In this appendix, we provide additional details on the methods and experiments discussed in Section 7 and Section 8. Explanations related to the dataset will be included in Section 9. We will also expand on the user study and present additional case studies in Sections 10 and 11. Finally, necessary explanations on multi-media material will be provided in Section 12. Limitations are discussed in Section 13.

7. Method Details

7.1. Motivation of Decomposing

Current V2As [24, 54] struggle with control accuracy due to incomplete visual cues in their control signals. Simply predicting mel (in low-resolution) from video increases complexity, and reduces performance in controlling audio generator (Table 3, row 1). To cope with this issue, we decompose mel into energy(E), semantic(S), and std.(D) (Table 3, row 2), which are then processed in quantization(S) or continuum(E&D), enhancing performance (Table 3, row 3&4) by balancing *completeness* and *complexity*.

7.2. Explanation of Mel-QCD

Figure 6 depicts 2D t-SNE visualization of S, derived from an audio features two sound events (gun shooting, shoot spreading, as shown in Figure 3). It shows S effectively distinguishes between sound events, and determines inter-event semantic differences.

8. Experimental Details

8.1. ControlNet

Our diffusion model is built upon the architecture of Auffusion [50], which fine-tunes the text-to-image generation model Stable-Diffusion-v1.5 [40] using text-audio pairs. By leveraging its pre-trained weights from the text-to-audio generation task, we incorporate the VAE encoder and UNet encoder to construct our ControlNet.

During the forward process of the model, a control map $\mathbf{C}_S \in \mathbb{R}^{K\times(T\times f_{mel})}$ is first repeated three times to accommodate the channel dimension of the images. This map is then downsampled to $(\frac{K}{8} \times \frac{(T\times f_{mel})}{8})$ using the VAE encoder.

Next, following [52], we pass the downsampled output through several trainable convolutional blocks, concluding with a layer that is initialized to zero. The processed output is then fed into the copied UNet encoder. At the end of each downsampling block, we retain the intermediate features and combine them with the skip features extracted from the main denoising UNet.

We freeze the parameters of the base model and introduced VAE encoder within ControlNet, and only train the copied UNet encoder.

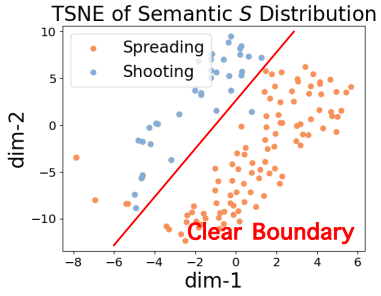

Figure 6. The t-SNE visualization for S.

8.2. Textual Inversion

For textual inversion, we employ the CLIP visual encoder to convert video frames $\mathbf{V} \in \mathbb{R}^{(T\times f_v)\times 3\times H\times W}$ into features with shape of $(T \times f_v) \times 768$, which are then processed by the Inversion Adaptor made up of one transformer layer and two MLP layers into 32 tokens.

8.3. V2X Predictor

We begin by using Synchformer to convert video frames $\mathbf{V} \in \mathbb{R}^{(T\times f_v)\times 3\times H\times W}$ into an embedding sequence with a shape of $((T \times f_v) \times 1024)$. This is followed by a two-layer MLP projector and four basic transformer blocks to generate the target signal embeddings.

Next, we employ three different signal prediction heads to transform embeddings into the corresponding signals. Notably, to account for scale invariance among the different losses, we optimize the three V2X predictors independently.

8.4. Training Details

During training, we first train the UNet encoder included within ControlNet on eight NVIDIA A100 GPUs with 80 Gb VRAM each for two days. With well trained ControlNet, we freeze it and introduce textual inversion related modules, which are trained for one day. In above training, we utilize an optimizer of AdamW with a learning rate of 1e-5, a batch size of 20 on each A100 card. For training the V2X predictors, we run each on eight NVIDIA L20 GPUs, each equipped with 48 GB of VRAM, for a duration of one day. To train this, we optimize it with an optimizer of AdamW with a learning rate of 3e-5, a batch size of 32 on each L20 card.

8.5. Inference Details

Following Auffusion [50], we utilize a diffusion sampler of PLMS [31] with a sampling step of 100 and incorporate the diffusion process with a classifier-free-guidance with a guidance scale of 7.5 during inference.

9. Dataset Related Explanations

9.1. Data Filtering

Since the official version of VGGSound consists of YouTube links, some of which have become inactive or have been replaced with new videos, we implemented a three-round data filtering process to ensure that only valid videos are included in our training set.

