EchoFoley: Event-Centric Hierarchical Control for Video Grounded Creative Sound Generation

Bingxuan Li[1,4], Yiming Cui[2], Yicheng He[2], Yiwei Wang[3], Shu Zhang[2], Longyin Wen[2], Yulei Niu[2]

[1] University of Illinois Urbana-Champaign [2] ByteDance, Intelligent Creation (USA)
[3] University of California, Merced [4] University of California, Los Angeles

https://echofoley.github.io/

Abstract

Sound effects build an essential layer of multimodal storytelling, shaping the emotional atmosphere and the narrative semantics of videos. Despite recent advancement in video–text–to–audio (VT2A), the current formulation faces three key limitations: (1) an imbalance between visual and textual conditioning that leads to visual dominance; (2) the absence of a concrete definition for fine-grained controllable generation; (3) weak instruction understanding and following, as existing datasets rely on brief categorical tags. To address these limitations, we introduce ***EchoFoley*** *(**E**vent-**C**entric **H**ierarchical c**O**ntrol), a new task designed for video-grounded sound generation with both event-level local control and hierarchical semantic control. Our symbolic representation for sounding events specifies when, what, and how each sound is produced within a video or instruction, enabling fine-grained controls like sound generation, insertion, and editing. To support this task, we construct* ***EchoFoley-6k****, a large-scale, expert-curated benchmark containing over 6,000 video–instruction–annotation triplets and 42,000 fine-grained sounding event annotations. Building upon this foundation, we propose EchoVidia, a sounding-event-centric agentic generation framework with slow-fast thinking strategy. Experiments show that EchoVidia surpasses recent VT2A models by 40.7% in controllability and 12.5% in perceptual quality.*

1. Introduction

Creative intelligence drives imagination to conceive stories and worlds beyond what we see. Recent advances in generative modeling have made this imagination increasingly tangible, allowing humans to use natural language to control highly realistic images and videos generation [3, 30, 33], even enabling counterfactual world synthesis. Yet, the ultimate virtual worlds we create go beyond silent films, and sound represents a crucial dimension of creativity. Besides, controllability is essential for creative content synthesis, including video-to-audio generation. A slight change in sound can entirely recast a visual scene, reshaping how audiences perceive motion, story, and atmosphere. As the example illustrated in Figure 1, consider a silent video where a cat meows twice, by adding a magical whoosh and transforming the second meow into a lion's roar, this ordinary scene tells a creative story: "*A cat greeted me with a gentle meow, then with my wizard magic, the cat can now suddenly roar with the mighty voice of lion.*" Generating such sound effect requires fine-grained controls beyond the video level.

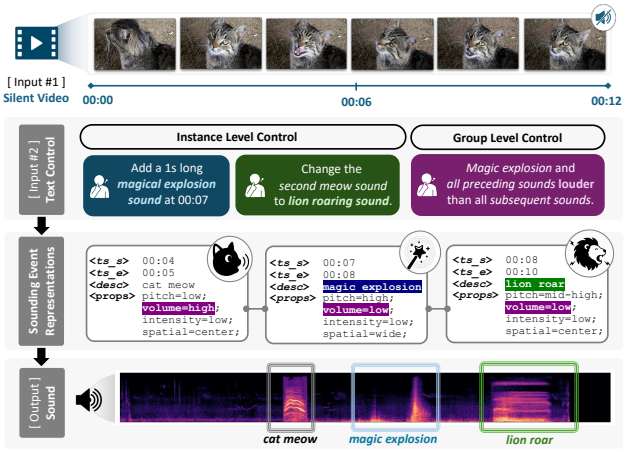

Figure 1. **Motivation of *EchoFoley***. In creative storytelling, sound shapes the story we perceive. Given a silent video, generating audio that matches this story-shaped imagination requires fine-grained control over how each sound is crafted and transformed over time. We formulate such task with event-centric hierarchical control (*e.g.*, instance control, group control), and propose effective solution.

While recent models have made progress toward control-

lable video-to-audio generation, existing approaches still fall short of the fine-grained creative control envisioned above. Current systems typically rely on optional text instructions to guide sound generation, such as short labels [8, 9, 46] (*e.g.*, "cat meowing") or short natural language descriptions or reasoning [26, 31, 32, 34, 41] (*e.g.*, "distant rumble of thunder becomes louder and more intense"). However, the current coarse instruction setting cannot support fine-grained controls or editing, especially for complex visual and temporal scenes. First, these video-level instructions operate at the granularity of a specific sound category and thus cannot distinguish multiple events of the same category. For example, "change the cat's meow to a louder sound" is ambiguous when the video contains several meow events. The instruction cannot specify which meow—*e.g.*, "the second meow at 00:07"—should be replaced with a lion roar or edited independently of the others. Second, the instructions on sound effects are limited to single aspects, for example the timbre of the sound (*e.g.*, "add a magic explosion"). Such instructions cannot support editing multiple attributes, like a combination of timbres, orders, durations, and volumes, at the same time (*e.g.*, "insert a 1-second magic explosion at 00:07 and make this explosion and all preceding sounds louder than all sub-sequent ones"). These limitations hindered the development of fine-grained controllable video-text-to-audio generation.

In this work, we present ***EchoFoley***, a new task designed for **E**vent-**C**entric and **H**ierarchical c**O**ntrol in video-grounded sound generation. Instead of video-level control, we focus on event-level control to disentangle a target sounding event from others. We further introduce a symbolic sounding event representation to define structure the event, enabling a hierarchical control to edit sounding effects according to a sound category, a single sounding event, or its specific attributes. To support this task, we construct ***EchoFoley-6k***, a high-quality, expertly curated benchmark comprising over 6,000 video–instruction–audio triplets with 42,000 densely annotated sounding events. Each example combines natural-language instructions about how sounds should evolve with event-level temporal annotations indicating when and where each event occurs.

While many VT2A methods have achieved impressive results, our preliminary studies on ***EchoFoley*** reveal that they remain limited in how they represent and control auditory imagination through natural language. To bridge this gap, we further propose *EchoVidia*, a training-free agentic framework with slow-fast thinking. *EchoVidia* enables hierarchical and interpretable auditory control—allowing models to reason about events, understand human intent, and generate creative, contextually aligned soundscapes. Experiments show that *EchoVidia* outperforms recent VT2A baselines by 40.7% in controllability and 12.5% in perceptual quality, demonstrating its effectiveness in fine-grained, instruction-guided sound generation.

In summary, the main contributions of our work are:

1. We formulate ***EchoFoley***, introducing a new paradigm for event-centric hierarchical control in video-grounded sound generation, and define a symbolic sounding-event representation that explicitly specifies when, what, and how each sound is produced within a video.
2. We construct ***EchoFoley-6k***, a large-scale, densely-annotated, and expertly-curated benchmark containing over 6,000 video–instruction-annotation triplets of data and over 42,000 dense sounding event annotations. We also provide a suite of metrics to support systematic evaluation of the proposed task.
3. We propose *EchoVidia*, a sounding-event-centric agentic generation framework with slow-fast thinking that enables fine-grained control. Experiments demonstrate significant improvements in controllability, semantic alignment, and quality over recent VT2A baselines.

2. Related Work

Audio-Video Correspondent Datasets. Learning robust multimodal representations relies on large-scale datasets that align visual and auditory signals. *VGGSound* [6] first established broad correspondences between videos and sound events, while *ego4Dsounds* [5], *AVVP* [38], and *ASVA* [45] extended this paradigm to egocentric, weakly supervised, and synchronized settings for richer temporal and spatial modeling. Audio–text corpora enable grounding of auditory semantics in language. *AudioSet* [15] and the *BBC Sound Effects Library* [1] provide broad sound category coverage, while *AudioCaps* [20] and *WavCaps* [29] pair audio clips with natural language captions for multimodal alignment. Our work advances the direction with fine-grained event-level correlations.

Multimodal Conditioned Audio Generation. Recent works demonstrate how cross-modal reasoning enables controllable multimodal content generation [22, 47, 50]. In the multimodality-conditioned audio generation domain, diffusion- and transformer-based models including *Seeing&Hearing* [42], *Diff-Foley* [27], *MultiFoley* [8], *MMAudio* [9], *HunyuanVideo-Foley* [34], *Hear-Your-Click* [24], *YingSound* [7], and *ThinkSound* [26] synthesize temporally aligned, video-conditioned audio. Yet current models remain over-optimized for visual alignment, struggle with complex multi-object scenes, and lack fine-grained controllability through text. Some recent works have attempted to adopt MLLMs for multi-stage V2A [32, 41]. Our work evaluate and advance this direction to fine-grained control with symbolic representation.

Sounding Event Localization. A key step toward controllable audio generation is accurately localizing where and when sound events occur. Early work by Tian et al. [37] established the audio–visual event localization task. Sub-

sequent studies advanced granularity and data scale: Hebbar *et al.* [17] curated a movie-based detection dataset, while Mahmud and Marculescu [28] introduced *AVE-CLIP* with pre-trained audio–language representations for temporal segmentation. Recent works [11, 14, 16, 49] explore weakly supervised, long-video, and open-vocabulary settings for more flexible event reasoning. Egocentric settings further enrich perception by coupling actions and sounds in first-person videos [4, 19]. Our work extends this direction toward fin-grained controlled audio generation.

Video Event Reasoning with Large Multimodal Models. Large multimodal models extend language understanding to spatiotemporal reasoning in videos. Models such as *Video-LLaMA* [44], *Video-LLaVA* [25], *MovieChat* [35], *VideoChat* [23], *Vidi* [36], *Vita* [12], and *Gemini 2.5* [10] enable detailed video understanding, editing, and instruction following through unified visual–language representations. Earlier work such as *Vid2Seq* [43] explored dense video captioning and temporal grounding, laying the foundation for reasoning-aware video–language alignment. Classical temporal reasoning and localization studies [13, 18, 21, 48] further established the importance of identifying moments and actions from natural language queries. Together, these advances inspire the integration of multimodal reasoning and temporal grounding in our setting.

