Orchestrating Audio: Multi-Agent Framework for Long-Video Audio Synthesis

Yehang Zhang[1*], **Xinli Xu**[1*], **Xiaojie Xu**[1*],
Doudou Zhang[1], **Li Liu**[1,2 †], **Yingcong Chen**[1,2 †]
[1] The Hong Kong University of Science and Technology (Guangzhou)
[2] The Hong Kong University of Science and Technology

Abstract

Video-to-audio synthesis, which generates synchronized audio for visual content, critically enhances viewer immersion and narrative coherence in film and interactive media. However, video-to-audio dubbing for long-form content remains an unsolved challenge due to dynamic semantic shifts, audio diversity and the absence of dedicated datasets. While existing methods excel in short videos, they falter in long scenarios (e.g., movies) due to fragmented synthesis and inadequate cross-scene consistency. We propose LVAS-Agent, a multi-agent framework that offers a coordinated, multi-component approach to long-video audio generation. Our approach decomposes long-video synthesis into four steps including scene segmentation, script generation, audio design and audio synthesis. To enable systematic evaluation, we introduce LVAS-Bench, the first benchmark with 207 professionally curated long videos spanning diverse scenarios. Experiments show that our method outperforms state-of-the-art V2A models in overall audio synthesis quality.

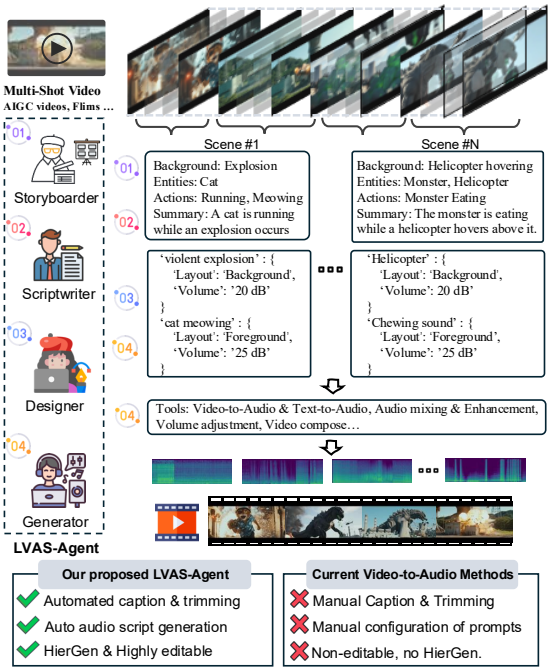

Figure 1: **We introduce LVAS-Agent, a multi-agent collaborative framework for end-to-end long video audio synthesis.** Built on VLM and LLM-based agents, it simulates real-world dubbing workflows, enabling automatic video script generation, audio design, and high-quality audio synthesis for long videos.

1 Introduction

Recent advances in diffusion models and large language models (LLMs) have greatly improved short-video dubbing by enabling synchronized and immersive audio-visual experiences. However, long-video dubbing remains challenging due to semantic complexity, cross-scene consistency, and dynamic content alignment. Existing models, tailored for short clips, struggle to maintain coherence across long durations and lack scalability to applications like film dubbing and AIGC video generation. Progress is further hindered by the absence of dedicated long-video audio synthesis datasets.

Existing video-to-audio methods fall into two categories: (1) training dedicated generators (e.g., SpecVQGAN (Iashin and Rahtu, 2021) and Foley-Crafter (Zhang et al., 2024a)) that capture short-term correlations but falter in scene transitions, and (2) text-driven models(e.g. SonicVisionLM (Xie et al., 2024), V2A-Mapper (Wang et al., 2024a)) that depend on textual inputs and perform poorly on implicit visual cues in long videos. These methods encounter common issues: (i) they lack mechanisms to capture long-range dependencies across dynamically changing scenes and are limited in handling multi-shot videos; (ii) they are overly sensitive to text prompts and often require manual prompt tuning to improve audio quality; (iii) they struggle to synthesize audio for longer video durations. Additionally, these methods rely on short-

[*]Equal contribution. [†] Co-corresponding author.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 22267–22282
November 4-9, 2025

video datasets that lack rich multi-sound annotations and sufficient cross-scene coverage, with each audio label typically containing only 2–4 words. A key question arises: *How can we leverage short-video dubbing priors to enable coherent and temporally aligned long-video synthesis without large-scale training data?* While naively segmenting long videos allows reuse of existing methods, it often breaks audio coherence and misaligns with the overarching semantics due to lack of global understanding.

To address this, we propose LVAS-Agent, a multi-agent framework that simulates professional dubbing workflows via structured role collaboration. The pipeline decomposes long-video audio synthesis into four tightly coupled stages: scene segmentation, video script generation, audio script design, and final audio synthesis (Figure 1). Each stage is assigned to a dedicated role: the **Storyboarder** segments scenes based on narrative and dubbing logic; the **Scriptwriter** generates video scripts by combining visual semantics with contextual cues; the **Designer** designs sound effects grounded in the video script; and the **Generator** retrieves audio label knowledge and orchestrates hierarchical audio synthesis using tools such as Video-to-Audio (V2A), Text-to-Audio (T2A), and audio editing.

Central to the system are two collaborative mechanisms: a decision-correction process for refining multi-shot video boundaries and optimizing scripts, and a generation-retrieval-optimization loop that iteratively aligns sound design with retrievable audio knowledge to achieve high-quality audio synthesis.

To enable systematic evaluation, we introduce **LVAS-Bench**, a curated dataset of 207 long videos across diverse scenarios such as urban scenes, combat, and animation.

Our contributions can be summarized as follows:

- We propose LVAS-Agent, a multi-agent system that structures long-video dubbing into collaborative roles and iterative processes.
- We release LVAS-Bench, the first dedicated long-video audio synthesis dataset, covering 207 professionally curated videos across diverse scenarios, enabling standardized benchmarking.
- Experiments demonstrate that LVAS-Agent improves semantic fidelity, timing accuracy, and audio distribution matching compared to prior methods.

2 Related Work

2.1 Video-to-Audio Generation

Video-to-audio generation, or dubbing, is a vital technique for enhancing auditory experiences and has advanced significantly with neural methods. Early models showed deep learning's potential in sound synthesis but were confined to specific genres (Chen et al., 2018, 2020; Mo et al., 2024; Zhou et al., 2018). Recent progress follows two main directions. One trains generators from scratch, including SpecVQGAN (Iashin and Rahtu, 2021) with cross-modal Transformers, Im2Wav (Sheffer and Adi, 2023) conditioned on CLIP features, Diff-Foley (Luo et al., 2023b) with contrastive pre-training, and MMAudio (Cheng et al., 2024) using flow-matching-based multimodal training. The other adapts text-to-audio models, e.g., Xing et al. (Xing et al., 2024) using ImageBind (Girdhar et al., 2023a), SonicVisionLM (Xie et al., 2024) using caption-based synthesis, V2A-Mapper (Wang et al., 2024a) translating visual to text embeddings, and FoleyCrafter (Zhang et al., 2024a) adding learnable modules to T2A models for end-to-end training. However, most methods still struggle with long videos, facing noise and audio-scene mismatches. Our method addresses this by introducing a video understanding and segmentation module to support long-form audio generation.

