
11-785 Project Final Report

EchoFrame: Capturing the Sound of Moments - Video2Audio Generation

Xuanrui Chen
Carnegie Mellon University
Pittsburgh, PA 15213
xuanruic@andrew.cmu.edu

Yiming Fu
Carnegie Mellon University
Pittsburgh, PA 15213
yimingfu@andrew.cmu.edu

Zian Pan
Carnegie Mellon University
Pittsburgh, PA 15213
zianp@andrew.cmu.edu

Lenghan Zhu
Carnegie Mellon University
Pittsburgh, PA 15213
lenghanz@andrew.cmu.edu

Abstract

In this project, we address the challenge of generating realistic audio from silent video inputs, with a specific emphasis on achieving high semantic relevance, and audio quality. We explore a cascaded model approach, utilizing pretrained multimodal embeddings—specifically CLIP for visual features and CLAP for audio embeddings—to bridge the gap between visual perception and auditory synthesis. Our proposed model, EchoFrame, consists of a learned mapping network that translates visual embeddings into audio embeddings, subsequently generating audio via a latent diffusion model. We evaluate our approach using perceptual and alignment-based metrics such as Fréchet Audio Distance (FAD) and Kullback-Leibler Divergence (KL), demonstrating improved performance over existing methods. Additionally, we contribute to the community by providing an organized version of the VGGSound dataset, along with publicly available visual and audio embeddings, supporting future research in audio-visual generation.

1 Motivation and Objective

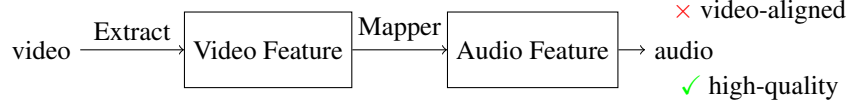
Video-to-audio generation is a compelling task that seeks to bridge the sensory gap between visual and auditory modalities, enabling machines to synthesize realistic soundscapes from silent videos. Human perception is inherently multimodal, and enriching videos with synchronized, contextually accurate audio can enhance immersion in applications like virtual reality, gaming, film production, and assistive technologies for the hearing impaired. Beyond entertainment, this capability has practical benefits in video restoration, generating missing audio in historical footage, and improving accessibility. Our objective in this project is to develop a robust video-to-audio generation model that can produce high-quality, temporally aligned, and semantically meaningful audio outputs. We aim to explore both cascaded and end-to-end model architectures, leveraging pre-trained vision-language models, diffusion-based audio synthesis, and multimodal learning techniques. Through extensive experimentation, we will assess the effectiveness of different approaches using perceptual and alignment-based metrics, ultimately striving for improved synchronization, realism, and diversity in the generated audio.

2 Related Work

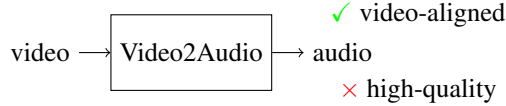
2.1 Literature Review

Generating audio from video is an actively researched area with subtasks that extend to generating sound from objects [1], generating music from human activities [2], music generation [3], generating natural sounds [4], and generating foley in the movie industry [5].

There are mainly two categories of models, as is shown in the chart below. **Cascaded Model** breaks the task into sequential, independently optimized stages, while **end-to-end model** learns a direct mapping from input to output with joint optimization.



(a) cascaded model structure



(b) end-to-end model structure

2.1.1 Cascaded Model

Due to the emergence of multimodal pre-trained visual language models, many researchers have taken advantage of their ability to extract rich semantics from vision by connecting VLMs to text-to-audio models or other Vocoders to form cascade structures. VATT [6] uses a large language model (LLM) with learnable project layer to map video features into a textual representation, along with VATT audio decoder, consist of a bi-directional transformer decoder utilizing masked token prediction approach to generate waveforms. V2A-Mapper [7] proposed to bridge the domain gap between CLIP and CLAP using a mapper network, then generate audio with pretrained audio generation model without training or tuning VLMs. FolyCrafter [8] integrates a pluggable module into a pre-trained text-to-audio (T2A) generator, using a semantic adapter for video-conditioned audio generation and a temporal controller for precise synchronization. There are also researchers aim to build a unified framework to cover multi-tasks involving vision and audio, including video-to-audio, audio-to-video and audio-visual generation [9, 10].

