Hear What Matters! Text-conditioned Selective Video-to-Audio Generation

Appendix

A. More Implementation Details

A.1. Video encoder

CLIP vs. Synchformer. We first analyze the role of each vision feature (*i.e.*, CLIP [54] and Synchformer [31]) used in MMAudio [9], which is also used in parameter initialization of our method. Understanding how each embedding contributes to generation quality is crucial for determining how to adapt the video encoder for text conditioning. Tab. A1 reports an ablation study by substituting each feature with its corresponding learned null embedding trained for classifier-free guidance. Interestingly, removing the CLIP [54] embedding actually improves the CLAP score by 0.038 while the DeSync score remains steady. This suggests that the CLIP embedding often introduces semantic distraction without conveying significant temporal information. In contrast, the IB and DeSync scores deteriorate significantly when the Synchformer embedding is removed. This shows that the Synchformer feature contributes both semantic and temporal information for reliable audio-video alignment.

Input configuration. Synchformer, consisting of audio encoder and video encoder, is learned to predict the temporal offset to evaluate audiovisual synchronization. In this experiment, we only use the video encoder for feature extraction, following the details of MMAudio. Note that the architecture of the video encoder follows the Motionformer with divided space-time attention [4, 51]. Given an input video of 8 seconds at 25 fps, we first divide it into segments with windowing (window size of 16, hop size of 8 frames), which results in 24 segments. Here, a batched video data has a shape [Batch, Segments, Channel, Height, Width] by resizing 224×224 resolution without center crop. Each video frame is patchified and flattened in rasterized order. After passing the video encoder, each segment results in 8 embeddings in the temporal axis with a hidden dimension of $D = 768$. The final video feature $\mathbf{v}$ of a minibatch has a shape of [Batch, Segments, $t = 8$, $D = 768$].

A.2. Text encoder

To extract text embeddings, we use Flan-T5-Base [10] [1] to condition the video encoder. For the audio generator, to reuse pretrained parameters from MMAudio, CLIP's text encoder [54] is employed.

A.3. Training

In the first training stage, we finetune the Synchformer [31] video encoder to condition text prompts. We use the pre-trained checkpoint `24-01-04T16-39-21` from the official implementation [2], trained on AudioSet [21] using a two-stage process consisting of audio-visual contrastive learning and offset estimation. Specifically, we train the initialized spatial attention pooling layer and a new trainable text cross-attention block, which is placed after the space-

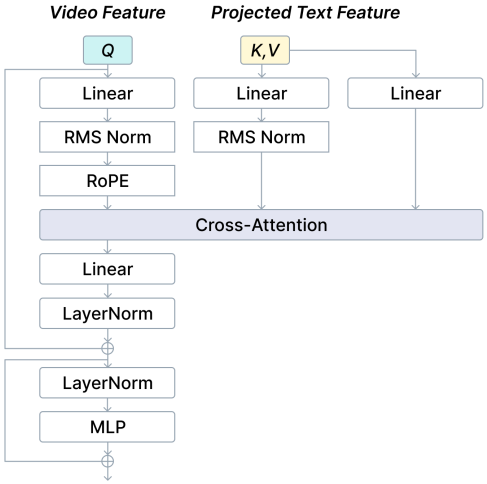


Figure A1. Detailed architecture of cross-attention used in student video encoder.

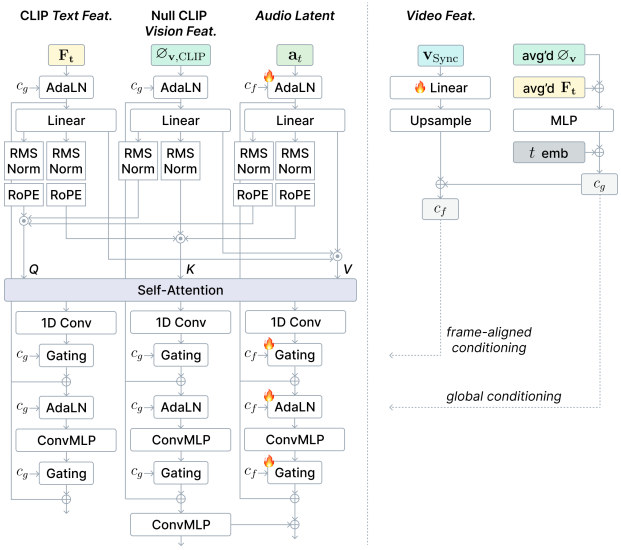


Figure A2. Detailed architecture of Multimodal transformer block in MM-DiT(Multi Modal Diffusion Transformer).