2.2 MLLMs for Video Understanding

Recent advances in vision foundation models (Dosovitskiy et al., 2020; Liu et al., 2021; Radford et al., 2021; Kirillov et al., 2023) have enabled the development of multimodal LLMs (MLLMs) (Liu et al., 2023; Tian et al., 2024; Zhang et al., 2023b), capable of language-guided visual understanding. This capability has extended to video understanding with models like VideoChat (Li et al., 2024a), Video-LLaMA (Zhang et al., 2023a), and Valley (Luo et al., 2023a). However, videos introduce temporal complexity and token-length challenges, often exceeding MLLMs' context limits. While frame sampling is commonly used, some works (e.g., Video-ChatGPT (Maaz et al., 2023)) propose efficient video feature representations.

For long video understanding, keyframe selection becomes critical. Approaches such as Kangaroo (Liu et al., 2024) and LLaVA-Video (Zhang et al., 2024b) use LLMs with expanded context windows, while others like MovieChat (Song et al., 2024) and MA-LMM (He et al., 2024) introduce

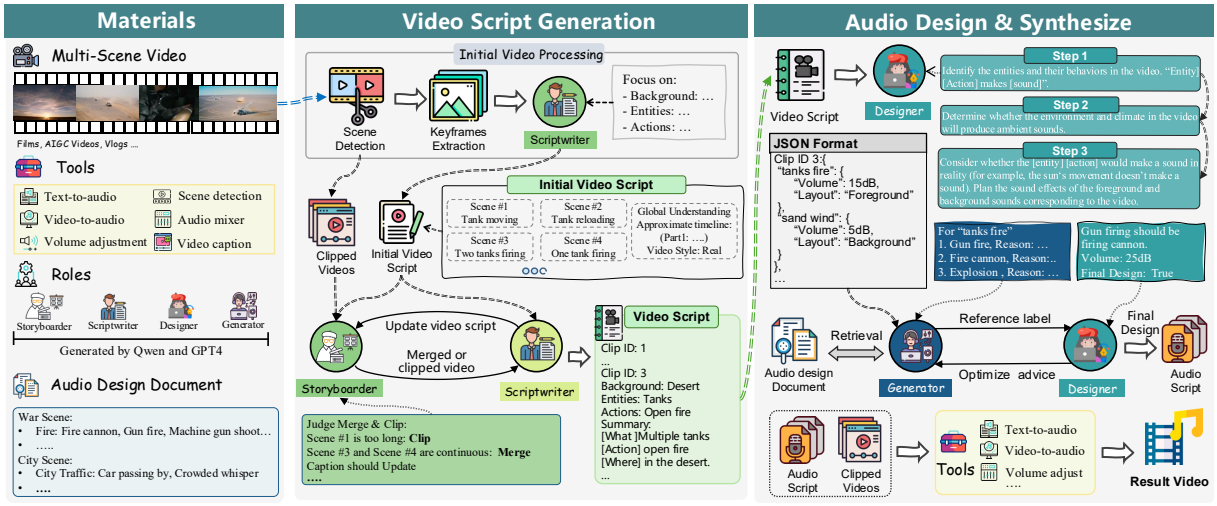

Figure 2: **Framework of LVAS-Agent**. Given the original video, Storyboarder and Scriptwriter collaborate through Decision and Correction to create a structured video script. The Designer and Generator complete multi-layered, high-quality sound synthesis through the Generate-Retrieve-Optimize mechanism.

memory systems or compact models for temporal compression. LongVLM (Weng et al., 2024) further reduces token load via token merging.

2.3 Multi-Agent Systems

LLM-based agent systems evolved from single-agent frameworks focused on tool integration (Li et al., 2023; Schick et al., 2023; Shen et al., 2023) to multi-agent collaborations. Inspired by concepts like the Society of Mind (Minsky, 1988), multi-agent systems (Park et al., 2023) tackle complex problems beyond single-agent capabilities. Frameworks like ChatDev (Qian et al., 2023), MetaGPT (Hong et al., 2023), and TransAgents (Wu et al., 2024) leverage simulated workflows where specialized agents collaborate, demonstrating superior reasoning and task completion on complex problems requiring interaction. Several agent-based approaches have also emerged in the audio domain. AudioGPT (Huang et al., 2024) supports tool-augmented audio generation but is limited to T2A tasks and cannot handle video-aligned sound synthesis. Audio-Agent (Wang et al., 2024c) focuses on general audio reasoning but lacks support for long, multi-shot videos and does not provide end-to-end V2A synthesis. In contrast, LVAS-Agent implements an end-to-end multi-agent framework specifically designed for long-video audio synthesis. By mimicking professional dubbing workflows, it effectively maintains audio continuity across shot transitions and achieves semantic alignment between video and audio. Through carefully designed agent collaboration, the system operates in a training-free manner and addresses common limitations of existing V2A models, such as unable to synthesize multiple audio tracks in complex scenes and poor cross-scene performance.

3 Method

3.1 Overview

By clearly defining agent roles, LVAS-Agent decomposes the video-to-audio synthesis task into two coordinated stages: video script generation and audio design. As shown in Figure 2, four specialized agents—**Storyboarder**, **Scriptwriter**, **Designer**, and **Generator**—collaborate to produce high-quality, multi-layered audio aligned with video content.

The process begins with video script generation. The **Storyboarder** segments the video into scenes, while the **Scriptwriter** analyzes both global and local content to generate descriptive captions for each scene. These agents interact through a *Decision-Correction* mechanism to iteratively refine scene boundaries and textual descriptions, resulting in a structured video script.

Subsequently, this structured script serves as the input for the audio design and synthesis stage, managed by the **Designer** and **Generator**. The **Designer** analyzes the video script to annotate detailed sound requirements, including foreground and background sound events, and their characteristics. The **Generator** then takes these annotations and, leveraging a knowledge base and synthesis models, produces the actual sound effects. The **Designer** and **Generator** collaborate through

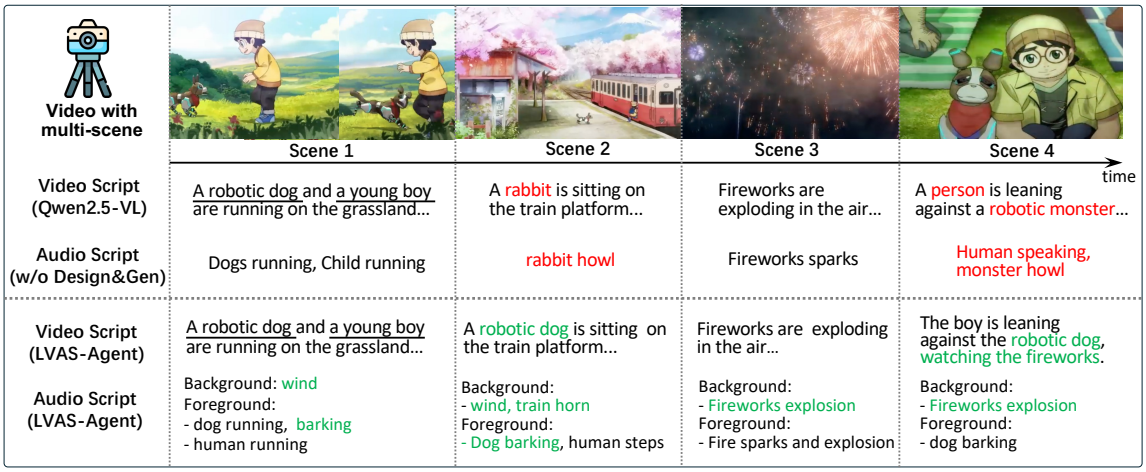

Figure 3: Comparison between the baseline and LVAS-Agent in video and audio script generation. The baseline misidentifies the same character as different labels, leading to inconsistent scripts and flat audio. In contrast, LVAS-Agent ensures character consistency and generates more accurate, diverse, and hierarchical audio.

