LocalSoundBench: Do Video-to-Audio Models Respond Locally to Visual Edits?

Anonymous authors
Paper under double-blind review

Abstract

Videos with several visible events need each sound to occur with its event. A local edit gives a controlled test: the changed event's audio should update while sounds elsewhere stay as they were. Global fidelity and synchronization metrics can miss this error because the expected sounds still occur somewhere in the clip. We introduce LocalSoundBench to test whether current video-to-audio (V2A) models can do both. Across 21 released settings, when models produce the requested sound, they often also alter the audio of footage that was never edited. This is not because they cannot produce the affected sounds: when we test these events in isolation and probe the models' internal features, the sound identities are still present. The models instead do not reliably keep each sound at the right time. We therefore propose *Regional Flow*, which, given the event regions, lets the model see the whole scene but uses information from each event only when generating audio at that event's time. With Regional Flow, interval accuracy rises by 5.9 to 39.1 points on three continuous-latent generators. Regional Flow also improves both edit accuracy and joint success on paired edits for all three models. For MMAudio, training with sounds aligned to their event times and supplying annotated sound and time pairs on continuous video both give gains in the same direction. These results suggest that when a model generates one soundtrack from the full video, scene context alone is not enough: the model needs explicit control over when each sound is produced. Our code is available in the Supplementary Material.

1 Introduction

In films, television programs, games, and online videos, Foley tracks provide footsteps, impacts, and object interactions aligned with visible events. Video-to-audio (V2A) generators now produce audio synchronized with these events (Luo et al., 2023; Cheng et al., 2025; Shan et al., 2025; Dai et al., 2026; Viertola et al., 2025), and recent methods introduce semantic, timing, or reference controls (Zhang et al., 2026a; Lee et al., 2025; Mo & Song, 2025; Li et al., 2026c;a). A video with several events can contain all expected sounds and still be wrong if those sounds occur in the wrong intervals. For example, a bark in the train interval and train noise in the dog interval are both errors even if the soundtrack is plausible overall. Generating audio for each source clip separately and splicing the outputs is a practical workaround that we call Window Stitch. It works when the source clips can be generated independently. It does not test whether the model itself can keep each sound with its event while retaining the full video as input. We test the value of the surrounding video on continuous EPIC-SOUNDS clips by comparing each full clip with a crop of four seconds centered on the same event. Cropping does not consistently improve interval accuracy and lowers MMAudio accuracy under SSAST, CLAP, and PANNs, showing that surrounding frames can provide useful evidence about the object, action, or scene (Appendix G).

A local visual replacement makes this question easy to test. If one visible event is replaced, only its audio should change. We call this behavior *local response*. When a train shot is replaced with a ringing bell, the edited interval should ring while an unchanged dog in a neighboring shot still barks (Figure 1). Keeping the other footage fixed lets us measure the requested update and any change elsewhere separately. Existing V2A benchmarks (Dixit et al., 2026; Zhang et al., 2026b; Yang et al., 2026b; Wei et al., 2026; Zhou et al., 2026; Li et al., 2026b) measure overall fidelity, coherence across events, or isolated physical variations. Global scores can remain high when the expected sounds occur somewhere in the clip. A synchronization score can follow the energy envelope without identifying