[1] https://huggingface.co/google/flan-t5-base

[2] https://github.com/v-iashin/Synchformer

Model	Audio Quality			Semantic Alignment			Temporal Alignment
	FAD↓	KAD↓	IS↑	KL↓	CLAP↑	IB↑	DeSync↓
MMAudio-S-16kHz [9]	**5.15**	**0.260**	14.53	**1.64**	0.197	**0.2927**	**0.486**
w/ null CLIP emb.	7.85	0.338	**18.95**	1.75	**0.235**	0.2670	0.492
w/ null Synchformer emb.	7.51	0.563	14.27	2.00	0.196	0.2394	1.243

Table A1. Performance of MMAudio [9] on VGGSound test set with different input visual feature combinations.

time attention blocks (see Fig. A1 for details). This process ensures parameter-efficient finetuning, updating only 14% (19M) of the 135M total parameters. We train for 50k steps with a batch size of 4 on a single NVIDIA RTX 4090. The base learning rate is set to 1e-4, with a 1k-step linear warmup schedule.

In the second training stage, we train the multimodal-conditioned audio generator. To efficiently train the large-scale generator, we take the initial parameters from the MMAudio-small-16k model [9]. Therefore, the architecture of the generator in **SELVA** is similar to MMAudio. Only 14% (22M) of the 157M total parameters are trained in our second stage. Concretely, we finetune the initial projection layer for the Synchformer video feature $\mathbf{v}_{\text{Sync}}$ and all adaLN-related layers that receive the video feature as input. Fig. A2 specifies those learnable layers within a single MM-DiT [18, 40] block of MMAudio. It is worth noting that a frame-aligned conditioning c_f is a function of the Synchformer video feature, while the global conditioning c_g is not. While MMAudio originally used the CLIP image feature $\mathbf{v}_{\text{CLIP}}$, we do not use this for conditioning by replacing with the null feature $\varnothing_{\mathbf{v}}$. We train for 25k steps with a batch size of 12 on a single NVIDIA RTX A6000. The base learning rate is set to 1e-5, also with a 1k-step linear warmup.

Common to both training stages, we utilize `bf16` automatic mixed precision (AMP). We employ the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and a weight decay of 1e-6, along with gradient clipping at a norm of 1. After training, we apply post-hoc EMA [33] with a relative width of $\sigma_{\text{rel}} = 0.05$.

B. VGG-MONOAUDIO

This section details the benchmark construction process and its resulting statistics.

B.1. Data collection

As mentioned in Section 4.1, we acquired 67 clean, mono-source videos through automatic filtering and manual curation. These videos cover 39 unique text labels spanning 8 sound categories in Tab. B2. Fig. B3 summarizes the category-wise statistics. To obtain clean, mono-source audio-video-text samples, we begin with the audio-visual event annotations from UnAV-100 [22], a dataset containing timestamped text labels for 100 sound categories across 10,000 videos. First, we identified 907 videos that are common to both UnAV-100 and the VGGSound testset. An automatic filtering step then removed clips annotated with more than one unique sound event. Subsequently, we performed a manual verification process, retaining clips that met three strict criteria: (1) the video contains a single audible sound source with minimal background noise or off-screen sound; (2) the sounding object is clearly visible; and (3) the text annotation precisely matches the sound event. When text annotations from UnAV-100 and VGGSound differed, we manually selected the more appropriate one.

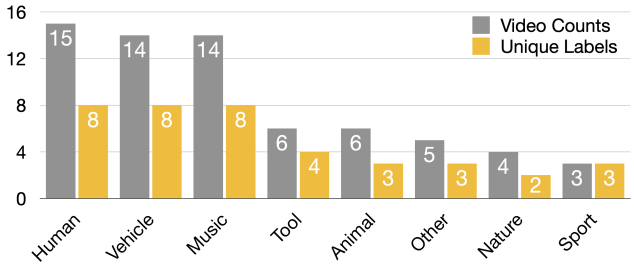
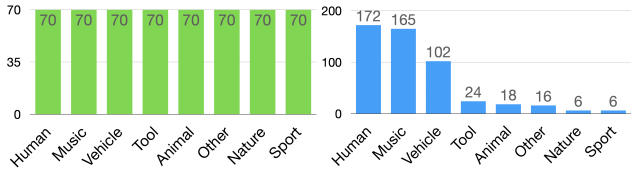


Figure B3. Statistics on single-source videos.