Algorithm 1: Decision-Correction

Input: Storyboarder agent $\mathbf{P}$, Scriptwriter agent $\mathbf{Q}$, Video $\mathbf{V}$
Output: Structured video script $\mathbf{T}$

$\mathbf{T} \leftarrow \emptyset$
$\{v_0, \ldots, v_n\} \leftarrow \mathbf{P}(\mathbf{V})$ // Shot Change Detection
$\{kf_0, \ldots, kf_n\} \leftarrow \mathbf{P}(\{v_0, \ldots, v_n\})$ // Keyframe Extraction
$\mathrm{U_V} \leftarrow \mathbf{Q}(\mathbf{V})$ // Understand global content, style features
for $i = 0$ **to** n **do**
  $\mathrm{U}_i \leftarrow \mathbf{Q}(kf_i)$
  if $i > 1$ **then**
    $D \leftarrow \mathbf{P}(\mathrm{U}_i, \mathrm{U}_{i-1}, \mathrm{U_V})$
    if $D = MERGE$ **then**
      $v_{i-1} \leftarrow merge\_segments(v_{i-1}, v_i)$
      $\mathbf{T} \leftarrow \mathbf{T} \cup [\mathrm{U}_{i-1}, \mathrm{U}_i]$
    else
      $\mathbf{T} \leftarrow \mathbf{T} \cup [\mathrm{U}_i]$
for $i = 0$ **to** n **do**
  $D \leftarrow \mathbf{P}(\mathrm{U}_i)$
  if $D = CLIP$ **then**
    $v_i, v_{i+1} \leftarrow clip\_segments(v_i)$
    $\mathbf{T} \leftarrow \mathbf{Q}(v_i, v_{i-1})$ // Update video script
$\mathbf{T} \leftarrow \mathbf{T} \cup \mathrm{U_V}$
return $\mathbf{T}$

a Generation-Retrieval-Optimization strategy, involving iterative refinement of the sound design and synthesis plan to ensure high-quality, multi-layered audio output that is coherent with the video content. The specific prompts for each of these agents are provided in Appendix A.2.

3.2 Video script generation

As shown in Figure 2, this paper proposes a structured video script generation method to assist in generating sound effects for full-length videos. The method addresses three core challenges: 1) current V2A methods usually have duration constraints; 2) current V2A methods struggle with scene and content transitions; 3) ensuring consistency between video captions and audio descriptions for coherent synthesis. To overcome these, we introduce a fine-grained video structuring approach, supported by collaboration between storyboarder and scriptwriter agents, as outlined in Algorithm 1. The specific design of these agents is detailed as follows.

Storyboarder is responsible for fine-grained video structuring. Its key functions include detecting scenes for coarse segmentation, extracting key frames using the K-Means clustering algorithm, and refining segment boundaries based on the Scriptwriter's video script. Scene detection uses an HSV color space transition method for rapid, frame-accurate segmentation. By extracting key frames from smaller segments, it captures more visual information compared to directly inputting the full video into a vision-language model, enhancing video comprehension. Storyboarder also collaborates with the Scriptwriter to decide whether segments should be further edited, considering video script.

Scriptwriter is a visual support agent responsible for comprehending both the full video and individual video segments. Recent video understanding tasks achieve comprehension by extracting information from visual contexts to derive semantic features (Li et al., 2024b) or by directly generating descriptive text (Fan et al., 2025). Textual descriptions of the video script make it easier to maintain consistency between video and audio descriptions.

Furthermore, converting the video into a structured script, distinct from video frames, enhances processing speed and significantly reduces the token count.

Decision-Correction As illustrated in Algorithm 1, this strategy is carried out through the collaboration of two agents: the Storyboarder ($\mathbf{P}$) and the Scriptwriter ($\mathbf{Q}$). **Storyboarder** segments the video into distinct scenes $[v_0, \ldots, v_n]$ based on shot transitions and extracts corresponding keyframe sets $\{[f_{1,1}^{\text{key}}, \ldots], \ldots, [f_{n,1}^{\text{key}}, \ldots]\}$. **Scriptwriter** first performs a global analysis of the entire video and then generates detailed captions for each segment based on its keyframes. Subsequently, $\mathbf{P}$ and $\mathbf{Q}$ jointly decide whether adjacent segments should be merged or split and whether captions require refinement, based on both global context and segment-level semantics. The collaboration results in a structured and coherent video script. As shown in Figure 3, the Decision-Correction mechanism enables the generated video script to better fit audio synthesis scenarios. It not only avoids audio description errors caused by multi-shot transitions but also supports the generation of off-screen audio labels.

3.3 Audio Design and Synthesis

This section presents the second stage of LVAS-Agent: audio design and synthesis (as shown in Figure 2). The design follows three key principles: (1)Analyzing video scripts for accurate audio descriptions, (2) improving generated audio quality by combining V2A, T2A, and audio editing tools, and (3) enabling editable audio planning. This stage adopts a collaborative framework involving two LLM-based agents—**Designer** and **Generator**—and integrates retrieval-augmented generation (RAG) and audio synthesis models to produce multi-layered, high-quality audio.

Designer annotates audio in the video script and collaborates with the Generator agent to finalize the audio design. Real-world dubbing often involves complex scenes with layered environmental sounds and diverse sound-producing actions. To address this, we introduce a Chain-of-Thought (CoT) reasoning mechanism, breaking the task into steps: identifying primary action sounds, analyzing background audio, and ensuring audio coherence. The Designer agent creates the initial audio design, covering foreground and background sounds, volume control, and sound descriptions. It then provides iterative feedback to the Generator to optimize the final audio plan.

Generator The Generator synthesizes audio based on the audio annotations obtained through collaboration with the Designer. It uses retrieval-augmented generation (RAG) with an audio label knowledge base, Video-to-Audio (V2A) and Text-to-Audio (T2A) models for synthesis, hierarchical mixing, and volume adjustment. RAG-based retrieval ensures high-quality synthesis, addressing the limitations of V2A models trained on the VGGSound dataset, which contains only 310 audio labels with 2-4 words each. When audio prompts match these predefined labels, the generated audio is more stable and higher quality.

