, we need to extract this information from it. In our setting, the feature representation should satisfy the following three properties:

- The feature effectively represents the acoustic structure information.
- The feature is robust to semantic changes during editing, ensuring it remains informative for generating the edited audio.
- The feature can be extracted at multiple levels of detail, enabling control over the amount of preserved structural information.

A common choice to extract audio features would be using a pretrained model. However, such a model generally does not satisfy the second property, as they are typically trained to be sensitive to differences in audio semantics (e.g., CLAP (Elizalde et al., 2023)). As a result, when used as "structure" features, they tend to overly preserve the original semantic content of the source audio (e.g., specific sound identities), which hinders semantic edits and can cause the editing to fail. In addition, extracting semantic features at a specified level of detail is challenging, which conflicts with the third property. Consequently, we design simpler statistical features of audio signals that satisfy all three properties. Statistical features, such as loudness, are less sensitive to semantic changes in audio content, making them more reliable for preserving structural information (García et al., 2025).

Implementation. Inspired by Sketch2Sound (García et al., 2025), our acoustic feature is based on the frame-wise loudness of the audio signal processed with a median filter for temporal smoothing. We follow a common definition of loudness, which is an A-weighted sum across the frequency bins in a magnitude spectrogram (Morrison et al., 2024). To extract acoustic features at various levels of detail, we recursively split the frequency bins of the magnitude spectrogram into two sub-ranges (e.g., low and high frequencies) and compute the loudness for each sub-range at every level. Let $\mathbf{s} \in \mathbb{R}^{F \times T_a}$ denote the spectrogram of the audio, where F is the number of frequency bins and T_a is the number of time-frames. At level l, $\mathbf{s}$ is divided into $\{\mathbf{s}_i^{(l)} \in \mathbb{R}^{F/2^l \times T_a}\}_{i=0,\ldots,2^l-1}$, and the acoustic features are calculated as follows:

$$\mathbf{a}_l = \text{concat}\left(\left\{A(\mathbf{s}_i^{(l)})\right\}_{i=0}^{2^l-1}\right), \tag{4}$$

where A is a function that computes the A-weighted sum, and concat denotes concatenation along the frequency axis.

As shown in Fig. 2, this process yields a hierarchy of acoustic features $\{\mathbf{a}_l\}$, from coarse to fine detail, allowing flexible control over structural information preservation. The level of detail can be controlled by masking out the features corresponding to finer details. Specifically, the acoustic features at the specified level of detail are given by:

$$\mathbf{a} = \text{concat}\left(\{\mathbf{a}_l \odot \mathbf{m}_l, \mathbf{m}_l\}_{l=0}^{L}\right), \tag{5}$$

where L is the maximum level, and $\mathbf{m}_l \in \mathbb{R}^{2^l \times T_a}$ denote a mask at level l. In the remainder of this paper, we refer to $\mathbf{a}$ as the acoustic features.

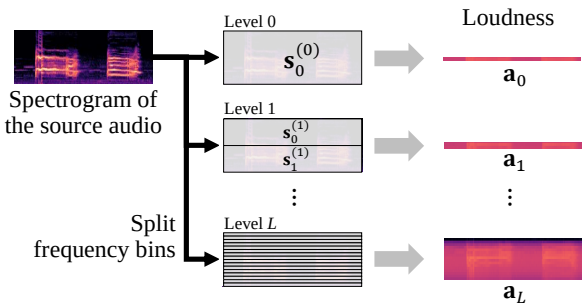


Figure 2: Hierarchical acoustic features.

During inference, the mask is defined as

$$\mathbf{m}_l = \begin{cases} \mathbf{1} & \text{if } l \leq l^*, \\ \mathbf{0} & \text{otherwise,} \end{cases} \tag{6}$$

where l^* is the specified level of detail, and $\mathbf{0}$ and $\mathbf{1}$ denote matrices of all zeros and ones, respectively. That is, elements at levels finer than l^* are masked out. During training, we apply further augmentation to the mask, as described in Section 3.4.

3.3 Model architecture

Our model shown in Fig. 3 is based on the MMAudio architecture (Cheng et al., 2025), with two key modifications to effectively handle information from the acoustic features.

Base architecture. The backbone of the base model is a multi-modal Transformer that processes noisy audio latent features, as well as conditional text and video features, to predict the velocity for audio generation through the flow matching framework. It comprises a sequence of MM-DiT blocks (Esser et al., 2024) followed by a sequence of audio-only single-modal DiT blocks. Additionally, the base model includes a conditional synchronization module to enhance audio-visual synchrony. This module extracts high frame-rate features from the input video using Synchformer (Iashin et al., 2024) and combines them with temporally-global conditional features extracted from text and video. The combined features are fed into the DiT blocks through adaptive layer normalization (Perez et al., 2018).