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

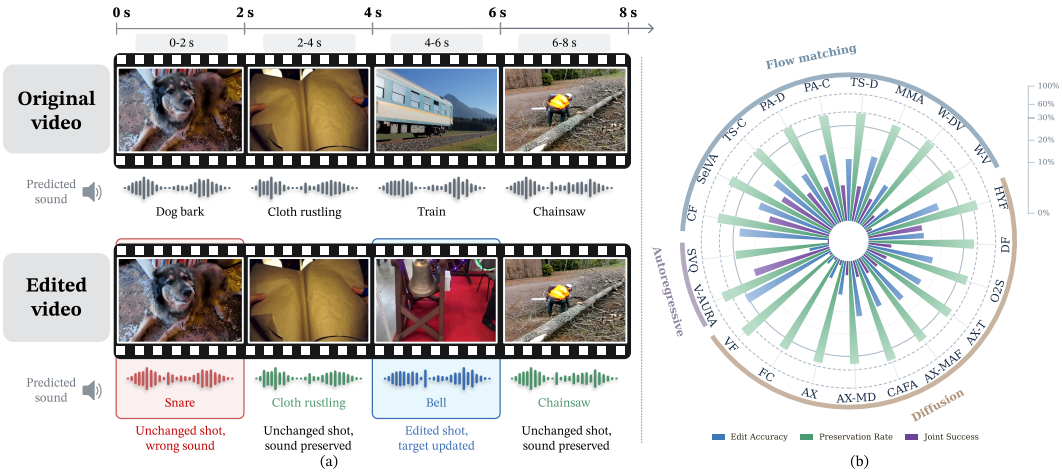

Figure 1: A local visual edit also changes sounds in unedited intervals. (a) Here, replacing the train shot with a bell updates the edited interval, but the unedited first shot changes from an initially correct dog bark to a snare. (b) Across 21 released settings under matched conditions, models preserve a median of 54.5% of initially correct unchanged sounds (**Preservation Rate**) but update only 14.0% of edited intervals (**Edit Accuracy**). **Joint Success**, which requires both in the same pair, has a median of 4.2%. Updating the edited shot and leaving the rest of the soundtrack intact therefore rarely happen together.

the sound at each event. We therefore introduce LocalSoundBench, a benchmark of paired videos that differ in exactly one visible event. **Edit Accuracy** checks whether the edited interval contains the requested sound, **Preservation Rate** checks whether initially correct sounds in unchanged intervals remain correct, and **Joint Success** requires both in the same pair. Across 21 released model settings, models rarely update the edited event and keep the sounds that were already correct at unchanged events: median joint success is only 4.2%.

The low joint success has several possible causes. A model may be unable to produce the affected sound, sampling may be unstable, or the replacement may disrupt the sequence. Many affected sounds are recognizable when generated alone, exact reruns remain stable, and a replacement from the same class still changes unedited intervals. When the same shots are reordered while the shot durations and sound set stay fixed, a shot often receives the sound of another shot in the same video. A linear probe also recovers the intended sound class from intermediate features. Reversing only the semantic or timing features moves the generated order toward the reversed order. Window Stitch then recovers many of these sounds by generating each source clip alone and making its audio play when the source event occurs. Its success supports the diagnosis. The models can produce the sounds and still respond to timing, but do not reliably keep each sound with its event when the full video contains several events.

Regional Flow tests whether this error can be reduced during generation from the full video. Given the event times, it lets each event affect the audio only at its own time. On unedited sequences, it raises interval accuracy by 5.9 to 39.1 points across MMAudio, HunyuanVideo-Foley XL, and Omni2Sound. On LocalSoundBench, it also raises edit accuracy and joint success for every model. For MMAudio, training with sounds aligned to their event times and supplying annotated sound and time pairs on continuous video both help keep each sound with its event. These results show that full video context is not enough. The model also needs control over when each sound is generated. Window Stitch recovers sounds through separate generations. Regional Flow instead keeps the full video during generation and uses each event at its own time. This retains surrounding frames that can help identify the object, action, or scene. Appendix C.4 explains when each method applies.

Our main contributions are:

- **A V2A Benchmark for Local Response:** We introduce LocalSoundBench to test whether a generator that receives the full video updates the edited event while keeping initially correct sounds with their unchanged events. Evaluating 21 released model settings shows that models rarely do both, exposing a widespread failure that standard metrics miss.