Building on this insight, all VGGSound labels were reorganized and reclassified into 20 common video scenarios. To enrich the labels, GPT-4 and human annotators added details such as typical scenarios and relevant objects. This resulted in 192 refined labels. The structured knowledge base allows the Generator to retrieve predefined labels, rather than relying on open-ended prompts. The Generator uses MMAudio (Cheng et al., 2024), a state-of-the-art model for both V2A and T2A tasks, as the primary audio generation tool. Background sounds are generated via T2A, while foreground sounds are produced via V2A. Finally, the Generator performs audio mixing, volume adjustment, and refinements based on the audio script.

Generation-Retrieval-Optimization The Designer and the Generator collaborate through GRO to optimize the audio tags in the audio script. The process is as shown in Algorithm 2. First, the Designer agent $\mathbf{D}$ formulates an initial sound design based on the video script. The Generator agent $\mathbf{S}$ then retrieves relevant knowledge from a sound synthesis database to generate a concrete implementation plan. This plan is reviewed by the Designer agent $\mathbf{D}$, who decides whether further refinement of the sound design is needed or if the plan is ready for final synthesis. Specifically, this process begins with an in-depth understanding of the video script, followed by iterative exchanges of feedback between the Designer agent $\mathbf{D}$ and the Generator agent $\mathbf{S}$. Through multiple iterations, the final sound synthesis plan is determined.

4 LVAS-Bench

Collection. We construct the first specialized **l**ong-**v**ideo **a**udio **s**ynthesis **bench**mark(**LVAS-Bench**). The benchmark contains 207 professionally cu-

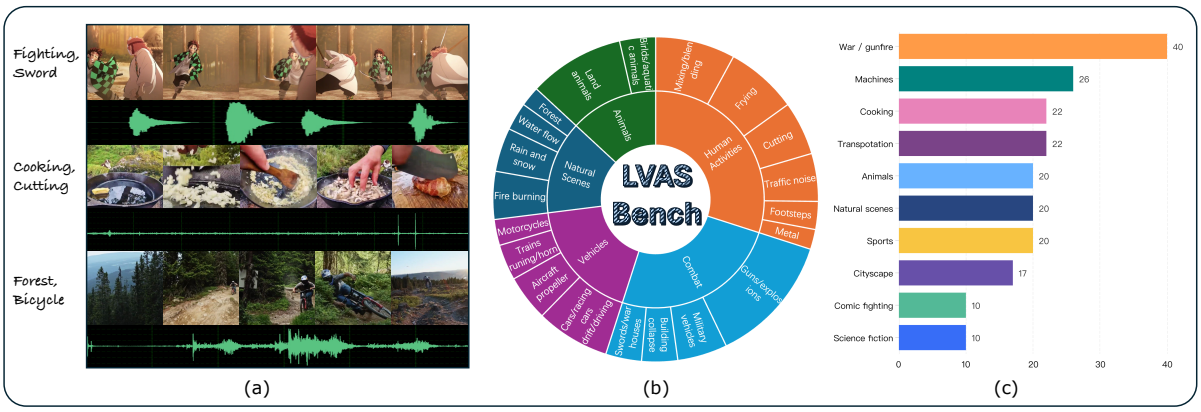

Figure 4: Our LVAS-Bench is presented in the following parts: (a) illustrates sample data from the benchmark, (b) provides statistical distributions of audio categories and sub-categories across the dataset, and (c) presents the statistics of video categories within the dataset.

Methods	Distribution Matching					Audio Quality	Semantic Align	Temporal Align
	$FD_{VGG}\downarrow$	$FD_{PANN}\downarrow$	$FD_{PASST}\downarrow$	$KL_{PANNs}\downarrow$	$KL_{PASST}\downarrow$	$IS_{PANNs}\uparrow$	IB-Score↑	DeSync↓
Baseline(FoleyCrafter)	6.60	61.36	637.82	2.54	2.65	4.79	0.28	1.24
Baseline(MMAudio)	8.96	52.94	589.27	2.01	1.91	3.92	0.29	0.60
LVAS-Agent(Ours)	**5.76**	**46.16**	**573.67**	**1.86**	**1.77**	4.28	**0.33**	**0.53**

Table 1: Comparison of different methods on various evaluation metrics. Lower values (↓) indicate better performance, while higher values (↑) indicate better quality.

Algorithm 2: Generation-Retrieval-Optimization Collaboration Strategy

Input: Designer agent $\mathbf{D}$, Generator agent $\mathbf{S}$, Video script $\mathbf{T}$, Maximum iterations $N_{\max}$
Output: Finalized sound synthesis plan $\mathbf{A}_{\text{final}}$
Initialization: $\mathbf{A}_{\text{init}} \leftarrow \mathbf{D}(\mathbf{T})$;
$\mathbf{A}_{\text{retrieved}} \leftarrow \mathbf{S}(\mathbf{A}_{\text{init}})$;
for $i = 1$ *to* $N_{\max}$ **do**
  $\mathbf{A}_{\text{reviewed}} \leftarrow \mathbf{D}(\mathbf{A}_{\text{retrieved}})$;
  if $\mathbf{D}$ *determines* $\mathbf{A}_{\textit{reviewed}}$ *is FINAL* **then**
    break; `// Exit early if finalized`
  $\mathbf{A}_{\text{modified}} \leftarrow \mathbf{D}(\mathbf{A}_{\text{reviewed}})$;
  $\mathbf{A}_{\text{retrieved}} \leftarrow \mathbf{S}(\mathbf{A}_{\text{modified}})$;
return $\mathbf{A}_{\text{reviewed}}$;

rated videos (with an average duration of 1 minute) sourced from three main origins: (1) film production archives with open licenses, (2) annotated documentary segments, and (3) procedurally generated synthetic scenes. Importantly, all videos in the dataset have pure sound effects, with no background noise or human speech. This creates a dataset of long-form, semantically rich videos with clear transitions, matched with corresponding pure sound-audio. Figure 4(a) illustrates representative video-audio cases.

Statistical Analysis. To ensure diversity, LVAS-Bench covers sufficient video and audio categories. Figure 4(b) visualizes the benchmark's audio types, encompassing five major classes (e.g., human activities) with numerous fine-grained subcategories. Figure 4(c) quantifies the distribution across 10 video-level categories, where instances such as the "cooking" category comprise 22 entries.

Benchmark Annotation. LVAS-Bench offers comprehensive global descriptions for each video and provides audio labels for its internal sub-segments. The time-stamped annotations indicate captions from specific seconds to specific seconds, while the global descriptions provide a detailed account of the entire long video. The specific annotation details are in the Appendix.