Modifications. To incorporate the acoustic features, we introduce two modifications to the base model:

- **Modulation of audio latents:** We add the processed acoustic features to the audio latents. Before the addition, we adjust the temporal length of the acoustic features by linear interpolation and

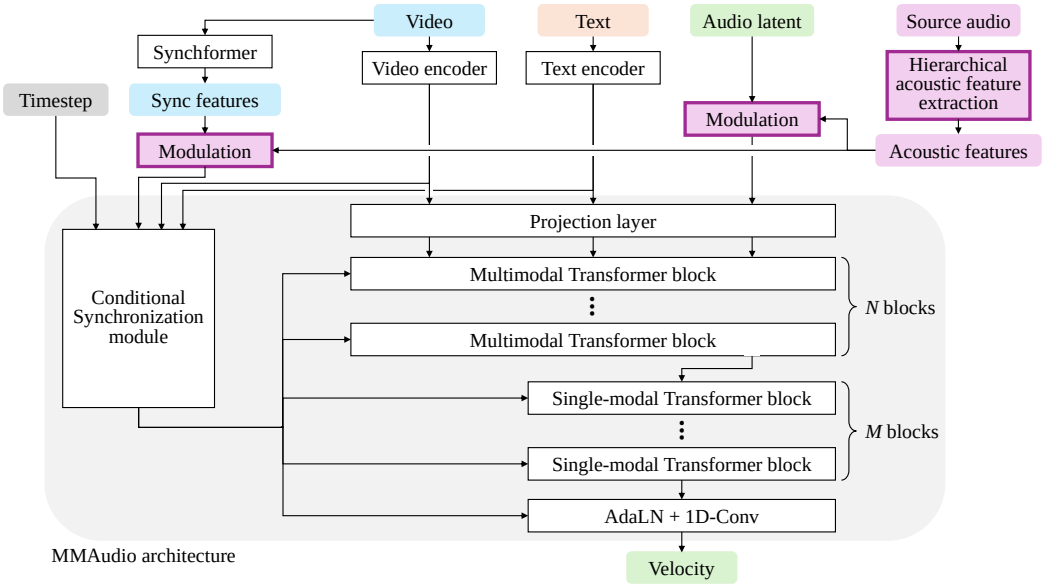


Figure 3: The architecture of our model. The major modifications from MMAudio are shown in purple.

process them with a trainable linear layer to match the dimensionality of the audio latents. This approach, which uses input addition or concatenation, is a common method for integrating conditional structural data into flow matching models and has been widely used in prior studies (Wang et al., 2024; Zhu et al., 2024).

- **Modulation of Synchformer features:** We also modulate the Synchformer features in a frame-wise manner using the acoustic features, which are processed with temporal length regulation and a feed-forward module (as in the DiT block). This temporally-local modulation is important because the Synchformer features strongly influence the generated audio in terms of temporal dynamics via adaptive layer normalization. To properly reflect the conditional acoustic information in the generation process, it is essential to modulate the Synchformer features before they are used in the base model.

These modules are initialized so that the modulation simply results in an identical function. During training, they are fixed in the first half of the training iterations and are set trainable there after. We adopt this two-stage training strategy to prevent the model from too much focusing on the acoustic features rather than the text prompt and the target video to generate the target audio.

3.4 Training with detail-temporal masking

We train our model through flow matching described in Section 2.1. Ideally, training would require sets of source audio, text prompt, target video, and target audio for each sample, but collecting such paired data with accompanying text and video at scale is challenging. Instead, we use standard text-audio-video data as a triplet of text prompt, target audio, and target video. To mimic the acoustic features of the source audio, we extract them from the target audio. As explained in Section 3.2, our acoustic features capture structural cues such as loudness envelopes across frequency bands rather than semantic content, and are thus largely invariant to semantic differences. This makes features extracted from the target audio a reasonable proxy for those from a semantically different source audio, enabling the model to learn the intended editing behavior without requiring true source-target audio pairs.

We jointly apply two kinds of masking to augment the acoustic features as shown in Fig. 4: detail masking and temporal masking. These two masking processes are introduced for separate purposes as described below.

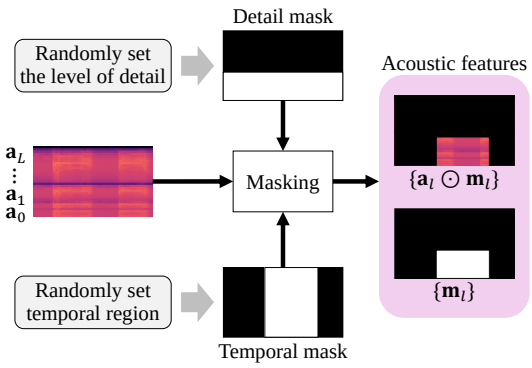


Figure 4: Detail-temporal masking of the acoustic features during training.

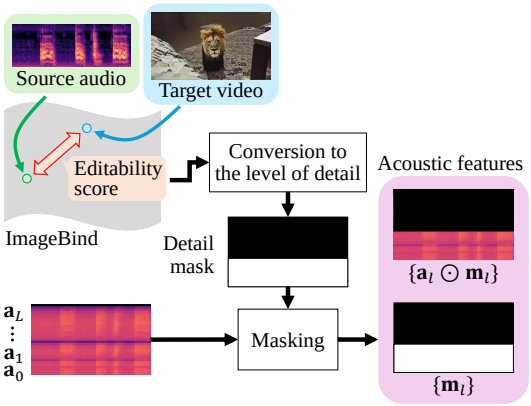


Figure 5: The adaptive conditioning based on the editability score.