3. Task: *EchoFoley*

EchoFoley aims to provide video-grounded sound generation with fine-grained controls at event level. To support such controllability, we first introduce a symbolic representation for sounding events that serves as an intermediate interface between natural language instructions and video-grounded audio generation. Based on this symbolic representation, we formulate an event-centric, hierarchically controllable video-grounded sound generation task.

3.1. Sounding Event Representation

We define *sounding events* as temporally-localized audio segments corresponding to actions or objects that are grounded in either the video content and the instruction context. We formulate the symbolic representation of a sounding event e as a structured tuple:

$$e = (\mathbf{t},\, d,\, \mathbf{p}),$$

where $\mathbf{t} = (t_{start}, t_{end})$ denotes the temporal location of the event along the video timeline, d is a semantic description about <subject, action, object> where object is optional, and $\mathbf{p}$ specifies controllable audio properties such as timbre, pitch, intensity, and spatialization.

3.2. Task Definition

Building on the symbolic formulation above, we define ***EchoFoley*** as producing audio tracks that reflect both the video context, and *faithfully satisfies the event-centric hierarchical control constraints* specified by the user's instruction. Given a video V and an instruction I, the corresponding set of sounding events $\mathcal{C}$ is denoted as:

$$\mathcal{C} = \{(\mathbf{t},\, d,\, \mathbf{p}) | V, I\}$$

The task reduced to video-audio generation if I is `Null`. The instruction can explicitly or implicitly specify the temporal location (*e.g.*, "`1 second-long` magical explosion at time stamp `0:03`", "`second` meow sound"), semantic description (*e.g.*, "`cat meows`", "`ball hits bottles`"), and attributes (*e.g.*, "`low pitch`", "`soft timbre`"), and any-level of combination.

We further organize the symbolic control space into hierarchical levels of sounding-event–centric control, where each level governs events at a different level of semantic and temporal abstraction. This hierarchy comprises three *control levels*:

- **Instance Level** — controls the properties of a single sounding event, such as emission or insertion of an event. (*e.g.*, "change the *second* meow into a lion roar").
- **Group Level** — coordinates multiple related events, enabling control over interactions, co-occurring actions, or repeated event sequences. (*e.g.*, "transform *all* cat meows in the video into lion-like vocalizations").
- **Video Level** — shapes the overall acoustic profile, balance, and distribution of all events throughout the video. (*e.g.*, "render the whole soundtrack with a cartoon-like audio aesthetic").

We also design three complementary *control types* independent to the control levels:

- **Temporal Control** — determines *when* a sounding event occurs and *how long* it lasts, regulating timestamps and durations (*e.g.*, "delay the explosion by one second").
- **Timbre Control** — specifies *what* an object should sound like by modifying auditory texture or identity (*e.g.*, "make the cat bark and the dog meow").
- **Volume Control** — adjusts *how strong or distant* a sound appears, manipulating volume depth (*e.g.*, "make the thunder louder").

This hierarchical design enables flexible and interpretable modulation of audio generation, from precise event-level adjustments to global scene-level control.

4. Benchmark: *EchoFoley-6k*

To enable systematic studies of the proposed task, we establish a comprehensive benchmark comprising (1) a large-scale, densely annotated dataset (***EchoFoley-6k***), and (2) an evaluation suite with both automatic and human assessments. In this section, we will introduce the dataset construction process (§4.1), summarize its key statistics (§4.2), and outline the evaluation protocols (§4.3).

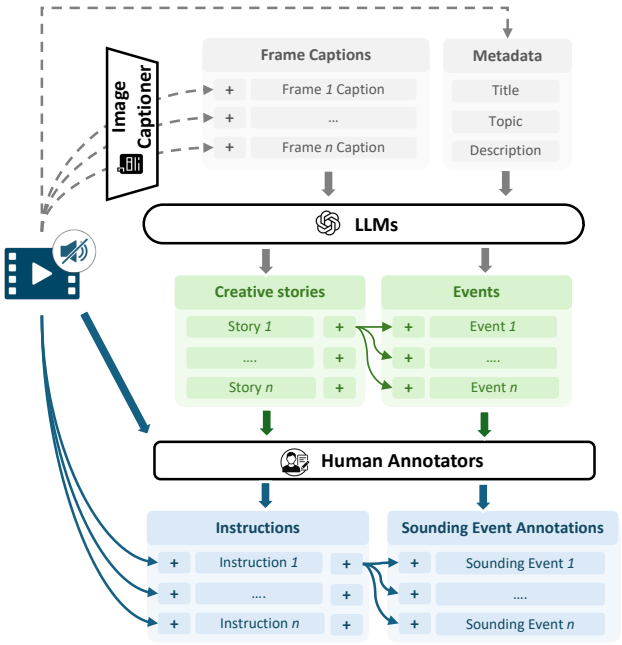

Figure 2. **Data Curation Pipeline** of ***EchoFoley-6k***

4.1. Data Curation

Figure 2 illustrates the data construction pipeline of ***EchoFoley-6k***. We sample candidate videos from VGGSound [6] and PE Video Dataset [2], followed by a series of processing and annotation steps (please refer to appendix for implementation details):

Step 1: Video Filtering. We begin with motion-centered videos where sound-producing interactions are visually evident—for example, an animal vocalizing, an object being struck, or an environment changing. This ensures that sounding events are grounded in the visible scene rather than arbitrary background audio.

Step 2: Metadata and Frame Captioning. For each video, we collect coarse metadata (title and short description) and generate frame-level visual captions that describe how the scene evolves moment by moment. These captions provide structured visual grounding and temporal cues for both narration and event identification.

Step 3: Story Proposal and Event Extractions. Using the metadata and captions, a large language model produces an imaginative story describing how sound could shape the scene and proposes an initial set of sounding events (*e.g.*, "where a meow starts, or when an object impact occurs"). These serve as high-level scaffolds rather than final labels.

Step 4: Human Modification. Human annotators then convert the creative story into concrete, fine-grained *instructions* describing how sounds should change over time (*e.g.*, "make the second meow sharper and more excited"), and refine the list of candidate sounding events by adjusting temporal boundaries and specifying auditory attributes such as intensity, pitch, and texture.

This process yields a dataset where each video is associated with multiple natural-language instructions and detailed sounding-event annotations.

Statistic	Number
Total Number of Videos	500
Video Topics	14
Avg Video Duration	11
Video Duration Range	6∼30 seconds
Total Sounding Events	3612
Avg. Sounding Events per Video	7.2
Total Instructions	6,000
Avg. Instructions per Video	12

Table 1. **Dataset Statistics for** ***EchoFoley-6k***.

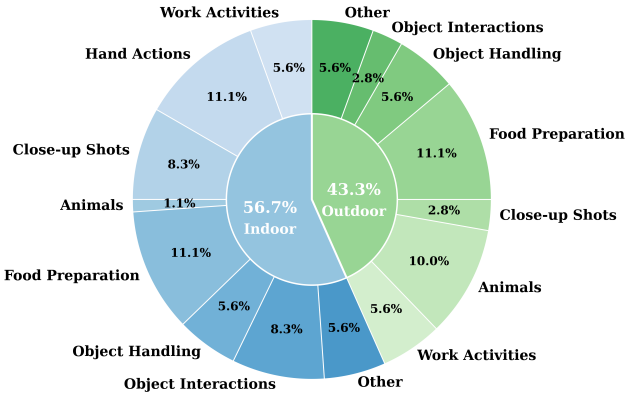

Figure 3. **Video Topic Distribution** in ***EchoFoley-6k***

4.2. Dataset Statistics

Table 1 and Figure 3 summarize the overall scale and diversity of ***EchoFoley-6k***. The benchmark comprises 6,000 video–instruction pairs and 42,000 fine-grained event annotations. Each video contains multiple instructions that provide temporal and attribute-level control cues, enabling detailed mapping between user intent and event-level sound synthesis. Across the dataset, videos span a wide range of visual scenes and auditory contexts (Figure 3). Table 1 summarizes the scale and statistics of ***EchoFoley-6k***.

4.3. Evaluation Suite

To assess objective performance, we design an evaluation suite with automatic and human evaluation metrics. More implementation details are in the appendix.

We design three automatic evaluation metrics to evaluate the controllability the event-level perspective, focusing when, what, and how the generated sound behaves.

When: Temporal Control. For each event e_i, we compare the ground-truth event duration $\mathbf{t} = (t_{start}, t_{end})$ with the predicted active region of the generated audio $\hat{\mathbf{t}}$

obtained from an AudioLLM-based onset/offset detector given the instruction. To measure how well the predicted event location aligns with the ground truth, we compute the intersection-over-union (IoU) of the two intervals:

$$\text{TempIoU}(e) = \frac{\left|\mathbf{t} \cap \hat{\mathbf{t}}\right|}{\left|\mathbf{t} \cup \hat{\mathbf{t}}\right|}.$$

The temporal controllability score for a single video is the average IoU across all events:

$$\text{TempCtl}(V, I) = \sum_{e \in \mathcal{C}} \text{TempIoU}(e),$$

where $\mathcal{C} = \{e|V, I\}$ is the ground-truth sounding event set given the video V and instruction I.

