# SonicVisionLM: Playing Sound with Vision Language Models

Zhifeng Xie[1,2], Shengye Yu[1], Qile He[1], Mengtian Li[1,2†]
[1]Shanghai University
[2]Shanghai Engineering Research Center of Motion Picture Special Effects
{zhifeng_xie,yussisy,shu_hql,mtli}@shu.edu.cn

2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 979-8-3503-5300-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/CVPR52733.2024.02537

## Abstract

*There has been a growing interest in the task of generating sound for silent videos, primarily because of its practicality in streamlining video post-production. However, existing methods for video-sound generation attempt to directly create sound from visual representations, which can be challenging due to the difficulty of aligning visual representations with audio representations. In this paper, we present* ***SonicVisionLM****, a novel framework aimed at generating a wide range of sound effects by leveraging vision-language models(VLMs). Instead of generating audio directly from video, we use the capabilities of powerful VLMs. When provided with a silent video, our approach first identifies events within the video using a VLM to suggest possible sounds that match the video content. This shift in approach transforms the challenging task of aligning image and audio into more well-studied sub-problems of aligning image-to-text and text-to-audio through the popular diffusion models. To improve the quality of audio recommendations with LLMs, we have collected an extensive dataset that maps text descriptions to specific sound effects and developed a time-controlled audio adapter. Our approach surpasses current state-of-the-art methods for converting video to audio, enhancing synchronization with the visuals, and improving alignment between audio and video components. Project page:* https://yusiissy.github.io/SonicVisionLM.github.io/

## 1. Introduction

The sound effects artists work with various types of sounds, including those visible on-screen (like footsteps or a car passing) and those audible but not visible (like background noises and heartbeats can enhance the video's authenticity and narrative). On-screen sounds match what is happening in the video, while off-screen sounds establish the ambience and provide additional information. Creating soundtracks for videos is a vital aspect of video production, but it can be labour-intensive for artists. Therefore, the video-sound generation task has gained notable attention.

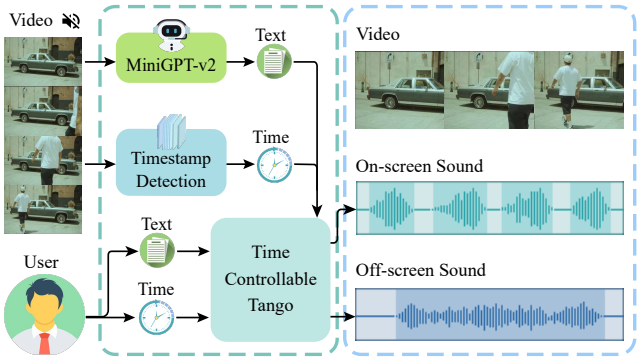


Figure 1. A model implements the automatic detection of on-screen sound generation and accepts the user's editing of text and time in the off-screen section. On-screen sound refers to audio that originates from visible actions within the video frame. Off-screen sound is not directly observable on the screen.

Although the recent approaches have made great efforts, the video-sound generation task is still challenging. For on-screen sounds, achieving semantic relevance and maintaining temporal synchronization continues to be a complex issue. It is hard to edit off-screen sounds. Current methods [14, 25, 32] primarily focus on the visual content to generate the corresponding sound, a subset of these methods [7, 9] considers editability. Nonetheless, the alignment between video and audio features is tricky, leading to deficiencies including 1) incorrect sound meanings and mismatched timing, 2) monotonous sound effects, and a lack of complex scenarios. Both lead to unsatisfactory results in the video-sound generation task.

We propose a novel framework named SonicVisionLM to solve the above deficiency, as shown in Fig. 1. SonicVisionLM is proposed by introducing three key components: video-to-text, text-based interaction, and text-to-audio generation. First, the video-to-text component focuses on generating sound effects for on-screen events. This step uses a VLM to identify appropriate sound descriptions from the input silent video. Following this, a timestamp detection network is trained to extract specific temporal information

[†]Corresponding author.

2575-7075/24/$31.00 ©2024 IEEE
DOI 10.1109/CVPR52733.2024.02537

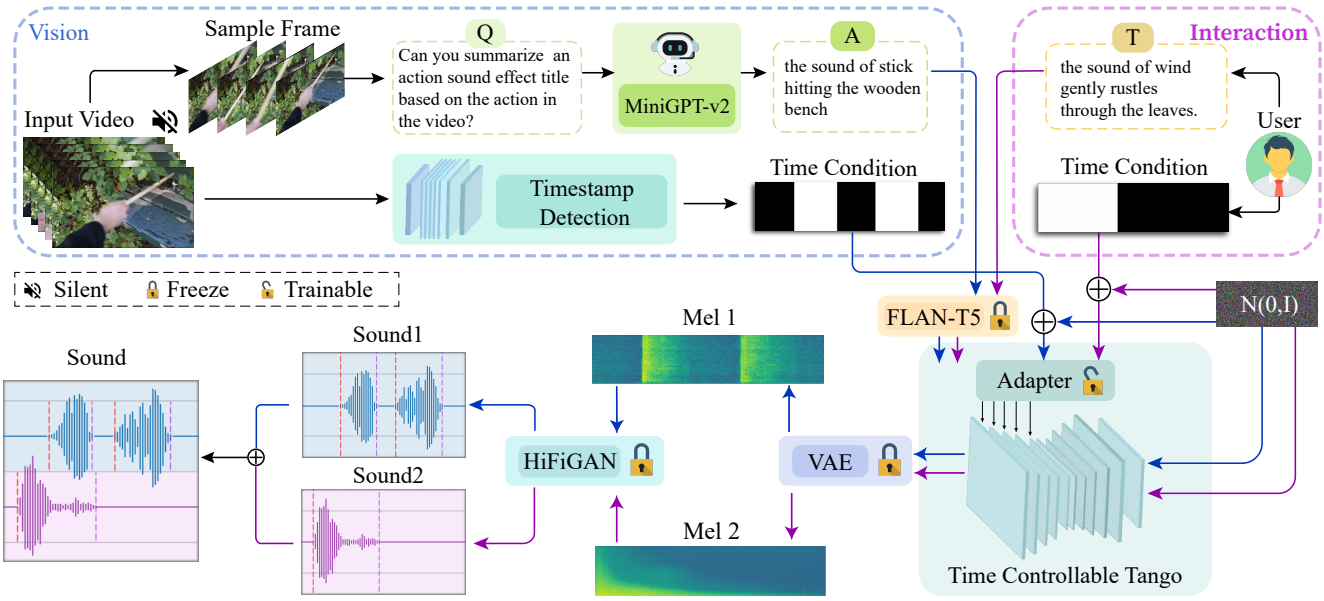


Figure 2. **SonicVisionLM's framework.** SonicVisionLM presents a composite framework designed to automatically recognize on-screen sounds coupled with a user-interactive module for editing off-screen sounds. The blue dashed box and arrows in the figure represent the visual automation workflow: First, a silent video goes into the system to determine the occurring events (text) and their timing (time). Then, this information conditions the generation of sounds matching the screen. The purple dotted box and arrows show how users can modify or add off-screen sounds.