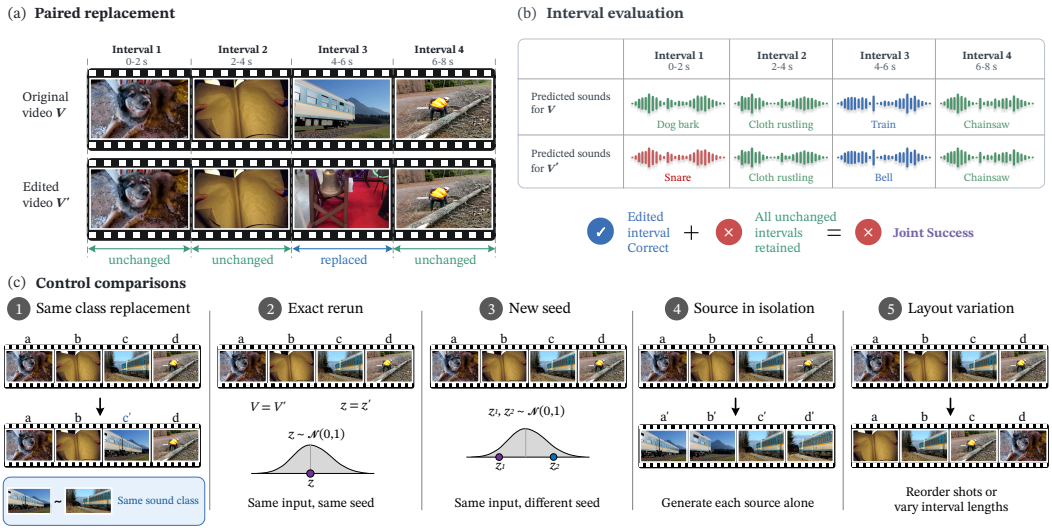

Figure 2: Benchmark overview. (a) Each pair differs in one shot while the remaining footage and generation settings are fixed. (b) **Edit Accuracy** checks the replaced interval, **Preservation Rate** checks initially correct sounds in unchanged intervals, and **Joint Success** requires both in the same pair. (c) Replacements within the same class, exact reruns, new seeds, isolated source clips, and layout controls vary one factor at a time. This design separates the intended update from collateral changes and ordinary sampling variation.

- **Controlled Diagnosis:** Controlled tests show that the failure is not an inability to produce the sounds or ordinary sampling variation. Intermediate features retain sound identity and respond to timing, but the models do not reliably keep each sound with its event.
- **Targeted Interventions at Inference and Training:** We propose *Regional Flow*, which keeps the full video and uses each event at its own time. It improves edit accuracy and joint success for three models. Aligned targets improve all three measures for MMAudio.

2 Benchmark Design

LocalSoundBench tests local Foley revision with matched video pairs. Each pair changes one visible event while keeping the other events and their timing fixed (Figure 2). This lets us score the requested update and any change elsewhere in the soundtrack separately.

2.1 Benchmark Construction

We begin with the VGGSound test pool (Chen et al., 2020). We remove invalid media, nearly silent clips, speech or music dominated audio, label conflicts, and events with unclear timing. We keep clips in which the action that produces the sound is visible, then manually review the clips and classes. The final benchmark contains 1,024 clips from 64 classes and 2,048 edit pairs. Each pair changes one event in a four-shot sequence. Appendix A describes the source selection and rendering. Appendix C.1 shows that the main result remains stable as more source groups are added.

The main test keeps event count and duration fixed. Later tests change the donor class, the number of edited events, shot order, or interval duration. Insertion and deletion use reviewed impact events from Greatest Hits (Owens et al., 2016). Class labels and event times define the scoring targets but are never given to the generator. Appendix A.2 describes the additional tests.

2.2 Paired Tests and Metrics

Tests. For each pair, we replace shot j and regenerate the complete soundtrack. The two runs share the other shots, event times, prompt, sampler, and initial noise. Only the replacement changes. A sequence error may also come from a sound that the model cannot produce. We test this by placing the same source clip in two matched contexts. **Alone** in Table 1 is the interval accuracy when the source clip appears by itself in a video lasting eight seconds. **In sequence** scores the same clip when three other shots are present. Both videos last eight seconds. We define $\Delta =$ In sequence $-$ Alone. A negative value means that other events make the sound harder to recognize.

Table 1: Main controlled results. CLAP results (%) for paired edits and for the same source clip generated alone or with three other shots. Alone scores the clip by itself. In sequence scores the same clip beside the other shots. Δ = In sequence − Alone, with bars ( ) proportional to Δ. Darker blue indicates higher values. Bold marks column maxima. PR and JS counts appear in Appendix Table 8.