What: Timbre Control. To assess whether the generated sound matches the intended semantic identity of event e, we compute the semantic similarity between the audio segment and the event's semantic description d. The audio segment of the event is extracted from the video's original audio A, grounded by its temporal location $\mathbf{t}$. We then calculate the CLAP similarity score [40] between $A_{\mathbf{t}}$ and d:

$$\text{CLAP}(e) = \text{sim}\big(A_{\mathbf{t}},\, d\big).$$

The timbre controllability score for a single video is obtained by averaging CLAP scores across all the events:

$$\text{TimbCtl}(V, I) = \sum_{e \in \mathcal{C}} \text{CLAP}(e).$$

How: Volume Control. Volume characterizes the perceptual attributes of a sound, reflecting contextual cues like strength, spatial distance, emotion, which should be aligned with video or instruction context. To eliminate the impact of reasonable volume variation, we define three loundness levels: [*low*, *medium*, *high*]. To evaluate volume controllability, we compute the relative loudness of each generated audio segment $A_{\mathbf{t}}$ to the entire audio sequence A as $\ell_{\mathbf{t}}$ by

$$\ell_{\mathbf{t}} = \begin{cases} \text{low}, & \dfrac{\text{Loudness}(A_{\mathbf{t}})}{\text{Loudness}(A)} < \tau_1, \\ \text{medium}, & \tau_1 \leq \dfrac{\text{Loudness}(A_{\mathbf{t}})}{\text{Loudness}(A)} < \tau_2, \\ \text{high}, & \dfrac{\text{Loudness}(A_{\mathbf{t}})}{\text{Loudness}(A)} \geq \tau_2. \end{cases}$$

where τ_1 and τ_2 are predefined thresholds and $\text{Loudness}(\cdot)$ is a function that extracts the volume of an audio segment.

We define the depth controllability score as the average equivalence between the loudness $\ell_{\mathbf{t}}^{\text{gt}}$ of the ground-truth audio segment and $\ell_{\mathbf{t}}$:

$$\text{VolCtl} = \sum_{e \in \mathcal{C}} \mathbf{1}\left[\, \ell_{\mathbf{t}} = \ell_{\mathbf{t}}^{\text{gt}} \,\right].$$

To complement automatic metrics, we further conduct human evaluations to assess perceptual aspects that are difficult to capture algorithmically. Human raters score each generated audio clip on a five-point Likert scale along the following dimensions:

- **Instruction Adherence:** How well the audio follows the user instruction, including requested changes in timing, timbre, or loudness.
- **Audio–Visual Coherence:** How consistent and synchronized the generated sounds are with the visual content, including object actions, motion, and event boundaries.
- **Perceptual Quality:** The overall naturalness, clarity, and realism of the audio as perceived in the video context.

5. Evaluation

We evaluate current video-text-to-audio models on our ***EchoFoley-6k*** benchmark to test their (1) event-level hierarchical controllability and (2) generated audio quality via automatic and human assessment.

5.1. Main Evaluation Setup

Models. We evaluated 8 recent open-source video-text-to-audio generation models that support both visual and textual conditioning, including MMAudio [9], ThinkSound [26], AudioGenie [31], and HunyuanVideo-Foley [34].

Evaluation Metrics. We evaluate models with the *proposed* automatic and human metrics introduced in Section 4.3. For objective controllability, we report three automatic metrics: **TemporalCtl**, **TimbreCtl**, and **VolCtl**, which quantify accuracy in temporal alignment, timbre manipulation, and loudness modulation, respectively. For human evaluation, we use three subjective metrics: **Instruction Adherence**, **Audio–Visual Coherence**, and **Perceptual Quality**, which collectively measure controllability, cross-modal alignment, and perceptual naturalness of the generated audio. To further assess the intrinsic quality of the generated audio, we adopt the **Audio Aesthetics Score (AES)** [39]. AES decomposes audio aesthetics into four interpretable dimensions:

- **Production Quality (PQ):** Technical fidelity, including clarity, dynamic range, freq. balance, and spatial realism.
- **Production Complexity (PC):** Richness and structural complexity of elements within the audio scene.
- **Content Enjoyment (CE):** Subjective enjoyability, reflecting artistic expressiveness, emotional impact, and listener engagement.
- **Content Usefulness (CU):** Practical utility of the audio, capturing its potential reusability in creative workflows.

For human evaluation, we randomly sample 50 video–instruction pairs and recruit 6 participants to rate each generated audio. The inter-annotator agreement for human evaluation is 0.62 (Cohen's kappa).

Model	Automatic (Controllability)			Automatic (Audio Quality)				Human Evaluation		
	TempCtl	**TimbCtl**	**VolCtl**	**PQ**	**PC**	**CE**	**CU**	**Instr.**	**A–V Coh.**	**Qual.**
MMAudio-S-16kHz	0.26	0.21	0.52	6.22	2.79	3.18	5.69	1.60	3.20	3.07
MMAudio-S-44.1kHz	0.30	0.24	0.55	6.25	3.00	3.21	5.47	2.00	3.53	3.13
MMAudio-M-44.1kHz	0.28	0.23	0.54	6.06	2.79	3.11	4.88	1.60	3.60	3.13
MMAudio-L-44.1kHz	0.29	0.23	0.56	5.97	2.84	2.99	5.08	1.93	3.53	3.13
ThinkSound	0.18	0.34	0.50	6.49	3.03	3.45	5.94	1.53	2.20	2.00
AudioGenie	0.27	0.23	0.58	6.22	2.79	3.18	5.69	1.47	3.47	3.47
HunyuanVideo-Foley-xl	0.41	0.46	0.67	6.47	3.48	3.46	5.64	2.60	**4.20**	3.73
HunyuanVideo-Foley-xxl	0.43	0.48	0.69	6.49	3.45	3.49	5.66	2.53	4.07	3.67
EchoVidia	**0.72**	**0.78**	**0.75**	**7.32**	**4.29**	**4.33**	**6.50**	**3.80**	3.93	**3.79**

Table 2. **Main Evaluation Results** with metrics for Controllability (**TempCtl**: Temporal Controllability, **TimbCtl**: Timbre Controllability, **VolCtl**: Volume Controllability), Audio quality (**PQ**: Production Quality, **PC**: Production Complexity, **CE**: Content Enjoyment, **CU**: Content Usefulness), and human evaluation (**Instr.**: Instruction Adherence, **A–V Coh.**: Audio–Visual Coherence, **Qual.**: Perceptual Quality). The best-performing value per metric is **bolded**, and the second-best value is underlined.

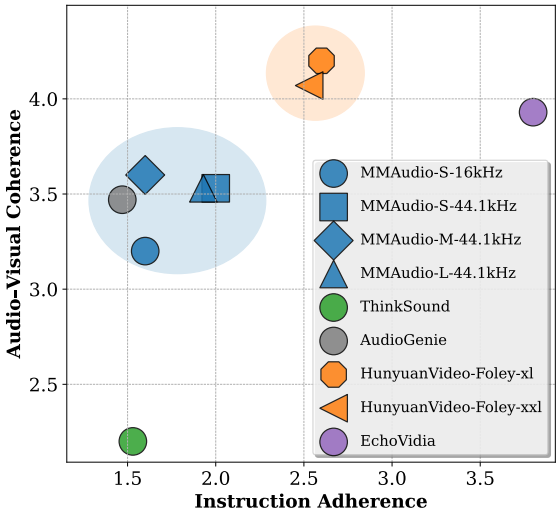

Figure 4. The relationship of Instruction Adherence (x-axis) plotted against Audio–Visual Coherence (y-axis) of models shows Visual Dominance Bias.

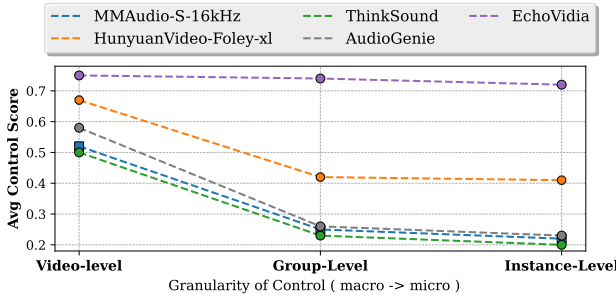

Figure 5. Average controllability scores for different levels of control granularity, from macro (global) to micro (atomic).

5.2. Main Evaluation Results and Analysis

Table 2 summarizes the performance of existing VT2A models across controllability, audio quality, and human evaluation metrics. We have three main insights as follows.

Lack of Controllability. Overall, current VT2A models demonstrate significant limitations in controllability, particularly for temporal and timbre dimensions. Temporal controllability remains low across all the baseline models (0.18∼0.43), revealing persistent difficulty in aligning generated audio with visual event timing. Timbre controllability shows similarly modest performance, with most models clustered around 0.21∼0.24, and only ThinkSound (0.34) and HunyuanVideo-Foley variants (0.46∼0.48) showing moderate gains. Volume controllability is comparatively higher (0.50∼0.69), suggesting that loudness modulation is inherently easier for current systems, though still far from reliable. Human evaluations reveal similar weaknesses, where annotators consistently report low *Instruction Adherence* (below 2.60 out of 5.0).

Visual Dominance with Text Neglect. Figure 4 reveals a consistent pattern across existing VT2A models: high Audio–Visual Coherence but low Instruction Adherence. This imbalance indicates a strong visual dominance effect—VT2A models tend to rely heavily on visual cues while largely ignoring the fine-grained controls specified in the text instruction. Most models cluster toward the upper–left region of the plot, showing that they are able to synchronize generated audio with visible events reasonably well (*e.g.*, Audio–Visual Coherence ranges in 3.2∼3.6), yet fail to follow detailed textual directives (*e.g.*, Instruction Adherence ranges in 1.4∼2.0). Models such as MMAudio illustrate this pattern most clearly, producing audio that matches what is seen on screen but rarely incorporates the requested temporal, timbral, or intensity adjustments. The HunyuanVideo-Foley variants achieve slightly higher adherence but still remain far from reliable, underscoring that current architectures do not effectively integrate textual control signals. This visual bias ultimately limits controllable generation: when visual cues and textual instructions conflict, models overwhelmingly favor the visual stream, disregarding the user-specified behaviors.

