FoleyDirector: Fine-Grained Temporal Steering for Video-to-Audio Generation via Structured Scripts

Supplementary Material

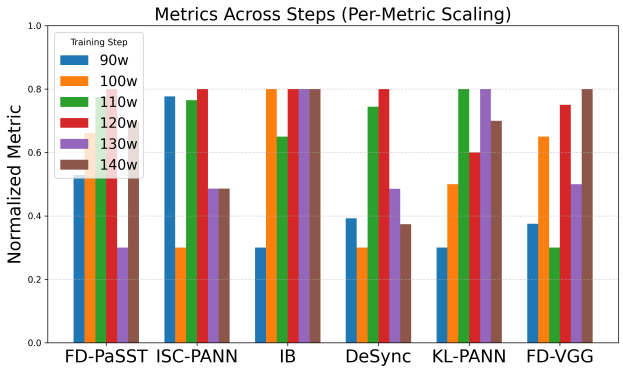


Figure 1. **Ablation on training iterations.** We show the model metrics after training for different numbers of iterations. The metrics are normalized to the range of 0.3–0.8, with higher values indicating better performance.

A. Model Architecture

DiT Architecture. Our model architecture is designed following the framework of MMAudio, consisting of stacked multi-stream and single-stream DiT blocks. The video input (8 frames per second) is encoded by CLIP to extract visual features, while the global text description is processed by CLIP's text encoder to obtain textual representations. The audio input is encoded and compressed using the VAE pretrained in MMAudio. All these modality-specific features are projected through their respective projectors and fed into the multi-stream DiT blocks. Additionally, the pooled global tokens from the video and text encoders, together with timestep and Synchformer features, are used to construct a temporally-aware global embedding, which conditions the adaptive LayerNorms within each DiT block.

In the DiT block, we further improve upon MMAudio's original design. After adaptive LayerNorm, temporal RoPE encodings are added to both video and audio features, and joint attention is performed across the three modalities (video, text, and audio). Following this joint attention, we introduce an additional SG-TFM module, which takes STS features as extra input and performs Temporal Script Attention, a type of joint attention with the audio features to inject fine-grained temporal-semantic information. After passing through four joint blocks, the fused audio features, which integrate visual, textual, and temporal cues, are processed by eight single-stream DiT blocks (as in MMAudio) to produce the final target flow prediction.

When users wish to render complex soundscapes with fine-grained control, we introduce an Bi-Frame Sound Synthesis Framework. In each joint block, the audio latent is copied into in-frame and out-of-frame branches. During rendering, we disable the influence of visual features on the out-of-frame branch, allowing it to be modulated solely by textual and temporal-semantic cues, while maintaining conditioning from the globally temporal Synchformer embedding. The in-frame branch, in contrast, follows the standard rendering process. The latent outputs from both the in-frame and out-of-frame SG-TFM modules are then segmented and reorganized according to the control intervals of the desired sound events, enabling synchronized yet independently controllable generation of in-frame and out-of-frame audio.

VAE and vocoder. In our framework, we adopt the audio representation and backbone modules directly from MMAudio, including its pretrained VAE encoder–decoder and BigVGAN vocoder. Following MMAudio, the audio waveform is first transformed into mel spectrograms by applying a short-time Fourier transform (STFT) and extracting the magnitude component. The resulting spectrograms are then encoded into latent representations using a pretrained variational autoencoder (VAE). During inference, generated latents are decoded back into mel spectrograms by the same VAE, which are subsequently converted into audio waveforms using the pretrained vocoder. The VAE follows the 1D convolutional network design of Make-An-Audio 2, employing a downsampling factor of 2 and trained with reconstruction, adversarial, and Kullback–Leibler (KL) divergence objectives. MMAudio applies the magnitude-preserving network design introduced in EDM2, replacing standard convolutional, normalization, addition, and concatenation layers with magnitude-preserving counterparts. This modification effectively stabilizes latent magnitudes without noticeably affecting reconstruction performance. For vocoder components, we use the 44.1 kHz version BigVGAN-v2.

B. Pipeline Details

We show the details of our data annotation pipeline, including Perception and Recognition, segment-level Classification.