2.1.2 End-to-End Model

Taming Visually Guided Sound Generation [4] In this paper, Iashin et al. proposes a model named SpecVQGAN, which used video inputs as guidance to generate audio outputs based on VQ-VAE and Transformer. The model first uses Vector Quantized Encoding to discretize Mel Frequency, and then uses Transformer block to perform auto-regressive sampling for audio token prediction, while the MelGAN is used to decode the audio token into audio wave. Although this method performs well in the audio quality and inference speed, it still suffers from problems like codebook collapse and high computational costs.

Masked Generative Video-to-Audio Transformers with Enhanced Synchronicity [11] In this paper, Pascual et al. introduces an end-to-end Video-to-Audio generation method based on Transformer and Neural Audio Codec. The model is trained using Masked Audio Token Learning, where video frames are processed through visual feature encoding and Transformer to generate audio tokens. These tokens are then converted into high-quality audio using a Neural Decoder, such as DAC, Descript Audio Codec. Compared to traditional cascaded methods, MaskVAT doesn't require intermediate text or audio labels, directly generating waveforms from video and achieving end-to-end

optimization. Although experiments demonstrate that MaskVAT outperforms existing SOTA methods in terms of audio quality, temporal synchronization, and inference efficiency, it suffers from problems like detail loss and unstable training process.

3 Methodology

3.1 Dataset (VGGSound)

The **VGGSound dataset** [12] is a large-scale collection of **200k video clips**, each lasting **10 seconds**, sourced from YouTube. It spans **309 diverse audio-visual classes**, categorized into broader groups such as *people, sports, nature, home, tools, vehicles, and music*. VGGSound is designed to ensure strong **audio-visual alignment**, leveraging **computer vision techniques** to filter out irrelevant noise. While substantially larger than VAS, it is less curated due to its automated data collection process.

Note: We have spent effort download this dataset and store it on PSC (Since it’s 300~400 GB huge size). Also, many downloads for VGGSound is deprecated on several websites. For this, we fixed the dataset files and also the labels in .CSV file. We posted the first perfectly organized VGGSound on Huggingface, which is already a big contribution to the community.

We also generated audio and visual embeddings for VGGSound, which were also public available open-source community usage.

Embeddings Huggingface Link

Huggingface Link

3.2 Evaluation Metric

Kullback-Leibler Divergence with PassT [13] Liu et al. [6] uses KL Divergence to evaluate the performance between the generated audio against the reference audio. They utilize the pretrained PaSST [13], an audio transformer classifier trained on the AudioSet-2M dataset. Specifically, they extract the classification output probabilities from PaSST and compute the KL divergence in a pairwise manner between the generated and real samples. The final KL divergence score is obtained by averaging the KL divergence values over all 15,446 samples. The formula is given by:

$$D_{KL}(P||Q) = \sum_{i=1}^n P_i \log \frac{P_i}{Q_i}$$

Fréchet Audio Distance (FAD) [14] FAD measures the distance between distributions of real and generated audio embeddings. It is commonly used in speech synthesis, music generation, and audio style transfer to assess how close the generated audio is to real-world samples. Liu et al.[6] also uses FAD for evaluation, which compares between representation extracted by **VGG-ish model** (different from FD) for the generated and reference audio.

$$FAD = \|\mu_r - \mu_g\|^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{\frac{1}{2}})$$

where:

- μ_r, μ_g are the means of the real and generated audio feature distributions,
- Σ_r, Σ_g are the covariance matrices of the real and generated audio features,
- $\|\mu_r - \mu_g\|^2$ is the squared Euclidean distance between the means,
- $\text{Tr}(\cdot)$ denotes the trace of a matrix,
- $(\Sigma_r \Sigma_g)^{\frac{1}{2}}$ is the square root of the product of the covariance matrices.