Model	Recall	F1 Score
Qwen3-VL-8B-Instruct	0.48	0.48
Qwen3-VL-8B-Thinking	0.44	0.50
Qwen3-VL-30B-Instruct	0.39	0.49
Qwen3-VL-30B-Thinking	0.45	0.53
OmniVinci	0.13	0.19
Gemini-2.5 Flash	0.66	0.50
Gemini-2.5 Pro	0.66	0.59
Qwen3-Omni-30B-Instruct	0.42	0.43
Qwen3-Omni-30B-Thinking	0.48	0.56
Gemini-2.5 Pro + SF	**0.83** $_{+0.17}$	**0.74** $_{+0.15}$
Qwen3-VL-30B-Thinking + SF	**0.54** $_{+0.09}$	**0.71** $_{+0.18}$

Table 3. **Task 1: Sounding Event Detection.** Evaluating the ability of models to enumerate and correctly identify all sounding events in a video. "+SF" denotes our proposed Slow–Fast reasoning strategy. Results are averaged over three runs.

The Finer-grained Level, the Worse Control. As shown in Figure 5, controllability decreases sharply as the control level becomes more fine-grained. All models achieve their highest scores under video control, where the instruction specifies coarse-grained or video-level behavior. However, performance drops substantially for group control, and even further for individual control, which requires precise manipulation of individual sounding events. This widening gap reveals that existing VT2A models struggle to localize, disentangle, and manipulate fine-grained event attributes.

5.3. Sounding Event Awareness

Beyond controllable audio generation, ***EchoFoley-6k*** also *naturally* functions as a rigorous benchmark for evaluating a model's **sounding event awareness**—its ability to detect and temporally localize sounding events in video. We design two complementary tasks to assess nine recent VideoLLMs and omni-modal foundation models.

Task 1: Sounding Event Detection. This task measures a model's capacity to exhaustively identify all sounding events occurring within a video, reflecting its fundamental awareness of audio-relevant visual cues. We evaluate nine models on 300 videos with human-verified sounding-event annotations with zero-shot, single-turn prompting protocol. Table 3 presents recall and F1 scores. Gemini-2.5 Pro demonstrates the highest performance. Omni-modal models generally outperform vision-centric VideoLLMs, indicating the importance of audio-aligned multimodal pretraining for event-level understanding.

Task 2: Sounding Event Localization. This task evaluates whether a model can not only detect sounding events but also accurately localize their temporal boundaries. Using the same 300 videos, we assess temporal alignment only for correctly predicted events. For each such event, we compare the model's predicted temporal span with the human-annotated ground truth and compute the IoU for measurement. Figure 6 presents IoU scores across models. All models exhibit notable boundary drift—indicating that while they often detect *which* events occur, they struggle to match the precise onset and offset times.

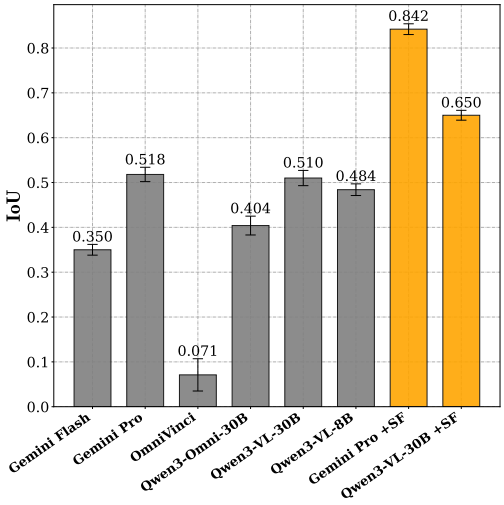

Figure 6. **Task 2: Sounding Event Localization.** Intersection-over-Union between predicted and ground-truth temporal spans, computed only on correctly detected events.

6. Method: *EchoVidia*

To address the limitations of existing video-to-audio generation models on lack of fine-grained controllability and vision dominance, we propose *EchoVidia*, a training-free agentic framework with slow-fast thinking. In this section, we first introduces the design and core components of the proposed method (§6.1), and then presents the experimental setup and results (§6.2).

6.1. Overview

Slow–Fast Thinking Strategy. As analyzed in Section 5.3, current VideoLLMs exhibit weak awareness of sounding events and struggle with accurate temporal grounding. Inspired by dual-process cognition where System 1 performs fast intuitive reasoning and System 2 conducts slow analytical reasoning, we first introduce a *slow–fast thinking* strategy (SF) to enhance event understanding and timestamp localization. The *fast thinking* pathway captures a global overview of the video in 1 fps, summarizing its high-level structure and coarse auditory context. The *slow thinking* pathway performs detailed reasoning by viewing its 16x slower-motion video. Specifically, we first downsample the video to 16 fps, then temporally stretch it by 16x to obtain the video at 1 fps), enabling precise event localization and attribute inference.

Architecture. The core of *EchoVidia* is an agentic framework that structures the generation process into three interconnected stages: reasoning, design, and synthesis. At its center is a VideoLLM-based agent that interacts with

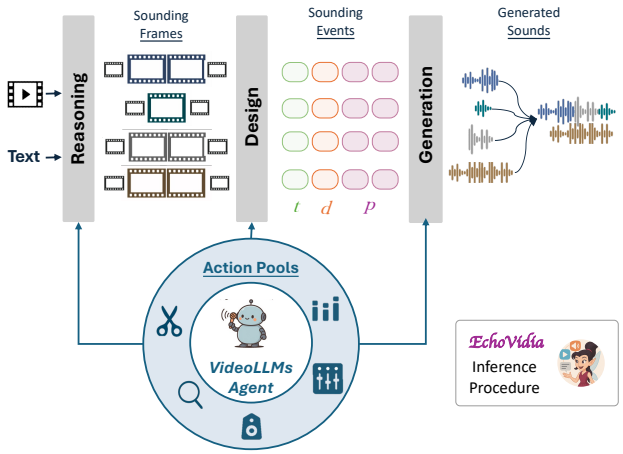

Figure 7. *EchoVidia* Inference Procedure

an *action pool*—a set of 12 atomic operations spanning visual reasoning and sound generation (see appendix for more details). Within this pool, video reasoning actions identify sounding events, retrieve temporal cues, and crop relevant visual segments. Sound design actions allow the agent to add, remove, or modify event representations, controlling their semantic attributes and temporal properties. Finally, generation actions synthesize, adjust, and mix the resulting audio layers, ensuring both temporal alignment and perceptual coherence between modalities.

Inference Procedure. As illustrated in Figure 7, during inference, the agent first analyzes the input video to identify potential sounding events and estimate their approximate timing. It then constructs a symbolic event plan describing how each event should sound and iteratively refines this plan through reasoning and editing actions. The finalized symbolic representation is passed to the sound generation module, which renders the corresponding audio conditioned on both visual and textual contexts.

6.2. Results

We compare our *EchoVidia* with other video-text-to-audio models following the evaluation settings in Section 5.

Performance on *EchoFoley*. *EchoVidia* achieves the best performance across most evaluation dimensions, outperforming every baseline model by a substantial margin, as shown in Table 2. For controllability, *EchoVidia* reaches 0.72 TemporalCtl, 0.78 TimbreCtl, and 0.75 DepthCtl, corresponding to average improvements of roughly 55% over the strongest baseline (outperforming the best baseline by +0.29, +0.30, and +0.19, respectively). Human evaluation further confirms the advantage: *EchoVidia* attains 3.80 Instruction Adherence, 3.93 Audio–Visual Coherence, and 3.79 Perceptual Quality, consistently outperforming the strongest baseline by +1.20, +0.40, and +0.32, respectively.

Figure 4 illustrates that *EchoVidia* achieves a *balanced integration of visual and textual conditioning*. Whereas existing models cluster in the upper-left region—showing reasonable Audio–Visual Coherence but poor Instruction Adherence—*EchoVidia* uniquely attains simultaneously high scores on both axes. This shows that *EchoVidia* not only synchronizes sound with visual content but also executes fine-grained textual controls faithfully, eliminating the visual-dominance bias observed in previous VT2A systems.

Ablation on Slow–Fast (SF) Thinking Strategy. As shown in Table 3 and Figure 6, models equipped with the SF reasoning strategy exhibit **substantial gains** in sounding-event awareness. For T1, SF boosts recall from 0.66 to 0.83 for Gemini-2.5 Pro and increases the F1 score from 0.59 to 0.74, representing the highest scores among all evaluated models. Similarly, Qwen3-VL-30B-Thinking benefits noticeably from SF, achieving a +0.15 gain in the F1 score. Beyond enumeration, SF also delivers strong gains in timestamp accuracy: for T2, the IoU of Gemini-2.5 Pro increases from 0.510 to 0.842, and Qwen3-VL-30B improves from 0.484 to 0.650, corresponding to more than a 60% relative improvement in temporal precision.

6.3. Border Impact

Our proposed framework opens up new opportunities for controllable and interpretable VT2A generation. By enabling fine-grained instruction-guided sound control, it can benefit multiple downstream applications, including video editing, film post-production, and accessible multimedia creation. The data curation pipeline for ***EchoFoley-6k*** and the *EchoVidia* model together provide a scalable data generation pipeline for synthesizing high-quality sounding videos with rich, aligned multimodal annotations. Such controllable generation pipelines not only enhance content creation but also serve as a valuable source of training data for large-scale *world models* and *omni-modal foundation models*, which aim to unify perception, reasoning, and generation across modalities.