(a) *Inter-class* (b) *Intra-class*

Figure B4. Statistics on VGG-MONOAUDIO.

B.2. Statistic

An exhaustive pairing of these 67 source videos yields a total pool of 3,750 potential inter-class pairs and 560 potential intra-class pairs. First, to construct the *Inter-class* VGG-MONOAUDIO, we sample 560 pairs from the 3,750 available, ensuring balance by target sound category as shown in Fig. B4a. We ensure a balanced selection of sample pairs across sound categories. When a category does not contain enough valid pairs, we fill the remaining slots by randomly sampling from the available pairs in that category. This results in the final test set of 560 inter-class pairs.

Human	
baby crying	baby laughter
child singing	male singing
people burping	people sneezing
people whispering	baby babbling
Vehicle	
car passing by	driving buses
driving motorcycle	engine knocking
fire truck siren	police car siren
train wheels squealing	airplane flyby
Music	
playing acoustic guitar	playing banjo
playing cello	playing electric guitar
playing harmonica	playing harp
playing zither	playing accordion
Tool	
lawn mowing	typing on computer keyboard
vacuum cleaner cleaning floors	chainsawing trees
Animal	
dog barking	sheep bleating
bird chirping	
Nature	
waterfall burbling	underwater bubbling
Sport	
rope skipping	skateboarding
basketball bounce	
Other	
firework banging	machine gun shooting
church bell ringing	

Table B2. List of 39 unique text labels in VGG-MONOAUDIO.

For the *Intra-class* VGG-MONOAUDIO, we manually filter the initial 560 candidate pairs to prevent semantic overlap. This step removes pairs where the target text prompt semantically subsumes the paired video's prompt. For instance, a pair with the target *'people whispering'* and the non-target *'baby mumbling'* would be removed, as the target label could also refer to the non-target video. This curation process results in a final set of 511 intra-class pairs, as summarized in Fig. B4b.

B.3. Pre-processing

The target frame is randomly placed on either the left or right side of the video. All videos are processed to a 1280×720 resolution, with video encoded using the H.264 codec and audio using the AAC codec. Each video clip is 8 seconds long, with a 25 fps and an audio sample rate of 16kHz.

C. Detailed Evaluation Setup

C.1. Baseline models

ReWaS [32]. ReWaS leverages a pretrained text-to-audio (TTA) model as its generator for text-conditioned V2A. The model first predicts the audio's energy curve from the input video and uses this curve as a condition for the TTA model. Since ReWaS natively generates 5-second audio, we adapt it for 8-second videos by splitting each video into two overlapping 5-second segments (0-5s and 3-8s). We generate audio for each segment independently and then construct the final 8-second track by merging the first 4 seconds of the first clip (0-4s) with the last 4 seconds of the second clip (4-8s). We use the official implementation[3] with default parameters.

VinTAGe [39]. VinTAGe is also a text-conditioned V2A model that aims to generate both on-screen and off-screen sounds that are semantically consistent with the text and video. As the model generates 10-second audio, we take the first 8 seconds for evaluation. We use the official code[4] and default parameters for ODE sampling during inference.

VOS+MMAudio. To implement segmentation-based models [45, 60] in our experimental setup, we employ the SoTA video object segmentation model, DEVA [8], and multimodal-conditioned audio generator model, MMAudio [9]. Similar to **SELVA**, this VOS-based pipeline takes a video and a text prompt as condition inputs to improve user controllability. DEVA first predicts a segmentation mask for each frame based on the text prompt. Pixels outside this predicted mask are zeroed out to form a masked video. Therefore, ideally, only the text-related target object is visible. This masked video is subsequently fed into MMAudio, along with the original text prompt, to generate the corresponding audio. We used the official DEVA implementation [5] with its default hyperparameters, which include leveraging SAM [36] for segmentation, applying semi-online temporal fusion of segmentation hypotheses, and disabling video resizing.