3.3 Data Pre-processing

Before using CLIP to extract features, we need to conduct several pre-processing techniques for the input videos:

1. Crop the video to 10 seconds, so that all videos can be time-aligned.
2. Resize the images at each frame, and crop the images around the center.
3. Normalize the input images using specific means and standard deviations.

These pre-processing methods can improve the performance of CLIP model.

3.4 Flowchart

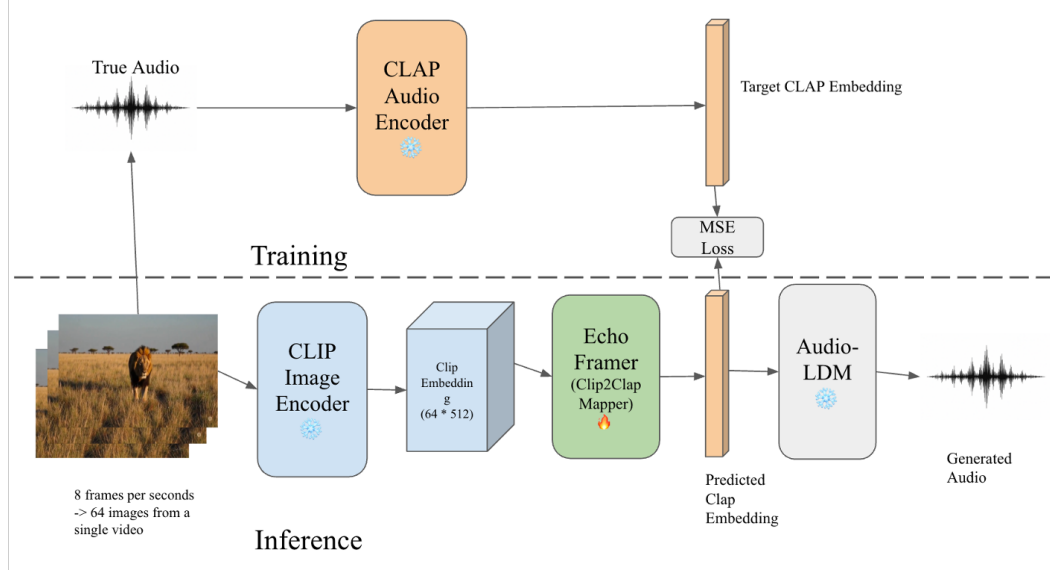


Figure 1: Training and Inference Flowchart

3.5 Model Components

- **Input Video (V):** The source data containing both visual frames and corresponding audio.
- **CLAP Audio Encoder (CLAP_{enc}):** A pre-trained model that extracts semantic audio features, generating a CLAP embedding (e_{clap}) from an audio signal.
- **CLIP Image Encoder (CLIP_{img}):** A pre-trained model that extracts semantic visual features, generating an image embedding (e_{clip}) from an image frame.
- **Echo Framer ($\mathcal{F}$):** A learnable module (Clip2Clap Mapper) designed to map a sequence of CLIP image embeddings to a corresponding predicted CLAP audio embedding.
- **Audio-LDM ($\mathcal{G}$):** A pre-trained latent diffusion model capable of generating audio conditioned on an input embedding (in this case, a CLAP embedding).
- **MSE Loss ($\mathcal{L}_{MSE}$):** Mean Squared Error loss function used for training the Echo Framer.

3.6 Training Process

The goal of the training process is to optimize the parameters (θ) of the Echo Framer ($\mathcal{F}_\theta$) to accurately predict the CLAP embedding of a video's audio track based solely on its visual frames.

