MMAudio: Taming Multimodal Joint Training for High-Quality Video-to-Audio Synthesis

Supplementary Material

Table of Contents

A. User Study

In addition to the objective metrics presented in Table 1, we have also performed a user study for subjective evaluation on the VGGSound [1] test set. For comparisons, we have selected our best model (MMAudio-L-44.1kHz) and four best baselines:

1. Seeing and Hearing [69], as it has the highest ImageBind (*i.e.*, best semantic alignment with videos) score, besides ours.
2. V-AURA [65], as it has the lowest DeSync (*i.e.*, best temporal alignment) with videos, besides ours.
3. VATT [40], as it has the lowest Kullback–Leibler divergence (*i.e.*, KL_{PANNs} and KL_{PaSST}), besides ours.
4. V2A-Mapper [66], as it has the lowest Fréchet distances (*i.e.*, FD_{PaSST}, FD_{PANNs}, and FD_{VGG}), besides ours.

We sample eight videos from the VGGSound [1] test set, after excluding videos that are of low-resolution (below 360p) or that contain human speech. In total, each participant evaluates 40 videos (8 videos $\times$ 5 methods). We group the samples for the same video, and randomly shuffle the ordering in each group to avoid bias. We ask each participant to rate the generation in three aspects using the Likert scale [36] (1-5; strongly disagree, disagree, neutral, agree, strongly agree) providing the following instructions:

(a) The audio is of **high quality**.
Explanation: An audio is low-quality if it is noisy, unclear, or muffled. In this aspect, ignore visual information and focus on the audio.

(b) The audio is **semantically aligned** with the video.
Explanation: An audio is semantically misaligned with the video if the audio effects are unlikely to occur in the scenario depicted by the video, *e.g.*, the sound of an explosion in a library.

(c) The audio is **temporally aligned** with the video.
Explanation: An audio is temporally misaligned with the video if the audio sounds delayed/advanced compared to the video, or when audio events happen at the wrong time (*e.g.*, in the video, the drummer hits the drum twice and stops; but in the audio, the sound of the drum keeps occurring).

In total, we have collected 920 responses in each of these aspects from 23 participants. Table A1 summarizes the results from the user study. MMAudio receives significantly higher ratings in all three aspects from the users, which aligns with the objective metrics presented in Table 1 of the main paper.

Method	Audio quality↑	Semantic alignment↑	Temporal alignment↑
Seeing&Hearing [69]	2.65±1.05	3.10±1.24	1.85±0.99
V-AURA [65]	3.59±1.02	3.70±1.17	3.65±1.16
VATT [40]	2.66±0.99	3.32±1.17	2.04±1.07
V2A-Mapper [66]	3.00±0.95	3.28±1.27	2.03±1.11
MMAudio-L-44.1kHz	**4.14±0.77**	**4.52±0.74**	**4.46±0.80**

Table A1. Average ratings for each method from the user study. We show mean±std in each aspect.

B. Comparisons with Movie Gen Audio

Recently, Movie Gen Audio [52] has been introduced for generating sound effects and music for input videos. While Movie Gen Audio's technical details are sparse, it represents the industry's current state-of-the-art video-to-audio synthesis algorithm. Its 13-billion parameters model has been trained on non-publicly accessible data that is $> 100\times$ larger than ours. Nevertheless, we compare MMAudio to Movie Gen Audio [52] to benchmark the differences between public and private models.

At the time of writing, the only accessible outputs from Movie Gen Audio are 527[2] generations in the "Movie Gen Audio Bench" dataset. All the videos from Movie Gen Audio Bench are generated by MovieGen [52], which we note is different from the distribution of real-world videos (*e.g.*, over-smoothed textures, slow motions). Since these are synthetic videos, there is no corresponding ground-truth audio. We run our best model MMAudio-L-44.1kHz on these videos and the corresponding audio prompts (which Movie Gen Audio also uses) and compare our generations with Movie Gen Audio.

Since there is no ground truth audio, among the standard metrics that we have used in the main paper, we can only evaluate Inception Score (IS, audio quality), IB-score (ImageBind [11] similarly, semantic alignment between video and audio), DeSync (misalignment predicted by SynchFormer [19] between video and audio), and CLAP [7, 68] (alignment between text and

[2]While the MovieGen technical report mentioned 538 samples, only 527 were released at the time of writing.

audio). Additionally, we have conducted a user study following the protocol of Appendix A, and have excluded audios with very low volume (cannot be heard clearly at a normal volume) generated by Movie Gen Audio to prevent bias. We sampled a total of 5 videos and received 230 responses in each of the aspects from 23 participants.

Table A2 summarizes our results. In subjective metrics, MMAudio is comparable to Movie Gen Audio – slightly worse in semantic alignment and slightly better in temporal alignment. In objective metrics, we observe the same trend – MMAudio and Movie Gen Audio obtain the same audio quality (IS) score, Movie Gen Audio has a better semantic alignment (IB-score and CLAP), and MMAudio has a better video-audio synchrony (DeSync).

			Subjective metrics			Objective metrics			
Method	Param	Training data	Audio qual.↑	Semantic align.↑	Temporal align.↑	IS↑	IB-score↑	CLAP↑	DeSync↓
Movie Gen Audio [52]	13B	$\mathcal{O}(1,000,000)$h	**3.93±0.92**	**4.36±0.74**	3.52±1.21	**8.40**	**36.26**	**0.4409**	1.006
MMAudio-L-44.1kHz	1.03B	$\sim 8,200$h	**3.93±0.89**	4.26±0.71	**3.62±1.03**	**8.40**	27.01	0.4324	**0.771**

Table A2. Comparisons between Movie Gen Audio and MMAudio in both subjective metrics (from user study) and objective metrics. For the subjective metrics, we show mean±std.

Further, in terms of IB-score, we find that MMAudio struggles more in some videos, while Movie Gen Audio delivers more consistent results. We plot the sorted IB-score comparing MMAudio and Movie Gen Audio in Figure A1 (left). Movie Gen Audio consistently performs better in the low-performance regime, but the gap narrows in the high-performance region. We believe this is due to our limited training data, which is unable to adequately cover the data in Movie Gen Audio Bench and thus falls short in unfamiliar video types. Note, our only video-audio dataset for training is VGGSound [1] which contains videos for 310 classes. We hypothesize that collecting open-world data beyond these classes can effectively reduce this performance gap. The same phenomenon occurs at a much smaller scale for the CLAP score, which might be because we use more audio-text data. Figure A2 shows examples where we obtain a substantially higher/lower IB-score on videos with concepts well/not well covered by the training data.