C.2. Metrics

To assess overall audio quality, we adopt three different metrics. Fréchet Audio Distance (FAD) [34] measures the Fréchet distance between Gaussian distributions fitted to audio embeddings from a reference set and a generated set. Kernel Audio Distance (KAD) [12], proposed as an unbiased and distribution-free alternative to FAD, also measures this set-wise embedding distance using the Maximum Mean Discrepancy (MMD) with a Gaussian RBF kernel. Inception Score (IS) [56] evaluates both the quality and diversity of generated samples by calculating the KL divergence between the conditional label distribution for individual samples and the marginal distribution across all samples.

For semantic alignment, we report KL divergence, CLAP [65], and ImageBind [23] scores. The Kullback-Leibler divergence (KL) measures audio semantic similarity using the audio classification distributions of the generated and ground-truth audio. CLAP and IB scores capture

[3] https://github.com/naver-ai/rewas

[4] https://github.com/sakshamsingh1/vintage_aud_gen

[5] https://github.com/hkchengrex/Tracking-Anything-with-DEVA

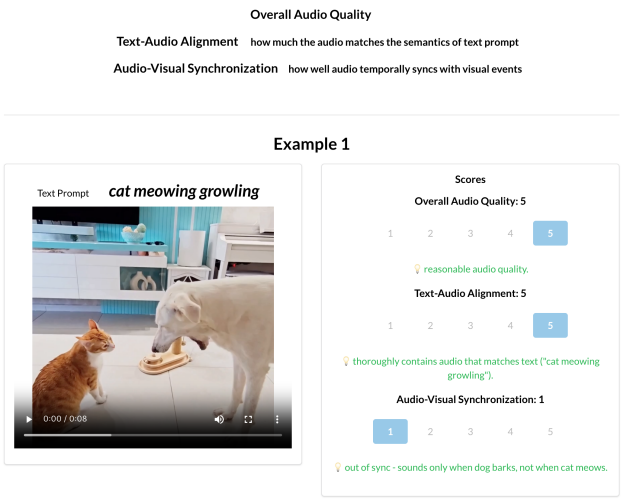


Figure C5. Tutorial example for human study to guide participants in rating audio-video-text pairs along with audio quality, text–audio alignment, and audio–video temporal alignment.

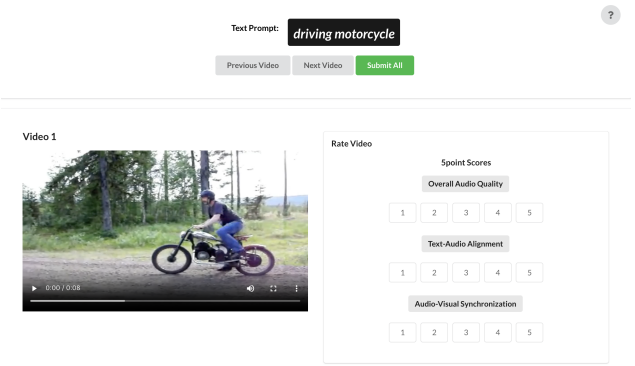


Figure C6. Web interface for human study, where participants rated audio–video-text pairs on three criteria.

global semantic similarity using cosine distance between text-audio and video-audio pairs, respectively.

Finally, to assess audio-video temporal alignment, we report the DeSync score [9], which is defined as the average predicted offset (in seconds) between the audio and video predicted by a pretrained Synchformer [31].

We use a pretrained PANNs [37] model to extract audio embeddings for FAD, KAD, IS, and KL, as the model's features have shown a high correlation with human perception of audio quality [12, 58]. All metrics were calculated using open-source toolkits, including `av-benchmark` [6] and `kadtk` [7].

[6] https://github.com/hkchengrex/av-benchmark
[7] https://github.com/YoonjinXD/kadtk

C.3. Human study

In Sec. 4.3, we conducted a human study to evaluate different text-conditioned V2A models. We provided a tutorial for each criterion (*i.e.*, AQ, TA, VA) with 4 examples, as shown in Fig. C5. After watching each video clip, the participants were asked to score each criterion on a 5-point Likert scale, as shown in Fig. C6.

C.4. Attention visualization

Fig. 3 visualizes the attention scores associated with the `[eos]` text token embedding by combining two attention maps: the text-guided cross-attention (with `[eos]` as the key), and the spatial-pooling map. Both are averaged over their respective attention heads, and multiplied element-wise. This final visualization reveals how much the text semantics contributed to the video feature at a specific time frame.

D. Additional Results

D.1. The number of `[SUP]` tokens

Tab. D3 shows the ablation result of all objective metrics on different numbers of `[SUP]` tokens.