Perception and Recognition. We use system instructions to prompt Qwen-Omni to generate a caption for the given audio, enabling it to form a holistic understanding of the audio content. Based on this overall comprehension, we

(a) Perception and Recognition

You are an audio-visual analysis expert. Given an audio clip (and optionally its corresponding video), perform the following steps:

1) Analyze the audio in the context of the video to form a rich overall perception, considering multiple aspects such as content, timbre, volume, pitch, texture, and environment.
2) Generate a concise and informative caption summarizing your perception: caption: ...
3) Based on the caption and the audio, identify and locate distinct sound events, including their timing.
4) List all detected sound categories in the format: category: sound1, sound2, ..., separated by commas.

(b) Segment-level Classification

You are an audio-visual analysis expert. Analyze the provided audio clip along with its corresponding video clip, and determine whether any of the specified categories are present.

1) For each category in the provided list, indicate whether the category is present in the clip ("Yes") or absent ("No"), based primarily on the audio, but ensure your conclusion is consistent with the visual content.
2) If a category is present, you may optionally provide additional details such as:
   a) Loudness: "soft", "medium", or "loud"
   b) Timbre: a brief description of the sound quality
   c) Desc: a brief caption of sound

Output the result in the following JSON format only

Figure 2. **System Prompt.** The system prompt we used in annotation pipeline

Model	Params	Distribution matching					Quality	Semantic	Temporal
		$\text{FD}_{\text{VGG}}\downarrow$	$\text{FD}_{\text{PANN}}\downarrow$	$\text{FD}_{\text{PaSST}}\downarrow$	$\text{KL}_{\text{PANN}}\downarrow$	$\text{KL}_{\text{PaSST}}\downarrow$	$\text{ISC}_{\text{PANN}}\uparrow$	IB↑	DeSync↓
Medium	621M	1.45	8.23	92.80	1.67	1.47	14.38	0.32	0.439
Large	1.03B	1.47	8.22	93.77	1.66	1.46	14.41	0.32	0.456

Table 1. Comparison of MMAudio models of different sizes on VGGSound-Director.All the model are 44KHZ.

further instruct Qwen-Omni to identify the types of sounds present in the audio and return their corresponding category names. Through this process, we obtain a fundamental understanding of the audio and discover the fine-grained sound events it contains. The system prompt we use is shown in Fig. 2 (a).

Segment-level Classification. We further perform **segment-level classification** to obtain fine-grained audio labels at the temporal segment level. Directly prompting Qwen-Omni to localize audio events yields unsatisfactory precision. Therefore, we reformulate the localization problem into a *binary classification task*. Specifically, we pre-segment each audio clip (and its corresponding video) into fixed-length intervals, and instruct Qwen-Omni to determine whether each predefined sound category is present within a given segment. If a sound is detected, Qwen-Omni is further asked to describe its content and timbre characteristics, thereby producing fine-grained segment-level text annotations. The system prompt used for this stage is illustrated in Fig. 2 (b).

C. More Analysis

The difference between other works. 1) The concept of **temporal control** refers to controlling the generation of both *on-screen* and *off-screen* sounds *within specific time periods* in our paper. Given a car video V_{car}, we can control the car horn at arbitrary time periods, which is an active control capability that **the other two works fail to achieve**. 2) **VTA-SAM** only extracts frame-level object features (not event features) via SAM, it can only extract the information indicating the probability of a car being present in each frame of V_{car}, unable to provide information for objects that are off-screen, and such object-level feature, rather than event-based feature, also fail to enable the control of the car horn at any arbitrary time. 3) **Video-Foley** *relies entirely on Videos* to extract RMS signals (thus cannot control off-screen sounds). The RMS signal only captures temporal intensity, lacks semantic-temporal information for multiple events to achieve fine-grained semantic-temporal control.

Details on inference latency. We will add test latency details in the revised version. We tested the average inference time on several 8-second video with MMAudio (**3.30s**) and our method (**3.76s**) on A800. It shows that we **introduces only about 12% additional inference overhead**, but achieves a **significant 102.6% performance improvement** (from 0.2378 to 0.4819) on Overall F1 score.

D. The Composition of STS

Through the above pipeline, we can obtain the audio events and their associated attribute descriptions (Loudness and Timbre) for each segment. We combine these descriptions using a simple template (f"{Desc}, {Loudness}, {Timbre}") to generate the Structured Temporal Scripts for each segment.