from the video. A key innovation within this framework is the design of a time-conditioned embedding, which is utilized to guide an audio adapter. After that, the text-based interaction component allows users to change the text and timestamps from a previous video-to-text component or to input new corresponding text-timestamp pairs for personalized sound design. Finally, the text-to-audio generation component accepts the text and timestamp conditions and inputs them into the LDM and adapter to generate diverse, time-synchronized, and controllable sounds. Simultaneously, We collect a text-to-single-sound dataset, named CondPromptBank, for sound effects caption and timing cues, comprising over ten thousand data points, covering 23 categories. The main contributions of this work are:

- We propose a novel framework called SonicVisionLM and collect a dataset CondPromptBank specifically for training a time-controllable adapter. It ensures the generated sound aligns perfectly with our text input and maintains precise timing control.
- We introduce three pioneering components: video-to-text, text-based interaction, and text-to-audio generation. This unique combination facilitates the automatic recognition of on-screen sounds while enabling user customization of off-screen sounds.
- The proposed framework achieves state-of-the-art results on conditional and unconditional video-sound generation tasks. The conditional task can be noticeably enhanced in all metrics. (IoU: 22.4→39.7)

## 2. Related Work

**Audio Generation** can be broadly classified into two categories. The *Text-to-Audio Generation* field includes Text-to-Speech (TTS) and Text-to-Music (TTM). Leading TTS models, such as FastSpeech2 [29] and NaturalSpeech [33], now produce speech virtually indistinguishable from human speech. In TTM, MusicLM [1], Noise2Music [12], MusicGen [6] and MeLoDy [21] are aimed to generate music segments from text, bringing innovation to music composition and synthesis. Models like AudioGen [19], AudioLDM [24], Tango [10], and Make-an-Audio [13] focuses on universal audio generation modeling. AudioGen [19] treats audio generation as a conditional language modelling task, while the other three models employ latent diffusion methods to accomplish sound generation. Current methods use datasets including sound effects, voices, and music, but practical applications use these elements separately. As the textual descriptions of time are subjective, videos are more intuitive and precise, so V2A requires more accurate semantic features and time control than T2A. Therefore, we have created a text-to-single-sound dataset called CondPromptBank with detailed semantic segmentation and temporal annotations that help models produce high-quality sound effects for videos. In *Video-to-Audio Generation* task, SpecVQGAN [14] utilizes a Transformer-based autoregressive model, drawing on ResNet50 or RGB+Flow features to generate sound. Im2Wav [32] uses a dual-transformer model conditioned on CLIP features for sound generation. CondFoleyGen [9] and VARIETYSOUND [7]

introduce tasks for controllable timbre generation. Diff-foley [25] uses contrastive audio-visual pretraining to align audio and visual features. ClipSonic [8] learns the text-audio correspondence by leveraging the audio-visual correspondences in videos and the multi-modal representation learned by pre-trained VLMs. However, the sounds generated by these methods often suffer from poor audio-visual synchronization, noticeable noises, and lack of editability. Unlike the works above, our model ensures audio-visual synchronization, enriches diversity, and supports personalized user edits. Our model provides a more comprehensive sound solution for video production.

**Diffusion Model** has been utilized for generating both mel-spectrogram generation [4, 28], and waveform generation [3, 20, 22]. However, their iterative generation process can be slow for high-dimensional data. Models such as AudioLDM [24], Make-An-Audio [13], and Tango [10] have successfully trained diffusion models within a continuous latent space. Nevertheless, achieving satisfactory results in controlling LDM for audio generation tasks remains challenging. This paper aims to introduce time control to ensure audio-visual synchronization.

**Vision Language Models** like ChatGPT-4, which demonstrated advanced multi-modal abilities and inspired vision-language LLMs. Vision-LLM [36] and LLaVA [34] focus on aligning image inputs with large language models Vicuna[5] exhibit similar multi-modal capabilities. Recent developments in this field include MiniGPT-v2 [2]. Kosmos-2 [27] demonstrates multi-modal LLMs' ability to perform visual grounding. In this paper, we first introduce the VLMs to the audio generation task.

# 3. Method

## 3.1. Overview

In this section, we introduce the framework of SonicVisionLM, as shown in Fig. 2. Before delving into the specific design details, we first briefly overview the preliminary knowledge (Sec. 3.2). Then, we introduce the Visual-to-Audio Event Detection Module (Sec. 3.3), which obtains textual descriptions of on-screen sounds through VLMs. Subsequently, we present the Sound Event Timestamp Detection Module (Sec. 3.4), designed to accurately detect the timing information through network architecture. Finally, we introduce the proposed time-controllable adapter as an extension of the audio diffusion model (Sec. 3.5), enabling the generation of multiple sounds that are semantically coherent and temporally aligned.