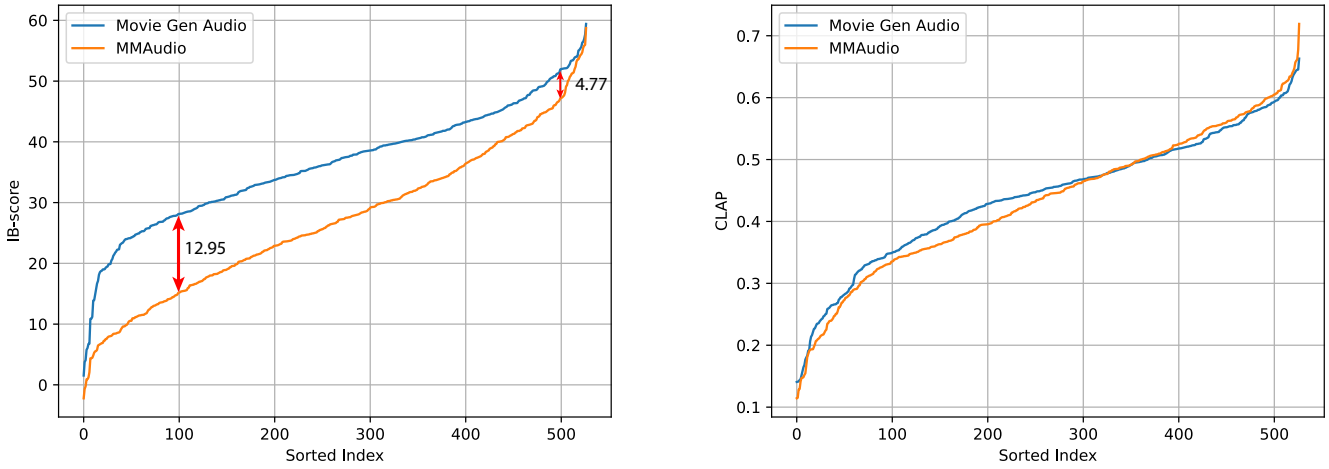

Figure A1. Sorted MMAudio and Movie Gen Audio performance scores in Movie Gen Audio Bench.

C. Evaluation on the Greatest Hits Dataset

To address any potential bias by using the model-based DeSync metric, we conduct an additional experiment to assess temporal alignment by comparing the onsets of generated audio with ground-truth labels. Concretely, we use the *Greatest Hits* [48] test set (244 videos) which contains videos with distinct and labeled sound events (*a drumstick hitting <object>*). Notably, neither our models nor the baselines have been trained on this dataset. We test both our method and baselines (using available code) on each video's first 8 seconds (due to the models' constraints): we extract onsets from the generated audio following [6] and compare them with the labeled sound events. We assess performance using accuracy, average precision (AP), and F1-score. We provide the results in Table 2 and visualize the spectrograms in Figure A8. MMAudio achieves significantly better performance in these *model-free* metrics. Note that a high AP (not accuracy or F1) can be achieved by generating very few onsets (*e.g.*, silent/noise), which is the case in Seeing&Hearing.

Audio prompt: rhythmic splashing and lapping of water
IB-score (Movie Gen Audio): 42.74
IB-score (MMAudio, ours): 53.95

Audio prompt: creamy sound of mashed potatoes being scooped
IB-score (Movie Gen Audio): 30.94
IB-score (MMAudio, ours): 10.52

Figure A2. Examples of videos in Movie Gen Audio Bench that are well/not well covered by our training data. Left: with a familiar concept in our training data (516 swimming videos in the VGGSound training set), MMAudio achieves a higher IB-score. Right: with an unfamiliar concept (there are no videos about mashed potatoes in VGGSound [1], according to the provided labels), MMAudio attains a significantly lower IB-score.

D. Ablations on Filling in Missing Modalities

Among our training data, VGGSound is the only tri-modal (with class names as text) dataset while all others are audio-text. For other data, we replace missing visual modalities (CLIP and Sync features) with end-to-end learnable embeddings ($\varnothing_v$ and $\varnothing_{syn}$) and missing text modalities with the empty string ($\varnothing_t$). We believe other methods to fill in missing modalities would be similarly effective since the deep net likely adapts. Indeed, replacing the missing modalities with either all learnable embeddings or zeros yields no significant difference (Table A3). Note, we also drop modalities randomly during training to enable classifier-free guidance, which enhances the model's robustness to missing modalities.

Method	FD_{PaSST}↓	IS↑	IB-score↑	DeSync↓
Ours	70.19	14.44	29.13	0.483
With all learnable	70.13	14.63	29.23	0.494
With zeros	69.91	14.60	29.22	0.496

Table A3. Comparisons of different methods to fill in missing modalities. As expected, there is no significant difference as the deep net learns to adapt.

E. Details on Data Overlaps

We note that there are training and testing data overlaps among commonly used datasets for video-to-audio generation. For example, AudioSet [9] is commonly used to train VAE encoders/decoders but it contains test set data from VGGSound [1] and AudioCaps [24]. Additionally, AudioCaps is often used to train text-to-audio models [70], which is then used as the backbone for video-to-audio models which evaluate on VGGSound [1] – however, part of the VGGSound test set overlaps with the AudioCaps training set. Moreover, AVSync15 [72], which is sometimes used jointly with VGGSound for training/evaluating video-to-audio algorithms [73], contains severe cross-contamination with VGGSound. This results in biased evaluations in both VGGSound and AVSync15. To our best knowledge, this data contamination is not yet addressed in the video-to-audio community. We thank Labb et al. [31] for raising this issue in the audio captioning field, which has helped us identify this problem.

Table A4 summarizes the observed overlaps. The overlaps with WavCaps [46] and Freesound [33] have been included as part of their release, which we do not repeat in our table.

We have carefully removed from our training data (AudioSet [9], AudioCaps [24], Clotho [5], Freesound [33], WavCaps [46], and VGGSound [1]) anything that overlaps with any of the test sets (VGGSound and AudioCaps). Additionally, we have also removed from our training data the test set of Clotho [5]. Since most baselines have been trained on VGGSound, we elect not to evaluate on AVSync15.

Test sets (number of samples)	Training sets			
	AudioSet	AudioCaps	VGGSound	AVSync15
AudioCaps (975)	580 (59.5%)	-	147 (15.1%)	-
VGGSound (15,496)	132 (0.9%)	13 (0.1%)	-	59 (0.4%)
AVSync-15 (150)	-	-	144 (96.0%)	-

Table A4. Overlaps between training and test sets of different datasets. The percentage denotes the proportion of overlapping data in the entire test set. "-" means that we did not compute this data (we do not train or test on AVSync15).