5 Experiment

5.1 Experiment Setup

Metrics We assess the generation quality in four different dimensions: distribution matching, audio quality, semantic alignment, and temporal alignment. 1) Distribution matching assesses the similarity in feature distribution between ground-truth audio and generated audio, under some embedding models. We compute Fréchet Distance (FD) and Kullback–Leibler (KL) distance. For FD, we adopt PaSST (Koutini et al., 2022) (FD_{PaSST}), PANNs (Kong et al., 2020) (FD_{PANNs}), and VGGish (Gemmeke et al., 2017) as embedding models. For the

Model Variation	Struct. Script (DC)	CoT (Designer)	RAG (Generator)	Iterative Ref. (GRO)	$FD_{VGG}\downarrow$	$FD_{PANNs}\downarrow$	$IS_{PNSS}\uparrow$	IB-Score↑	DeSync↓
M0: Baseline					7.81	84.37	1.69	0.281	0.361
M1: M0 + DC	✓				7.47	79.85	1.77	0.315	0.349
M2: M1 + CoT	✓	✓			7.39	77.82	1.74	0.323	0.337
M3: M2 + RAG	✓	✓	✓		7.24	76.19	1.94	0.336	0.301
M4: LVAS-Agent	✓	✓	✓	✓	**7.22**	**73.94**	**2.16**	**0.341**	**0.283**

Table 2: **Ablation Study of LVAS-Agent.** Evaluating LVAS-Agent performance on LVAS-Bench by progressively adding key components and interaction strategies. Abbreviations denote: DC, Decision-Correction for video script generation; CoT, Chain-of-Thought for audio design by the Designer agent; RAG, Retrieval-Augmented Generation by the Generator agent; and GRO, Generation-Retrieval-Optimization for iterative audio label refinement.

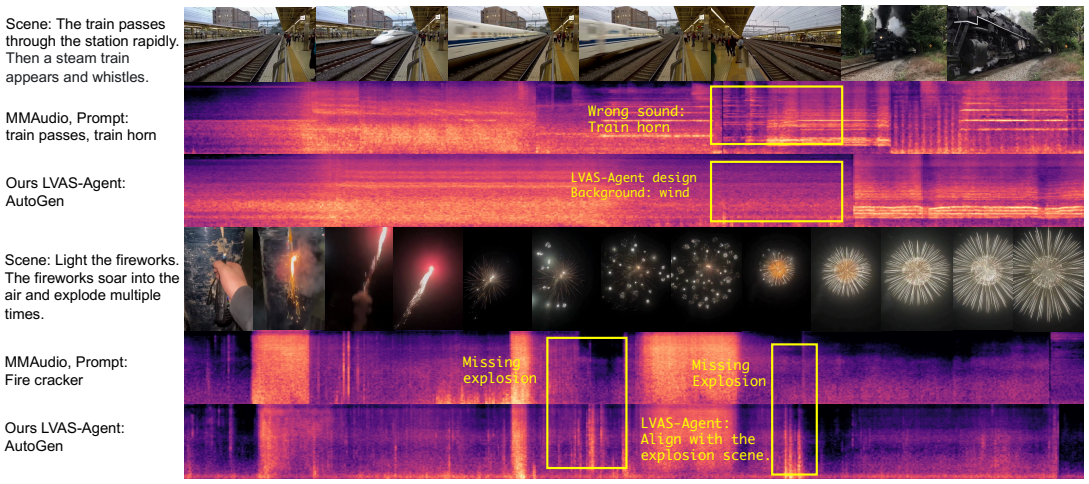

Figure 5: We visualize the spectrograms of generated audio (by sota V2A method and our method). LVAS-Agent demonstrates superior performance in synthesizing long video audio, ensuring seamless scene transitions without errors or missing sounds.

KL distance, we adopt PANNs (KL_{PANNs}) and PaSST (KL_{PaSST}) as classifiers. 2) We use PANNs as the classifier, following Wang et al. (Wang et al., 2024b), to assess generation audio quality without the need for comparison with the ground truth, utilizing the inception score. 3) Semantic alignment is measured using ImageBind (Girdhar et al., 2023b), following Viertola et al. (Viertola et al., 2024), by extracting visual features from the input video and audio features from the generated audio, then computing the average cosine similarity as the IB-score. 4) Temporal alignment: We use synchronization score (DeSync) to assess audio-visual synchrony. DeSync is predicted by Synchformer (Huang et al., 2023) as the misalignment (in seconds) between the audio and video.

Data. Since our method focuses on the task of sound effect synthesis for long videos, which consist of shorter video-audio pairs, are not suitable for evaluation. Therefore, this paper uses the proposed LVAS-Bench to assess the performance of the Agent-System.

Baselines. To handle sound effect synthesis for long videos, the baseline first segments the video using a scene change detection method based on HSV color variation. To align with the capabilities of existing V2A models, each segment is further limited to a maximum duration of 10 seconds. We employ two state-of-the-art V2A models, FoleyCrafter (Zhang et al., 2024a) and MMAudio (Cheng et al., 2024), to generate audio for each segment. FoleyCrafter supports audio generation for clips up to 10 seconds, while MMAudio performs best on clips of similar duration due to the characteristics of its training data. The final audio is obtained by concatenating the outputs from each segment.

Implementation Details. In our experiments, all LLM-based agents use the Qwen API (Qwen et al., 2025) with the “qwen-max" model to simulate different agent roles. The visual support agent is implemented using the locally deployed “Qwen2.5-VL-7B" model. The retrieval-augmented generation for the predefined audio description knowledge base is built on LlamaIndex and powered by a “qwen-plus" model.

5.2 Main Results

The evaluation metric comparison results are shown in Table 1, where LVAS-Agent outperforms the baseline methods across all metrics in four key dimensions, achieving state-of-the-art performance. Additionally, we visualize and compare the audio waveforms generated by different methods. The quantitative results demonstrate that our approach enables the existing V2A base models to generate higher-quality audio in long videos with enhanced semantic and temporal consistency, all without additional training.

As shown in the visualized spectrogram comparison in Figure 5, LVAS-Agent dynamically adjusts the video captions for each segment based on the underlying video semantics. This allows the system to generate diverse yet semantically aligned audio for multi-scene videos, effectively avoiding mismatched or inappropriate audio synthesis. Furthermore, by intelligently trimming long multi-shot videos, LVAS-Agent reduces the length of video tokens, ensuring that the V2T model captures key audio-related content without omission. Through well-structured audio script design, LVAS-Agent significantly enhances audio quality and audiovisual alignment of V2A models in a training-free manner. The video and audio script examples are shown in Figure 7.

5.3 Ablation Study

An ablation study on 20 LVAS-Bench videos (As shown in Table 2) was conducted to evaluate the impact of individual modules within the LVAS-Agent framework. Starting from a baseline (M0) without any specialized components, each configuration progressively introduces one module. Adding the Structured Scripting module with Decision-Correction (M1) improves initial segmentation and script quality, leading to better alignment and reduced desynchronization. Introducing the Designer's Chain-of-Thought reasoning (M2) enhances the coherence of sound effect descriptions, further improving alignment and reducing audio inconsistency. Incorporating Retrieval-Augmented Generation (RAG) in M3 enables more precise audio label retrieval, resulting in noticeable gains in semantic accuracy and synchronization. Finally, the full model (M4) integrates Iterative Refinement via a Generation-Retrieval-Optimization (GRO) loop, which significantly boosts all evaluation metrics through adaptive feedback between the Designer and Generator.