D.2. VGGSound test set

Tab. D4 summarizes the performance of state-of-the-art text-conditioned V2A models on the VGGSound [5] test set. **SELVA** achieves results comparable to MMAudio, showing improved semantic alignment but slightly lower temporal alignment. This stems from the nature of VGGSound original test set, which is not curated for selective sound generation and often contains text-irrelevant sound events in videos. Consequently, DeSync may favor holistic generation models (*i.e.*, MMAudio) that reproduce these extraneous sounds over selective generation methods (*i.e.*, **SELVA**). ReWaS [32] and VinTAGe [39] underperform in all aspects, particularly in temporal alignment. This is likely because they rely heavily on the text modality: ReWaS leverages a pretrained text-to-audio model, while VinTAGe is trained to generate both on-screen and off-screen sounds based on text descriptions. Additionally, we observe that FAD follows the trend of KAD in Tab. D4 (dataset size: 15k), unlike in Tab. 1 (dataset size: 0.5k). This discrepancy arises because FAD is a biased estimator sensitive to sample size.

D.3. Scaling up the model

To analyze scalability, we evaluate larger versions of our audio generator $\mathcal{G}$ in Tab. D5. Following the model configurations of MMAudio [9], the parameter counts for our small, medium, and large variants are 157M, 621M, and 1.03B, respectively. While generating audio at a higher sample

# of [SUP]	Audio Quality			Semantic Alignment			Temporal Alignment
	FAD↓	KAD↓	IS↑	KL↓	CLAP↑	IB↑	DeSync↓
Inter-class							
0	51.4	**0.637**	12.95	**1.79**	0.289	0.3272	0.756
1	**51.2**	0.655	12.97	1.82	0.290	0.3263	0.760
3	**51.2**	0.638	12.94	1.81	**0.292**	**0.3289**	0.745
SELVA/ 5	51.7	0.676	**13.07**	1.85	**0.292**	0.3251	**0.721**
7	51.9	0.659	13.02	1.84	0.289	0.3233	0.759
Intra-class							
0	**36.3**	0.485	9.74	**1.01**	0.281	0.3277	0.676
1	37.4	0.510	**9.78**	1.03	**0.284**	0.3296	0.675
3	36.5	**0.474**	9.72	1.03	0.282	0.3255	0.683
SELVA/ 5	37.0	0.492	9.62	1.04	0.280	0.3262	**0.639**
7	37.2	0.495	9.65	1.03	0.280	**0.3300**	0.675

Table D3. Ablation on the number of [SUP] tokens. Since DeSync has been dramatically changed in this ablation, we adopt five tokens as the default configuration.

Model	Audio Quality			Semantic Alignment			Temporal Alignment
	FAD↓	KAD↓	IS↑	KL↓	CLAP↑	IB↑	DeSync↓
ReWaS [32]	19.96	1.626	7.66	2.42	0.182	0.1825	1.275
VinTAGe [39]	15.96	1.185	8.30	4.91	0.217	0.0486	1.263
MMAudio-S-16k [9]	**7.85**	**0.338**	18.95	**1.75**	0.235	0.2670	**0.492**
SELVA	8.30	0.365	**21.09**	1.76	**0.243**	**0.2688**	0.541

Table D4. Performance of state-of-the-art models on VGGSound [5] test set. Even though **SELVA** outperforms those methods on VGG-MONOAUDIO, **SELVA** still shows comparable performance on noisy VGGsound test samples.

(a) *"Cat Meowing"* (b) *"Dog Barking"*

Figure D7. Attention visualization for [eos] token over real-world video frame in the last block without (left) / with (right) [SUP] tokens. Each subcaption denotes the corresponding target prompt.

rate necessitates more parameters, scaling the model generally improves overall performance. Note that the results of **SELVA**-S-16k are reported for all other experiments. The limited performance gain of **SELVA**-L-44k likely stems from the restricted VGGSound; scaling to larger datasets could better exploit its capacity.

D.4. More qualitative examples

Fig. D7 additionally provides an attention visualization of the [eos] token on a real-world video. By introducing our supplementary token [SUP], our model focuses more effectively on the target sound source while suppressing attention toward the paired object.

Fig. D8 illustrates the mel-spectrograms of audios inferred by different models, alongside their corresponding video frames. The white dotted curve indicates the root-mean-squared (RMS) audio amplitude.