F. Details on the Audio Latents

As mentioned in the main paper, we obtain the audio latents by first transforming audio waveforms with the short-time Fourier transform (STFT) and extracting the magnitude component as mel spectrograms [57]. Then, spectrograms are encoded into latents by a pretrained variational autoencoder (VAE) [27]. During testing, the generated latents are decoded by the VAE into spectrograms, which are then vocoded by a pretrained vocoder [35] into audio waveforms. Table A5 tabulates our STFT parameters and latent information.

For the VAE, we follow the 1D convolutional network design of Make-An-Audio 2 [15] with a downsampling factor of 2 and trained with reconstruction, adversarial, and Kullback–Leibler divergence (KL) objectives. We note that the default setting leads to extreme values in the latent at the end of every sequence ($\pm 10\sigma$ away). To tackle this problem, we have applied the magnitude-preserving network design from EDM2 [22], by replacing the convolutional, normalization, addition, and concatenation layers with magnitude-preserving equivalents. While this change removes the extreme values, it leads to no significant empirical performance difference. We train the 16kHz model on AudioSet [9], following Make-An-Audio 2 [15]. For the 44.1kHz model, we increase the hidden dimension from 384 to 512 and train it on AudioSet [9] and Freesound [33] to accommodate the increased reconstruction difficulty due to a higher sampling rate.

For vocoders, we use the BigVGAN [35] trained by Make-An-Audio 2 [15] in our 16kHz model. For our 44.1kHz model, we use BigVGAN-v2 [35] (the `bigvgan_v2_44khz_128band_512x` checkpoint).

Model variants	Latent frame rate	# latent channels	# mel bins	# FFTs	Hop size	Window size	Window function
16kHz	31.25	20	80	1024	256	1024	Hann
44.1kHz	43.07	40	128	2048	512	2048	Hann

Table A5. Short-time Fourier transform (STFT) parameters and latent information.

G. Network Details

G.1. Model Variants

Our default model generates 16kHz audio encoded as 20-dimensional, 31.25fps latents (following Frieren [67]), with $N_1 = 4, N_2 = 8, h = 448$. We refer to this default model as 'S-16kHz'. To faithfully capture higher frequencies, we also train a 44.1kHz model ('S-44.1kHz') that generates 40-dimensional, 43.07fps latents while all other settings are identical to the default. To scale up the high-frequency model, we first double the hidden dimension to match the doubled latent dimension, *i.e.*, we use $N_1 = 4, N_2 = 8, h = 896$ and refer to this model using 'M-44.1kHz'. Finally, we scale the number of layers, *i.e.*, $N_1 = 7, N_2 = 14, h = 896$ and refer to this model via 'L-44.1kHz'. These model variants are summarized in Table A6.

G.2. Projection Layers

We use projection layers to project input text, visual, and audio features to the hidden dimension h and for initial aggregation of the temporal context.

Text feature projection. We use a linear layer that projects to h, followed by an MLP.

Clip feature projection. We use a linear layer that projects to h, followed by a ConvMLP with a kernel size of 3 and a padding of 1.

Model variants	Params	# multimodal blocks N_1	# single-modal blocks N_2	Hidden dim h	Latent dim	Time (s)
S-16kHz	157M	4	8	448	20	1.23
S-44.1kHz	157M	4	8	448	40	1.30
M-44.1kHz	621M	4	8	896	40	1.35
L-44.1kHz	1.03B	7	14	896	40	1.96

Table A6. Summary for different MMAudio model variants. Time is the total running time to generate one sample with a batch size of one after warm-up and excludes any disk I/O operations on an H100 GPU.

Sync feature projection. We use a 1D convolutional layer with a kernel size of 7 and a padding of 3 that projects to h, an SELU [28] activation layer, followed by a ConvMLP with a kernel size of 3 and a padding of 1.

Audio feature projection. We use a 1D convolutional layer with a kernel size of 7 and a padding of 3 that projects to h, an SELU [28] activation layer, followed by a ConvMLP with a kernel size of 7 and a padding of 3.

G.3. Gating

The gating layers are similar to the adaptive normalization layers (adaLN). Each global gating layer modulates its input $y \in \mathbb{R}^{L \times h}$ (L is the sequence length) with the global condition c_g as follows:

$$\text{Gating}_g(y, c_g) = y \cdot \mathbf{1}\mathbf{W}_g(c_g). \tag{A1}$$

Here, $\mathbf{W}_g \in \mathbb{R}^{h \times h}$ is an MLP, and $\mathbf{1}$ is a $L \times 1$ all-ones matrix, which "broadcasts" the scales to match the sequence length L – such that the same condition is applied to all tokens in the sequence (hence global).

Similarly, for per-token gating layers, the frame-aligned conditioning c_f is injected into the audio stream for precise feature modulation via

$$\text{Gating}_f(y, c_f) = y \cdot \mathbf{W}_f(c_f), \tag{A2}$$

where $\mathbf{W}_f \in \mathbb{R}^{h \times h}$ is an MLP. Different from Equation (A1), the scales are applied per token without broadcasting,

G.4. Details on Synchronization Features

We use the visual encoder of Synchformer [19] to extract synchronization features. We use the pretrained audio-visual synchronization model trained on AudioSet, provided by Iashin et al. [19]. As input, we obtain frames at 25 fps. Synchformer partitions these frames into overlapping clips of 16 frames with stride 8 and produces features of length 8 for each clip. Thus, for a video of length T_{sec} seconds, the sequence length of the synchronization features is

$$L_{\text{sync}} = 8 \left(\left\lfloor \frac{25 T_{\text{sec}} - 16}{8} \right\rfloor + 1 \right). \tag{A3}$$

The corresponding feature fps is

$$\text{FPS}_{\text{sync}} = \frac{L_{\text{text}}}{T_{\text{sec}}}. \tag{A4}$$

In this paper, we experimented with $T_{\text{sec}} = 8$ and $T_{\text{sec}} = 10$. In both cases, FPS_{sync} is exactly 24. Additionally, we introduce a learnable positional embedding of length 8 (matching the number of features in each clip processed by Synchformer) that is added to the Synchformer features, as illustrated in Figure A3.

G.5. Illustration of the "sum sync with visual" Ablation

Figure A4 illustrates the network architecture for the "sum sync with visual" ablation in the "conditional synchronization module" paragraph. The visual features are upsampled using the nearest neighbor to match the frame rate of the synchronization features. This architecture has a worse FD_{PaSST}, IB-score, synchronization (DeSync) but a better inception score (IS), which we hypothesize is due to the increased number of visual tokens in the upsampling step, leading to finer-grained computations.