1. Target CLAP Embedding Generation:

- Extract the true audio signal (A_{true}) from the input video V .
- Process the true audio through the CLAP Audio Encoder to obtain the target CLAP embedding:

$$e_{clap_target} = \text{CLAP}_{enc}(A_{true})$$

2. Predicted CLAP Embedding Generation:

- Sample N frames $\{f_1, f_2, \dots, f_N\}$ from the input video V . (Based on our implementation, $N = 64$, sampled at 8 frames per second).
- Process each frame f_i through the CLIP Image Encoder to obtain a sequence of image embeddings:

$$E_{clip} = \{\text{CLIP}_{img}(f_1), \text{CLIP}_{img}(f_2), \dots, \text{CLIP}_{img}(f_N)\}$$

where each $\text{CLIP}_{img}(f_i) \in \mathbb{R}^{D_{clip}}$ (e.g., $D_{clip} = 512$ based on the diagram).

- Feed the sequence of CLIP embeddings E_{clip} into the Echo Framer ($\mathcal{F}_\theta$) to generate the predicted CLAP embedding:

$$e_{clap_pred} = \mathcal{F}_\theta(E_{clip})$$

3. Parameter Update:

- Calculate the Mean Squared Error (MSE) between the target and predicted CLAP embeddings:

$$\mathcal{L}_{MSE} = ||e_{clap_target} - e_{clap_pred}||_2^2$$

- Update the parameters θ of the Echo Framer $\mathcal{F}_\theta$ using gradient descent based on the calculated loss $\mathcal{L}_{MSE}$. The objective is to minimize this loss:

$$\min_{\theta} \mathcal{L}_{MSE}$$

3.7 Inference Process

During inference, the trained Echo Framer is used along with the CLIP Image Encoder and Audio-LDM to generate audio for a given input video V .

1. CLIP Embedding Extraction:

- Sample N frames $\{f_1, f_2, \dots, f_N\}$ from the input video V .
- Process each frame f_i through the CLIP Image Encoder to obtain the sequence of image embeddings E_{clip} :

$$E_{clip} = \{\text{CLIP}_{img}(f_1), \text{CLIP}_{img}(f_2), \dots, \text{CLIP}_{img}(f_N)\}$$

2. Predicted CLAP Embedding Generation:

- Feed the sequence of CLIP embeddings E_{clip} into the trained Echo Framer $\mathcal{F}_{\theta^*}$ (where θ^* represents the optimized parameters) to generate the predicted CLAP embedding:

$$e_{clap_pred} = \mathcal{F}_{\theta^*}(E_{clip})$$

3. Audio Generation:

- Input the predicted CLAP embedding e_{clap_pred} into the pre-trained Audio-LDM generator $\mathcal{G}$:

$$A_{gen} = \mathcal{G}(e_{clap_pred})$$

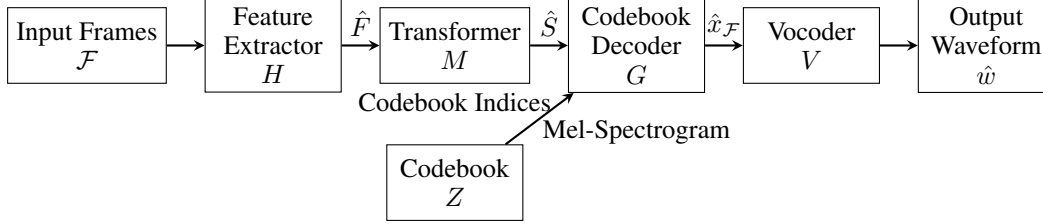
- The output A_{gen} is the final generated audio corresponding to the input video V .

4 Baselines and Extensions

4.1 A reproduced work: SpecVQGAN

The paper introduces an efficient approach for generating high-fidelity sounds from visual cues across multiple classes. The key innovation is a three-stage pipeline:

1. **Perceptually-Rich Spectrogram Codebook** - A variant of VQGAN for audio spectrograms
2. **Vision-based Conditional Cross-modal Autoregressive Sampler** - A transformer that samples from the codebook
3. **Spectrogram Vocoder** - Converts spectrograms to waveforms



4.2 EchoFrame

Here we have tried three different versions of V2AMapper, utilizing BiLSTM, MLP and Transformer respectively. Below is our model structure table.