- **Detail masking.** For each training sample, we randomly determine the level of detail of the acoustic features. Since this corresponds to randomly choosing a proper mask for the acoustic features as described in Section 3.2, we refer to this augmentation as detail masking. This masking enables the model to work with various levels of details, which means that we can control how much the original acoustic structure in the source audio should be preserved in the target audio at the inference time. We also include a choice to totally mask out the entire acoustic features in the setting, enabling classifier-free guidance for more accurate structure preservation. In the experiments, the probability of masking out the entire acoustic features was set to 0.1; otherwise, the mask level was selected with a probability proportional to 2^{-l}.

- **Temporal masking.** We also apply random masking along the temporal dimension to prevent the model from excessively relying on the acoustic features to generate the target audio. Since the acoustic features contain temporally rich information of the audio, especially when the level of detail is high, the model often focuses more on aligning with the acoustic features rather than text and video. This leads to degraded text fidelity and audio-visual alignment, which is not desirable for the audio editing. To alleviate this issue, we mask out the acoustic features at randomly selected time frames during training. It enforces the model to jointly reflect both the acoustic features and the other conditional features to generate the target audio. In the experiments, temporal masking was applied with a probability of 0.5. Once applied, the masking boundaries were sampled from a uniform distribution over the entire time frames.

3.5 Generation with adaptive conditioning

Once the model is trained, we can generate the target audio by solving the ODE as described in Section 2.1. Following the prior works on multi-conditional generation (Brooks et al., 2023; Chen et al., 2024; Kushwaha & Tian, 2025), we adopt classifier-free guidance with multiple guidance terms as shown below:

$$\begin{aligned}\tilde{u}_\theta(\mathbf{x}_t, t, \mathbf{C}, \mathbf{a}) = u_\theta(\mathbf{x}_t, t, \varnothing, \varnothing) &+ w_1(u_\theta(\mathbf{x}_t, t, \mathbf{C}, \varnothing) - u_\theta(\mathbf{x}_t, t, \varnothing, \varnothing)) \\ &+ w_2(u_\theta(\mathbf{x}_t, t, \mathbf{C}, \mathbf{a}) - u_\theta(\mathbf{x}_t, t, \mathbf{C}, \varnothing)),\end{aligned} \tag{7}$$

where $\mathbf{a}$ and $\mathbf{C}$ represent the acoustic features extracted from the source audio and a pair of text and video conditions, respectively, and w_1 and w_2 are coefficients to control the strength of two guidance terms. The first guidance term enhances the fidelity to the text and video conditioning, while the second term improves the structure preservation through referring to the acoustic features. In the experiments, we used $w_1 = 7.5$ and $w_2 = 0.5w_1$.

During generation, the model adaptively selects the appropriate level of detail for the acoustic features depending on the estimated difficulty of the audio editing task. This selection is performed deterministically based on an editability score, as described below. We term this process adaptive conditioning.

Motivation of adaptive conditioning. The optimal level of detail depends on how difficult the audio editing task is. For easy edits, where the source audio is already similar to the target (e.g., bird chirping → seagull chirping), preserving detailed acoustic features helps maintain structural fidelity without sacrificing alignment with the target prompt and video. In contrast, for difficult edits, where significant changes are needed (e.g., cat meowing → lion roaring), retaining too much detail from the source is less effective, much of the original acoustic structure will be altered or lost. Therefore, our method adaptively adjusts the level of acoustic detail based on the estimated editing difficulty for each sample, aiming to balance structural preservation and target fidelity.

Implementation of adaptive conditioning. To implement adaptive conditioning, we estimate an editability score that represents how easy the audio editing is expected to be. Specifically, we compute the audio-visual semantic similarity between the source audio and the target video. We use ImageBind (Girdhar et al., 2023) to embed the audio and video clips in a common feature space and then calculate the cosine similarity between these embeddings. Since ImageBind can only handle relatively short clips (e.g., two seconds), we use a sliding window approach with a window size of two seconds without overlapping, repeating this process and averaging the similarities across all time frames.

Given the editability score s, we determine the level of detail by quantizing the normalized score as:

$$l^* = \text{round}\left(l_{\max} \cdot \text{clamp}\left(\frac{s - s_{\min}}{s_{\max} - s_{\min}}\right)\right), \tag{8}$$

where clamp$(*)$ is a function that clamps the input to the range $[0, 1]$, $s_{\min}$ and $s_{\max}$ are the expected maximum and minimum scores, respectively, and $l_{\max}$ denote the maximum level of detail. For example, an easy edit (s close to $s_{\max}$) results in a higher level of detail being preserved, while a difficult edit (s close to $s_{\min}$) results in less detail being retained. The acoustic features $\mathbf{a}$ used for generation are then computed by applying the mask $\{\mathbf{m}_l\}$ with this l^* to the source audio, as described in Section 3.2.

We control the structure preservation in our model by changing the value of $l_{\max}$. Allowing a larger level of detail leads to more accurate preservation of the original audio structure. On the other hand, we fix both $s_{\max}$ and $s_{\min}$ so that they represent the two extreme cases of the editing difficulty. Specifically, we estimate $s_{\max}$ as the average editability score of genuine audio-video pairs, which correspond to the easiest case where no editing is needed, and $s_{\min}$ as the average score of random audio-video pairs, which correspond to the most difficult case where the two modalities are entirely unrelated.