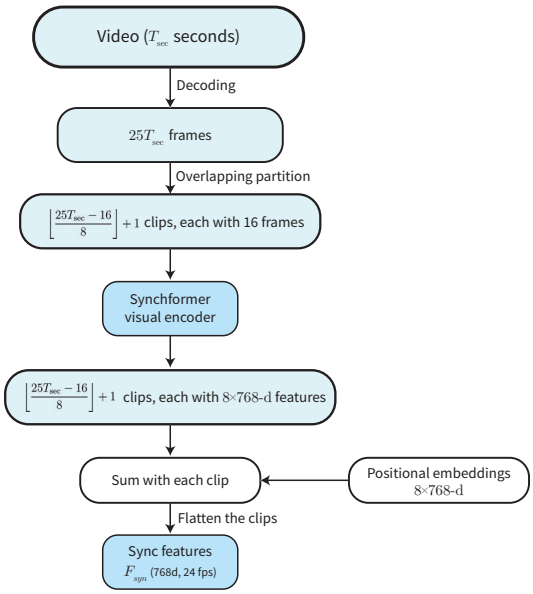

Figure A3. Synchformer feature extraction.

G.6. Visualization of Aligned RoPE

To visualize the effects of using aligned RoPE [59], we compare the dot-product affinity of two sequences $\mathbf{1}^{250\times C}$ and $\mathbf{1}^{64\times C}$ when RoPE is applied. Here, 250 represents the audio sequence length (31.25 fps for 8 seconds), 64 represents the visual sequence length (8 fps for 8 seconds), and $C = 64$ is the channel size. Concretely, we visualize

$$\text{RoPE}_{\text{default}}(\mathbf{1}^{250\times C})\left(\text{RoPE}_{\text{default}}(\mathbf{1}^{64\times C})\right)^T, \tag{A5}$$

and,

$$\text{RoPE}_{\text{aligned}}(\mathbf{1}^{250\times C})\left(\text{RoPE}_{\text{aligned}}(\mathbf{1}^{64\times C})\right)^T, \tag{A6}$$

in Figure A5. Temporal alignment is attained when we use aligned RoPE.

H. Training Details

Training setup. Unless otherwise specified, we used the same set of hyperparameters for all model sizes. To train the models, we use the base learning rate of 1e-4, with a linear warm-up schedule of 1K steps, for 300K iterations, and with a batch size of 512. We use the AdamW optimizer [26, 41] with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and a weight decay of 1e-6. If the default $\beta_2 = 0.999$ was used instead, we notice occasional training collapse (to NaN). For learning rate scheduling, we reduce the learning rate to 1e-5 after 80% of the training steps, and once again to 1e-6 after 90% of the training steps. For model exponential moving average (EMA), we use the post-hoc EMA [22] formulation with a relative width $\sigma_{\text{rel}} = 0.05$ for all models. For training efficiency, we use `bf16` mixed precision training, and all the audio latents and visual embeddings are precomputed offline and loaded during training. Table A7 summarizes the training resources we used for each model size.

Model	Number of GPUs used	Number of hours to train	Total GPU-hours
MMAudio-S-16kHz	2	22	44
MMAudio-S-44.1kHz	2	26	52
MMAudio-M-44.1kHz	8	21	168
MMAudio-L-44.1kHz	8	38	304

Table A7. The amount of training resources used for each model size. H100 GPUs are used in all settings.

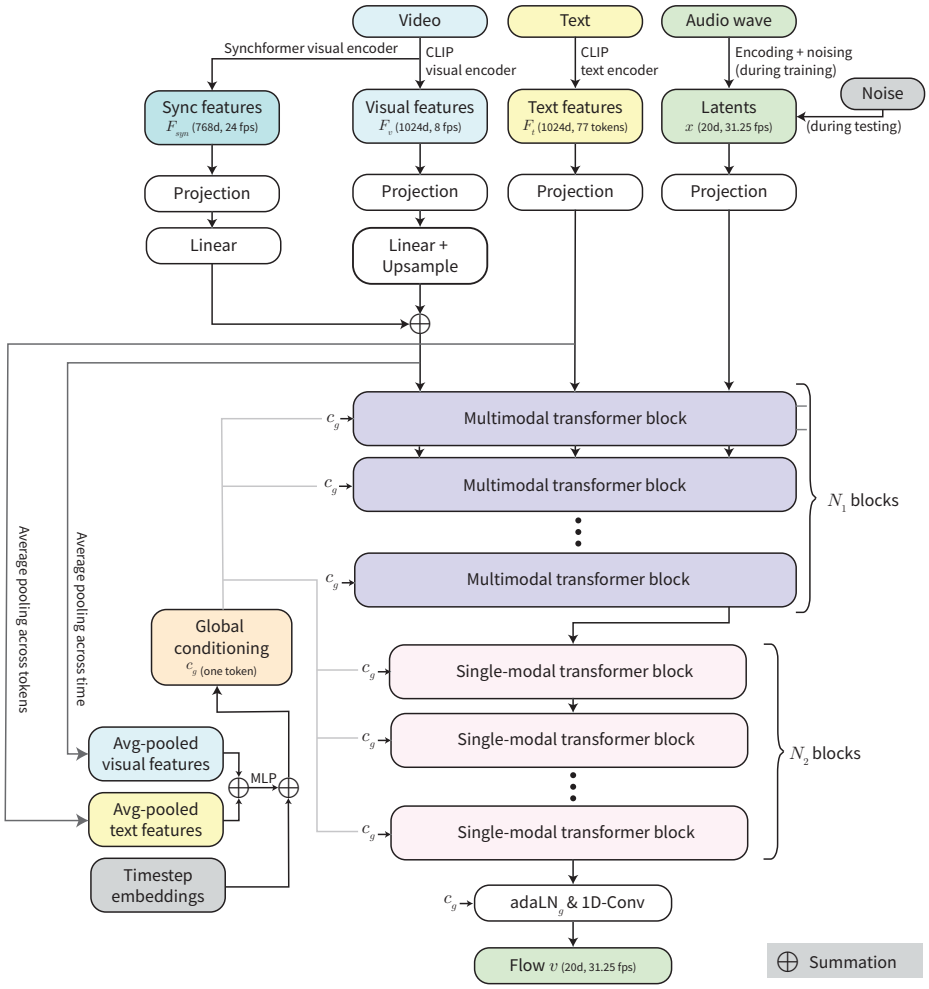

Figure A4. Illustration of the "sum sync with visual" ablation.