## 3.2. Preliminaries

**Audio Diffusion Model.** The text-prompt encoder encodes the input description $\tau \in R^{L \times d_t}$ of the sound, where $L$ is the token count and $d_t$ and is the token-embedding size. The latent diffusion model (LDM) is used to construct the audio prior $z_0$ with the guidance of text encoding $\tau$. This essentially reduces to approximating the true prior $q(z_0 \mid \tau)$ with parameterized $p(z_0 \mid \tau)$. LDM can achieve the above through forward and reverse diffusion processes. The forward diffusion is a Markov chain of Gaussian distributions with scheduled noise parameters $0 < \beta_1 < \beta_2 < \cdots < \beta_N < 1$ to sample noisier versions of $z_0$, where $N$ is the number of forward diffusion steps. For each step $n$, we define $\alpha_n = 1 - \beta_n$, and calculate the cumulative product $\bar{\alpha}_n = \prod_{i=1}^{n} \alpha_n$.The diffusion equations are described as follows:

$$q(z_n \mid z_{n-1}) = \mathcal{N}\left(\sqrt{1-\beta_n} z_{n-1}, \beta_n \mathbf{I}\right), \quad (1)$$

$$q(z_n \mid z_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_n} z_0, (1-\bar{\alpha}_n)\mathbf{I}\right), \quad (2)$$

where the noise term $\epsilon$ and the final step of the forward process yields $z_N$ follow a Gaussian distribution, specifically $\epsilon, z_N \sim \mathcal{N}(0, I)$. The reverse process denoises and reconstructs $z_0$ through text-guided noise estimation $(\hat{\epsilon}_\theta)$ using following loss function:

$$\mathcal{L}_{DM} = \sum_{n=1}^{N} \gamma_n \mathbb{E}_{\epsilon_n \sim \mathcal{N}(\mathbf{0},\mathbf{I}), z_0} \left\| \epsilon_n - \hat{\epsilon}_\theta^{(n)}(z_n, \tau) \right\|_2^2, \quad (3)$$

After training LDM, we generate audio latent by sampling through the reverse process with $z_N \sim \mathcal{N}(0, I)$, conditioned on the given textual representation $\tau$. Its reverse dynamics are shown below:

$$p_\theta(z_{n-1} \mid z_n, \tau) = \mathcal{N}\left(\mu_\theta^{(n)}(z_n, \tau), \tilde{\beta}^{(n)}\right), \quad (4)$$

$$\mu_\theta^{(n)}(z_n, \tau) = \frac{1}{\sqrt{\alpha_n}}\left[z_n - \frac{1-\alpha_n}{\sqrt{1-\bar{\alpha}_n}} \hat{\epsilon}_\theta^{(n)}(z_n, \tau)\right], \quad (5)$$

$$\tilde{\beta}^{(n)} = \frac{1-\bar{\alpha}_{n-1}}{1-\bar{\alpha}_n}\beta_n. \quad (6)$$

The noise estimation $(\hat{\epsilon}_\theta)$ is parameterized with U-Net framework [30] with a cross-attention component to include the text guidance $\tau$.

After that, the decoder of audio VAE [18] constructs a mel-spectrogram feature from the latent audio representation $\hat{z}_0$. This mel-spectrogram feature is conveyed to a vocoder to generate the final audio.

## 3.3. Visual-to-Audio Event Understanding Module

The diversity of sound is influenced by various factors such as its source, actions, the environment, and more. At the same time, these factors are often included in the description of visual images. Inspired by the widespread use of VLMs [23, 37, 39], we chose MiniGPT-v2 [2] to process visual information and generate descriptions of sounds. Recognizing that MiniGPT-v2 was initially developed for single-image understanding and its limitations in conveying dynamic information, we have adapted the

LLaMA-2 [34] conversation template design to suit a multi-modal instructional framework. To not miss sound events, we encoded four random video frames with time-sensitive prompts (temporal cues):

```
First      ,< Img >< ImageFeature >< Img >.
Then       ,< Img >< ImageFeature >< Img >.
After that,< Img >< ImageFeature >< Img >.
Finally    ,< Img >< ImageFeature >< Img >.
[Task Identifier ] Instruction
```

### 3.4. Sound Event Timestamp Detection Module

In practice, sound artists need to manually determine the point in time when a sound event starts and ends and then adjust the appropriate sound effect to the correct position. This judgment is usually based on current visual information. To simplify the process, we use the sound event timestamp detection module to detect the timestamp of the sound event in the video inspired by the CondFoleyGen [9]. Instead of using a hand-crafted approach to transfer sounds from the conditional audio, We use a ResNet(2+1)-D18 [35] visual network to capture timestamps as time-conditional inputs to the LDM, trained on paired video and timestamp.

The workflow begins with feeding a sequence of silent video frames $V_f$ into the detection network. The network then outputs a binary vector $V_{ct}$ representing predictions for each frame, derived from a fully connected layer post-pooling. The ground truth $V_{ct}$ is obtained: Function $P$ detects audio $a$ in each frame, applying a threshold $x$ to reduce noise effects. Sounds within 0.02s across consecutive frames are considered a single event. This method identifies the start $x_{start}$ and end points $x_{end}$ of sound segments, as delineated in Eq. 7. Based on timestamps, we construct $T_{ct}$, as depicted in Eq. 8, where 1 indicates the presence of sound, and 0 indicates the absence. The process equations are as follows:

$$x_{start}, x_{end} = P\left(a_c\right), \tag{7}$$

$$T_{ct} = \begin{cases} 1 & ,if \;\; t \;\; in \;\; [x_{start}, x_{end}], \\ 0 & ,else. \end{cases} \tag{8}$$

Finally, $T_{ct}$ is adjusted to correspond with the video frames' duration, represented as $V_{ct}$. We employ binary cross-entropy loss to penalize inaccuracies in time prediction. Given that an input video may contain multiple sound events, each sound event's weights are based on its duration relative to the total sound duration in the video. The binary vector $T_{ct}$ serves as input for the Time-controllable Latent Diffusion Model. The whole process and network structure are shown in Fig. 3.

### 3.5. Time-controllable Latent Diffusion Model

In our experimentation, we observed that the results generated were semantically inaccurate and temporally unsynchronized. This issue often arose when utilizing audio-visual datasets to train end-to-end models, limiting the task's practical application. We attribute these shortcomings to the complexity of the sound sources and the poor audio quality of the audio-visual dataset used for training. To address this, we use text to bridge audio and video and then introduce time control in the T2A generation model. In the visual domain, many works are based on the ControlNet [40] architecture, which can finer control the generation of images or videos by manipulating the input image conditions of neural network blocks. However, unlike the visual domain's inherent intuitiveness, audio features exhibit temporal continuity and are inherently more complex and abstract. Thus, selecting appropriate audio features to guide the generation process poses a greater challenge.