Overall, each component meaningfully contributes to the system's performance, and the cumulative design effectively balances complexity with output quality.

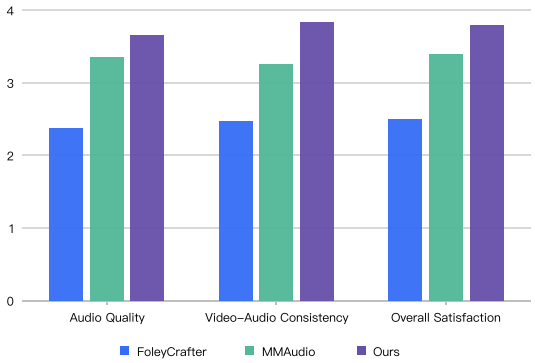

Figure 6: User study comparing our method with baselines across different aspects. Higher values indicate greater user preference.

5.4 User Study

We conducted a user study involving 30 participants to evaluate our method in comparison with FoleyCrafter (Zhang et al., 2024a) and MMAudio (Cheng et al., 2024). Participants were asked to listen to 10 audio samples generated by each method and rate them on a scale of 1 to 5 across three dimensions: "Audio Quality," "Video-Audio Consistency," and "Overall Satisfaction." Higher scores indicate better performance. As illustrated in Figure 6, the results of the user study demonstrate that our method outperforms the two baseline approaches across all evaluated aspects.

6 Conclusion

We present LVAS-Agent, a multi-agent framework that systematically tackles long-video dubbing challenges through role-specialized collaborative agents. By decomposing the workflow into scene segmentation, script generation, sound design, and hybrid synthesis, our method overcomes limitations in semantic continuity and temporal alignment inherent to existing approaches. We also release the first dedicated long-video audio synthesis dataset, covering 207 professionally curated videos, named LVAS-Bench. Experimental results demonstrate superior performance in distribution matching, audio quality, and alignment metrics on LVAS-Bench. For future work, we aim to develop a large-scale, finely annotated dataset of long-video audio to further advance the development of long-video dubbing models.

7 Limitations

Our work has several limitations. First, the video script generation in LVAS-Agent relies on the video understanding capability of vision-language models. Errors in video interpretation can lead to cascading inaccuracies in the subsequent audio script generation, resulting in mismatches between the synthesized audio and the actual video content. Second, LVAS-Agent segments multi-shot videos based on audio continuity. When extremely short clips (e.g., under 0.5 seconds) appear, they are merged with adjacent segments due to limitations of current V2A and T2A models, which struggle to synthesize audio for such brief durations. However, this merging may involve visually diverse shots, reducing audio quality. Finally, although we introduce LVAS-Bench to evaluate the performance of V2A models on long-video audio synthesis, the dataset size remains limited due to the challenges of collecting high-duration, multi-shot videos with clean and isolated sound effects.

References

Kan Chen, Chuanxi Zhang, Chen Fang, Zhaowen Wang, Trung Bui, and Ram Nevatia. 2018. Visually indicated sound generation by perceptually optimized classification. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0.

Peihao Chen, Yang Zhang, Mingkui Tan, Hongdong Xiao, Deng Huang, and Chuang Gan. 2020. Generating visually aligned sound from videos. *IEEE Transactions on Image Processing*, 29:8292–8302.

Ho Kei Cheng, Masato Ishii, Akio Hayakawa, Takashi Shibuya, Alexander Schwing, and Yuki Mitsufuji. 2024. Taming multimodal joint training for high-quality video-to-audio synthesis. *arXiv preprint arXiv:2412.15322*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, and 1 others. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. 2025. Videoagent: A memory-augmented multimodal agent for video understanding. In *European Conference on Computer Vision*, pages 75–92. Springer.

Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. 2017. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. 2023a. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15180–15190.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. 2023b. Imagebind one embedding space to bind them all. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15180–15190.

Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. 2024. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13504–13514.

Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, and 1 others. 2023. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 3(4):6.

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

Rongjie Huang, Mingze Li, Dongchao Yang, Jiatong Shi, Xuankai Chang, Zhenhui Ye, Yuning Wu, Zhiqing Hong, Jiawei Huang, Jinglin Liu, and 1 others. 2024. Audiogpt: Understanding and generating speech, music, sound, and talking head. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 23802–23804.

Vladimir Iashin and Esa Rahtu. 2021. Taming visually guided sound generation. *arXiv preprint arXiv:2110.08791*.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. 2023. Segment Anything. *arXiv preprint arXiv:2304.02643*.

Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D Plumbley. 2020. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2880–2894.

Khaled Koutini, Jan Schlüter, Hamid Eghbal-zadeh, and Gerhard Widmer. 2022. Efficient training of audio transformers with patchout. In *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*, pages 2753–2757. ISCA.

Chenliang Li, Hehong Chen, Ming Yan, Weizhou Shen, Haiyang Xu, Zhikai Wu, Zhicheng Zhang, Wenmeng Zhou, Yingda Chen, Chen Cheng, and 1 others. 2023. Modelscope-agent: Building your customizable agent system with open-source large language models. *arXiv preprint arXiv:2309.00986*.

KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. 2024a. Videochat: Chat-centric video understanding. *Preprint*, arXiv:2305.06355.

KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. 2024b. Videochat: Chat-centric video understanding. *Preprint*, arXiv:2305.06355.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual Instruction Tuning. *arXiv preprint arXiv:2304.08485*.

Jiajun Liu, Yibing Wang, Hanghang Ma, Xiaoping Wu, Xiaoqi Ma, Xiaoming Wei, Jianbin Jiao, Enhua Wu, and Jie Hu. 2024. Kangaroo: A powerful video-language model supporting long-context video input. *arXiv preprint arXiv:2408.15542*.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022.

Ruipu Luo, Ziwang Zhao, Min Yang, Junwei Dong, Da Li, Pengcheng Lu, Tao Wang, Linmei Hu, Minghui Qiu, and Zhongyu Wei. 2023a. Valley: Video assistant with large language model enhanced ability. *Preprint*, arXiv:2306.07207.

Simian Luo, Chuanhao Yan, Chenxu Hu, and Hang Zhao. 2023b. Diff-foley: Synchronized video-to-audio synthesis with latent diffusion models. *Advances in Neural Information Processing Systems*, 36:48855–48876.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2023. Video-chatgpt: Towards detailed video understanding via large vision and language models. *Preprint*, arXiv:2306.05424.

Marvin Minsky. 1988. *Society of mind*. Simon and Schuster.

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

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, and 1 others. 2023. Chatdev: Communicative agents for software development. *arXiv preprint arXiv:2307.07924*.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551.

Roy Sheffer and Yossi Adi. 2023. I hear your true colors: Image guided audio generation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36:38154–38180.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, and 1 others. 2024. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232.

Yunjie Tian, Tianren Ma, Lingxi Xie, Jihao Qiu, Xi Tang, Yuan Zhang, Jianbin Jiao, Qi Tian, and Qixiang Ye. 2024. Chatterbox: Multi-round multimodal referring and grounding. *arXiv preprint arXiv:2401.13307*.