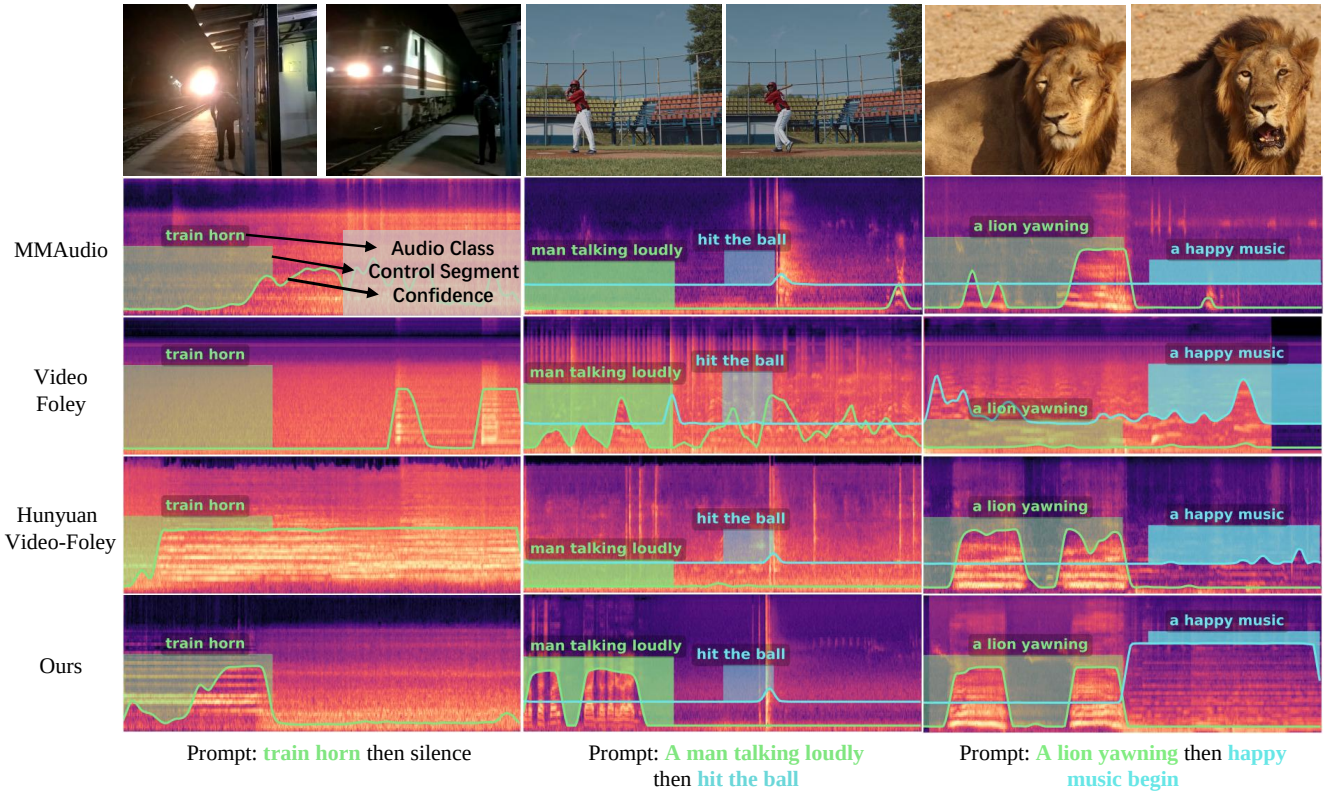


Figure 3. **Visual Results in DirectorBench.** We present several results from DirectorBench.

E. Benchmark Details

In this section, we present the details of the benchmark we constructed. The benchmarks used in our evaluation include **DirectorBench** and **VGGSound-Director**.

DirectorBench. DirectorBench primarily evaluates a model's controllability. To build this benchmark, we collected videos from the VGGSound test set and the Pexels website, and manually annotated various control instructions. The annotation process is as follows:

1. **Anchoring existing sound events.** All annotations are based on sound events that can be visually identified, so we first locate visible sound events in each video.
2. **Temporal control.** This category examines the model's ability to control when sounds occur and includes:
   (a) Allowing certain sound events to occur normally.
   (b) Enforcing silence for specific sound events.
   (c) For sound events with uncertain onset (e.g., distant fireworks or thunder), assigning a randomly selected segment as the sounding period.
3. **Counterfactual control.** This category evaluates combined control over both sound attributes and sounding periods, including counterfactual sound properties, off-screen sounds, and mixtures of counterfactual and normal sound events.