7. Conclusion

In conclusion, we introduced ***EchoFoley***, a new task focusing on fine-grained video-grounded sound generation with event-level hierarchical control. By shifting the focus from coarse video-level prompts to symbolic, temporally grounded sounding events, our formulation provides a principled way to specify *what* sound should occur, *when* it should occur, and *how* it should evolve. To support systematic study of this task, we constructed ***EchoFoley-6k***, a large-scale, expertly curated benchmark and metrics. We developed *EchoVidia*, a training-free agentic framework with slow–fast thinking that significantly improves controllability, grounding fidelity, and perceptual quality. Future research may integrate our event-centric formulation into end-to-end trainable models, and extend the symbolic event representation to more application to further broaden the

creative generation capabilities of generative model. We hope this work inspires future research toward omni-modal generative intelligence that can both understand and recreate the multimodal richness of the real world.

Acknowledgments

We thank Andong Deng, Zhenfang Chen, Dawei Du, Sijie Zhu for their insightful feedback and discussion on the work.

References

[1] BBC Sound Effects — sound-effects.bbcrewind.co.uk. https://sound-effects.bbcrewind.co.uk. [Accessed 22-10-2025]. 2

[2] Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, et al. Perception encoder: The best visual embeddings are not at the output of the network. In *NIPS*, 2025. 4

[3] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. 1

[4] Changan Chen, Kumar Ashutosh, Rohit Girdhar, David Harwath, and Kristen Grauman. Soundingactions: Learning how actions sound from narrated egocentric videos. In *CVPR*, 2024. 3

[5] Changan Chen, Puyuan Peng, Ami Baid, Sherry Xue, Wei-Ning Hsu, David Harwath, and Kristen Grauman. Action2sound: Ambient-aware generation of action sounds from egocentric videos. In *ECCV*, 2024. 2

[6] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *ICASSP*, 2020. 2, 4

[7] Zihao Chen, Haomin Zhang, Xinhan Di, Haoyu Wang, Sizhe Shan, Junjie Zheng, Yunming Liang, Yihan Fan, Xinfa Zhu, Wenjie Tian, et al. Yingsound: Video-guided sound effects generation with multi-modal chain-of-thought controls. *arXiv preprint arXiv:2412.09168*, 2024. 2

[8] Ziyang Chen, Prem Seetharaman, Bryan Russell, Oriol Nieto, David Bourgin, Andrew Owens, and Justin Salamon. Video-guided foley sound generation with multimodal controls. In *CVPR*, 2025. 2

[9] Ho Kei Cheng, Masato Ishii, Akio Hayakawa, Takashi Shibuya, Alexander Schwing, and Yuki Mitsufuji. Mmaudio: Taming multimodal joint training for high-quality video-to-audio synthesis. In *CVPR*, 2025. 2, 5

[10] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 3, 2

[11] Baoyu Fan, Lu Liu, Xiaochuan Li, Runze Zhang, Liang Jin, and Jin Zhang. Fine-grained audio–visual event localization. *IEEE Transactions on Neural Networks and Learning Systems*, 2025. 3

[12] Chaoyou Fu, Haojia Lin, Zuwei Long, Yunhang Shen, Meng Zhao, Yifan Zhang, Shaoqi Dong, Xiong Wang, Di Yin, Long Ma, et al. Vita: Towards open-source interactive omni multimodal llm. *arXiv:2408.05211*, 2024. 3

[13] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. TALL: temporal activity localization via language query. In *ICCV*, 2017. 3

[14] Junyu Gao, Mengyuan Chen, and Changsheng Xu. Collecting cross-modal presence-absence evidence for weakly-supervised audio-visual event perception. In *CVPR*, 2023. 3

[15] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *ICASSP*, 2017. 2

[16] Tiantian Geng, Jinrui Zhang, Qingni Wang, Teng Wang, Jinming Duan, and Feng Zheng. Longvale: Vision-audio-language-event benchmark towards time-aware omni-modal perception of long videos. In *CVPR*, 2025. 3

[17] Rajat Hebbar, Digbalay Bose, Krishna Somandepalli, Veena Vijai, and Shrikanth Narayanan. A dataset for audio-visual sound event detection in movies. In *ICASSP*, 2023. 3

[18] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan C. Russell. Localizing moments in video with natural language. In *ICCV*, 2017. 3

[19] Chao Huang, Yapeng Tian, Anurag Kumar, and Chenliang Xu. Egocentric audio-visual object localization. In *CVPR*, 2023. 3

[20] Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. Audiocaps: Generating captions for audios in the wild. In *NAACL-HLT*, 2019. 2

[21] Jie Lei, Tamara L. Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. In *NIPS*, 2021. 3

[22] Bingxuan Li, Yiwei Wang, Jiuxiang Gu, Kai-Wei Chang, and Nanyun Peng. Metal: A multi-agent framework for chart generation with test-time scaling. In *ACL*, 2025. 2

[23] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv:2305.06355*, 2023. 3

[24] Yingshan Liang, Keyu Fan, Zhicheng Du, Yiran Wang, Qingyang Shi, Xinyu Zhang, Jiasheng Lu, and Peiwu Qin. Hear-your-click: Interactive video-to-audio generation via object-aware contrastive audio-visual fine-tuning, 2025. 2

[25] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. In *EMNLP*, 2024. 3

[26] Huadai Liu, Jialei Wang, Kaicheng Luo, Wen Wang, Qian Chen, Zhou Zhao, and Wei Xue. Thinksound: Chain-of-thought reasoning in multimodal large language models for audio generation and editing. In *NIPS*, 2025. 2, 5

[27] Simian Luo, Chuanhao Yan, Chenxu Hu, and Hang Zhao. Diff-foley: Synchronized video-to-audio synthesis with latent diffusion models. In *NIPS*, 2023. 2

[28] Tanvir Mahmud and Diana Marculescu. Ave-clip: Audioclip-based multi-window temporal transformer for audio visual event localization. In *WACV*, 2023. 3

[29] Xinhao Mei, Chutong Meng, Haohe Liu, Qiuqiang Kong, Tom Ko, Chengqi Zhao, Mark D Plumbley, Yuexian Zou, and Wenwu Wang. Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:3339–3354, 2024. 2

[30] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024. 1

[31] Yan Rong, Jinting Wang, Guangzhi Lei, Shan Yang, and Li Liu. Audiogenie: A training-free multi-agent framework for diverse multimodality-to-multiaudio generation. In *ACM Multimedia*, 2025. 2, 5

[32] Ciara Rowles, Varun Jampani, Simon Donné, Shimon Vainer, Julian Parker, and Zach Evans. Foley control: Aligning a frozen latent text-to-audio model to video. *arXiv preprint arXiv:2510.21581*, 2025. 2

[33] Chitwan Saharia, William Chan, Saurabh Saxena, Lili Li, Jay Whang, Emily Denton, Gabriel Goh, Antoine Sablayrolles, Ishan Misra, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *NIPS*, 2022. 1

[34] Sizhe Shan, Qiulin Li, Yutao Cui, Miles Yang, Yuehai Wang, Qun Yang, Jin Zhou, and Zhao Zhong. Hunyuanvideo-foley: Multimodal diffusion with representation alignment for high-fidelity foley audio generation. *arXiv preprint arXiv:2508.16930*, 2025. 2, 5

[35] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Xun Guo, Tian Ye, Yan Lu, Jenq-Neng Hwang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *CVPR*, 2024. 3

[36] Vidi Team, Celong Liu, Chia-Wen Kuo, Dawei Du, Fan Chen, Guang Chen, Jiamin Yuan, Lingxi Zhang, Lu Guo, Lusha Li, et al. Vidi: Large multimodal models for video understanding and editing. *arXiv preprint arXiv:2504.15681*, 2025. 3

[37] Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. Audio-visual event localization in unconstrained videos. In *ECCV*, 2018. 2

[38] Yapeng Tian, Dingzeyu Li, and Chenliang Xu. Unified multisensory perception: Weakly-supervised audio-visual video parsing. In *ECCV*, 2020. 2

[39] Andros Tjandra, Yi-Chiao Wu, Baishan Guo, John Hoffman, Brian Ellis, Apoorv Vyas, Bowen Shi, Sanyuan Chen, Matt Le, Nick Zacharov, Carleigh Wood, Ann Lee, and Wei-Ning Hsu. Meta audiobox aesthetics: Unified automatic quality assessment for speech, music, and sound. 2025. 5

[40] Yusong Wu*, Ke Chen*, Tianyu Zhang*, Yuchen Hui*, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *ICASSP*, 2023. 5

[41] Zhifeng Xie, Shengye Yu, Qile He, and Mengtian Li. Sonicvisionlm: Playing sound with vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26866–26875, 2024. 2

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

[43] Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In *CVPR*, 2023. 3

[44] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *EMNLP*, 2023. 3

[45] Lin Zhang, Shentong Mo, Yijing Zhang, and Pedro Morgado. Audio-synchronized visual animation. In *ECCV*, 2024. 2

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

[47] Songyan Zhao, Bingxuan Li, Yufei Tian, and Nanyun Peng. Reffly: Melody-constrained lyrics editing model. In *NAACL*, 2025. 2

[48] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, 2018. 3

[49] Jinxing Zhou, Dan Guo, Ruohao Guo, Yuxin Mao, Jingjing Hu, Yiran Zhong, Xiaojun Chang, and Meng Wang. Towards open-vocabulary audio-visual event localization. In *CVPR*, 2025. 3

[50] Yu Zhou, Bingxuan Li, Mohan Tang, Xiaomeng Jin, Te-Lin Wu, Kuan-Hao Huang, Heng Ji, Kai-Wei Chang, and Nanyun Peng. Contrastive visual data augmentation. In *ICML*, 2025. 2

Appendix Contents

A. Details on Benchmark: EchoFoley-6k

A.1. Data Example

Each data point in EchoFoley-6k follows the triplet format ⟨video, instruction, sounding-event annotations⟩.