4 Experiments

4.1 Dataset and evaluation metrics

Training datasets. Following MMAudio (Cheng et al., 2025), we jointly used several text-video-audio and text-audio datasets to train our model: VGGSound (Chen et al., 2020), AudioCaps (Kim et al., 2019), Clotho (Drossos et al., 2020), and WavCaps (Mei et al., 2024). All audio clips were resampled to 16 kHz, and 8-second segments were cropped without overlap for each training sample.

Evaluation dataset. We used the AvED-Bench dataset (Lin et al., 2025), which has particularly been constructed to evaluate audio-visual editing performance. The AvED-Bench dataset consists of 110 ten-second clips covering various types of natural audio-visual events. For each clip, a pair of the source and target prompts is given to specify the editing task. Audio and video were resampled to 16 kHz and 20 fps, respectively, and used as the source audio and video in our experiments.

Metrics. We assess the quality of edited audio from three different perspectives:

- **Audio-visual alignment (IB-AV and DeSync)** assesses the alignment between the target audio and the target video. We used ImageBind to evaluate coarse-grained alignment, as described in Section 3.5, and DeSync (Cheng et al., 2025; Iashin et al., 2024) to evaluate fine-grained alignment.

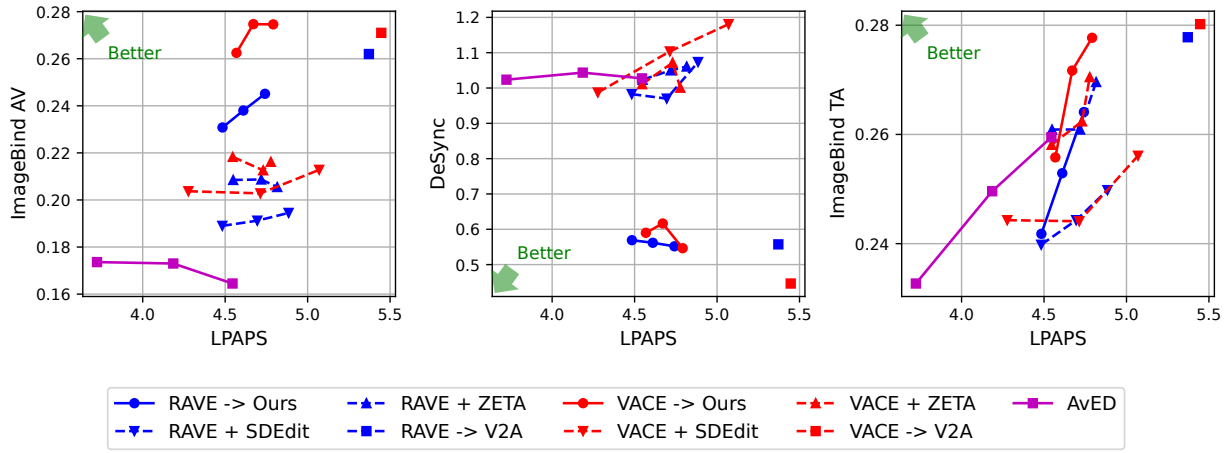


Figure 6: Experimental results on the AvED-Bench dataset. The horizontal axis represents structure preservation, where lower values are better. In the left and middle plots, the vertical axis shows coarse-grained audio-visual alignment (higher is better) and fine-grained audio-visual alignment (lower is better), respectively. The right plot shows text fidelity, where higher values are better. For each method except $* \rightarrow$V2A, we varied the hyperparameters to obtain results with different degrees of structure preservation.

- **Structure preservation (LPAPS)** assesses how much the acoustic structure of the source audio is preserved in the target audio. Following the prior studies (Manor & Michaeli, 2024; Lin et al., 2025), we used LPAPS (Iashin & Rahtu, 2021) to evaluate the structure preservation.
- **Text fidelity (IB-TA)** assesses the fidelity of the target audio to the given target prompt using ImageBind. Since ImageBind can only handle two-second audio clips, we cropped five non-overlapping audio sub-clips from the target audio and extracted ImageBind features from each sub-clip. Then, we computed the cosine similarity between those features and the feature extracted from the prompt. To aggregate the multiple similarities, we used max pooling, because the audio event described in the prompt is often temporally local (e.g., dog barking) and thus could be missed in several sub-clips.