In total, we collected **100** videos, and for each video we annotated roughly **1–4** control scenarios per category, yielding **395** control samples. For evaluation, we use a grounding model to localize sound events. Based on the target sounding periods we annotated, we determine each event's sounding period and infer the silent intervals. We then measure the overlap between each predicted sound interval (including silence) and the target interval. Segments with an overlap greater than **0.5** are considered correct predictions, which are then used to compute **Precision**, **Recall**, and **F1-score**.

VGGSound-Director. VGGSound-Director focuses on evaluating generation quality. Following the annotation procedure described above, we labeled the STS data on the VGGSound test set and obtained **2.2K** test samples. On this benchmark, we follow MMAudio and compute audio-quality-related metrics, including distributional quality, perceptual quality, semantics, and audio-visual alignment.

F. More Results

DirectorBench. As shown in Fig. 3, we present a subset of the control results in DirectorBench. Consistent with the main text, we display the mel spectrograms of audio generated by different methods, annotate the target sound cat-

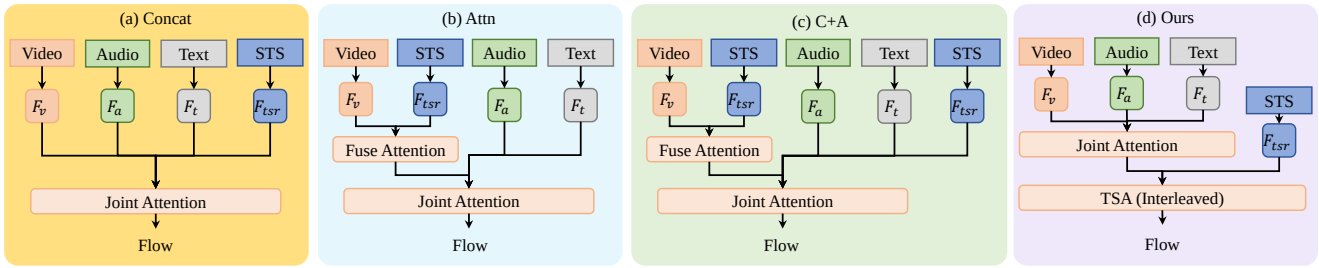


Figure 4. **Different Architecture.** We present the architectural diagrams of different design variants of SG-TFM. (a) concatenates the STS token with features from other modalities; (b) first fuses the STS token with video features using a Fuse-Attention layer; (c) combines the previous two approaches; and (d) shows the design we currently adopt.

egories, and overlay the Grounding model's confidence for the temporal localization of each sound. The target control time intervals are highlighted above the confidence curves. We observe the following phenomena: 1) Our method improves the accuracy of temporal control. In the leftmost example, we control the time interval for a train horn; other methods either produce a prolonged horn or trigger it at incorrect times. 2) Our method can effectively synthesize offscreen sounds. In the middle example, we aim to create a story of an athlete shouting to motivate themselves before hitting a ball; our method successfully synthesizes the offscreen audio while maintaining a reasonable hitting timing. In the rightmost example, we synthesize background music.

VGGSound-Director. We show the mel spectrograms of audio generated by different baselines alongside the mel spectrograms of the corresponding ground-truth audio. We also compute and annotate the L1 similarity (sim) between each method's mel and the ground-truth mel. The visualization results are presented in Fig. 5, 7 and 8. For special videos, such as black screens, text overlays, or transitions, where obvious visual cues are absent, our method synthesizes audio that more closely matches the ground truth, demonstrating its controllability. Additionally, for cases like police sirens or sounds from children facing away from the camera, our method generates audio that is noticeably closer to the real results.

Audio results. We have also included in the supplementary materials ZIP a portion of our generated audio results, as well as some outputs from SOTA generators. For each case, we provide a text file describing the control requirements, challenges, and corresponding results.

G. More Experiments

The Choice of Base-Model. We evaluated MMAudio models of different sizes (MMAudio-Medium-44K and MMAudio-Large-44K). As shown in the Tab. 1, although the Large model introduces substantially more parameters and requires greater training resources, it does not yield significant performance improvements. Therefore, we choose to use the more lightweight Medium model for training.

Drop	$FD_{VGG}\downarrow$	$KL_{PANN}\downarrow$	$ISC_{PANN}\uparrow$	IB↑	DeSync↓
0.0	1.51	1.31	14.63	0.33	0.467
0.1	1.17	1.42	14.84	0.33	0.432
0.3	1.61	1.37	14.63	0.32	0.456
0.5	1.67	1.42	14.81	0.32	0.461

Table 2. Comparison of MMAudio models of different sizes on VGGSound-Director. All models are 44kHz.