Model	EA	PR	JS	Alone	In sequence	Δ
Flow matching						
MMAudio	17.6	50.4	5.4	**70.6**	20.4	**−50.2**
ControlFoley	27.2	67.5	15.0	42.6	24.4	**−18.2**
ThinkSound†	14.9	57.8	7.0	45.1	17.0	**−28.1**
PrismAudio†	14.0	58.4	7.5	31.0	14.6	**−16.4**
ThinkSound‡	20.1	58.1	8.2	59.1	20.5	**−38.6**
PrismAudio‡	18.5	57.2	7.1	46.6	20.0	**−26.6**
SelVA	21.1	67.9	13.2	35.3	22.9	**−12.3**
Woosh-VFlow	6.0	49.0	1.2	13.6	5.8	**−7.8**
Woosh-DVFlow	9.2	57.5	3.1	26.9	9.6	**−17.3**
Diffusion						
Omni2Sound	16.8	50.9	4.2	64.6	17.3	**−47.2**
HunyuanVideo-Foley XL	21.2	69.5	11.6	47.5	22.7	**−24.8**
FoleyCrafter	4.1	52.0	0.6	5.4	5.4	0.0
Video-Foley	2.0	**81.0**	0.6	3.0	2.8	**−0.2**
CAFA	10.4	54.5	3.3	14.4	9.9	**−4.5**
Diff-Foley	12.1	52.6	4.5	31.8	13.3	**−18.5**
AudioX	6.7	51.4	1.1	37.9	9.6	**−28.3**
AudioX-Turbo	15.3	49.8	3.4	55.8	14.2	**−41.6**
AudioX-MAF	13.3	48.3	3.4	54.7	13.3	**−41.4**
AudioX-MAF-MMDiT	12.3	46.8	2.8	50.2	13.4	**−36.8**
Autoregressive						
V-AURA	**33.6**	78.2	**22.6**	57.1	**34.2**	**−22.9**
SpecVQGAN	3.9	29.2	0.0	18.2	6.6	**−11.5**

†Direct prompt. ‡Visual reasoning prompt.

Additional controls replace several shots, insert or delete an event, and vary interval duration. Appendix A.2 gives the matched construction.

Metrics. CLAP (Elizalde et al., 2023) and BEATs (Chen et al., 2023) classify the sound in each interval. PANNs (Kong et al., 2020) and a blinded Qwen2.5-Omni judge that receives only audio provide separate checks. **Edit Accuracy** (**EA**) is the share of pairs whose edited interval contains the new sound. **Preservation Rate** (**PR**) is the share of unchanged intervals that were correct before the edit and remain correct after it. **Joint Success** (**JS**) requires both in the same pair. JS is the strictest of the three measures: one failed unchanged interval makes the pair fail. The scores credit the sound class rather than an exact waveform. For unedited sequences, *interval accuracy* is the share of intervals classified as their intended class, and *sequence accuracy* is the share of sequences whose intervals are all correct. Confidence intervals account for clips drawn from the same source. Appendix B.3 gives the classification procedure, and Appendix B.4 gives the formal definitions and effective counts. We also report standard V2A metrics: FD-PaSST (Kilgour et al., 2019; Koutini et al., 2022) for the distribution of generated audio, ImageBind (Girdhar et al., 2023), LAION-CLAP (Wu et al., 2023), and MS-CLAP (Elizalde et al., 2023) for agreement over complete clips, and DeSync from Synchformer (Iashin et al., 2024) for temporal offset.

2.3 Models and Inputs

The primary comparison covers 21 released model and input settings from flow matching, diffusion, and autoregressive systems (Cheng et al., 2025; Dai et al., 2026; Shan et al., 2025; Yang et al., 2026a; Zhang et al., 2026a; Lee et al., 2025; Benita et al., 2025; Luo et al., 2023; Viertola et al., 2025; Liu et al., 2025; 2026; Lee et al., 2026; Hadjeres et al., 2026; Tian et al., 2026a;b). Each system keeps its released visual, motion, synchronization, and event inputs. The standard test uses the benchmark video and the direct prompt, without a class label or interval boundary. ThinkSound and PrismAudio are also tested with their visual reasoning prompt. Missing, silent, and undecodable outputs count as errors. Appendix B.2 lists the exact settings.

Together, this setup compares each model under its released inputs while asking one local question: whether a visual change alters only the sound tied to that change. EA records the requested update, PR records whether the unchanged sounds remain correct, and JS requires both to succeed. These

measures distinguish a model that ignores the edit from one that makes the update but also changes sounds that should remain intact.

3 Diagnosing Local Revision Failures

A paired edit measures two effects at once: the requested change at the edited shot and any collateral change in the soundtrack produced by regenerating the full video. A low score alone cannot tell us why the collateral change occurred. The model may never have learned the missing sound, stochastic sampling may change it on every rerun, the new event may disrupt the sequence, or several visible events may compete for the same audio intervals. We test these explanations in that order.

3.1 A Local Edit Changes Unedited Sounds

Consider four shots showing a dog, rustling cloth, a train, and a chainsaw. Replacing the train with a bell should change only the third interval. Table 1 reports EA, PR, and JS for this test. Its last three columns ask whether a low sequence score simply means that the model cannot generate the sound even when it appears alone.