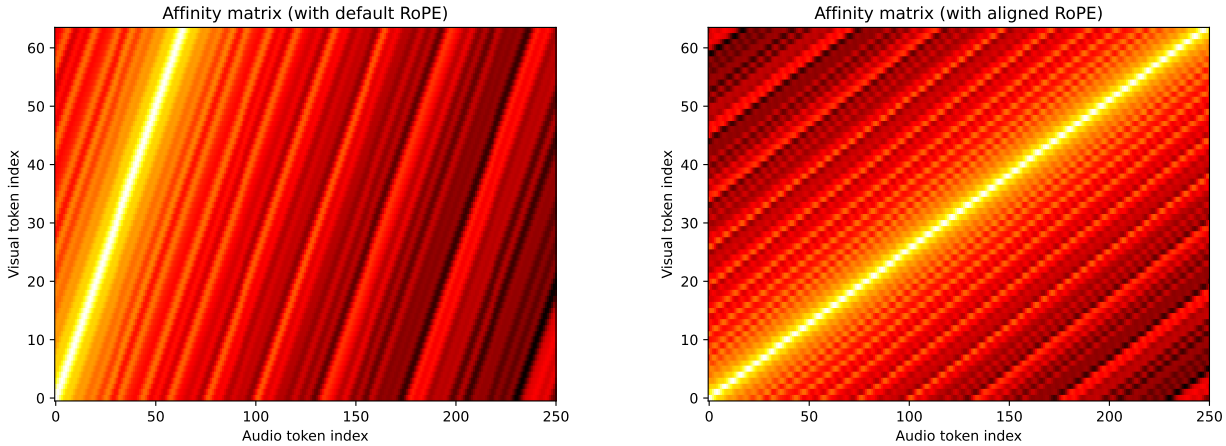

Figure A5. Affinity visualizations between two sequences with different frame rates when default/aligned RoPE embeddings are used. **Left**: with default RoPE, the sequences are not aligned. **Right**: with our proposed aligned RoPE, we attain temporal alignment.

Balancing multimodal training data. Since we have significantly more audio-text training data (951K) than audio-text-visual data (180K), we balance the dataset by duplicating the audio-text-visual samples before random shuffling in each epoch.

By default, we apply a 5X duplication for a rough 1:1 data sampling ratio. For the "medium" and "large" models, we reduce the duplication ratio to 3X to mitigate overfitting.

Duplicated videos. We observe VGGSound dataset [1] contains duplicated videos, likely due to multiple uploads of the same video to YouTube under different video IDs. For instance, videos `4PjEi5fFD6A` (in training set) and `FhaYvI1yrUM` (in test set) are the same video.[3] In Appendix E, we remove train-test sets overlaps by comparing the video IDs, though this method does not eliminate repeated uploads. Since prior works have been trained on the same dataset, our training scheme remains a fair comparison.

I. Additional Visualizations

We provide generated samples and comparisons with state-of-the-art methods on our project page `https://hkchengrex.com/MMAudio/video_main.html`. Below, we provide additional spectrogram visualizations comparing our method with prior works in Figures A6 to A8.

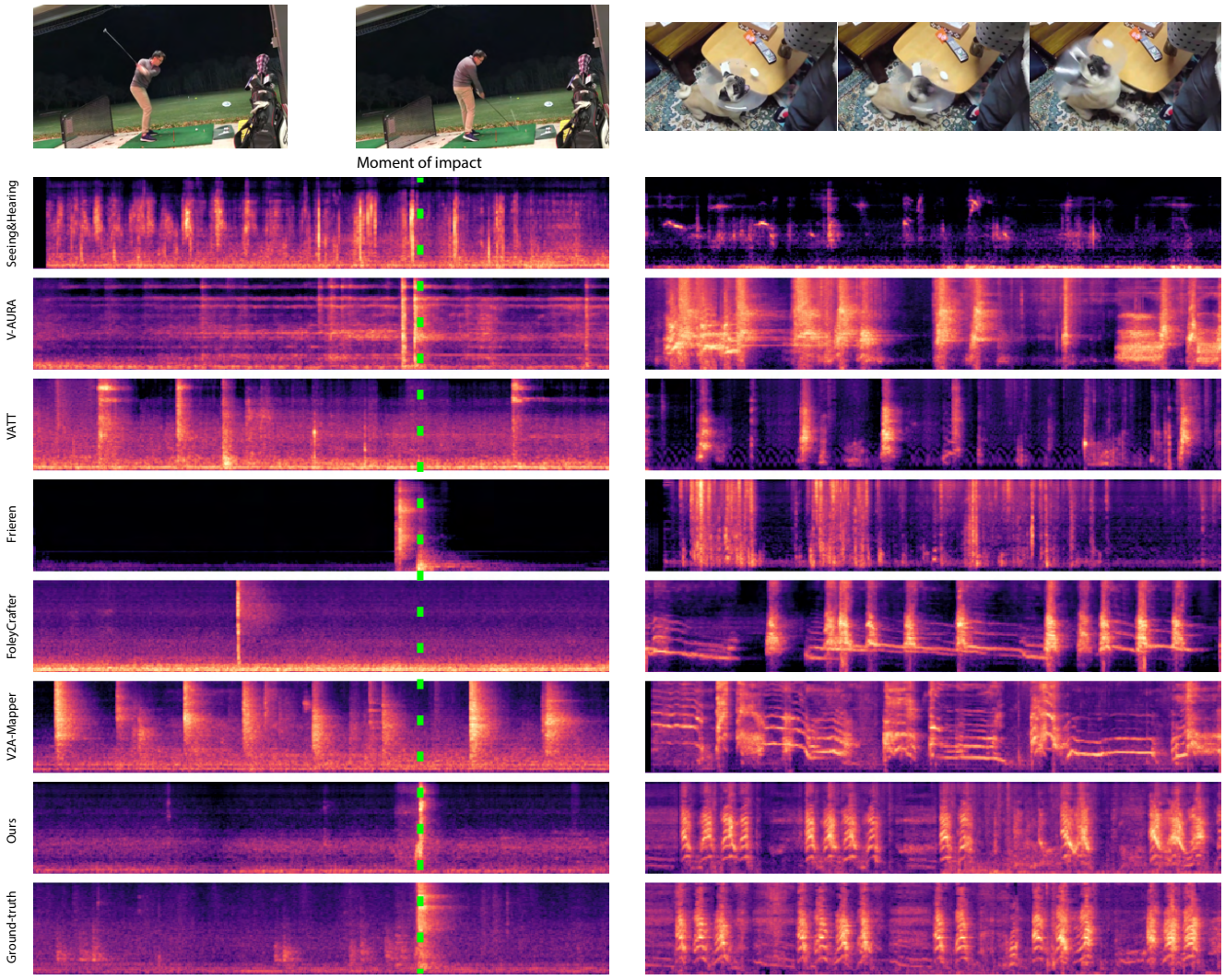

Figure A6. Left: our method can precisely capture the distinct audio event of striking a golf ball. Right: a dog barks in successive bursts. Our generation does not line up with the ground-truth as precisely due to the ambiguous nature of video-to-audio generation, but does capture the rapid bursts.