Model	Hidden layers	(Hyper-)Parameters
V2AMapperBiLSTM	1 × BiLSTM layer, followed by 1 × Sequential Block: LayerNorm (LN) → 2 × V2ABlock: Linear & LN (512 → 1024) → ReLU → Dropout (0.1) → Linear & LN (1024 → 512) → Linear & LN (512 → 512)	BiLSTM hidden size: 512 Intermediate dim: 1024 Final output dim: 512 Total Parameters: 3,943,936 Input shape: (1, 64, 512) Learning rate: 0.001 Batch size: 2048
V2AMapperMLP	Linear & LN (512 → 2048) → 2 × Sequential Block: Linear (2048 → 4096) → ReLU → Dropout (0.1) → Linear & LN (4096 → 2048) → 1 × V2ABlock: Linear (2048 → 2048) → ReLU → Dropout (0.1) → Linear & LN (2048 → 2048) → Linear & LN (2048 → 2048) → Linear & LN (2048 → 512)	Residual Connect (In V2ABlock) Total Parameters: 35,679,744
V2AMapperTransformer	Linear (512 → 768) → 1 × TransformerEncoder Block: 4 × TransformerEncoderLayer → Linear (768 → 512)	Hidden Dimension: 768 Number of Heads: 12 Number of Layers: 4 Total Paramters: 29,189,888

Table 1: Model architecture and hyperparameters of different versions of V2AMapper

5 Results and Analysis

5.1 Results

Model	VISUAL FEATURE	Mapper Arch	Metrics	
			FAD ↓	MKL ↓
SpecVQGAN	ResNet50	\	9.40	7.03
Im2Wav	CLIP	\	11.44	5.20
Diff-Foley	CAVP	\	9.87	6.43
EchoFrame	CLIP	Bi-LSTM	7.38	6.29
EchoFrame	CLIP	Transformer	7.23	6.22
EchoFrame	CLIP	MLP w/o Res	6.30	5.83
EchoFrame	CLIP	MLP w/ Res	6.28	5.87

Table 2: Experiment results of different models

5.2 Discussion

In this section, we will discuss about our our experiment results, as is summarized in the table above.

5.2.1 Quantitative Evaluation and Ablation Studies

To evaluate the performance of our proposed Echo Frame model, we conducted comparative experiments against several established baseline methods: SpecVQGAN, Im2Wav, and Diff-Foley. Concurrently, we performed an ablation study to analyze the impact of different architectures within our Echo Frame mapper component, specifically comparing Bidirectional Long Short-Term Memory (Bi-LSTM), Transformer, Multi-Layer Perceptron without residual connections (MLP w/o Res), and MLP with residual connections (MLP w/ Res). All Echo Frame variants utilized CLIP visual features.

Quantitative evaluation was performed using two standard metrics: Fréchet Audio Distance (FAD) and Mean Kullback-Leibler Divergence (MKL), where lower scores indicate better performance in matching the distribution of real audio features. The results are summarized in 2.

The findings clearly demonstrate the efficacy of the Echo Frame approach compared to the baselines, achieving lower FAD scores across all its variants. The ablation study further reveals that the choice of mapper architecture is significant. Notably, the Echo Frame model configured with an MLP mapper incorporating residual connections (MLP w/ Res) achieved the best FAD score of 6.28. While the Im2Wav baseline obtained the lowest MKL score (5.20), our top-performing variant (MLP w/ Res) registered a competitive MKL of 5.87, indicating a strong overall performance, particularly in terms of FAD.

The model demonstrated a strong capability to generate relevant audio for unseen videos from the VGG-Sound test set, indicating successful learning of audio-visual correspondences within this domain. An illustrative example of generated audio for a VGG-Sound video can be accessed here: [Google Drive](#).