Ilpo Viertola, Vladimir Iashin, and Esa Rahtu. 2024. Temporally aligned audio for video with autoregression. *Preprint*, arXiv:2409.13689.

Heng Wang, Jianbo Ma, Santiago Pascual, Richard Cartwright, and Weidong Cai. 2024a. V2a-mapper: A lightweight solution for vision-to-audio generation by connecting foundation models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15492–15501.

Yongqi Wang, Wenxiang Guo, Rongjie Huang, Jiawei Huang, Zehan Wang, Fuming You, Ruiqi Li, and Zhou Zhao. 2024b. Frieren: Efficient video-to-audio generation network with rectified flow matching. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Zixuan Wang, Yu-Wing Tai, and Chi-Keung Tang. 2024c. Audio-agent: Leveraging llms for audio generation, editing and composition. *arXiv preprint arXiv:2410.03335*.

Yuetian Weng, Mingfei Han, Haoyu He, Xiaojun Chang, and Bohan Zhuang. 2024. Longvlm: Efficient long video understanding via large language models. *arXiv preprint arXiv:2404.03384*.

Minghao Wu, Jiahao Xu, and Longyue Wang. 2024. Transagents: Build your translation company with language agents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 131–141.

Zhifeng Xie, Shengye Yu, Qile He, and Mengtian Li. 2024. Sonicvisionlm: Playing sound with vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26866–26875.

Yazhou Xing, Yingqing He, Zeyue Tian, Xintao Wang, and Qifeng Chen. 2024. Seeing and hearing: Open-domain visual-audio generation with diffusion latent aligners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7151–7161.

Hang Zhang, Xin Li, and Lidong Bing. 2023a. Video-llama: An instruction-tuned audio-visual language model for video understanding. *Preprint*, arXiv:2306.02858.

Shilong Zhang, Peize Sun, Shoufa Chen, Min Xiao, Wenqi Shao, Wenwei Zhang, Kai Chen, and Ping Luo. 2023b. GPT4RoI: Instruction Tuning Large Language Model on Region-of-Interest. *arXiv preprint arXiv:2307.03601*.

Yiming Zhang, Yicheng Gu, Yanhong Zeng, Zhening Xing, Yuancheng Wang, Zhizheng Wu, and Kai Chen. 2024a. Foleycrafter: Bring silent videos to life with lifelike and synchronized sounds. *arXiv preprint arXiv:2407.01494*.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. 2024b. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*.

Yipin Zhou, Zhaowen Wang, Chen Fang, Trung Bui, and Tamara L Berg. 2018. Visual to sound: Generating natural sound for videos in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3550–3558.

A Appendix

A.1 LVAS-Bench Dataset Construction

All video clips used in LVAS-Bench are reportedly sourced from publicly accessible content on YouTube. As potential data sources, we initially identified and collected a range of video types, including themed ASMR videos (e.g., cooking, sports, work, city walks, and nature scenes), post-processed documentary segments (mainly featuring animals and military topics), and selected vlogs. A key challenge in constructing this dataset was obtaining "clean audio" clips—segments free from prominent speech or background music. To address this, approximately 500 video clips of varying lengths were collected initially. During the filtering process, clips containing speech or music were discarded. Videos were also trimmed to remove transitions and on-screen subtitles. The final dataset consists of 207 curated clips.

For annotation, the LVAS-Agent first automatically generated draft scripts for the processed clips. These drafts were then manually reviewed and corrected. Scene segmentation boundaries were refined, and each finalized segment was annotated with explicit sound effect labels, categorized into foreground and background sounds. Typically, each segment contains no more than three labels per category. An example annotation is shown in Figure 7.

A.2 Key Prompts

This section presents the key prompts for the agent roles within the LVAS-Agent framework. The key prompt settings for the Agent roles are as follows: This prompt defines the decision-making process for the Storyboarder agent to determine whether two video segments should be merged.

Storyboarder: Judge Merge Instructions

You are a professional video editor responsible for determining whether two video clips should be merged based on their textual descriptions. Your decision must ensure narrative and audiovisual continuity in the final edit. Decision Criteria for Merging. Analyze the relationship between the last video clip, the current video clip, and the overall video timeline by considering:

1. Content Continuity

- Are the two clips depicting the same ongoing scene or event? - Does the transition between clips maintain a logical progression?

2. Scene & Environment Consistency

- Do both clips occur in the same setting? - Example of discontinuity: - A clip in an outdoor park followed by a scene inside a church → Not continuous - A lake with a single tree vs. a lake with snow-capped mountains → Different settings, not continuous - Example of continuity: - A dog standing on stairs → A dog jumping down the stairs → Same entity, continuous event

3. Action & Sound Consistency

- Are the actions and sounds from one clip naturally leading into the next? - Example of discontinuity: - Chopping vegetables followed by sprinkling seasoning → Both are cooking-related but produce different sounds, not continuous - Example of continuity: - Tank firing → Tank shooting → Essentially the same action and sound, should be merged - Two people dueling with knives from different perspectives → Same fight, should be merged

Processing Steps

1. Analyze the Text Descriptions:
- Compare the last clip's caption, the current clip's caption, and the whole video's context.
- Determine whether the clips belong to the same scene or action sequence.
2. If merging is required:
- Generate a new unified description combining both clips.
3. If merging is NOT required:
- Assess whether the current video description needs adjustments for clarity and continuity based on the global context. - If no modifications are needed, retain the original description.

This prompt guides the Scriptwriter agent in analyzing individual video clips and generating structured descriptions.

Scriptwriter: Analyze Video Clips

You are a video analyst with expertise in understanding and interpreting various video clips.

Task Description

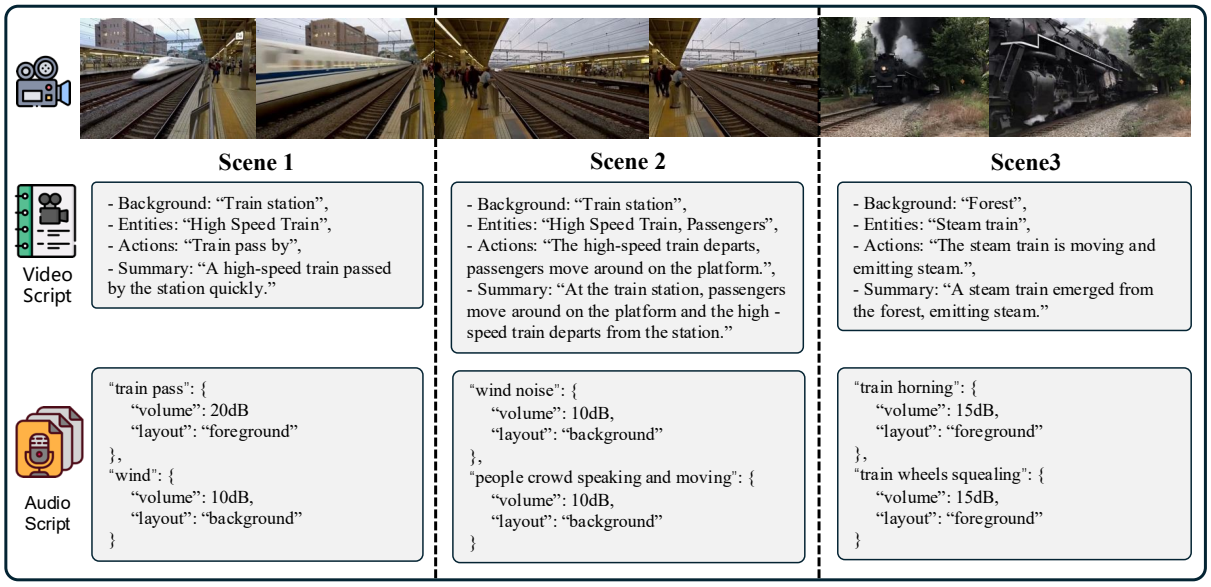

Figure 7: Examples of Video Script and Audio Script generated by LVAS - Agent. As annotations for LVAS - Bench after manual inspection of segment partitioning, video and audio scripts.