[3] Other uploads of this video that are *not* part of the VGGSound dataset include `1MQkMdlBezY` and `vHmRikW9axQ`.

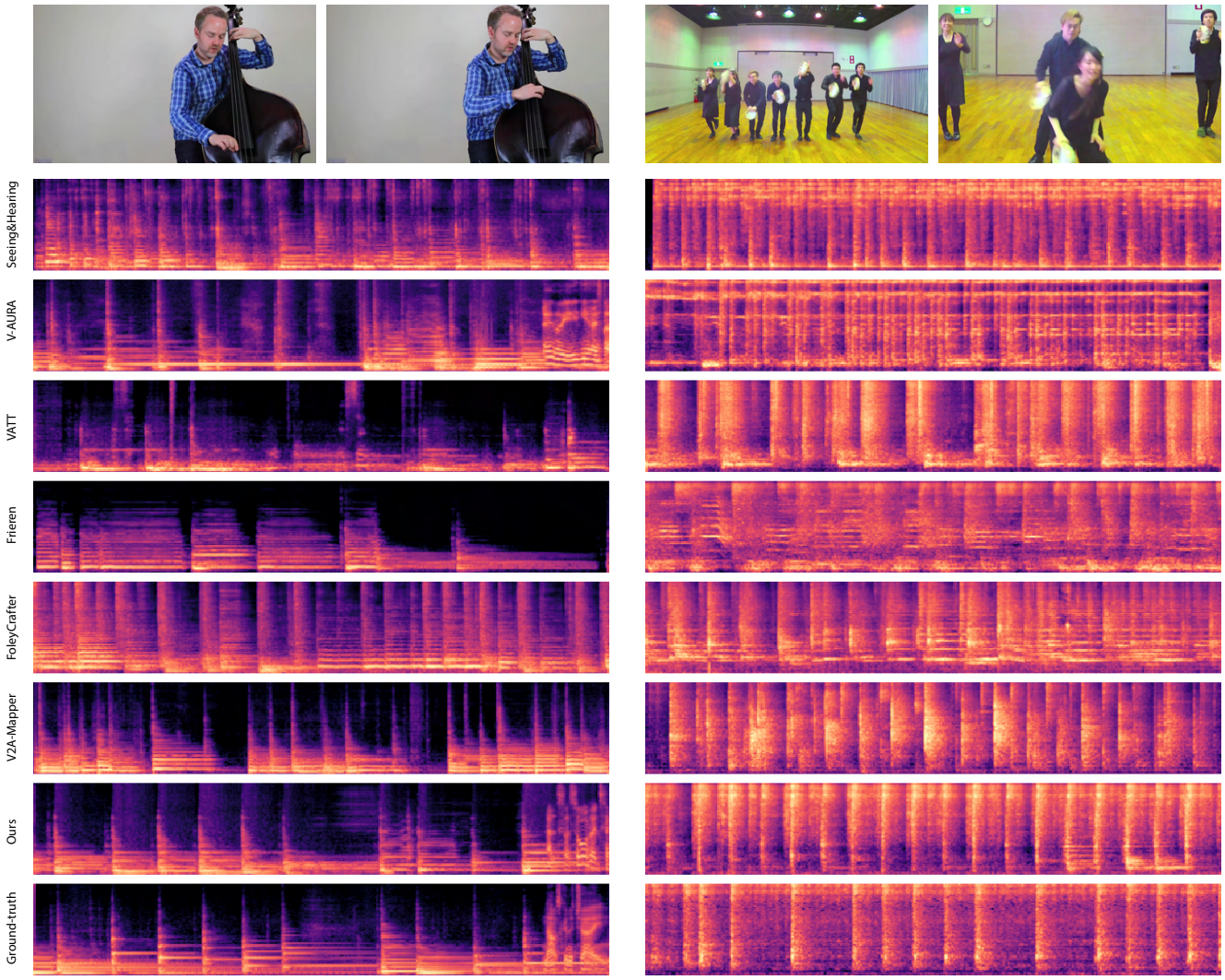

Figure A7. Left: when visible audio events (*e.g.*, when a string is played) can be clearly seen, MMAudio captures them much more precisely than existing methods. Right: in a complex scenario, MMAudio does not always generate audio aligned to the ground-truth (as common in the generative setting) but the generation is often still plausible.