The main failure is that the two requirements rarely hold together. Across the 21 settings, median EA is 14.0%, median PR is 54.5%, and median JS is only 4.2%. PR alone can be misleading because a model that rarely responds to an edit also has fewer chances to disturb the old soundtrack. JS instead asks for the behavior an editor needs: change the requested event and leave the rest intact.

The collateral changes do not simply erase sounds. Many initially correct sounds switch to another recognized class, while others become too weak to recognize (Appendix Figure 7). A class switch suggests that the sound may still be present but attached to a different event. This raises a narrower question: can the model produce the affected sound, and if so, where does it place it?

3.2 The Sounds Remain, but Appear at the Wrong Events

The last three columns show that MMAudio can still produce many of these sounds. Its accuracy rises from 20.4% in the sequence to 70.6% when the clip is generated alone, and the same difference remains as more source groups are added (Table 1 and Appendix Figure 6). Generating the clip alone changes the full visual context, so this test does not explain why the sounds are lost in a sequence.

We next return to paired sequences and remove simpler explanations. Exact reruns remain stable, and changing the seed gives nearly the same interval accuracy, so ordinary sampling variation does not explain the collateral changes. Replacing a shot with another example from the same class still changes untouched intervals, so the edit does not need to introduce a new sound label (Appendix Table 10). The same problem also appears under insertion, deletion, unequal intervals, and larger edits (Appendix Table 11). These tests leave two plausible causes: the hard cuts or competition among several visible events.

The repeated source control separates these two causes. It fixes the duration and cut positions, then changes only whether the shots repeat one source clip or show different source clips. Repetition is easier at every sequence length (Figure 3(a)). Since the cuts are identical, the added difficulty comes from keeping several sounds apart. This test still changes how many different sounds appear, so we next hold the complete sound set fixed.

The shot order test changes only position while keeping the clips, classes, duration, sound set, and generation settings fixed. Across 24 orders, the gap between the best and worst order ranges from 25.9 to 61.0 CLAP points (Figure 3(b)). Predictions often switch to another sound from the same video. Once the content and sampling are fixed, the remaining change is which sound appears during each event.

3.3 The Blind Spot of Standard Metrics

The order test also explains why standard metrics can miss the error. A global score can still find a sound after it moves to the wrong event, and a synchronization score can still match an unchanged energy envelope. We test this directly by moving each of four sounds to a wrong interval. Interval accuracy falls from 99.6% to 0.4%, while three global similarities rise slightly (Figure 3(c)). DeSync

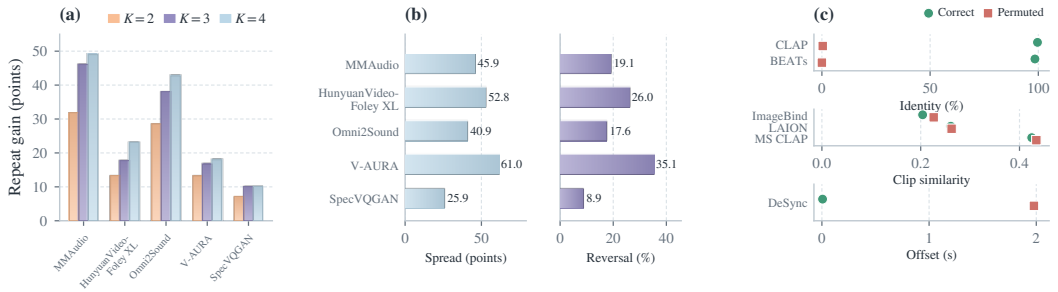

Figure 3: Sounds appear at the wrong event when several events share a video, and clip-level metrics miss the error. (a) Cuts alone do not explain the errors. With the same duration and cuts, repeating one source clip gives higher CLAP interval accuracy than showing K different clips (repeat gain). (b) Position alone changes the result. With the four clips fixed, spread is the best minus worst interval accuracy over 24 orders, and reversal is the share of interval predictions whose correctness flips when the order is reversed. (c) Clip-level similarity misses the error. Moving every reference sound to a wrong interval drops interval accuracy (Identity) to near zero but leaves ImageBind, LAION-CLAP, and MS-CLAP nearly unchanged. DeSync responds because the energy envelope moves with each sound. Panels (a) and (b) use five models and 256 source groups.