- **Video**: a motion-centered clip (6–30 seconds) where sound-producing interactions are visually evident.
- **Instruction**: a natural-language directive indicating how sounds should be generated, edited, or transformed. Instructions may specify instance-, group-, or video-level control (e.g., "change the second meow into a lion roar" or "make all prior sounds louder").
- **Sounding-Event Annotations**: a set of symbolic sounding events $e = (t, d, p)$, where $t = (t_{\text{start}}, t_{\text{end}})$ denotes the temporal span, d is a semantic description, and p contains controllable auditory attributes (pitch, timbre, loudness, intensity, spatialization).

Figure 1 in the main paper shows an example illustrating how temporal boundaries, semantic descriptions, and controllable properties interact with instructions.

A.2. Comparison with Existing Datasets

Existing audio–visual datasets primarily support coarse audio modeling and do not provide fine-grained event boundaries or hierarchical controllability. EchoFoley-6k fills this gap by jointly providing:

- dense event-level temporal boundaries,
- symbolic, multi-attribute sounding-event representations,
- natural-language instructions for hierarchical control.

Dataset	Modality	Scale	Temporal Ann.	Instructions
VGGSound	V–A	200k	No	No
AudioSet	A–T	2M	No	No
AVVP	V–A	4k	Weak	No
EchoFoley-6k	V–I–E	**6k**	**Yes (dense)**	**Yes**

Table 4. Comparison between EchoFoley-6k and representative audio–visual datasets. V: video, A: audio, T: text, I: instruction, E: event annotations.

EchoFoley-6k is constructed through a hybrid LLM-assisted and human-refined workflow. Large language models first propose creative narratives and coarse candidate event lists, offering high-level interpretations of possible auditory scenes. Human annotators then refine these outputs by adjusting event boundaries to frame-level precision, correcting and enriching semantic labels and object–action descriptions, and annotating detailed auditory attributes such as pitch, timbre, loudness, intensity, and spatialization. They also rewrite and standardize natural-language instructions to ensure clarity and controllability. Compared to datasets that rely solely on short categorical sound tags, EchoFoley-6k provides structured, interpretable, and temporally precise annotations that are essential for fine-grained, controllable sound generation.

A.3. Data Curation Details

The full data construction pipeline is illustrated in Figure 2 of the main paper. Below we describe each stage in detail.

A.3.1. Curation Process Details

1. **Video Filtering**. We begin with motion-centered videos containing visually identifiable sound-producing actions (e.g., collisions, vocalizations, material interactions). Videos with ambiguous, off-screen, or purely ambient sounds are removed to ensure that sounding events remain visually grounded.
2. **Metadata and Frame Captioning**. For each candidate video, we extract its metadata (title and textual description) and generate frame-level visual captions at 16 fps using Gemini 2.5 pro[10]. These captions summarize evolving object configurations, motions, and interactions, providing structured visual grounding for downstream event extraction and reasoning.
3. **Story Proposal and Event Extraction**. Using GPT-5, we generate imaginative narratives describing plausible auditory interpretations of the scene. The model also proposes an initial list of hypothesized sounding events, each with an approximate temporal region and a short semantic description. These machine-generated proposals serve as high-level scaffolds and are not treated as final annotations.

4. **Human Refinement**. Human annotators transform the model-generated narratives into executable, fine-grained instructions; refine all event boundaries using frame-by-frame inspection; remove any hallucinated or visually unsupported events; and annotate rich, multi-attribute sound properties (e.g., pitch, timbre, volume, intensity, spatialization). This stage yields temporally accurate and instruction-aligned sounding-event annotations of high quality.

A.3.2. Human Refinement Instruction

Context. Human annotators verify and refine AI-generated instructions and their corresponding manipulated sound annotations. The goal is to ensure that the final dataset is logically coherent, unambiguous, and free of AI errors.

General Principles. Annotators are provided with a silent video, an AI-generated instruction, and the corresponding manipulated annotation file. They are asked to validate the following three aspects:

- *Logical Coherence.* The instruction must be plausible within the visual world of the video. The described sound should be something that could reasonably occur in the depicted environment, even if it is not currently present. The key question is: *Could this sound plausibly occur in this scene?*
- *Instructional Clarity.* The instruction must be specific and unambiguous, allowing an annotator to execute it without guessing the intended operation. The key question is: *Do I know exactly what action to take without having to infer missing details?*
- *Annotation Accuracy.* The manipulated annotation file must be an exact execution of the validated instruction, with no inconsistencies in timing, wording, or structure. The key question is: *Is the final annotation a perfect realization of the instruction, with zero errors?*

Core Annotation Tasks.

T1. Verify the Instruction. Annotators first decide whether the instruction is both plausible and clear:

- **Plausibility.** Annotators watch the video and check whether the instruction is logically grounded in the scene. Instructions that cannot be reconciled with the visual context (e.g., adding an animal sound to a scene with no animals or outdoor context) are rejected.
- **Clarity.** Annotators read the instruction and determine whether it is precise enough to be executed. Instructions that are vague or subjective (e.g., "make the sound more exciting") are flagged, with a brief comment explaining the source of ambiguity.

T2. Verify and Correct the Annotation. If the instruction passes T1, annotators then meticulously verify and correct the AI-generated manipulated annotation:

- **Timestamps.** For instructions that shift or rescale time (e.g., "start 1.5 seconds earlier"), annotators recompute the new timestamps themselves (e.g., 00:05.000 $\rightarrow$ 00:03.500) and confirm that all start and end times are exact and consistent.
- **Descriptions.** When textual descriptions are modified, annotators check that the changes are integrated fluently (e.g., adding "metallic" yields "a metallic chirp" rather than ungrammatical phrasing) and that the semantics remain faithful to the instruction.
- **Structure.** When events are added, removed, or reordered, annotators scan the full event list to ensure that only the intended events have been modified, and that no unrelated events have been accidentally altered or deleted.

This procedure ensures that every example in the dataset results from a valid, human-verified instruction and a perfectly aligned manipulated annotation.

A.4. Automatic Evaluation Metrics Implementation Details

A.4.1. Temporal Control

To evaluate how accurately the generated audio follows the intended timing of each sounding event, we measure temporal controllability by comparing the annotated event interval with the event's predicted start and end times extracted from the generated audio.

Audio preprocessing. The generated waveform is resampled to 16 kHz and converted into a 64-bin log-mel spectrogram using 25 ms Hann windows with a 10 ms hop size. The spectrogram is normalized and used as input to the temporal localization module.

Gemini-based onset/offset prediction. For temporal localization, we directly query a Gemini-based audio–text model with the generated audio segment and the textual description of the event. Gemini is prompted to determine when the described sound first appears and when it ends, returning a predicted start and end time in seconds. We use the following prompt to guide Gemini in predicting the temporal boundaries of each event:

```
You are an expert audio analyst.
Given an audio clip and a textual description of a sound event,
identify when this event starts and when it ends.

The event description is:
"{EVENT_DESCRIPTION}"

Please listen to the audio and return:
- The start time of this event (in seconds)
- The end time of this event (in seconds)

If the event occurs multiple times, choose the occurrence
that best matches the description. If unsure, choose the
most prominent occurrence.

Output your answer in the following JSON format:
{
  "start_time": <float, seconds>,
  "end_time": <float, seconds>
}
```

Interval overlap estimation. The predicted interval is compared with the annotated interval to assess alignment. Strong overlap indicates precise temporal localization, while little or no overlap reflects temporal drift. The temporal controllability score for a video is computed by averaging the alignment scores across all instruction-relevant events.

```
Pseudocode for Temporal Controllability (TempCtl)

Input:
    - Ground-truth events C
    - For each event: annotated timestamp
    - Generated audio A_hat
    - Gemini-based boundary predictor D_time

Procedure:
    score = 0
    For each event e in C:
        # Ask Gemini to predict event boundaries
        t_pred = D_time(A_hat, e.description)

        # Compute overlap with ground truth
        inter_len = intersection_length(annotated(e), t_pred)
        union_len = union_length(annotated(e), t_pred)

        if union_len > 0:
            temp_score = inter_len / union_len
        else:
            temp_score = 0

        score += temp_score

Output:
    TempCtl = score / |C|
```

A.4.2. Timbre Control

Timbre controllability measures whether the generated audio segment for each event matches the intended sound identity specified by the instruction.

Audio cropping. For each event, the annotated start and end times are converted into sample indices and used to extract the corresponding waveform segment. If the segment is longer than the analysis window, a sliding window strategy is used; if shorter, zero-padding is applied.

CLAP-based semantic alignment. We employ a CLAP audio–text encoder to compute timbre similarity. The cropped audio is resampled to 48 kHz, converted to mono, amplitude-normalized, and fed to CLAP's audio encoder. The event description is processed by the CLAP text encoder. Both encoders output normalized embeddings, and a cosine similarity

score reflects how well the generated sound matches the desired auditory identity.