In the first round, we downloaded 199,176 videos and aimed to filter out links that have been replaced with new content. We used ImageBind to extract video, audio, and VGGSound embeddings based on textual class names. We sorted the scores for these three types of embeddings and discarded video indexes where all scores fell within the lower 20% range. As a result, we retained approximately 180,000 videos.

Next, we conducted a more fine-grained filtering process to eliminate videos with inconsistent visual, acoustic, and textual descriptions. We employed Aurora-Cap [5] and Qwen-Audio [8] to generate captions for the video and audio respectively. Using the original textual captions along with the generated ones, we utilized a Large Language Model (LLM), Llama-3 70b [12], to verify whether the three types of captions described the same sound effect. This process reduced our dataset to around 80,000 videos.

In the following step, we again used the same LLM, prompted with the three textual descriptions, to identify and remove videos featuring human talking. This step left us with approximately 56,000 videos, which became the final version of our dataset.

However, we noticed that some videos within this 56,000 set included sounds that did not correlate well with the visual content, making it challenging to infer sound effects from visual features. To mitigate this issue and improve the training of our V2X predictors, we conducted a manual filter on the 56,000 videos, selecting those with strong audio-video correlations. Ultimately, we narrowed down the dataset to about 22,000 videos, which were used to train the V2X predictors.

9.2. Data Split

To ensure a fair comparison with prior models trained on the officially split VGGSound training set, we must also isolate a test set. To achieve this, we select from the 56,000 videos intended for the official VGGSound test set to create our own test set, which will comprise 1,100 high-quality videos.

10. User Study

In this section, we describe a user study conducted with a selection of 100 videos from our test set. To facilitate comparisons with other methods, we present volunteers with a seed video along with the generated audio outputs from various algorithms, asking them to rank these audio samples.

Table 8. **Average User Rankings Across Three Evaluation Dimensions.** For the ranking index, a lower value indicates better performance in the evaluation metrics.

Method	Overall Score	Quality	Synchronization	Semantic
Im2Wav	4.64	3.92	4.82	5.18
DiffFoley	4.99	5.66	3.67	5.63
VTA-LDM	2.39	2.22	2.00	2.95
Seeing-and-Hearing	5.09	5.83	5.71	3.73
FoleyCrafter	3.81	3.01	3.59	4.82
Ours	**1.48**	**1.13**	**1.82**	**1.50**

And we report the Average User Ranking (AUR) in Table 8.

The user study evaluates the generated content across three key dimensions: generation quality, temporal synchronization, and semantic consistency. To guide the participants in their assessments, we provide the following prompts:

- **Generation Quality**: Rank the following audio samples based on their quality in representing the sound event.
- **Temporal Synchronization**: Rank the following videos according to how well the audio aligns with the video content in terms of synchronization with the sound event.
- **Semantic Consistency**: Rank the following videos based on how well the audio aligns semantically with the video content and the sound event `[sound event]`.

The results are presented in Table 8. As indicated in the table, our proposed method outperforms the other approaches across all three evaluation dimensions. Additionally, it is noteworthy that VTA-LDM consistently achieves the second-best performance across all metrics.

In the quality dimension, our method ranks highest with a score of 1.13, reaffirming its superiority in audio quality representation. VTA-LDM secures second place with a score of 2.22, demonstrating commendable quality, though not as high as "Ours." Seeing-and-Hearing ranks lowest in this dimension, with a score of 5.83, highlighting a significant gap from the top performers.

Regarding the synchronization metric, our method again takes first place with a score of 1.82, maintaining a strong performance in synchronizing audio with video content. VTA-LDM continues to show solid performance with a score of 2.00, solidifying its position as a competitive alternative. Im2Wav scores 4.82, indicating moderate performance, while DiffFoley scores lower at 3.67. Seeing-and-Hearing scores 5.71, aligning with its overall ranking and indicating synchronization issues.

In terms of semantic consistency, our method achieves a score of 1.50, reflecting effective semantic alignment with video content. VTA-LDM scores 2.95, demonstrating a good semantic connection, albeit not as robust as "Ours." Im2Wav and Seeing-and-Hearing fall behind with scores of 5.18 and 3.73, respectively, suggesting lower effectiveness

in maintaining semantic coherence. FoleyCrafter scores 4.82, indicating it also struggles with semantic alignment compared to the top methods.