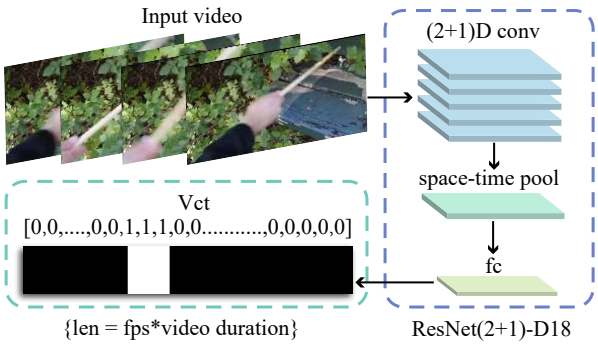


Figure 3. **Sound Event Timestamp Detection Module.** The network analyzes the video's features to output a binary vector corresponding to the video's frame count. Within this vector, sections marked in white (value of 1) mean sound presence, and those in black (value of 0) indicate sound absence.

In this paper, we propose a embedding called **Audio Time-condition Embedding**, denote as $A_{ct}$, which is constructed through the following procedure: During the training phase, the adapter extracted Mel-spectrogram features from audio waveform $a$ and then normalized it, denoted as $a_{mel} = mel(a)$. For $a_c = max(mel(a), T_{ct})$, we then initialize $a_c$ to match $a_{mel}$'s dimensions but with zero values, then fill $a_c$'s Mel channels with the maximum values based on the corresponding frames where $T_{ct}$ equals 1. Finally, $A_{ct}$ is derived by encoding the embedding of $a_c$ via the encoder $E_a$, and described as follows:

$$A_{ct} = E_a\left(\max\left(\mathrm{mel}\left(a_c\right), T_{ct}\right)\right). \tag{9}$$

Inspired by ControlNet [40], we have developed a network architecture named **Time-controllable Adapter**. Then, we integrate the audio time-condition embedding $A_{ct}$ with the text embedding $\tau$ and target audio embedding into the neural network block. This integration facilitates joint training for the time-controllable adapter. $A_{ct}$ is fed solely into the adapter, while text embedding $\tau$ inputs into both Tango and the adapter.

Tango's denoising model, akin to UNet, includes encoder $F$, middle block $M$, and decoder $G$. $F$ and $G$ have 12 corresponding blocks. Tango's outputs of the encoder's

$i-th$ block and decoder's $j-th$ block are $f_i$ and $g_j$, with $m$ for the single middle block. The adapter mimics Tango's encoder and middle layers as $F'$ and $M'$. Adapter's outputs, like $f_i'$ and $m'$, are marked with ($'$). Following this, the output of the time-controllable adapter is concatenated with the output of the corresponding decoder block during the decoding process. For example, $f_1$ from the 1st encoder layer adds to $g_{12}$ of the decoder, making $i+j=13$. To achieve it, we ensure that all Tango's elements are kept frozen while modifying the input of the $i-th$ block of the decoder as:

$$\begin{cases} Concat\left(m+m^{'}, f_j + zero\left(f_j^{'}\right)\right) & i=1, j=12. \\ Concat\left(g_{i-1}, f_j + zero\left(f_j^{'}\right)\right) & 2 \leqslant i \leqslant 12, i+j=13. \end{cases} \tag{10}$$

$zero(\cdot)$ is one zero convolutional layer, facilitating trainable and fixed neural network block connections. Its weights evolve from zero to optimized values during training. This approach not only retains the Tango's capability for generating audio, trained on billions of audio-text pairs, but also enables the model to comprehend the guidance provided by the time control embedding, resulting in temporally controllable outcomes.

Table 1. Distribution of CondPromptBank Categories

| Category | % | Category | % | Category | % |
|---|---|---|---|---|---|
| Household Daily | 14.11 | Transportation Vehicles | 10.42 | Impacts Crashes | 10.10 |
| Foley | 8.24 | Human Elements | 7.77 | Industrial | 6.58 |
| Weapons War | 5.83 | Cartoon Comical | 4.90 | Sports | 4.43 |
| Animals Insects | 4.04 | Instruments | 3.68 | Water Liquid | 3.27 |
| Technology | 2.70 | Horror | 2.41 | Emergency | 2.20 |
| Public Places | 1.87 | Sound Design Effects | 1.69 | Doors Windows | 1.56 |
| Fire Explosions | 1.49 | Nature Weather | 1.02 | Leisure | 0.84 |
| Multimedia | 0.47 | Bells | 0.37 | | |

# 4. Experiments

## 4.1. Experiment Settings

**Dataset.** Since Tango's training datasets include varied audio types like speech, sound effects, and music, it tends to produce mixed audio outputs. Our task requires distinct handling of these audio types, ensuring each sound event is separate. To meet our specific needs, we developed *CondPromptBank*, a high-quality dataset of single sound effects crafted for training time-controlled adapters. This dataset consists of 10,276 individual data entries, each with a sound effect, title, and start/end timestamps. Each sound is typically 10 seconds or shorter, sourced from freely available sound effect libraries and websites. During the collection, we focused on 23 common categories of sound effects and manually filtered out low-quality data with noise and mixed sources. The category distribution shows Tab. 1. To enhance the textual descriptions with precise details, we further annotated the sound effect text labels with fine-grained information based on sound characteristics. Now, each label not only identifies the audio source but also describes the associated actions in detail. We believe the division into single audio sources is crucial, as videos provide a more concrete expression of content than text, thus requiring more precise sound representation.

**Implementation details.** To train our complete model, we first train the adapter on CondPromtBank, then train the timestamps detection net on Greatest Hits [26] and Countix-AV dataset [41]. We trained the adapter for approximately 200 epochs with a batch size 32 and a learning rate of $3.0 \times 10^{-5}$ using Adam [17]. Our model has two versions: small and full. The difference between them is the version of pre-trained parameters used for Tango. We trained the timestamps detection net for 70 epochs with a batch size of 24 and a learning rate of $1.0 \times 10^{-5}$. The training of the adapter takes approximately five days, and the training of the timestamps detection network requires approximately one day using one NVIDIA A6000 GPU.

## 4.2. Conditional Generation Task Results

We tested our model using the Greatest Hits [26] dataset, which has 977 videos of drumsticks interacting with different objects, lasting 11 hours. This dataset is divided into two types of actions and 17 types of materials. This detailed categorization helps check if our model can change sound types based on these details but still match the target action. We used the same test settings as CondFoleyGen [9] and compared our results with theirs.

**Evaluation Metrics.** For the conditional generation task, we use the following five objective metrics to evaluate the performance of the model: *CLAP-top*, *Onset Acc* [9], *Onset AP* [9], *Time Acc*, and *IoU*.