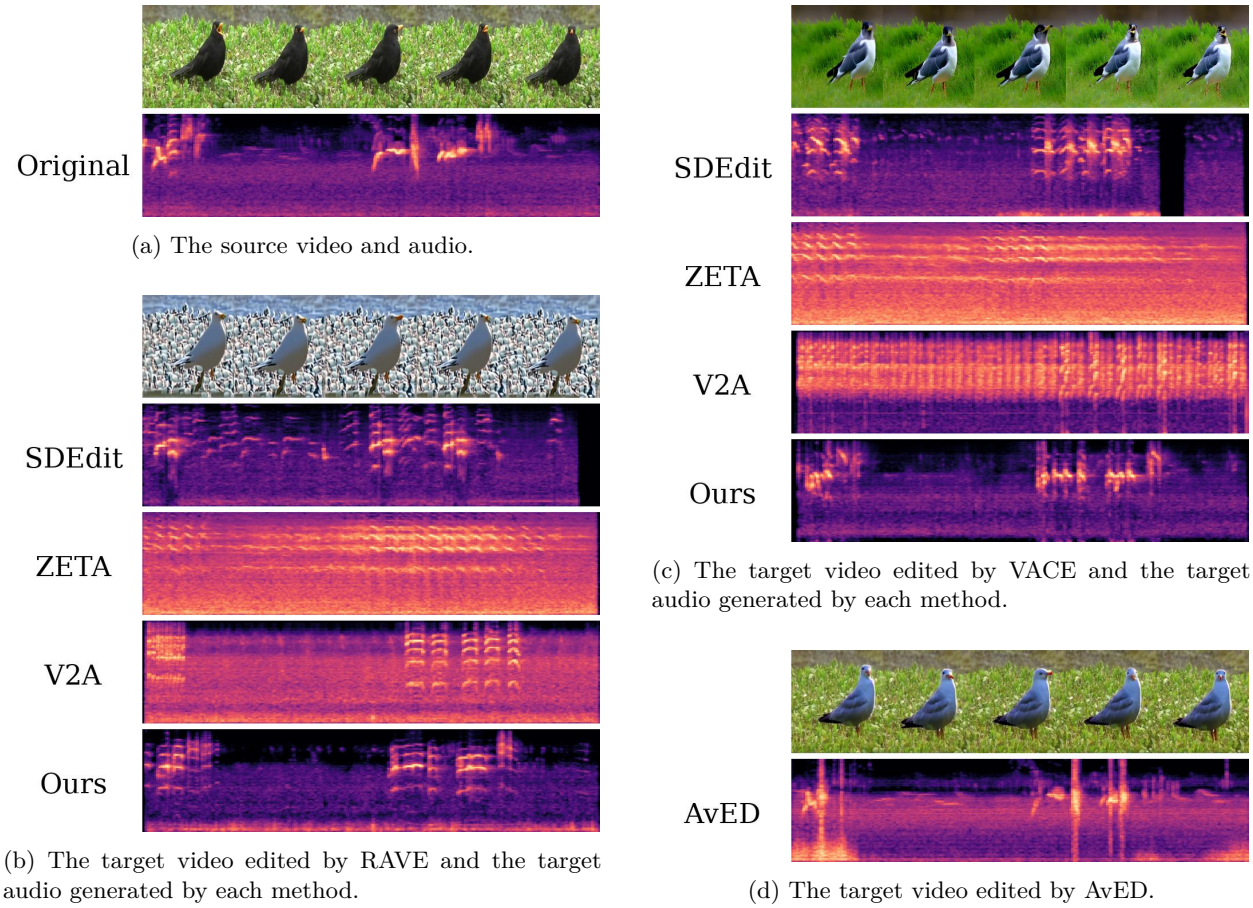
Regarding the edited video, we evaluated its quality following the VBench protocol (Huang et al., 2024).

4.2 Setup

We evaluated several methods representing three main approaches to audio-visual editing: independent, sequential, and joint.

Independent approach. In this approach, audio and video are edited separately. For audio editing, we employed SDEdit (Meng et al., 2022) and ZETA (Manor & Michaeli, 2024), both adapted to work with Stable Audio Open (Evans et al., 2025), a state-of-the-art audio generation model. For video editing, we used RAVE (Kara et al., 2024), which is based on Stable Diffusion 1.5 (Rombach et al., 2022), and VACE (Jiang et al., 2025), which utilizes the latest video generation model, Wan 2.1 (Wan et al., 2025). Among the various editing tasks supported by VACE, we used the "structure transfer" setting. To control the structure preservation in SDEdit and ZETA, we varied the intermediate timestep at which we inject the source audio information.

Sequential approach. In this approach, audio and video are edited in sequence. Our proposed method belongs to this category: we apply it to videos processed by RAVE and VACE (RAVE/VACE $\rightarrow$ Ours). We also evaluated our method with fully masked acoustic features, which completely remove source audio information from conditioning. This setting corresponds to applying a naive video-to-audio model to the edited videos (RAVE/VACE $\rightarrow$ V2A). In both settings, we used the Euler method as the ODE solver with 25 timesteps for sample generation.



(a) The source video and audio.

(b) The target video edited by RAVE and the target audio generated by each method.

(c) The target video edited by VACE and the target audio generated by each method.

(d) The target video edited by AvED.

Figure 7: Examples of the generated audio and video. The target caption is *"a seagull is chirping."*

Joint approach. In this approach, audio and video are edited jointly. We chose AvED (Lin et al., 2025) as the state-of-the-art baseline requiring no spatial control information, such as a target object mask. We controlled the structure preservation by varying the guidance scale used in the score distillation.

4.3 Main results

Figure 6 summarizes the performance of all editing methods. In all plots, the horizontal axis indicates structural preservation (LPAPS). The left plot reports coarse-grained audio-visual alignment (IB-AV), the middle plot reports fine-grained audio-visual alignment (DeSync), and the right plot reports target-audio text fidelity (IB-TA). Table 1 reports the results of video quality evaluation, and Fig. 7 presents qualitative examples. More examples are provided in the supplementary material.