5.2.2 Training

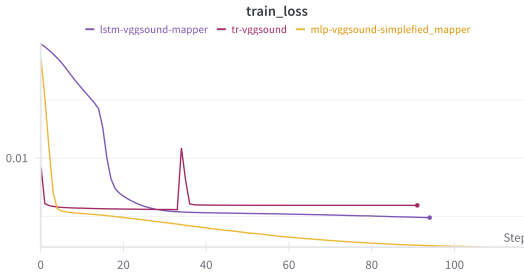


Figure 2: Training Loss

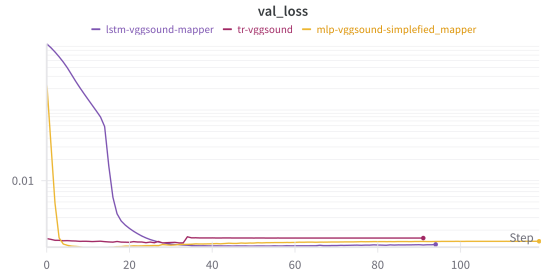


Figure 3: Validation Loss

During training, we observed that the MLP-based mapper converges much faster, with a rapid drop in training loss, compared to its LSTM and Transformer counterparts, which show slightly slower convergence. Notably, the Transformer-based V2A Mapper exhibits fluctuations during training, suggesting instability. We also found that increasing model complexity often leads to worse performance—for instance, a deeper Transformer-based mapper performs worse than a shallower one. This suggests that more complex models are prone to overfitting, and that the underlying mapping between the CLIP and CLAP embedding spaces may be relatively simple and does not require high model complexity to learn effectively.

5.2.3 Generalization Capability

Furthermore, we tested the model’s generalization capacity on video content generated by state-of-the-art text-to-video AI models, which often lack native audio tracks. The pipeline successfully generated plausible audio for a video produced by OpenAI’s Sora model, showcasing its applicability beyond the distribution of the training data. This example is available for review at: [Google Drive](#). These

qualitative findings suggest that our model learns a generalizable mapping from visual dynamics to corresponding sound characteristics.

5.2.4 Limitations

Despite the promising results, a key limitation observed during evaluation relates to the model’s dependence on the semantic coherence of the audio-visual pairs in the training data. When presented with visual scenes where the originally associated audio lacks a strong or direct semantic link (or if such unrelated pairs were present in training), the model struggles to generate contextually appropriate sound. This suggests that the model currently lacks robust ‘imaginative’ capabilities to infer plausible audio for arbitrary visual inputs independent of learned training correlations. The generation quality appears sensitive to the inherent semantic relationship strength within the source training examples.

5.3 Future Work

Several promising directions for future research emerge from this study:

Architectural Exploration and Ablation Studies: Further investigation into alternative neural network architectures for the Clip2Clap mapping component (Echo Framer) is warranted. Exploring different recurrent, convolutional, or transformer-based designs could potentially yield improvements in capturing complex temporal dependencies between visual Cues and audio events. Additionally, conducting more extensive ablation experiments would help rigorously quantify the impact of specific design choices and model components.

Direct Audio-Visual Contrastive Learning: An alternative avenue involves exploring end-to-end training paradigms based on audio-visual contrastive learning. Directly optimizing models to align video frame representations and corresponding audio segment representations in a shared multimodal embedding space could potentially lead to more robust and semantically richer mappings compared to relying solely on pre-trained unimodal encoders.

Enhanced Evaluation Protocols: While objective metrics provide valuable insights, future work should incorporate comprehensive subjective evaluation methodologies. Utilizing human raters to assess critical aspects such as the perceptual quality, semantic relevance to the video content, and temporal synchronization of the generated audio is essential for a more complete understanding of the model’s performance and limitations in real-world perception.

6 Conclusion

This project successfully developed and demonstrated a computationally efficient pipeline for generating audio directly from silent video input. By mapping visual features extracted via CLIP to auditory representations in the CLAP embedding space using a trained Echo Framer module, our approach provides a practical method for synthesizing corresponding sound. A key contribution of this work lies in addressing the common absence of synchronized audio in AI-generated video content, thereby offering a means to enhance the realism and immersive quality of such media. The proposed framework serves as a valuable step towards bridging the gap between visual and auditory modalities in generative modeling.