**Quantitative Results.** SonicVisionLM-full demonstrates superior performance over CondFoleyGen across all metrics, as seen in Tab. 2. It achieves leading scores of 36.8% and 42.8% on *CLAP-top* versions, exceeding any CondFoleyGen model by more than 20%. This result suggests that it is better at matching sounds to text prompts. Our model significantly surpasses CondFoleyGen in audio-visual synchronization, evidenced by an 11% improvement in *Onset AP* and a 16% rise in *IoU*. Moreover, we see a 6% enhancement in onset accuracy and an 8.3% increase in timing precision. These advancements illustrate our model's outstanding accuracy in sound event detection and its effectiveness in synchronizing generated sounds with the input video. We think CondFoleyGen relies on the audio-visual synchronization module [15] to improve time accuracy, requiring many samples to re-ranking the sounds. In contrast, our model gets higher synchronization during the generation process with fewer samples.

**Qualitative Results.** As shown in Fig. 4 (row 4, 6, 7), we compare the timestamped positional distance to the target sound. Our results closely match the target sound, demonstrating a high degree of accuracy. However, CondFoleyGen often produces results with the wrong number of sounds and with a large difference in the positional distance.

| Method | $CLAP - top_{general}\uparrow$ | $CLAP - top_{unfused}\uparrow$ | Onset Acc ↑ | Onset AP↑ | Time Acc↑ | IoU↑ |
|---|---|---|---|---|---|---|
| CondFoleyGen(old)* [9] | 17.3 | 15.6 | 21.6 | 67.0 | 35.5 | 23.3 |
| CondFoleyGen(new) * [9] | 16.3 | 13.6 | 19.2 | 68.6 | 37.4 | 22.4 |
| Ours-small | 29.6 | 28.0 | 19.4 | 77.03 | 27.8 | 35.6 |
| Ours-full | **36.8** | **42.8** | **27.6** | **78.1** | **43.8** | **39.7** |

Table 2. **Conditional Generation Task Quantitative Results.** *CLAP-top* metric evaluates a model's ability to control sound content. It calculates the percentage of times the sound samples generated by all models ranked in the top 1 according to the CLAP ranking, divided by the total number of samples. We used two versions of the CLAP models for evaluation: $CLAP_{general}$ and $CLAP_{unfused}$. *Onset Acc* and *Time Acc* are both metrics based on the number of sound occurrences, with *Onset Acc* focusing on the onset count and *Time Acc* on the count of time intervals. We measure the average accuracy of predicting onsets within 0.1s of ground truth to assess the timing of generated onsets. *IoU* is calculated by computing the intersection and union of these vectors. These metrics collectively allow us to comprehensively evaluate the accuracy of the generated sounds in terms of both timing and content. The variant *"old"* corresponds to the prior codebook, while variant *"new"* matches the updated codebook. The old model was trained on 192-width spectrograms, and the new one was trained on 2s waveforms. "* "denotes the data sourced from the official code and is based on experiments conducted with our local configurations. Underline denotes the worst performance. **Boldface** denotes the best performance. The six metrics are all measured in percentage.

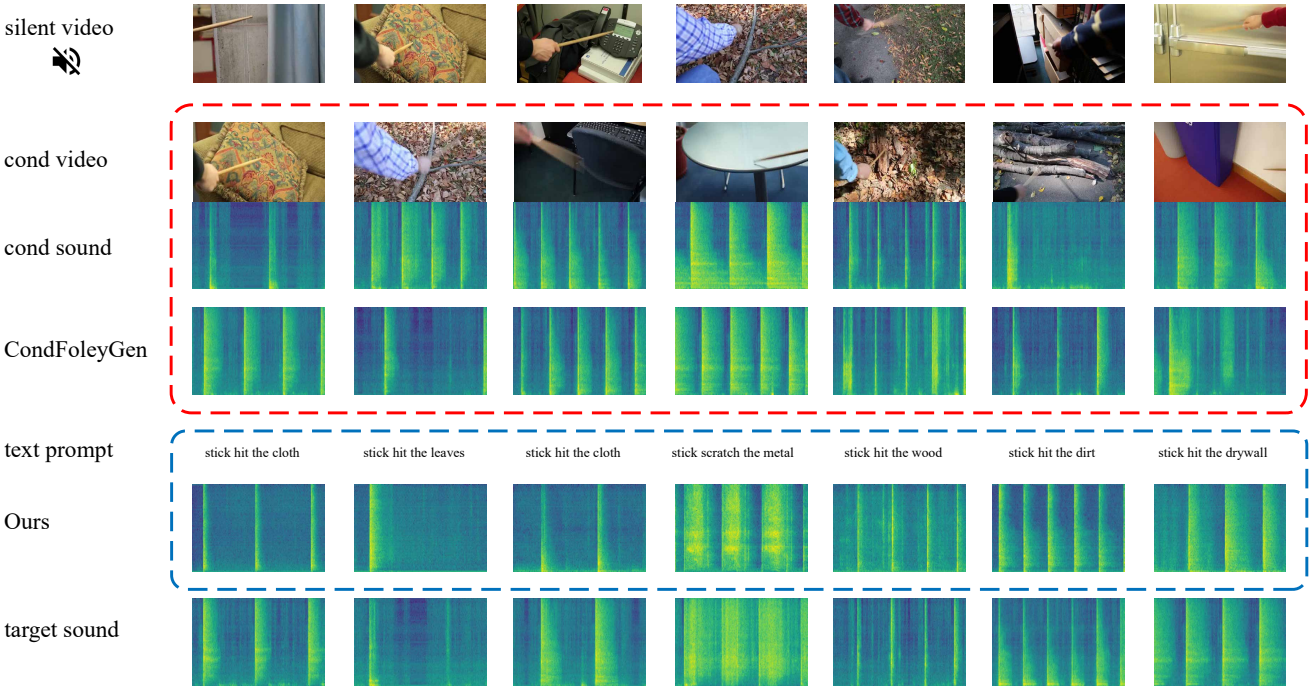


Figure 4. **Conditional Generation Task Qualitative Results.** The red dashed boxes are the conditional audio inputs and generated results for CondFoleyGen, and the blue dashed boxes are the conditional text inputs and results corresponding to SonicVisionLM.