Here are several observations we obtained through this experiment:

- **Benefit of sequential approach.** The sequential editing methods achieved significantly better performance in audio-visual alignment (higher IB-AV and lower DeSync) as they leverage the latest video-to-audio architecture when generating the edited audio. In contrast, the independent editing methods achieved resulted in poor audio-visual alignment due to the lack of a mechanism for aligning edited audio and video. AvED also failed to achieve high audio-visual alignment due to the difficulty of jointly editing both audio and video. In addition, as shown in Table 1, both RAVE and VACE outperform AvED in four out of five metrics for video quality, supporting the benefit of leveraging recent video editing models in our sequential pipeline.

Table 1: Video quality evaluation. SC: subject consistency, BC: background consistency, DD: dynamic degree, MS: motion smoothness, AQ: aesthetic quality.

	SC ↑	BC ↑	DD ↑	MS ↑	AQ ↑
AvED	0.84	0.92	0.72	0.96	0.42
RAVE	**0.91**	**0.94**	0.54	**0.98**	0.47
VACE	0.90	0.93	**0.59**	**0.98**	**0.54**

Table 2: Subjective evaluation results (mean score ± 95% confidence interval). For methods other than AvED, we used their VACE-based versions.

	Text fidelity	Audio-visual alignment	Structure perservation
ZETA	3.0 (± 0.5)	2.6 (± 0.7)	2.5 (± 0.7)
V2A	3.2 (± 0.6)	3.3 (± 0.6)	3.2 (± 0.5)
AvED	2.7 (± 0.4)	2.5 (± 0.6)	3.1 (± 0.4)
Ours	**3.7** (± 0.5)	**3.8** (± 0.3)	**3.9** (± 0.2)

- **Advantage of our method.** Simply applying video-to-audio to the edited video results in poor structure preservation, because naive video-to-audio generation cannot take the source audio into account. On the other hand, our method effectively maintains the structure of the source audio while achieving high audio-visual alignment and text fidelity. Note that our method shows slightly worse DeSync scores than the naive video-to-audio baseline, which reflects an inherent trade-off between structure preservation and fine-grained audio-visual synchronization. Conditioning on the source audio constrains the model to retain the original temporal structure, whereas the naive baseline can freely align the audio to the video at the cost of losing this structure. Our method achieves a favorable balance, gaining substantial structure preservation with only a marginal loss in fine-grained synchronization.
- **Dependency on the video editing method.** Compared with VACE → ours, the advantage of RAVE → ours is marginal especially in terms of text fidelity. This is due to the relatively lower performance of RAVE in video editing. RAVE sometimes fails to generate a reasonable video that is well aligned with the given text prompt. In such cases, our model struggles to handle target videos and text prompts that semantically contradict each other, leading to degraded quality of the edited audio.

4.4 Subjective test

We also conducted a subjective test to evaluate the perceptual quality of the edited videos. In this test, each rater first watches and listens to the source video and then views each edited version. The rater is asked to rate the quality of each edited video on a scale from 1 (poor) to 5 (excellent) in terms of the following three aspects: audio-text fidelity, audio-visual alignment, and audio structure preservation. In total, eleven raters participated in the test, and each rater evaluated six different videos.

Table 2 shows the results of the subjective test. Our method achieved the highest scores across all metrics. Consistent with the main evaluation results (Fig. 6), the proposed method and V2A achieved high audio-text fidelity and audio-visual alignment. In contrast, for audio structure preservation, unlike the main evaluation results, ZETA and AvED showed low scores. This suggests that, when the edited audio is experienced together with the video, perceived consistency with the video strongly influences subjective ratings.

4.5 Ablation study

We also performed an ablation study to clarify how much each newly introduced component contributes to enhancing the quality of edited audio. Specifically, we examine our method to remove the Synchformer feature modulation (SM) described in Section 3.3, the temporal masking (TM) in Section 3.4, and the adaptive conditioning (AC) in Section 3.5. In the case without AC, we used a fixed level of detail in all samples and varied it to control the overall degree of structure preservation. In addition, we also tested our model without providing the input video. This corresponds to the case where audio editing is performed independently of video editing.

Figure 8 shows the results of the ablation study.