Given a video clip, please perform the following tasks: - Analyze the clip, identifying entities, their actions, and the video scene. Provide text descriptions as required by the output format. The entities must be real and present in the video. - The full video description provided is rough and not entirely accurate. You need to first analyze the current clip and then summarize it, considering the existing full description.

Constraints

- The analysis is strictly limited to the provided video clip; avoid speculating or using background information beyond the video. - The summary must be strictly based on the video content, without personal assumptions or creative additions. - Background sounds typically include weather conditions (e.g., rain, snow, thunderstorms) or real-world sounds (e.g., crowd parades, train horns). - Avoid using abstract atmospheres like those of a futuristic city, forest, or the universe as background sounds.

Input Data

Video

Output Format

- Background:
- Entity:
- Action:
- Video caption:

This prompt instructs the Scriptwriter agent to analyze the full video and produce a structured scene summary and timeline.

Scriptwriter: Analyze whole video

You are a professional video editor with expertise in scene analysis, timeline construction, and event identification.

Task Description

Given a video, analyze its content to provide a detailed summary of the following:

1. Identify the main scenes and their sequence, highlighting key events and actions.
2. Construct an approximate timeline of the video, emphasizing transitions between key moments or actions.
3. Summarize the video’s content in a structured and coherent manner.

Input

A video clip.

Output Format

- Scene Summary: (Describe the key scenes and their sequence in detail)
- Timeline: (Provide a detailed summary of key events, highlighting transitions between scenes)

This prompt enables the Designer agent to generate an initial set of sound effect annotations from a video description.

Designer: Generate Initial Audio Script

You are a sound effects specialist. Your task is to generate precise and realistic sound effect descriptions for a video clip based on its textual description, just like a professional foley artist. Your output should be structured in JSON format.
Follow these steps carefully to ensure accurate and contextually appropriate sound effect descriptions:
1. Identify Sound-Producing Entities & Actions
- Extract key entities (e.g., people, animals, objects) and their actions from the video description.
- Only describe actions that naturally produce a sound. For example, a car accelerating makes a sound, but a sunset does not.
- Format: [Entity] makes [adjective] sound or [Action] makes sound.
2. Determine Background Ambience
- If the environment contributes to the soundscape (e.g., wind, rain, ocean waves), describe it as the background audio.
- Avoid vague terms such as tense atmosphere or futuristic hum—use concrete environmental sounds.
- Background audio should be clearly distinguishable from main sounds.
3. Prioritize Primary vs. Secondary Sounds
- If the scene has a dominant action sound (e.g., car racing), it should be the main audio, while secondary sounds (e.g., crowd cheering) should be background if necessary.
4. Determine Sound Output Based on Reality
Choose the most accurate option based on the video description:
- Option 1: If no entity or ambient sound is relevant $\rightarrow$ "audio": [], "background": [].
- Option 2: If there is only an ambient sound $\rightarrow$ "background": [ambient sound], "audio": [].
- Option 3: If entities/actions produce sound and there is ambient noise $\rightarrow$ "audio": [entity sound], "background": [ambient sound].
5. Avoid Redundant Sounds
- Do not repeat the same sound in both "audio" and "background".
- Example: "background": ["wind"], "audio": ["wind noise"] is redundant—only keep "audio": ["wind noise"].

This prompt tasks the Designer agent with reviewing and validating audio label suggestions proposed by the Generator.

Designer: Correction Audio Label

Your current task is to critically review audio label suggestions proposed by the Generator Agent. Your goal is to ensure the highest quality and most appropriate audio design, not to passively accept recommendations.
Inputs:
1. Current Audio Design: Your existing audio plan for the video segment, including your original designer label for each sound event, its timing, and foreground or background attributes.
2. Video Script: The textual description of the current video scene, detailing visuals, actions, and mood.
3. Generator Suggested Labels: A list of generator suggested label from the Generator, intended as replacements for your original designer label, supposedly retrieved from its refined knowledge base.

Task & Evaluation Mandate: For each generator suggested label, you must perform a ruthless evaluation against your original designer label and the Video Script. Your judgment must be sharp and uncompromising. Consider:
1. Semantic Precision & Contextual Relevance: Is the Generator's suggestion genuinely accurate and contextually fitting for the specific visual events and narrative in the Video Script?
2. True Necessity & Justifiable Improvement: Is this change actually necessary? Does it offer a significant, tangible improvement in the potential audio output, or is it a trivial, pedantic alteration?
Output: For each sound event where the Generator proposed a replacement, you must provide:
- Original_Designer_Label: Your initial label.
- Generator_Suggested_Label: The Genera-

tor's proposed label. - Decision: Choose one:
- ACCEPT SUGGESTION
- REJECT SUGGESTION
- MODIFY SUGGESTION (If modifying, also provide Modified Labels)
- KEEP ORIGINAL
- Justification: A brief, direct explanation for your decision
- Final_Label_For_Synthesis: The definitive label to be used.

This prompt defines how the Generator agent retrieves refined audio labels from a reference source based on video context.

Generator: Retrieval Audio Label

Your task is to retrieve the most appropriate and specific audio labels from a predefined reference document based on the provided video descriptions and original audio descriptions.

Guidelines for Retrieval & Labeling:

1. Strict Compatibility: The retrieved audio label must be highly compatible with both the video description and the original audio description. If no suitable label is found, do not provide a replacement.
2. Replacement Strategy:
- Prioritize semantic similarity when suggesting replacements. - If no exact match is found, focus on the type of sound produced, disregarding the sound source. Example adjustments: - Searching for building explosion, but only volcanic explosion is available $\rightarrow$ Output explosion - Searching for tank firing, but no exact match exists $\rightarrow$ Find related artillery firing labels - Searching for airplane engine roar, but no exact match exists $\rightarrow$ Look for airplane-related sounds, as airplane noise originates from its engine
3. Context Awareness: Consider both video captions and raw audio descriptions for accurate label selection.
4. Strict Label Set Adherence: Stay strictly within the available labels in the reference document.
5. Handling 'None' Labels: If the raw audio description is ["None"], retain ["None"] without suggesting alternatives.