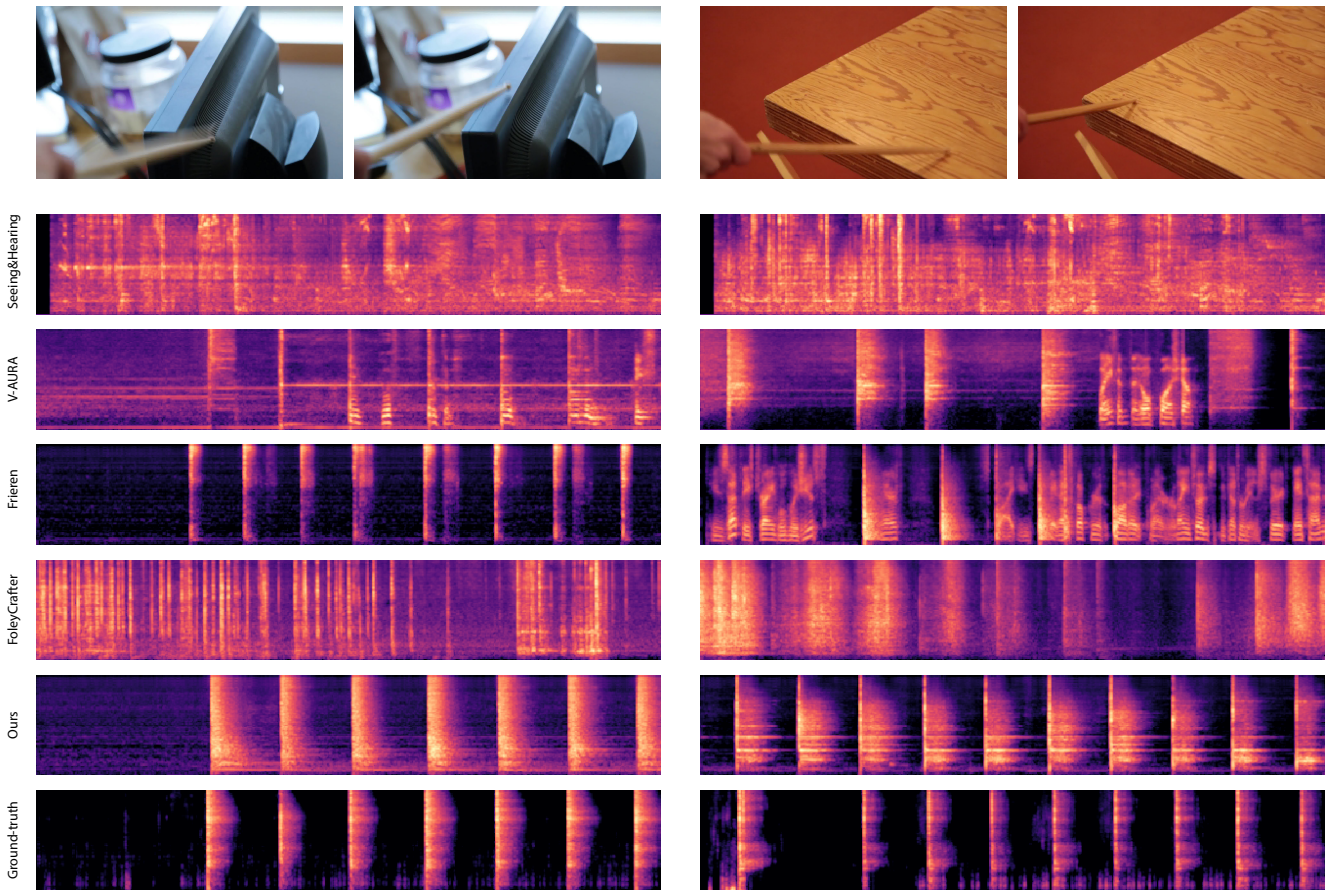

Figure A8. Comparisons of prior works with MMAudio on the Greatest Hits [48] dataset.

References

[1] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *ICASSP*, 2020. 1, 4, 5, 6, 12, 13, 14, 19

[2] Peihao Chen, Yang Zhang, Mingkui Tan, Hongdong Xiao, Deng Huang, and Chuang Gan. Generating visually aligned sound from videos. *TIP*, 2020. 2

[3] Ziyang Chen, Prem Seetharaman, Bryan Russell, Oriol Nieto, David Bourgin, Andrew Owens, and Justin Salamon. Video-guided foley sound generation with multimodal controls. *arXiv*, 2024. 2

[4] Yoonjin Chung, Junwon Lee, and Juhan Nam. T-foley: A controllable waveform-domain diffusion model for temporal-event-guided foley sound synthesis. In *ICASSP*. IEEE, 2024. 2

[5] Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. Clotho: An audio captioning dataset. In *ICASSP*. IEEE, 2020. 5, 14

[6] Yuexi Du, Ziyang Chen, Justin Salamon, Bryan Russell, and Andrew Owens. Conditional generation of audio from video via foley analogies. In *CVPR*, 2023. 2, 13

[7] Benjamin Elizalde, Soham Deshmukh, and Huaming Wang. Natural language supervision for general-purpose audio representations. In *ICASSP*, 2024. 12

[8] 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 *ICML*, 2024. 1, 3, 4

[9] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *ICASSP*. IEEE, 2017. 1, 5, 14, 15

[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. 7

[11] 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 *CVPR*, 2023. 1, 2, 5, 12

[12] Moayed Haji-Ali, Willi Menapace, Aliaksandr Siarohin, Guha Balakrishnan, Sergey Tulyakov, and Vicente Ordonez. Taming data and transformers for audio generation. *arXiv preprint arXiv:2406.19388*, 2024. 6, 7

[13] Moayed Haji-Ali, Willi Menapace, Aliaksandr Siarohin, Ivan Skorokhodov, Alper Canberk, Kwot Sin Lee, Vicente Ordonez, and Sergey Tulyakov. Av-link: Temporally-aligned diffusion features for cross-modal audio-video generation. *arXiv*, 2024. 2

[14] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 5

[15] Jiawei Huang, Yi Ren, Rongjie Huang, Dongchao Yang, Zhenhui Ye, Chen Zhang, Jinglin Liu, Xiang Yin, Zejun Ma, and Zhou Zhao. Make-an-audio 2: Temporal-enhanced text-to-audio generation. *arXiv preprint arXiv:2305.18474*, 2023. 7, 15

[16] 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*, 2023. 7

[17] Zhiqi Huang, Dan Luo, Jun Wang, Huan Liao, Zhiheng Li, and Zhiyong Wu. Rhythmic foley: A framework for seamless audio-visual alignment in video-to-audio synthesis. *arXiv preprint arXiv:2409.08628*, 2024. 2

[18] Vladimir Iashin and Esa Rahtu. Taming visually guided sound generation. In *BMVC*, 2021. 2, 5

[19] Vladimir Iashin, Weidi Xie, Esa Rahtu, and Andrew Zisserman. Synchformer: Efficient synchronization from sparse cues. In *ICASSP*. IEEE, 2024. 1, 2, 4, 5, 12, 16

[20] Yujin Jeong, Yunji Kim, Sanghyuk Chun, and Jiyoung Lee. Read, watch and scream! sound generation from text and video. *arXiv preprint arXiv:2407.05551*, 2024. 1, 2, 4, 5, 6

[21] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 1

[22] Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *CVPR*, 2024. 15, 17

[23] Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fréchet audio distance: A metric for evaluating music enhancement algorithms. In *Interspeech*, 2018. 1

[24] Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. Audiocaps: Generating captions for audios in the wild. In *NAACL-HLT*, 2019. 5, 6, 14

[25] Gwanghyun Kim, Alonso Martinez, Yu-Chuan Su, Brendan Jou, José Lezama, Agrim Gupta, Lijun Yu, Lu Jiang, Aren Jansen, Jacob Walker, et al. A versatile diffusion transformer with mixture of noise levels for audiovisual generation. In *NeurIPS*, 2024. 2

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 17

[27] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 3, 15

[28] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. *NeurIPS*, 2017. 16

[29] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D Plumbley. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *TASLP*, 2020. 5

[30] Khaled Koutini, Jan Schlüter, Hamid Eghbal-Zadeh, and Gerhard Widmer. Efficient training of audio transformers with patchout. In *Interspeech*, 2022. 5

[31] Etienne Labb, Thomas Pellegrini, and Julien Pinquier. Conette: An efficient audio captioning system leveraging multiple datasets with task embedding. *TASLP*, 2024. 14

[32] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024. 3

[33] LAION-AI. Audio dataset project. https://github.com/LAION-AI/audio-dataset, 2024. 14, 15

[34] Junwon Lee, Jaekwon Im, Dabin Kim, and Juhan Nam. Video-foley: Two-stage video-to-sound generation via temporal event condition for foley sound. *arXiv preprint arXiv:2408.11915*, 2024. 2

[35] Sang-gil Lee, Wei Ping, Boris Ginsburg, Bryan Catanzaro, and Sungroh Yoon. Bigvgan: A universal neural vocoder with large-scale training. In *ICLR*, 2023. 3, 15