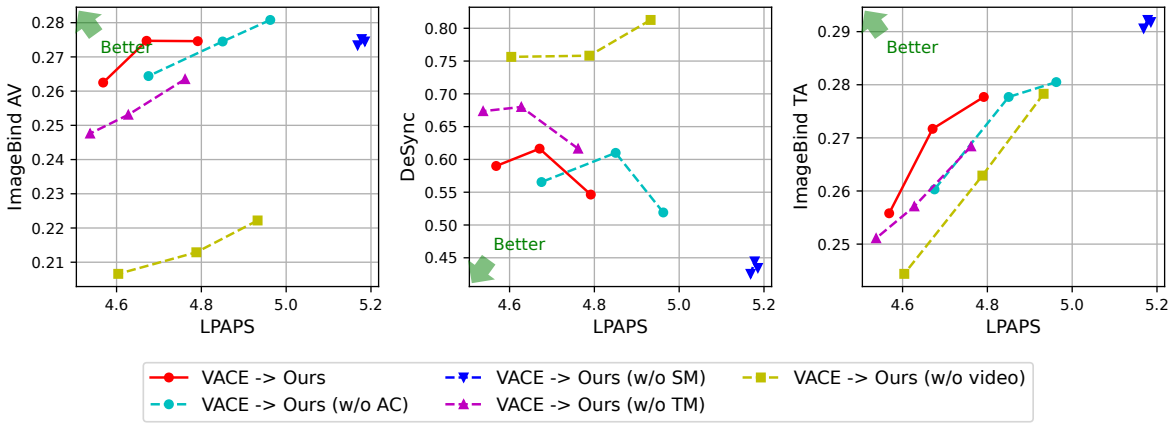


Figure 8: Ablation study. AC, SM, and TM denote adaptive conditioning, Synchformer feature modulation, and temporal masking, respectively.

- **Without SM**, the generated sounds result in similar LPAPS across various levels of detail, while achieving high performance in both audio-visual alignment and text fidelity. This indicates that modulating Synchformer features is important for reflecting conditional acoustic information in the generated target audio.
- **Without TM**, the quality of the generated sounds degrades in both audio-visual alignment and text fidelity. This highlights that augmenting the acoustic features using temporal masking contributes to effective training of our model, even though such masking is never used during inference.
- **Without AC**, LPAPS is generally higher than with AC, while performance in audio-visual alignment and text fidelity remains similar. This shows that adaptively controlling the level of detail based on editability contributes to accurate preservation of the acoustic structure while maintaining the quality of the generated audio.
- **Without input video**, the audio-visual alignment drops significantly. This indicates that video conditioning provides a clear gain in audio-visual alignment beyond source-audio conditioning alone.

Limitations. Since our pipeline is sequential, the generation quality can be degraded by errors in video editing. As shown in Fig. 6, using more advanced video editing methods leads to better performance; thus, careful selection of the video editing method is important for our pipeline.

5 Conclusion

We introduced a novel audio-visual editing pipeline that first applies an existing video editing method, followed by audio editing that aligns with the video edits. To enable this, we developed a new video-to-audio model that uses the source audio as a conditional input. Our model adaptively extracts acoustic features from the source audio based on an editability score and incorporates them into the generation process to maximize retention of the original audio structure. Experimental results demonstrate that our model achieves high audio-visual alignment in the edited audio-video pairs while effectively preserving the acoustic characteristics of the source audio. As future work, we plan to extend the pipeline to support a broader range of editing tasks, such as object deletion and addition, which may require further advances in acoustic structure preservation.

References

Matan Ben-Yosef, Tavi Halperin, Naomi Ken Korem, Mohammad Salama, Harel Cain, Asaf Joseph, Anthony Chen, Urska Jelercic, and Ofir Bibi. Avcontrol: Efficient framework for training audio-visual controls. *arXiv preprint arXiv:2603.24793*, 2026.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18392–18402, 2023.

Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 721–725. IEEE, 2020.

Shoufa Chen, Mengmeng Xu, Jiawei Ren, Yuren Cong, Sen He, Yanping Xie, Animesh Sinha, Ping Luo, Tao Xiang, and Juan-Manuel Perez-Rua. Gentron: Diffusion transformers for image and video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6441–6451, 2024.

Ho Kei Cheng, Masato Ishii, Akio Hayakawa, Takashi Shibuya, Alexander Schwing, and Yuki Mitsufuji. Mmaudio: Taming multimodal joint training for high-quality video-to-audio synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 28901–28911, 2025.

Nathaniel Cohen, Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. Slicedit: Zero-shot video editing with text-to-image diffusion models using spatio-temporal slices. In *International Conference on Machine Learning*, pp. 9109–9137. PMLR, 2024.

Yuren Cong, Mengmeng Xu, Shoufa Chen, Jiawei Ren, Yanping Xie, Juan-Manuel Perez-Rua, Bodo Rosenhahn, Tao Xiang, Sen He, et al. Flatten: optical flow-guided attention for consistent text-to-video editing. In *The Twelfth International Conference on Learning Representations*, 2024.

Google DeepMind. Veo 3. `https://deepmind.google/models/veo/`, 2025.

Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. Clotho: An audio captioning dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 736–740. IEEE, 2020.

Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. 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.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Zach Evans, Julian D Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Stable audio open. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Xiang Fan, Anand Bhattad, and Ranjay Krishna. Videoshop: Localized semantic video editing with noise-extrapolated diffusion inversion. In *European Conference on Computer Vision*, pp. 232–250. Springer, 2024.

Ruiqi Gao, Emiel Hoogeboom, Jonathan Heek, Valentin De Bortoli, Kevin Patrick Murphy, and Tim Salimans. Diffusion models and gaussian flow matching: Two sides of the same coin. In *The Fourth Blogpost Track at ICLR 2025*, 2025.

Hugo Flores García, Oriol Nieto, Justin Salamon, Bryan Pardo, and Prem Seetharaman. Sketch2sound: Controllable audio generation via time-varying signals and sonic imitations. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. In *The Twelfth International Conference on Learning Representations*, 2024.