Aggregation. Scores across all events described or modified by the instruction are averaged to produce the timbre controllability score.

```
Pseudocode for Timbre Controllability (TimbCtl)

Input:
    - Event set C
    - For each event: timestamp, description
    - Generated audio A_hat
    - CLAP module

Procedure:
    score = 0
    For each event e in C:
        segment = crop_audio(A_hat, e.timestamp)
        seg_proc = CLAP_preprocess(segment)
        sim = CLAP(seg_proc, e.description)
        score += sim

Output:
    TimbCtl = score / |C|
```

A.4.3. Volume Control

Volume controllability evaluates whether the generated audio reflects the intended loudness pattern for each sounding event, relative to the global loudness of the track.

Perceptual loudness extraction. We compute loudness using a perceptual RMS measure based on the ITU-R BS.1770 K-weighting filter. This produces a perceptually grounded measure for both the full generated audio and each event segment.

Relative loudness classification. To make the evaluation invariant to global gain differences, each event's loudness is normalized by the global loudness of the audio. The resulting ratio is mapped into "low", "medium", or "high" categories using two thresholds calibrated on the development set.

Scoring. A prediction is correct if the generated loudness category matches the annotated label. The final score is the proportion of correctly classified events.

```
Pseudocode for Volume Controllability (VolCtl)

Input:
    - Event set C
    - For each event: timestamp, loudness label
    - Generated audio A_hat
    - Loudness(): perceptual RMS function
    - Thresholds tau1, tau2

Procedure:
    global_L = Loudness(A_hat)

    correct = 0
    For each event e in C:
        seg = crop_audio(A_hat, e.timestamp)
        seg_L = Loudness(seg)
        r = seg_L / global_L

        if r < tau1:      pred = "low"
        elif r < tau2:    pred = "medium"
        else:             pred = "high"

        if pred == e.loudness_label:
            correct += 1

Output:
    VolCtl = correct / |C|
```

A.5. Human Evaluation Details

A.5.1. Survey Design

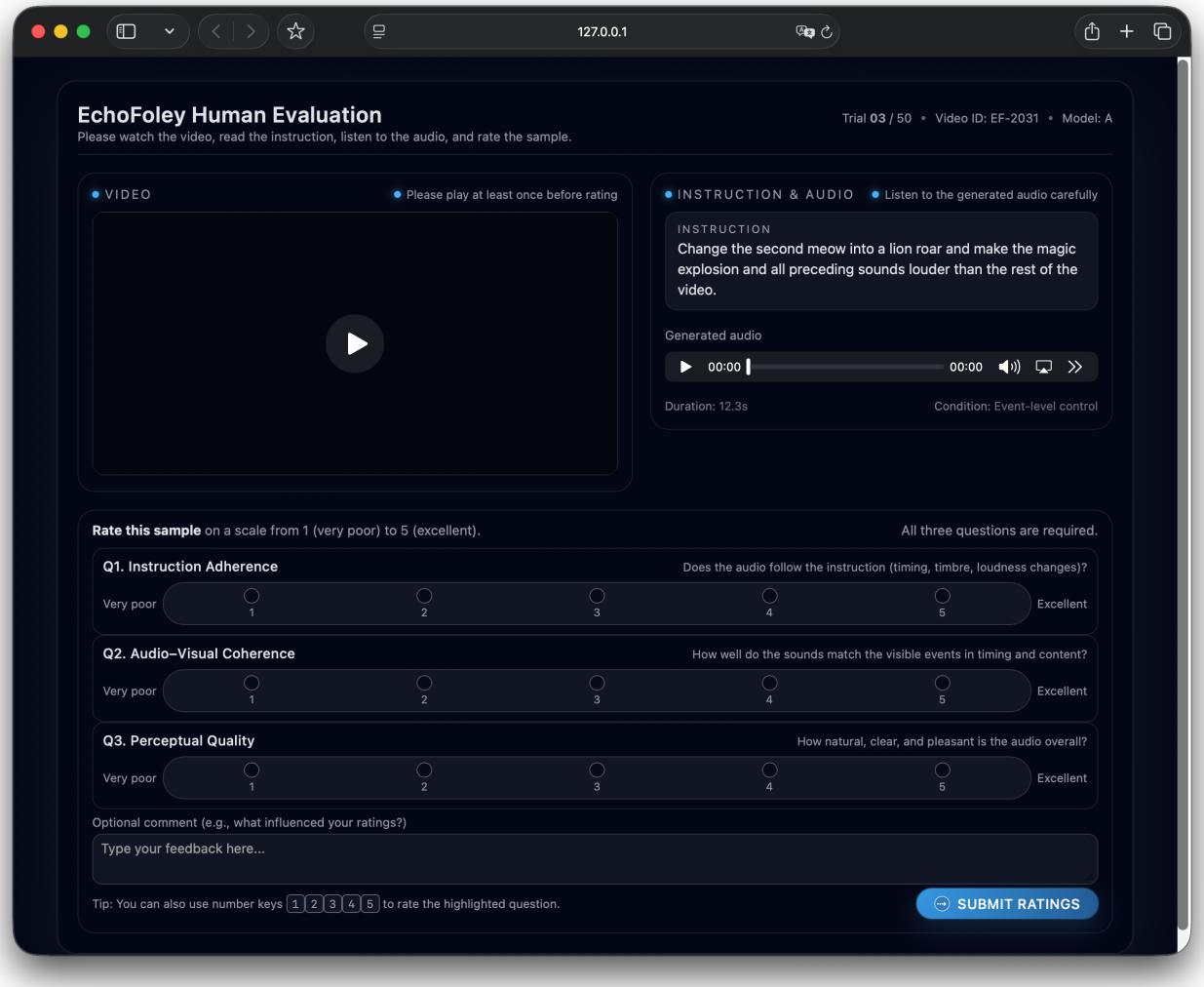

Figure 8. Human Evaluation UI

We design a three-part human evaluation to assess perceptual aspects of controllable video-to-audio generation that are difficult to capture algorithmically. Annotators rate each generated audio clip on a 1–5 Likert scale along the following dimensions:

- **Instruction Adherence:** How well the generated audio follows the user's instruction, including requested changes in timing, timbre, or loudness.
- **Audio–Visual Coherence:** How consistent the audio is with the visual content, including synchronization with object actions, motion dynamics, and event boundaries.
- **Perceptual Quality:** The overall naturalness, clarity, and realism of the generated audio within the video context.

Annotators are also provided with the original silent video, the instruction, and the generated audio to ensure consistent evaluation conditions.

A.5.2. Human Evaluation Procedure

We randomly sample 50 video–instruction pairs from EchoFoley-6k and generate outputs from all evaluated models. Six annotators independently rate each audio clip using the survey described above. To minimize potential bias, annotators are not informed of the identity of the model that produced each clip, and all clips are presented in randomized order.

For each metric, we compute the average score across annotators and clips. Inter-annotator agreement is measured using Cohen's kappa, resulting in a value of 0.62, indicating substantial agreement across raters. This procedure provides a reliable perceptual assessment complementing the automatic controllability metrics.

A.6. Evaluation Experiment Setup Details

For sound generation evaluation, we random choose 100 cases from ***EchoFoley-6k*** for evaluation. The baseline model configuration remain default for all models. Note that since AudioGenie is a huge frameowrk contains irrelevant packaged and models for other tasks (e.g. Text-to-Music, Text-to-Speech, etc.), for the sake of depolyment efficency, we implement the minimal version with only Video-Text-To-Audio functions avalibile, the remaining structure remain unchanged.

For sounding event awareness evaluation, we run 3 times for all unique videos in our dataset for each model, and take the average. We use all the default parameters (e.g. temprature, top-p, etc.) for the model.

B. Details on Method: *EchoVidia*

This section provides additional details of the proposed *EchoVidia* framework, including the slow–fast thinking strategy, action-pool architecture, implementation details, and limitations.

B.1. Slow{Fast Thinking Strategy Details

The Slow–Fast (SF) Thinking strategy is designed to compensate for the limited sounding-event awareness found in current VideoLLMs. As detailed in Sec. 6 of the main paper, SF integrates two complementary temporal reasoning pathways:

Fast Thinking (Global View). We extract a 1 fps version of the video, preserving only coarse temporal structure. This compressed view encourages the VideoLLM to capture global scene dynamics, high-level semantic context, and broad sequencing of potential sounding events. These global cues help the model form initial hypotheses regarding (1) what categories of events may occur, (2) their approximate ordering, and (3) the overall auditory context (e.g., repetitive actions, scene transitions).

Slow Thinking (Fine-Grained View). To support precise timestamp localization, we downsample the input video to 16 fps and then temporally stretch it by a factor of 16×. This effectively presents the model with an ultra-slow-motion view, enabling accurate inspection of subtle motions (e.g., object impacts, mouth articulations, or momentary gestures) that correspond to sounding events. The slow-stream reasoning significantly improves the detection of event boundaries and facilitates fine-grained attribute inference (e.g., intensity, pitch proxy, or spatial clues).

Integration. The VideoLLM processes both views independently. We aggregate their outputs by: (a) merging candidate sounding events; (b) reconciling coarse timestamps with fine-grained slow-view refinements; and (c) resolving inconsistencies by prioritizing slow-view boundaries when conflict arises. This integration forms the event plan used in the symbolic representation.

B.2. Framework Architecture Details

EchoVidia is implemented as an agentic multi-stage pipeline in which a VideoLLM-based controller sequentially invokes a set of atomic operations, referred to as the *action pool*. Instead of relying on a monolithic forward pass, EchoVidia decomposes video-to-audio generation into three explicit phases—*reasoning*, *sound design*, and *synthesis*. Each phase is realized through modular actions with well-defined inputs, outputs, and side effects. This decomposition enables the agent to iteratively refine symbolic sounding-event representations, perform targeted corrections, and maintain global temporal consistency.