The Impact of Training Iterations. The Fig. 1 shows the model performance after different numbers of training iterations. For ease of comparison, all metrics are normalized to the range of 0.3–0.8, with higher values representing better performance—even for metrics such as FD and KL. It can be seen that the red sample (representing 1.2M training iterations) achieves relatively strong performance across almost all metrics. Therefore, we ultimately selected 1.2M iterations, which takes approximately 72 hours.

The Impact of STS drop ratio. We evaluated the effect of different STS drop ratios on quality. The results are shown in the Tab. 2. It can be seen that dropping STS tokens with a probability of 0.1 and replacing them with empty text tokens achieves strong performance across most metrics. Therefore, we adopted this as the default configuration for CFG

The Design of Architecture. As shown in the Fig. 4, we tested different architectures for incorporating the STS token. In (a), we concatenate the STS token with features from other modalities and perform the attention operation; in (b), we first fuse STS with video features through an additional attention mechanism to inject temporal information into the video features, and then apply MMAudio's Joint Attention; in (c), we combine the previous two approaches: the STS token is first fused with video features via attention, and then concatenated with other modality features and the STS features for the attention operation; (d) illustrates the approach we ultimately adopt, where an extra at-

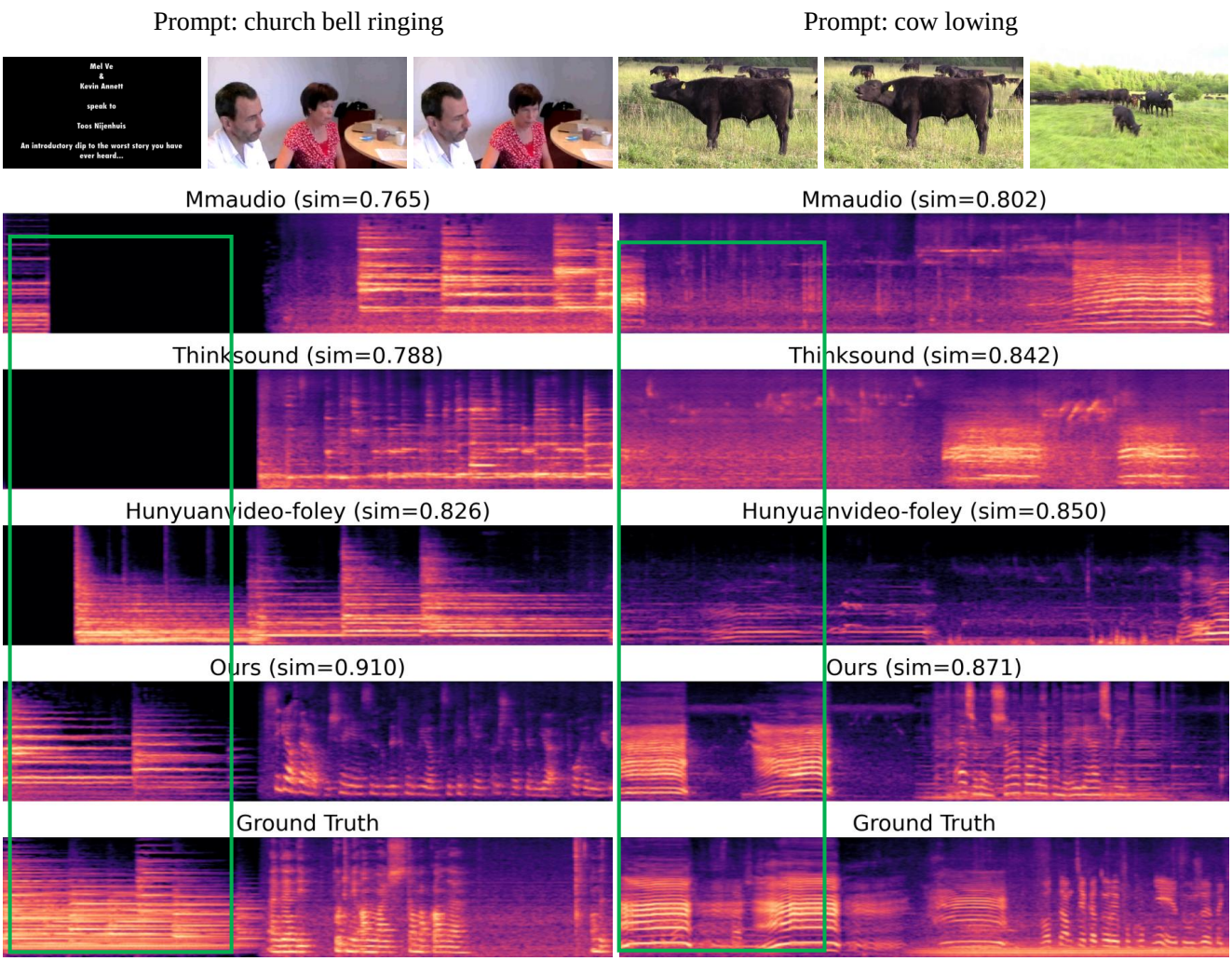
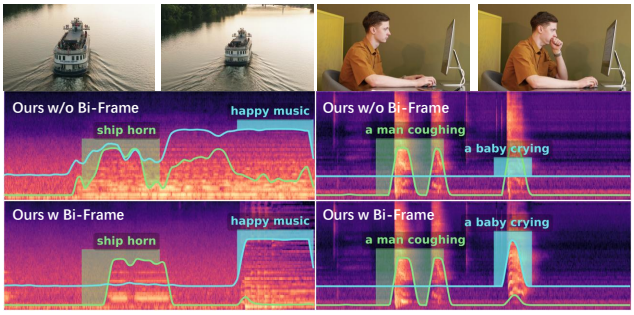