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

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15180–15190, 2023.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21807–21818, 2024.

Vladimir Iashin and Esa Rahtu. Taming visually guided sound generation. In *British Machine Vision Conference*. BMVA Press, 2021.

Vladimir Iashin, Weidi Xie, Esa Rahtu, and Andrew Zisserman. Synchformer: Efficient synchronization from sparse cues. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5325–5329. IEEE, 2024.

Yuhang Jia, Yang Chen, Jinghua Zhao, Shiwan Zhao, Wenjia Zeng, Yong Chen, and Yong Qin. Audioeditor: A training-free diffusion-based audio editing framework. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. *arXiv preprint arXiv:2503.07598*, 2025.

Ozgur Kara, Bariscan Kurtkaya, Hidir Yesiltepe, James M Rehg, and Pinar Yanardag. Rave: Randomized noise shuffling for fast and consistent video editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6507–6516, 2024.

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

Saksham Singh Kushwaha and Yapeng Tian. Vintage: Joint video and text conditioning for holistic audio generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 13529–13539, 2025.

Zitong Lan, Yiduo Hao, and Mingmin Zhao. Guiding audio editing with audio language model. *arXiv preprint arXiv:2509.21625*, 2025.

Xirui Li, Chao Ma, Xiaokang Yang, and Ming-Hsuan Yang. Vidtome: Video token merging for zero-shot video editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7486–7495, 2024.

Susan Liang, Chao Huang, Yapeng Tian, Anurag Kumar, and Chenliang Xu. Language-guided joint audio-visual editing via one-shot adaptation. In *Proceedings of the Asian Conference on Computer Vision*, pp. 1011–1027, 2024.

Yan-Bo Lin, Kevin Lin, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Chung-Ching Lin, Xiaofei Wang, Gedas Bertasius, and Lijuan Wang. Zero-shot audio-visual editing via cross-modal delta denoising. *arXiv preprint arXiv:2503.20782*, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *11th International Conference on Learning Representations, ICLR 2023*, 2023.

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. In *International Conference on Machine Learning*, pp. 21450–21474. PMLR, 2023.

Haohe Liu, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Qiao Tian, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D Plumbley. Audioldm 2: Learning holistic audio generation with self-supervised pretraining. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:2871–2883, 2024a.

Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. Video-p2p: Video editing with cross-attention control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8599–8608, 2024b.

Hila Manor and Tomer Michaeli. Zero-shot unsupervised and text-based audio editing using ddpm inversion. In *International Conference on Machine Learning*, pp. 34603–34629. PMLR, 2024.

Xinhao Mei, Chutong Meng, Haohe Liu, Qiuqiang Kong, Tom Ko, Chengqi Zhao, Mark D Plumbley, Yuexian Zou, and Wenwu Wang. Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32: 3339–3354, 2024.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

Max Morrison, Cameron Churchwell, Nathan Pruyne, and Bryan Pardo. Fine-grained and interpretable neural speech editing. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pp. 187–191, 2024.

OpenAI. Sora 2. `https://platform.openai.com/docs/models/sora-2`, 2025.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 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.

Fengyuan Shi, Jiaxi Gu, Hang Xu, Songcen Xu, Wei Zhang, and Limin Wang. Bivdiff: A training-free framework for general-purpose video synthesis via bridging image and video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7393–7402, 2024.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Yongqi Wang, Wenxiang Guo, Rongjie Huang, Jiawei Huang, Zehan Wang, Fuming You, Ruiqi Li, and Zhou Zhao. Frieren: Efficient video-to-audio generation network with rectified flow matching. *Advances in Neural Information Processing Systems*, 37:128118–128138, 2024.

Yuancheng Wang, Zeqian Ju, Xu Tan, Lei He, Zhizheng Wu, Jiang Bian, et al. Audit: Audio editing by following instructions with latent diffusion models. *Advances in Neural Information Processing Systems*, 36:71340–71357, 2023.

Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 7623–7633, 2023.

Manjie Xu, Chenxing Li, Duzhen Zhang, Dan Su, Wei Liang, and Dong Yu. Prompt-guided precise audio editing with diffusion models. In *International Conference on Machine Learning*, pp. 55126–55143. PMLR, 2024.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yanming Zhang, Jun-Kun Chen, Jipeng Lyu, and Yu-Xiong Wang. V2edit: Versatile video diffusion editor for videos and 3d scenes. *arXiv preprint arXiv:2503.10634*, 2025.

Haojie Zheng, Shuchen Weng, Jingqi Liu, Siqi Yang, Boxin Shi, and Xinlong Wang. Audio-sync video instance editing with granularity-aware mask refiner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23150–23160, 2026a.

Haojie Zheng, Yixin Yang, Siqi Yang, Shuchen Weng, and Boxin Shi. Instructav2av: Instruction-guided audio-video joint editing. *arXiv preprint arXiv:2605.18467*, 2026b.

Yixuan Zhu, Wenliang Zhao, Ao Li, Yansong Tang, Jie Zhou, and Jiwen Lu. Flowie: Efficient image enhancement via rectified flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13–22, 2024.