This indicates that our model has a higher visual-audio synchronization than CondFoleyGen. As shown in Fig. 4 (column 3, 7), we compare the sound shape to the conditional sound. Even though our model has not learned any timbre on the Greatest Hits dataset, our results are still very similar to the conditional sound. As shown in Fig. 4 (column 1, 7), The mel-spectrograms of the CondFoleyGen results do not produce sounds that are similar to, and sometimes blurred.

### 4.3. Unconditional Generation

To evaluate the task of unconditional sound generation, considering that our LDM has not been trained on audio-visual datasets, we have chosen to perform quantitative evaluation and qualitative evaluation on two datasets: Greatest Hits [26] and CountixAV [41], which are zero-shot tasks for all models. We use two state-of-the-art V2A models as baselines: SpecVQGAN [14] and DIFF-FOLEY [25].

**Evaluation Metrics.** For *objective evaluation*, we have employed three metrics as [14, 25]: *Inception Score (IS)* [31], *Frechet Distance (FID)* [11], and *Mean KL Divergence (MKL)* [14]. For *subjective evaluation*, as the [9], we conduct user evaluations for the three critical components: *overall audio quality (OVL)*, *alignment with the input video (REL)*, and *time synchronization (Time-sync)*.

| Dataset / Method | Metric | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Greatest Hits | | | | Countix-AV | | | |
| | MKL↓ | FID ↓ | IS↑ | IoU↑ | MKL↓ | FID ↓ | IS↑ | IoU ↑ |
| SpecVQGAN* [14] | <u>6.80</u> | <u>82.4</u> | <u>2.17</u> | 25.8 | 7.39 | <u>34.1</u> | 5.3 | 34.9 |
| DIFF-FOLEY† [25] | 5.68 | **20.0** | 3.84 | <u>22.0</u> | **4.9** | **15.9** | <u>5.2</u> | <u>31.3</u> |
| Ours-small | 6.46 | 31.9 | **3.88** | 36.6 | <u>10.0</u> | 21.7 | **15.1** | 37.5 |
| Ours-full | **4.67** | 24.9 | 3.26 | **39.5** | 9.71 | 19.7 | 12.7 | **42** |

Table 3. **Unconditional Sound Generation Quantitative Results.** IS assesses the quality and diversity of generated samples, FID measures distribution-level similarity, and MKL measures paired sample-level similarity. "* "denotes the data sourced from the official code and is based on experiments conducted with our local configurations. "† "denotes that data is obtained from our adjusted official code. <u>Underline</u> denotes the worst performance. **Boldface** denotes the best performance. IoU metric is measured in percentage.

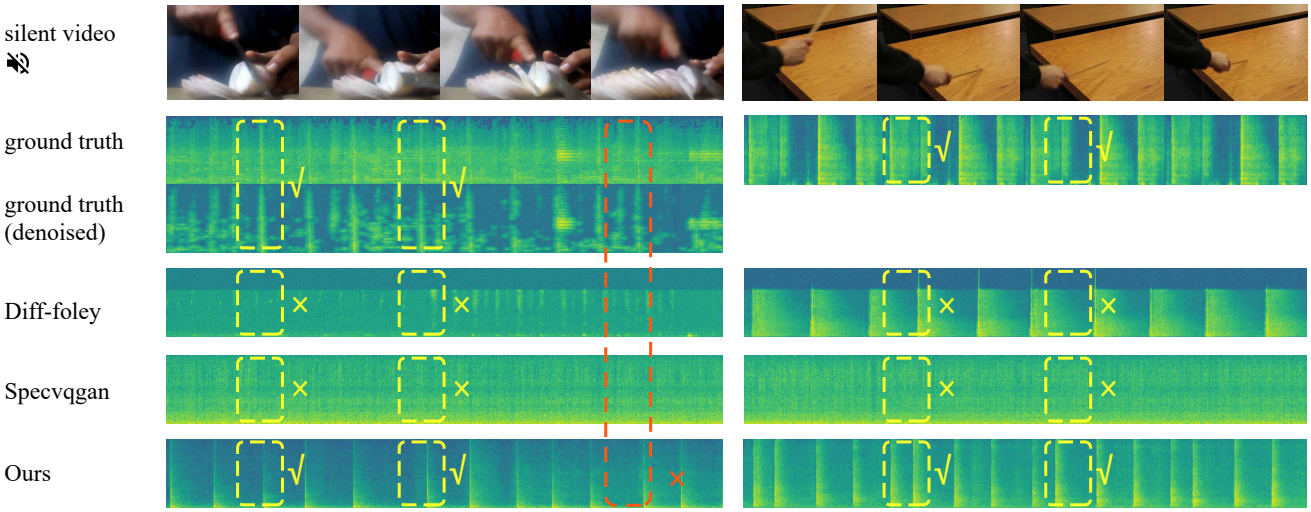


Figure 5. **Unconditional Generation Task Qualitative Results.** The left example is from CountixAV, and the right one is from Greatest Hits. We're comparing them side by side. The dashed box highlights examples of both good and bad results we generated.

| Method | OVL↑ | REL↑ | Time-sync ↑ |
|---|---|---|---|
| SpecVQGAN* [14] | <u>37</u> | <u>25</u> | <u>31</u> |
| DIFF-FOLEY† [25] | 48 | 64 | 58 |
| Ours | **75** | **69** | **87** |

Table 4. **Subjective Results.** Following [9], we invited the 300 English-proficient evaluators to rate 30 randomly selected audio samples from three perspectives: *OVL*, *REL*, and *Time-sync*. Scores were averaged on a 1-100 scale. "*" denotes the data sourced from the official code and is based on experiments conducted with our local configurations. "†" denotes data obtained from our adjusted official code. <u>Underline</u> denotes the worst performance. **Boldface** denotes the best performance.