Figure 5. **Visual Results in VGGSound-Director.** We present several results from VGGSound-Director, comparing the mel-spectrograms generated by our method with those from other approaches and with the ground-truth audio. We also compute the L1 similarity between each generated mel-spectrogram and the ground truth.

tention layer fuses STS with the already fused audio features. The experimental results are shown in the Tab. 3. We observe that the concatenation approach achieves lower KL scores but relatively moderate IS and DeSync; the attention-based method achieves the highest ISC but performs poorly on DeSync, likely because the global information in STS partially disrupts the instantaneous temporal properties of video features. The combined approach performs slightly worse than simple concatenation. Finally, our method not only achieves better performance on most metrics but also *allows the SG-TFM module to be optionally discarded to switch between V2A generation and temporal control*, resulting in a more flexible framework with higher overall metrics.



Ship horn then a happy music begin

A man coughing then baby crying

Figure 6. **Visual Results in DirectorBench.** We present several results from DirectorBench.

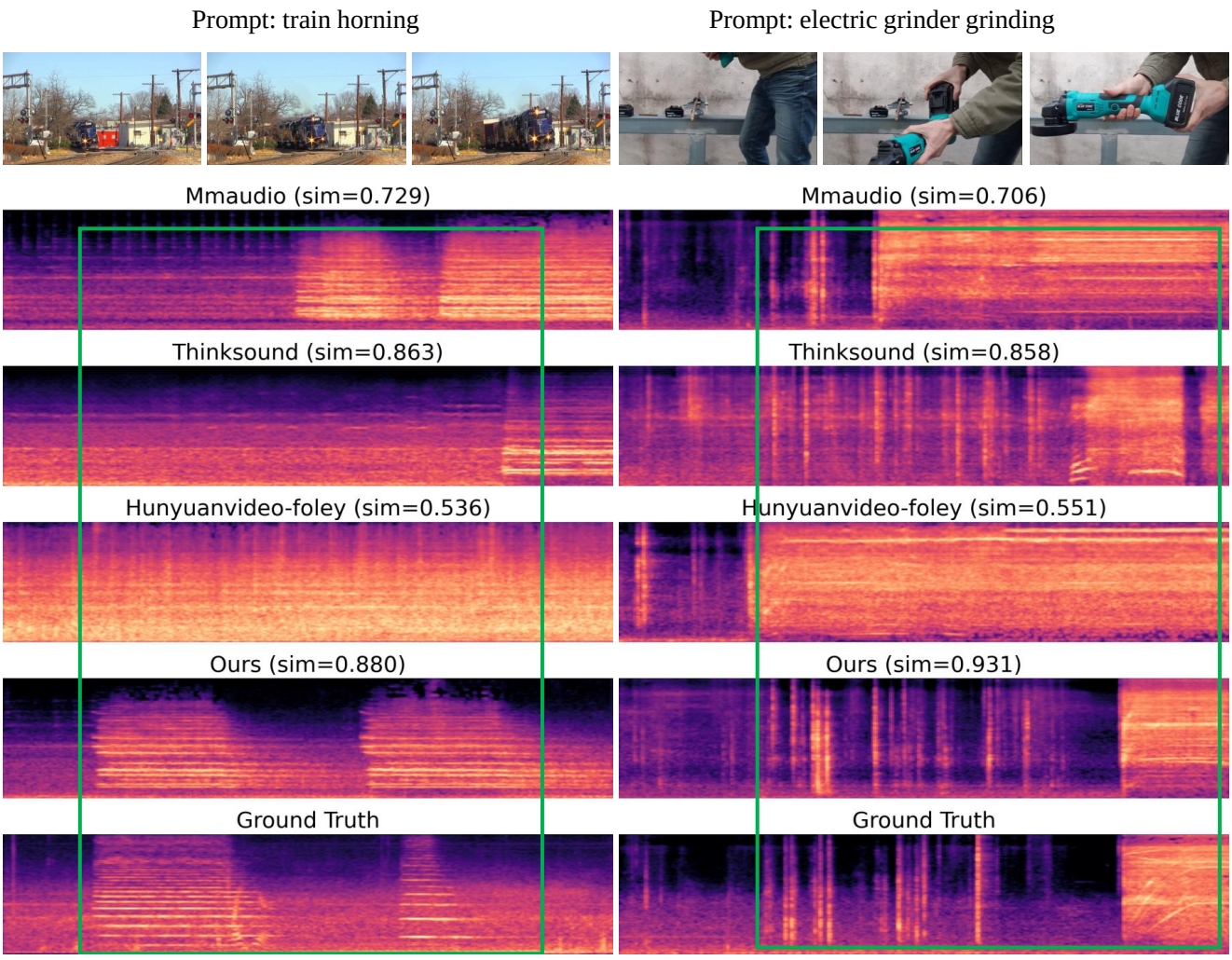


Figure 7. **Visual Results in VGGSound-Director.** We present several results from VGGSound-Director, comparing the mel-spectrograms generated by our method with those from other approaches and with the ground-truth audio. We also compute the L1 similarity between each generated mel-spectrogram and the ground truth.