7 Team contribution, Git REPO

- Zian Pan: Building Pipeline, Running Ablation, Prepare Dataset, Literature Review, Video Recording, Report Writing, Drinking Wushiland, Providing Psychological Support
- Lenghan Zhu: Building Pipeline, Running Ablation, Prepare Dataset, Literature Review, Video Recording, Report Writing, Drinking Wushiland, Providing Psychological Support
- Yiming Fu: Building Pipeline, Running Ablation, Prepare Dataset, Literature Review, Video Recording, Report Writing, Drinking Wushiland, Providing Psychological Support
- Xuanrui Chen: Literature Review, Topic/Title Selection, Report Writing, Drinking Wushiland, Discussing with Professor

Github Repo (Public Currently)

References

- [1] Andrew Owens, Phillip Isola, Josh McDermott, Antonio Torralba, Edward H Adelson, and William T Freeman. Visually indicated sounds. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2405–2413, 2016.
- [2] Kun Su, Xiulong Liu, and Eli Shlizerman. Audeo: Audio generation for a silent performance video. *Advances in Neural Information Processing Systems*, 33:3325–3337, 2020.
- [3] Kun Su, Judith Yue Li, Qingqing Huang, Dima Kuzmin, Joonseok Lee, Chris Donahue, Fei Sha, Aren Jansen, Yu Wang, Mauro Verzetti, et al. V2meow: Meowing to the visual beat via music generation. *arXiv preprint arXiv:2305.06594*, 2023.
- [4] Vladimir Iashin and Esa Rahtu. Taming visually guided sound generation. *arXiv preprint arXiv:2110.08791*, 2021.
- [5] Xinhao Mei, Varun Nagaraja, Gael Le Lan, Zhaoheng Ni, Ernie Chang, Yangyang Shi, and Vikas Chandra. Foleygen: Visually-guided audio generation. In *2024 IEEE 34th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2024.
- [6] Xiulong Liu, Kun Su, and Eli Shlizerman. Tell what you hear from what you see—video to audio generation through text. *arXiv preprint arXiv:2411.05679*, 2024.
- [7] Heng Wang, Jianbo Ma, Santiago Pascual, Richard Cartwright, and Weidong Cai. V2a-mapper: A lightweight solution for vision-to-audio generation by connecting foundation models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15492–15501, 2024.
- [8] Yiming Zhang, Yicheng Gu, Yanhong Zeng, Zhening Xing, Yuancheng Wang, Zhizheng Wu, and Kai Chen. Foleyrafter: Bring silent videos to life with lifelike and synchronized sounds. *arXiv preprint arXiv:2407.01494*, 2024.
- [9] Kun Su, Xiulong Liu, and Eli Shlizerman. From vision to audio and beyond: A unified model for audio-visual representation and generation. *arXiv preprint arXiv:2409.19132*, 2024.
- [10] Lei Zhao, Linfeng Feng, Dongxu Ge, Fangqiu Yi, Chi Zhang, Xiao-Lei Zhang, and Xuelong Li. Uniform: A unified diffusion transformer for audio-video generation. *arXiv preprint arXiv:2502.03897*, 2025.
- [11] Santiago Pascual, Chunghsin Yeh, Ioannis Tsiamas, and Joan Serra. Masked generative video-to-audio transformers with enhanced synchronicity. In *European Conference on Computer Vision*, pages 247–264. Springer, 2024.
- [12] H. Chen, W. Xie, and A. Zisserman. Vggsound: A large-scale audio-visual dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.
- [13] Khaled Koutini, Jan Schlüter, Hamid Eghbal-Zadeh, and Gerhard Widmer. Efficient training of audio transformers with patchout. *arXiv preprint arXiv:2110.05069*, 2021.
- [14] 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.