11. Case Study

In this section, we present visualized mel spectrograms generated by various methods in Figure 7. Please refer to the attached file for the generated videos.

12. Generated Cases

To enhance the reader's experience in listening to the generated audio, we have included the videos along with their corresponding generated audio in a PowerPoint presentation. Please refer to the attached file for access.

13. Limitations

We recognize that our experimental evaluation is constrained by computational limitations and the difficulty of acquiring large datasets. Future work will aim to advance industrial-level model training to improve the applicability and scalability of our approach.

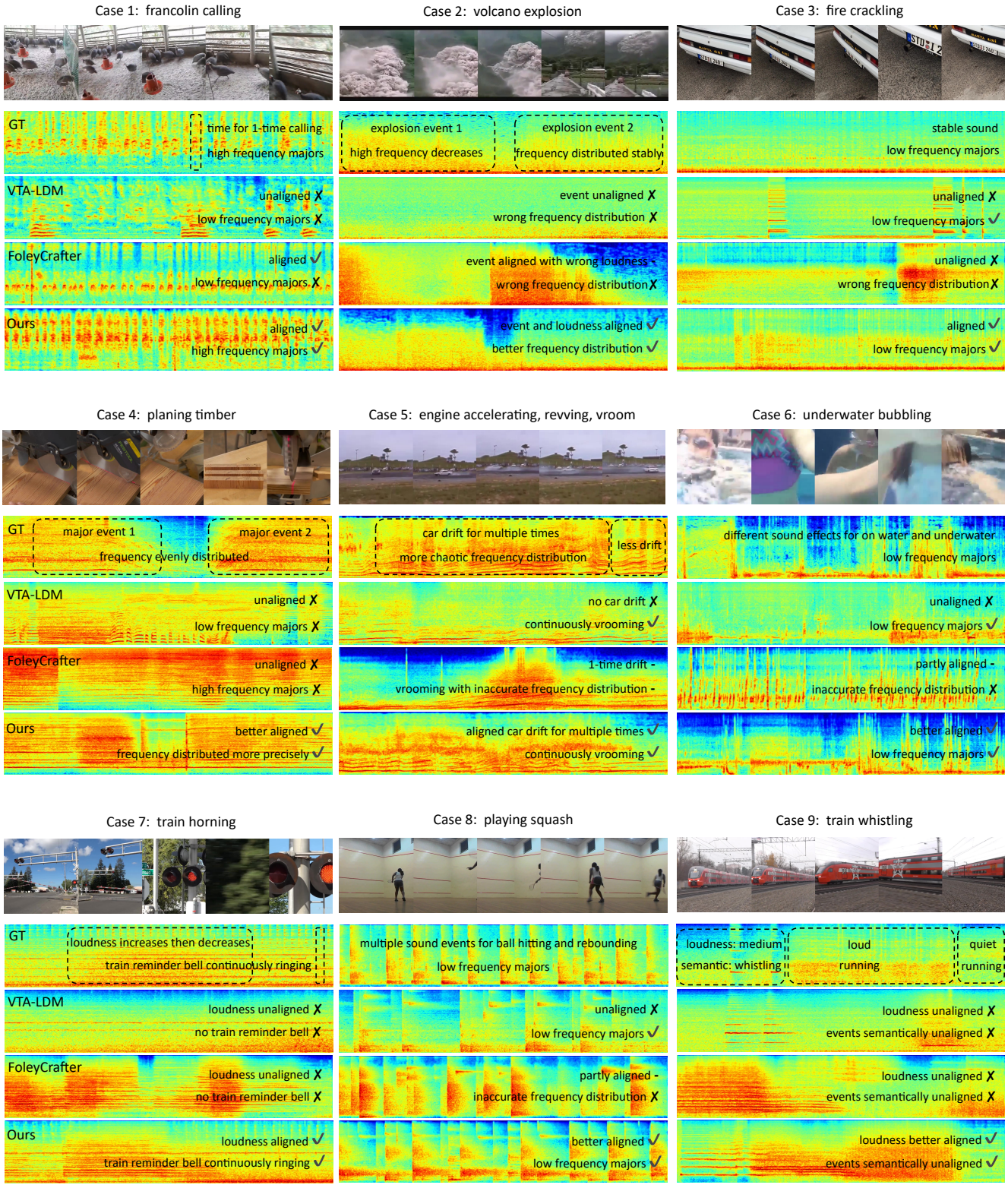

Figure 7. Case study for comparison on VGGSound test-set.