The Impact of Bi-Frame. We also visualized the results of the Bi-Frame framework. It can be observed that without the Bi-Frame strategy, when synthesizing certain off-screen or counterfactual sounds, the corresponding audio in our method is affected by visual information, leading to weaker control or attribute leakage. By employing the Bi-Frame framework, this issue is flexibly addressed, improving both controllability and flexibility.

H. Training Details

We train our model for **1200K** iterations with a batch size of **16**. We train our model on 8×A800 GPUs (40 GB each) using the MMAudio-medium architecture, with a total training time of approximately 72 hours. During training, we randomly drop TSR features with a probability of 0.1 We set an initial learning rate of 2.0×10^{-5}. We use the AdamW optimizer and a cosine learning rate decay schedule. We apply a weight decay of 1.0×10^{-6} and clip gradients at a maximum norm of **1.0**. For training efficiency, we use bf16 mixed precision training, and all the audio latents and visual embeddings are precomputed offline and loaded during training.

I. Limitations and Future work

Limitation. Although our approach enhances the controllability of the generation process, it still has several limitations. **1)** Our framework is built upon the MMAudio architecture, and thus its performance is inherently constrained by the limitations of MMAudio itself. **2)** Since the training audio in current V2A datasets is relatively simple, our model struggles to produce satisfactory results when synthesizing highly complex audio, due to both data limita-