Fig. D8a highlights the superior selective performance of **SELVA**. Given the target "dog barking", MMAudio erroneously generates both the barking and a train squealing sound, with the latter correlating with the paired (non-target) video. The VOS baseline fails to capture the last barking event. In contrast, **SELVA** faithfully generates only the dog barking sound, well-synchronized with the target video.

Example in Fig. D8b demonstrates the temporal synchronization capability of **SELVA**. MMAudio again fails at selection, generating undesired male speech that stems from the paired video on the right. The VOS baseline, while correctly generating the bus sound, fails to capture its temporal dynamics (e.g., the volume change of the bus approaching and passing). We hypothesize this is due to the vision encoder's deteriorated capability; by removing the background, it loses crucial contextual information, such as the bus's size change relative to the stationary background, which implies its motion. **SELVA** successfully captures these temporal dynamics while selectively generating the

Model	Audio Quality			Semantic Alignment			Temporal Alignment
	FAD↓	KAD↓	IS↑	KL↓	CLAP↑	IB↑	DeSync↓
Inter-class							
SELVA-S-16k	51.7	0.676	13.07	1.85	0.292	0.3251	0.721
SELVA-S-44k	52.7	0.561	13.93	1.82	0.305	0.3490	0.739
SELVA-M-44k	53.9	0.548	14.53	1.69	0.315	0.3586	0.695
SELVA-L-44k	52.6	0.595	14.28	1.76	0.305	0.3517	0.691
Intra-class							
SELVA-S-16k	37.0	0.492	9.62	1.04	0.280	0.3262	0.639
SELVA-S-44k	38.3	0.340	10.22	1.12	0.297	0.3457	0.695
SELVA-M-44k	39.0	0.360	10.34	1.05	0.295	0.3472	0.646
SELVA-L-44k	38.5	0.346	10.31	1.13	0.294	0.3436	0.659

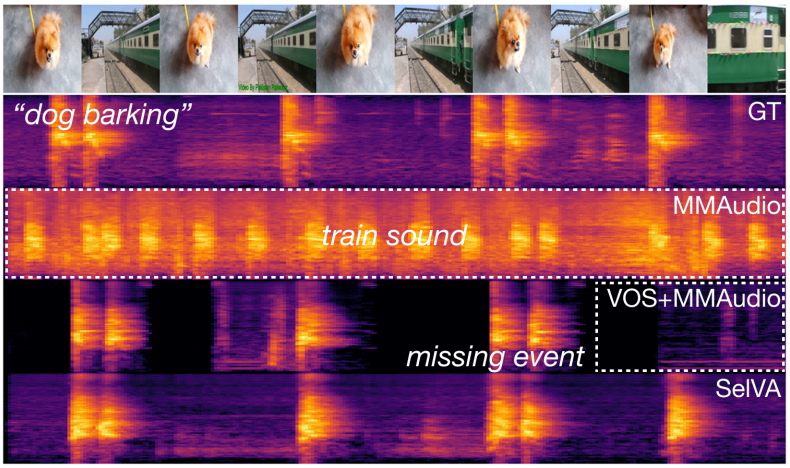
Table D5. Performance of **SELVA** with different sizes on VGG-MONOAUDIO. All methods used text prompts corresponding to the target videos. The **best** scores are shown in bold, and the second-best scores are underlined.

correct sound.

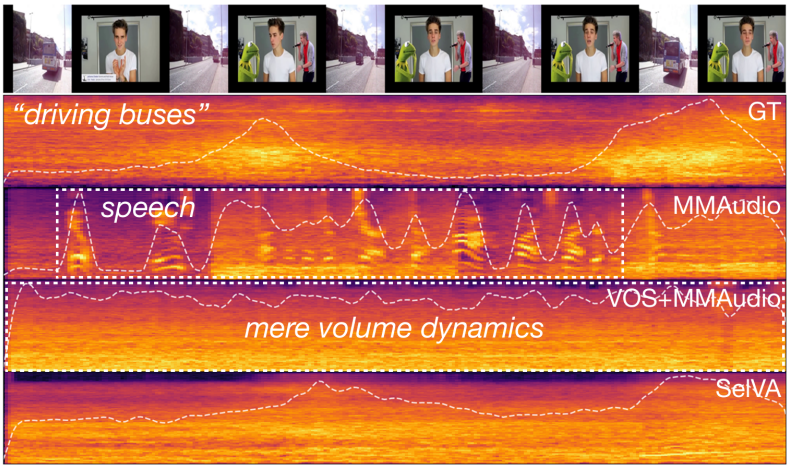
Fig. D8c showcases the semantic-level, cross-modal understanding of **SELVA**. This intra-class example pairs a target "baby crying" video with a "child singing" video. The task requires the model to semantically ground the text prompt, ignoring the visually present but undesired "child singing" event. Both MMAudio and the VOS baseline fail, generating mumble sounds synchronized with the non-target child on the right. This failure is expected for the VOS baseline, as DEVA [8] performs object-level segmentation and cannot semantically distinguish between the two subjects based on the text. In contrast, **SELVA** successfully leverages its text-conditioned vision encoder to generate the correct, synchronized crying sound.