**Quantitative Results.** In our experiments, the choice of vocoder significantly influenced the *FID* metric. We upgraded from DIFF-FOLEY's weak Griffin-Lim vocoder to the superior MelGAN to ensure fairness. Despite our model using 64 Mel filters compared to MelGAN's 80, our performance on the Greatest Hits dataset excels in *MKL*, IS, and IoU metrics, as shown in Tab. 3. This underscores our model's precise time control and high-quality sound generation, demonstrating its exceptional capability to produce sounds that align closely with the ground truth. In the CountixAV dataset, SonicVisionLM-small outperforms baselines by nearly 10 points in the IS metric, highlighting its exceptional sound quality. Despite slightly lower *MKL* and *FID* scores compared to baselines, we do not view this as a drawback. Unlike the Greatest Hits dataset, which was recorded in high quality with specialized recording equipment, the CountixAV dataset is sourced from diverse YouTube videos. It often includes sounds marred by low-quality background noise or mixed sound sources, especially human vocals. *MKL* and *FID* metrics emphasize similarity to the ground truth. Since DIFF-FOLEY was trained on similar audio-visual data, its tendency to mix multiple sounds in response to complex visual information explains its advantage in these metrics. Our model, targeting sound events from specific actions and filtering out irrelevant auditory information, diverges from the ground truth, a deliberate choice to enhance sound event relevance. The *IoU* metrics show that our model makes sound at the correct times and remains silent otherwise. To further argue our point, we conducted a subjective assessment of three aspects of *OVL*, *REL*, and *Time-sync*. As shown in Tab. 4, our model performed sig-

| Method | FD↓ | FAD↓ | KL↓ | IoU↑ |
|---|---|---|---|---|
| tango-small* [10] | 44.9 | 11.5 | 3.47 | 41.3 |
| Ours-small | 26.2 | **2.05** | **2.30** | **71.6** |
| tango-full* [10] | 44.3 | 10.87 | 3.14 | 39.3 |
| Ours-full | **25.4** | 3.19 | 2.32 | 65.7 |

Table 5. **Time Condition Ablation Results.** "* "denotes the data sourced from the official code and is based on experiments conducted with our local configurations. Underline denotes the worst performance. **Boldface** denotes the best performance. IoU metric is measured in percentage.

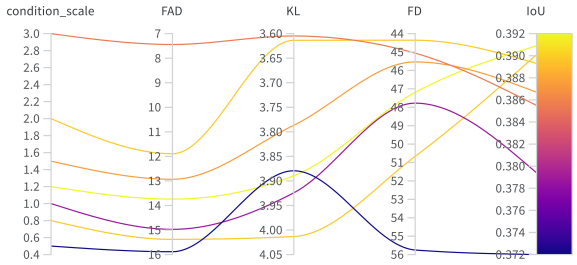


Figure 6. **Conditioning-scale Ablation Study.** In the chart, the higher the indicators are, the better they are. The brightness of the colour also represents a good or bad performance.

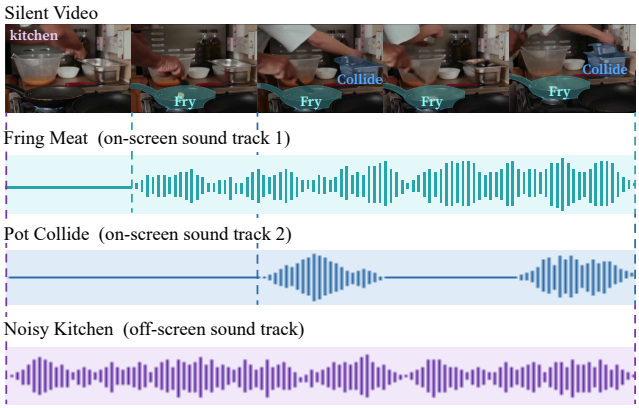


Figure 7. **Multi-soundtracks generation example.** In the video, the blue and green masks show different soundtracks. The timing of on-screen sounds (corresponding to visible actions) aligns accurately with their occurrence in the frames. Conversely, off-screen sounds (not visible in the video) lack precise timing.

nificantly better than the baseline models in all metrics.

**Qualitative Results.** For clearer time point comparison, we denoised the data from the CountixAV. Fig. 5 reveals our generated sounds are more precise than the baseline models. Furthermore, our model can capture finer-grained visual transformations, capturing the signal despite sudden changes in the frequency of movement and producing the correct sound. In contrast, SpecVQGAN results are full of noise. DIFF-FOLEY can sometimes generate clear sounds, but it is not aligned with the visual-sound synchronization.

### 4.4. Ablation Study

**Time condition.** *Evaluation Metrics.* We employed objective metrics from Tango [10], such as the *Frechet Audio Distance (FAD)* [16] and *Frechet Distance(FD)* [24] to measure the distribution difference between generated audio and ground truth without the need for any reference audio samples. *KL divergence* [19, 38] assesses their similarity based on the broad concepts present in the original and the generated audio signals.

*Result.* we compared our model with the Tango [10]. As shown in Tab. 5, both in the small and full versions, our model outperforms Tango in terms of IoU, which validates the effectiveness of the time-controllable adapter.

**Conditioning-scale.** As seen in the Fig. 6. The sound quality improves in the range from 0.6 to 2.0 on the conditioning scale as it lines up better with the IoU. But at 3.0, control goes down a lot. The best mix of good sound quality and steadiness is found at a conditioning scale equal to 2.0.

### 4.5. Multi-soundtracks generation

As shown in Fig. 7, SonicVisionLM works like a sound designer in traditional video post-production. First, it looks at the visual information to find the key sounds needed. For example, it picks out the hissing of cooking meat and the clinking of pots. Then, if the story needs it, the sound designer adds off-screen sounds. This might include extra kitchen sounds like distant chopping, simmering pots, or the buzz of kitchen machines, making it feel more natural. Lastly, SonicVisionLM puts these audio tracks in the right place on the video timeline. This ensures the sounds match perfectly with what is happening in the video, creating a smooth audio-visual experience. More detailed results can be found in the Supplementary Materials.

## 5. Conclusion

In this paper, we propose SonicVisionLM, which utilizes the capabilities of powerful vision-language models (VLMs). When provided with a silent video, SonicVisionLM first identifies events within the video using a VLM to suggest possible sounds that match the video content. SonicVisionLM demonstrates outstanding performance in both conditional and unconditional generation tasks. Further, we tested its efficacy in post-production, focusing on automatic recognition of on-screen sounds and personalized editing for off-screen sounds. Extensive experiments show the superior performance of our method.

**Limitations.** While SonicVisionLM has achieved adequate results, refinements in the visual understanding and timestamp detection parts are still required. Regarding various multimedia contexts, expanding the diversity and range of audio generation control are still the key demands of applications. Thus, the model should enrich the audio-visual experience and broaden applicability in future attempts.