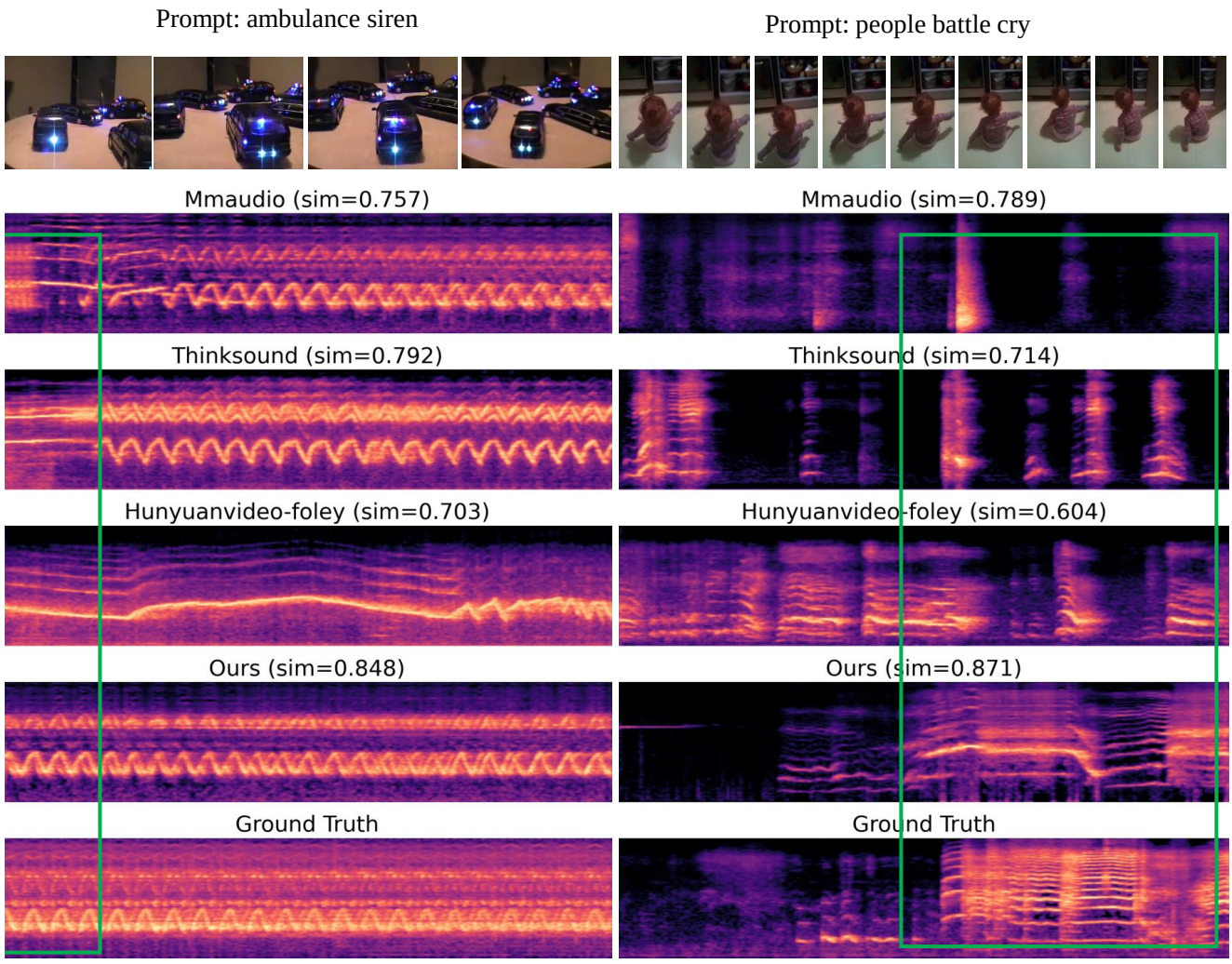


Figure 8. **Visual Results in VGGSound-Director.** We present several results from VGGSound-Director, comparing the mel-spectrograms generated by our method with those from other approaches and with the ground-truth audio. We also compute the L1 similarity between each generated mel-spectrogram and the ground truth.

Arch	FD_{VGG} ↓	KL_{PANN}↓	ISC_{PANN}↑	IB↑	DeSync↓
Concat	1.29	1.26	14.62	0.33	0.476
Attn	1.74	1.54	15.49	0.32	0.535
C+A	1.43	1.30	14.59	0.32	0.476
Ours	1.17	1.42	14.84	0.33	0.432

Table 3. Comparison of MMAudio models of different sizes on VGGSound-Director. All models are 44kHz.

tions and model capacity. **Future Work.** In traditional V2A tasks, the audio data and its annotations are relatively simple. This means that conventional V2A methods focus on a small set of audio event combinations, with only a single audio sound per time segment. We argue that though our method also only focus on the single event per segment, **the decoupling concept underlying Bi-Frame also holds potential for addressing multi-audio multi-track event combinations**. For example, one could replace Bi-Frame's off-screen/on-screen decoupling with a decoupling of individual audio events. For each audio event, a separate track is assigned and the audio is synthesized under STS control. During fusion, we can adjust the fusion weight of each track within the segment to achieve multi-event, multi-track synthesis.