D.5. Limitation of CLAP score

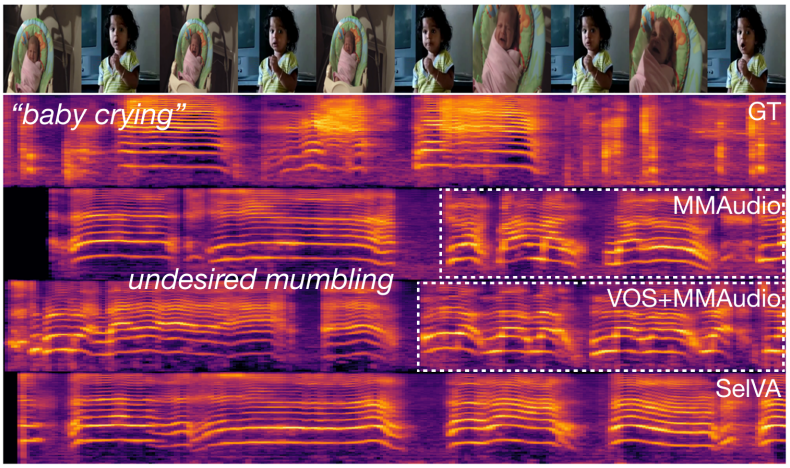
Fig. D9 highlights a limitation of the CLAP score [65] in capturing semantic text-audio alignment compared to human perception. In general, the VOS baseline was more likely to generate off-screen sounds, an error we attribute to the lack of background pixel information. Such artifacts may not be captured by the CLAP score. We argue that the CLAP encoder trained on noisy audio-text pairs may not penalize the presence of such non-diegetic sounds if they occurred frequently in its training data. However, we found that human annotators are highly sensitive to those artifacts, in that CLAP (0.344) of VOS is comparable to that of **SELVA** (0.349), but the temporal alignment scores (TA) differ substantially. This result demonstrates that human study is still essential to access V2A generation methods.



(a) ***"Dog barking"*** paired with *"Train wheels squealing"*.



(b) ***"Driving buses"*** paired with *"Male singing"*.



(c) ***"Baby crying"*** paired with *"Child singing"*.

Figure D8. Qualitative performance comparison with V2A methods in VGG-MONOAUDIO.

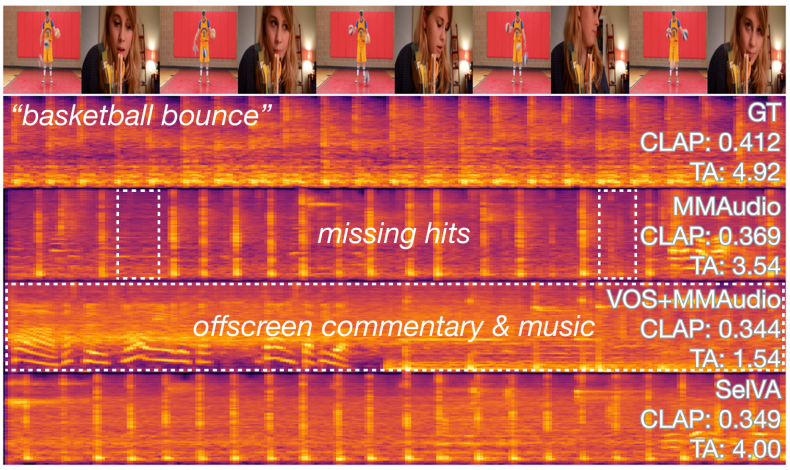


Figure D9. ***"Basketball bounce"*** paired with *"People whispering"*. There is a large discrepancy between the CLAP score and the human-annotated temporal alignment (TA) score.