[36] Rensis Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 1932. 12

[37] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 2

[38] 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*, 2023. 3, 5, 6

[39] 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. *TASLP*, 2024. 7

[40] Xiulong Liu, Kun Su, and Eli Shlizerman. Tell what you hear from what you see – video to audio generation through text. In *NeurIPS*, 2024. 2, 4, 5, 6, 12

[41] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *ICLR*, 2019. 17

[42] Simian Luo, Chuanhao Yan, Chenxu Hu, and Hang Zhao. Diff-foley: Synchronized video-to-audio synthesis with latent diffusion models. In *NeurIPS*, 2024. 2

[43] Navonil Majumder, Chia-Yu Hung, Deepanway Ghosal, Wei-Ning Hsu, Rada Mihalcea, and Soujanya Poria. Tango 2: Aligning diffusion-based text-to-audio generations through direct preference optimization. In *ACM MM*, 2024. 7

[44] Xinhao Mei, Varun Nagaraja, Gael Le Lan, Zhaoheng Ni, Ernie Chang, Yangyang Shi, and Vikas Chandra. Foleygen: Visually-guided audio generation. *arXiv preprint arXiv:2309.10537*, 2023. 2

[45] Xinhao Mei, Gael Le Lan, Haohe Liu, Zhaoheng Ni, Varun K Nagaraja, Anurag Kumar, Yangyang Shi, and Vikas Chandra. Towards temporally synchronized visually indicated sounds through scale-adapted positional embeddings. In *Audio Imagination: NeurIPS 2024 Workshop AI-Driven Speech, Music, and Sound Generation*, 2024. 2

[46] 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. *TASLP*, 2024. 1, 5, 14

[47] Shentong Mo, Jing Shi, and Yapeng Tian. Text-to-audio generation synchronized with videos. *arXiv preprint arXiv:2403.07938*, 2024. 2

[48] Andrew Owens, Phillip Isola, Josh McDermott, Antonio Torralba, Edward H Adelson, and William T Freeman. Visually indicated sounds. In *CVPR*, 2016. 6, 13, 21

[49] Santiago Pascual, Chunghsin Yeh, Ioannis Tsiamas, and Joan Serrà. Masked generative video-to-audio transformers with enhanced synchronicity. In *ECCV*, 2024. 2

[50] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018. 1, 4

[51] Karin Petrini, Sofia Dahl, Davide Rocchesso, Carl Haakon Waadeland, Federico Avanzini, Aina Puce, and Frank E Pollick. Multisensory integration of drumming actions: musical expertise affects perceived audiovisual asynchrony. *Experimental brain research*, 198: 339–352, 2009. 1

[52] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024. 6, 12, 13

[53] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICLR*, 2021. 1, 4

[54] Yong Ren, Chenxing Li, Manjie Xu, Wei Liang, Yu Gu, Rilin Chen, and Dong Yu. Sta-v2a: Video-to-audio generation with semantic and temporal alignment. *arXiv preprint arXiv:2409.08601*, 2024. 2

[55] Ludan Ruan, Y Ma, Huan Yang, Huiguo He, Bei Liu, Jianlong Fu, Nicholas Jing Yuan, Qin Jin, and Baining Guo. Mm-diffusion: Learning multi-modal diffusion models for joint audio and video generation. 2023 ieee. In *CVPR*, 2023. 2

[56] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NeurIPS*, 2016. 1, 5

[57] Stanley Smith Stevens, John Volkmann, and Edwin Broomell Newman. A scale for the measurement of the psychological magnitude pitch. *The journal of the acoustical society of america*, 1937. 3, 15

[58] Robynn J Stilwell. The fantastical gap between diegetic and nondiegetic, 2007. 1

[59] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 2024. 4, 8, 17

[60] Kun Su, Kaizhi Qian, Eli Shlizerman, Antonio Torralba, and Chuang Gan. Physics-driven diffusion models for impact sound synthesis from videos. In *CVPR*, 2023. 2

[61] Zineng Tang, Ziyi Yang, Mahmoud Khademi, Yang Liu, Chenguang Zhu, and Mohit Bansal. Codi-2: In-context interleaved and interactive any-to-any generation. In *CVPR*, 2024. 2

[62] Zineng Tang, Ziyi Yang, Chenguang Zhu, Michael Zeng, and Mohit Bansal. Any-to-any generation via composable diffusion. In *NeurIPS*, 2024. 2

[63] Alexander Tong, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Kilian Fatras, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023. 2

[64] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 4

[65] Ilpo Viertola, Vladimir Iashin, and Esa Rahtu. Temporally aligned audio for video with autoregression. *arXiv preprint arXiv:2409.13689*, 2024. 2, 5, 6, 12

[66] 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 *AAAI*, 2024. 1, 2, 5, 6, 12

[67] Yongqi Wang, Wenxiang Guo, Rongjie Huang, Jiawei Huang, Zehan Wang, Fuming You, Ruiqi Li, and Zhou Zhao. Frieren: Efficient video-to-audio generation with rectified flow matching. In *NeurIPS*, 2024. 1, 3, 5, 6, 15

[68] Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *ICASSP*, 2023. 6, 12

[69] Yazhou Xing, Yingqing He, Zeyue Tian, Xintao Wang, and Qifeng Chen. Seeing and hearing: Open-domain visual-audio generation with diffusion latent aligners. In *CVPR*, 2024. 1, 2, 5, 6, 12

[70] Jinlong Xue, Yayue Deng, Yingming Gao, and Ya Li. Auffusion: Leveraging the power of diffusion and large language models for text-to-audio generation. *arXiv preprint arXiv:2401.01044*, 2024. 14

[71] Qi Yang, Binjie Mao, Zili Wang, Xing Nie, Pengfei Gao, Ying Guo, Cheng Zhen, Pengfei Yan, and Shiming Xiang. Draw an audio: Leveraging multi-instruction for video-to-audio synthesis. *arXiv preprint arXiv:2409.06135*, 2024. 2

[72] Lin Zhang, Shentong Mo, Yijing Zhang, and Pedro Morgado. Audio-synchronized visual animation. *ECCV*, 2024. 14

[73] Yiming Zhang, Yicheng Gu, Yanhong Zeng, Zhening Xing, Yuancheng Wang, Zhizheng Wu, and Kai Chen. Foleycrafter: Bring silent videos to life with lifelike and synchronized sounds. *arXiv preprint arXiv:2407.01494*, 2024. 1, 2, 5, 6, 14

[74] Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiaxi Cui, HongFa Wang, Yatian Pang, Wenhao Jiang, Junwu Zhang, Zongwei Li, et al. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment. In *ICLR*, 2024. 2