Execution Flow. During inference, the agent begins with a high-level reasoning pass in which it identifies sounding events, estimates their temporal structure, and formulates an initial symbolic plan. It then repeatedly applies sound-design actions to modify the event plan in accordance with user instructions, ensuring correct temporal alignment and attribute-level control. Finally, the agent invokes generation actions to render each event as audio and perform mixing operations to synthesize the final output waveform.

Action Abstractions. EchoVidia treats each action in the pool as a callable transformation governed by a standardized interface:

- **Input:** a structured state object consisting of the video clip, optional cropped subclips, the current symbolic event plan, and the user instruction.
- **Operation:** an atomic transformation that updates either (i) the *event latent state* (i.e., symbolic representation), or (ii) the *audio latent state* (i.e., generator instructions), or (iii) the *visual latent state* (i.e., crop or resample metadata).
- **Output:** an updated state object that becomes the input for the next action.

This design enables multi-step planning, rollback of intermediate errors, and flexible recomposition of operations depending on the complexity of the instruction.

Action Pool. Table 5 summarizes the full set of actions used by the agent. Video reasoning actions support fine-grained temporal grounding and visual context extraction; sound-design actions manipulate the symbolic representation of sounding events; and sound-generation actions interface with the audio synthesis backend. The modular nature of the action pool allows the agent to construct arbitrarily complex behaviors through few-shot prompting rather than bespoke training.

Action Category	Description
Video Reasoning Actions	
Query sounding event	Retrieve sounding events based on a semantic query (e.g., "the second meow").
Query timestamp	Estimate onset/offset timestamps or refine event boundaries.
Crop video footage	Extract a subclip for localized inspection during slow–fast reasoning.
Adjust video speed	Present the video at altered framerates (slow-motion or accelerated) to improve temporal precision.
Sound Generation Actions	
Generate audio	Synthesize an audio segment given its symbolic description and temporal span.
Tune audio volume	Adjust loudness or relative gain for a target event.
Mix audio tracks	Merge event-level audio layers with crossfading and temporal alignment.
Sound Design Actions	
Add event	Insert a new sounding event into the event plan.
Delete event	Remove an existing event from the symbolic representation.
Modify event description	Change semantic attributes (e.g., "cat meow" $\rightarrow$ "lion roar").
Modify event time	Adjust event timestamps or duration while preserving ordering constraints.
Modify event properties	Edit timbre or perceptual attributes (volume, pitch, intensity, spatial width).

Table 5. Action pool used by the EchoVidia agent.

Agent Control Logic. The agent executes actions under a deliberative control loop:

1. **Perception:** invoke video reasoning actions using the slow–fast thinking strategy to identify candidate events.
2. **Planning:** assemble and refine the symbolic event plan through multiple design actions, conditioned on both the video and user instruction.
3. **Verification:** re-query timestamps or re-evaluate event descriptions if inconsistencies or contradictions arise.
4. **Synthesis:** call audio generation actions to produce event-level waveforms and mix them into the final audio track.

This explicit planning–editing–synthesis loop allows EchoVidia to handle complex hierarchical instructions such as temporal reordering, multi-event transformations, and fine-grained attribute manipulation.

B.3. Implementation Details

B.3.1. Prompt Design

We design a structured prompt template that guides the VideoLLM through the reasoning and refinement steps. The full template is included in the supplementary source code, but the key components are:

- **Video Context Prompt:** summarizing global scene information and requesting enumeration of potential sounding events.
- **Slow–Fast Fusion Prompt:** asking the model to reconcile global and fine-grained predictions.
- **Event Plan Prompt:** instructing the model to output symbolic events in the (t, d, p) structure.
- **Editing Prompt:** applying user instructions for event insertion, modification, or deletion.
- **Generation Prompt:** specifying how the symbolic plan should be converted to audio-generation commands.

B.3.2. Prompt Template

Video Context Prompt.

```
SYSTEM ROLE:
You are a Video Understanding Specialist. You analyze visual content
and extract only information relevant to sound-producing events.

INPUT:
<VIDEO_FRAMES_1FPS>

TASK:
1. Describe the global scene (1{2 sentences).
2. Enumerate all visually identifiable actions that can produce sound.
3. For each action, provide:
   - a short semantic label (e.g., "cat meow", "object impact")
   - the rough order index (1, 2, 3, ...) WITHOUT timestamps
   - a short justification (why it may produce sound)

OUTPUT FORMAT (STRICT):
EVENT_LIST = [
  {index: i, label: "...", justification: "..."},
  ...
]
Do NOT include timestamps or any speculation unrelated to visible motion.
```

Slow–Fast Fusion Prompt.

```
SYSTEM ROLE:
You are a Temporal Fusion Expert. Your job is to merge two event
streams into one consistent timeline.

INPUT:
FAST_VIEW_EVENTS:
<LIST_FROM_1FPS>

SLOW_VIEW_EVENTS:
<LIST_FROM_SLOW_MOTION>

TASK:
1. Merge events with similar semantics from both lists.
2. Refine event timing using SLOW_VIEW when available.
3. Assign timestamps (t_start, t_end) in seconds.
4. Remove duplicates and ensure chronological ordering.

OUTPUT FORMAT (STRICT):
MERGED_EVENTS = [
  {label: "...", t_start: x.xx, t_end: y.yy},
  ...
]
No explanations. Only the list above.
```

Event Plan Prompt.

```
SYSTEM ROLE:
You are the Sounding Event Structuring Agent. You convert events
into symbolic representations for controllable audio generation.

INPUT:
MERGED_EVENTS:
<list from fusion step>

TASK:
For each event, construct e = (t, d, p):
 - t = (t_start, t_end)
 - d = {subject: ?, action: ?, object: optional}
 - p = {pitch: ?, volume: ?, intensity: ?, spatial: ?}

Rules:
- Infer d from visual cues only.
- Use DEFAULT for p attributes if uncertain.
- All fields must exist.

OUTPUT FORMAT (STRICT):
EVENT_PLAN = [
  {
    t: (x.xx, y.yy),
    d: {subject: "...", action: "...", object: "..."},
    p: {pitch: "...", volume: "...", intensity: "...", spatial: "..."}
  },
  ...
]
No free-form text. Only structured results.
```

Editing Prompt.

```
SYSTEM ROLE:
You are the Sound Design Controller. You update an existing symbolic
event plan based on user instructions.

INPUT:
USER_INSTRUCTION:
"<instruction text>"

CURRENT_EVENT_PLAN:
<symbolic plan>

TASK:
1. Identify all referenced events.
2. Apply the required edits using ONLY:
   - ADD_EVENT
   - DELETE_EVENT
   - MODIFY_DESCRIPTION
   - MODIFY_TIME
   - MODIFY_PROPERTIES
3. Validate chronological ordering and value ranges.

OUTPUT FORMAT (STRICT):
UPDATED_EVENT_PLAN = [
  {t: (...), d: {...}, p: {...}},
  ...
]
No reasoning statements. Only the updated plan.
```

Generation Prompt.

```
SYSTEM ROLE:
You are the Audio Generation Planner. You convert symbolic events
into commands for the audio synthesis backend.

INPUT:
FINAL_EVENT_PLAN:
<symbolic plan>

TASK:
For each event, produce a generator command block with:
- event_id
- synthesis_prompt (derived from d and p)
- t_start / t_end
- acoustic properties

Then produce mixing instructions specifying:
- layering
- crossfades
- loudness normalization
- global effects

OUTPUT FORMAT (STRICT):
GENERATION_COMMANDS = [
  {
    event_id: i,
    synthesis_prompt: "<text prompt>",
    t_start: x.xx,
    t_end: y.yy,
    properties: {volume: ..., pitch: ..., intensity: ..., spatial: ...}
  },
  ...
]

MIXING_INSTRUCTIONS = {
  layering: "...",
  crossfade: "...",
  loudness_normalization: "...",
  global_effects: "..."
}
Only the structures above. No additional commentary.
```

B.3.3. Base Models

VideoLLMs. We use **Gemini 2.5 Pro** as the primary VideoLLM for event reasoning due to its strong temporal understanding and multimodal grounding. No additional fine-tuning is applied.

Audio Diffusion Model. We use **Stable Audio** (v2) as the base diffusion model for sound synthesis, leveraging its prompt-conditioning and high dynamic range. Event-level segments are generated individually and mixed with crossfading to maintain continuity.

B.3.4. Computation Resources Requirement

EchoVidia is entirely training-free. All evaluations were conducted using:

- **GPU:** A single NVIDIA A100 (80GB).
- **CPU:** 32-core Xeon server for lightweight preprocessing.
- **Latency:** Per-sample inference takes 120–270 seconds for VideoLLM reasoning and 6–12 seconds for audio generation, depending on instruction complexity and number of events.

B.4. Limitations

While EchoVidia improves controllability and semantic alignment, several limitations remain:

- **Latency.** The multi-stage reasoning and synthesis pipeline increases inference time, particularly when many events are present.

- **Dependency on VideoLLM Awareness.** Event accuracy is bounded by the VideoLLM's ability to perceive subtle motions or occluded interactions.
- **Limited Event Interaction Modeling.** The symbolic structure handles events independently and does not fully capture physical interactions or complex auditory mixtures.
- **Diffusion Model Biases.** Stable Audio may generate artifacts or stylistic biases, especially for rare or highly specific sound textures.

Future work may integrate EchoVidia into end-to-end trainable architectures and extend the symbolic representation to capture richer auditory phenomena such as reverberation, multi-source interference, and dynamic spatialization.

C. Motivation Illustration and Qualitative Examples

Please see the attached video. In the video, we present Figure 1 with video illustration, and compare the performance of *EchoVidia* with MMAudio-L-44.1kHz, AudioGenie, ThinkSound, and HuanyuanVideo-Foley-XXL.