## References


[1] Andrea Agostinelli, Timo I Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, et al. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*, 2023. 2

[2] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*, 2023. 3

[3] Zehua Chen, Xu Tan, Ke Wang, Shifeng Pan, Danilo Mandic, Lei He, and Sheng Zhao. Infergrad: Improving diffusion models for vocoder by considering inference in training. In *ICASSP*, pages 8432–8436. IEEE, 2022. 3

[4] Zehua Chen, Yihan Wu, Yichong Leng, Jiawei Chen, Haohe Liu, Xu Tan, Yang Cui, Ke Wang, Lei He, Sheng Zhao, et al. Resgrad: Residual denoising diffusion probabilistic models for text to speech. *arXiv preprint arXiv:2212.14518*, 2022. 3

[5] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2023. 3

[6] Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. *NeurIPS*, 2024. 2

[7] Chenye Cui, Zhou Zhao, Yi Ren, Jinglin Liu, Rongjie Huang, Feiyang Chen, Zhefeng Wang, Baoxing Huai, and Fei Wu. Varietysound: Timbre-controllable video to sound generation via unsupervised information disentanglement. In *ICASSP*, pages 1–5. IEEE, 2023. 1, 2

[8] Hao-Wen Dong, Xiaoyu Liu, Jordi Pons, Gautam Bhattacharya, Santiago Pascual, Joan Serrà, Taylor Berg-Kirkpatrick, and Julian McAuley. Clipsonic: Text-to-audio synthesis with unlabeled videos and pretrained language-vision models. In *WASPAA*, pages 1–5. IEEE, 2023. 3

[9] Yuexi Du, Ziyang Chen, Justin Salamon, Bryan Russell, and Andrew Owens. Conditional generation of audio from video via foley analogies. In *CVPR*, pages 2426–2436, 2023. 1, 2, 4, 5, 6, 7

[10] Deepanway Ghosal, Navonil Majumder, Ambuj Mehrish, and Soujanya Poria. Text-to-audio generation using instruction-tuned llm and latent diffusion model. *arXiv preprint arXiv:2304.13731*, 2023. 2, 3, 8

[11] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 30, 2017. 6

[12] Qingqing Huang, Daniel S Park, Tao Wang, Timo I Denk, Andy Ly, Nanxin Chen, Zhengdong Zhang, Zhishuai Zhang, Jiahui Yu, Christian Frank, et al. Noise2music: Text-conditioned music generation with diffusion models. *arXiv preprint arXiv:2302.03917*, 2023. 2

[13] Rongjie Huang, Jiawei Huang, Dongchao Yang, Yi Ren, Luping Liu, Mingze Li, Zhenhui Ye, Jinglin Liu, Xiang Yin, and Zhou Zhao. Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models. In *ICML*, pages 13916–13932. PMLR, 2023. 2, 3

[14] Vladimir Iashin and Esa Rahtu. Taming visually guided sound generation. In *BMVC*, 2021. 1, 2, 6, 7

[15] Vladimir Iashin, Weidi Xie, Esa Rahtu, and Andrew Zisserman. Sparse in space and time: Audio-visual synchronisation with trainable selectors. In *BMVC*, 2022. 5

[16] Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fr\'echet audio distance: A metric for evaluating music enhancement algorithms. *arXiv preprint arXiv:1812.08466*, 2018. 8

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3

[19] Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi. Audiogen: Textually guided audio generation. In *ICLR*, 2022. 2, 8

[20] Max WY Lam, Jun Wang, Rongjie Huang, Dan Su, and Dong Yu. Bilateral denoising diffusion models. *arXiv preprint arXiv:2108.11514*, 2021. 3

[21] Max WY Lam, Qiao Tian, Tang Li, Zongyu Yin, Siyuan Feng, Ming Tu, Yuliang Ji, Rui Xia, Mingbo Ma, Xuchen Song, et al. Efficient neural music generation. *NeurIPS*, 36, 2024. 2

[22] Sang-gil Lee, Heeseung Kim, Chaehun Shin, Xu Tan, Chang Liu, Qi Meng, Tao Qin, Wei Chen, Sungroh Yoon, and Tie-Yan Liu. Priorgrad: Improving conditional denoising diffusion models with data-dependent adaptive prior. In *ICLR*, 2021. 3

[23] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023. 3

[24] 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 *ICML*, pages 21450–21474. PMLR, 2023. 2, 3, 8

[25] Simian Luo, Chuanhao Yan, Chenxu Hu, and Hang Zhao. Diff-foley: Synchronized video-to-audio synthesis with latent diffusion models. *NeurIPS*, 2024. 1, 3, 6, 7

[26] Andrew Owens, Phillip Isola, Josh McDermott, Antonio Torralba, Edward H Adelson, and William T Freeman. Visually indicated sounds. In *CVPR*, pages 2405–2413, 2016. 5, 6

[27] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023. 3

[28] Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail Kudinov. Grad-tts: A diffusion probabilistic model for text-to-speech. In *ICML*, pages 8599–8608. PMLR, 2021. 3

[29] Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Fastspeech 2: Fast and high-quality end-to-end text to speech. *arXiv preprint arXiv:2006.04558*, 2020. 2

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015. 3

[31] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *NeurIPS*, 29, 2016. 6

[32] Roy Sheffer and Yossi Adi. I hear your true colors: Image guided audio generation. In *ICASSP*, pages 1–5. IEEE, 2023. 1, 2

[33] Xu Tan, Jiawei Chen, Haohe Liu, Jian Cong, Chen Zhang, Yanqing Liu, Xi Wang, Yichong Leng, Yuanhao Yi, Lei He, et al. Naturalspeech: End-to-end text-to-speech synthesis with human-level quality. *IEEE TPAMI*, 2024. 2

[34] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 3, 4

[35] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, pages 6450–6459, 2018. 4

[36] Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, et al. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *NeurIPS*, 2023. 3

[37] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. In *ICLR*, 2023. 3

[38] Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu. Diffsound: Discrete diffusion model for text-to-sound generation. *TASLP*, 2023. 8

[39] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *EMNLP*, pages 543–553, 2023. 3

[40] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, pages 3836–3847, 2023. 4

[41] Yunhua Zhang, Ling Shao, and Cees GM Snoek. Repetitive activity counting by sight and sound. In *CVPR*, pages 14070–14079, 2021. 5, 6