Table 2: Listeners preserve unchanged sounds more often than they hear the requested update. EA, PR, and JS for paired edits use decisive judgments. Appendix D.3 gives the protocol and counts.

Model	EA (%)	PR (%)	JS (%)
MMAudio	28.6	74.4	19.0
HunyuanVideo-Foley XL	75.0	96.4	67.9
Omni2Sound	24.0	100.0	24.0
V-AURA	52.2	88.1	52.2
All models	**46.4**	**90.3**	**42.3**

also reports no offset for some incorrect soundtracks when the envelope stays fixed. These metrics measure whether the right sounds and acoustic timing remain in the clip, but not whether each sound occurs with the right event. Interval accuracy exposes the error because it checks the sound during each event separately.

BEATs, PANNs, and a Qwen2.5-Omni judge that receives only audio give the same model ordering (Appendix Figure 8 and Table 14). We next compare these automatic decisions with a blinded listening audit.

Listeners accept some weak sounds or sounds with altered timbre that an audio classifier rejects, so their absolute scores are higher. Yet they still preserve unchanged sounds more often than they hear the requested update (Table 2). Automatic and human judgments therefore show the same gap between updating and preserving sounds.

Recovering the sounds. The diagnosis makes a testable prediction. If several events interfere during joint generation, Window Stitch should recover the missing sounds. It raises MMAudio interval accuracy from 20.4% to 67.7% (Appendix Table 13). This confirms that the model can produce the sounds when the shots are generated separately. It does not explain how to keep each sound with its event during joint generation. We next ask whether the model loses the sound, loses its timing, or retains both but still generates the sound at the wrong time.

4 Controlling When Each Sound Is Generated

Section 3 shows that sounds are often produced at the wrong events, but it does not show what fails inside the generator. The model may lose the sound identity, lose its timing, or retain both and place the sound incorrectly. We test these possibilities in order. First, we look for sound and timing information inside the model. We then ask whether limiting the visible context keeps them together. Finally, we test whether using each event only at its own time improves the generated audio.

4.1 Sound and Timing Remain, but Sounds Still Appear at the Wrong Time

We first ask whether the sound identity has been lost. A linear probe recovers the intended class from intermediate and predicted states in MMAudio, HunyuanVideo-Foley XL, and Omni2Sound, even

when the final interval prediction is wrong. At the model output, probe accuracy ranges from 36.8% to 65.5%, well above the shuffled controls (Figure 5(a) and Appendix Figure 14). The intended sound therefore remains present inside the model.

A readable feature does not show that the generator uses it. We test this by reversing the semantic or timing inputs while keeping the other settings fixed. The generated order moves toward the reversed input in all three tested models (Appendix Table 18). V-AURA provides a separate check because it generates audio from earlier to later times. A later edit changes only 5.2% of earlier predictions but 45.1% of later predictions (Appendix Figure 12). Sound identity is therefore still present, and timing still guides generation. The remaining difficulty is producing each sound when its event occurs.

Attention offers one possible explanation. An audio position in MMAudio reads nearby video tokens, but it also reads earlier and later shots (Figure 4). Several events can therefore affect the same output position. If broad visual access causes the error, limiting that access should keep each sound at the right time.

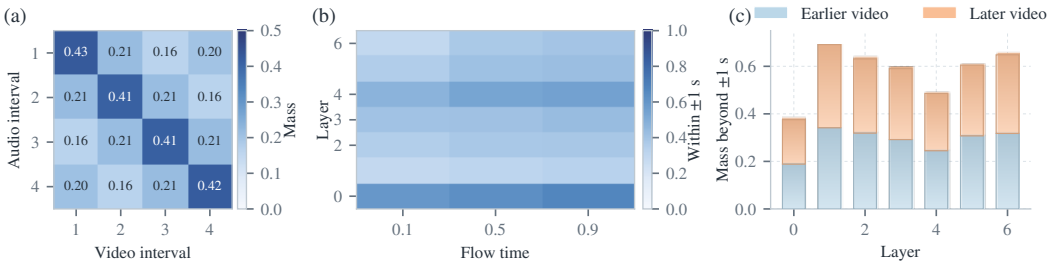

Figure 4: MMAudio reads video from outside the current event. (a) Attention averaged within each audio and video interval. (b) Attention mass within one second of the same time. (c) Remaining mass on earlier and later video. The maps show which video frames each audio position reads. Reading a frame does not mean that its sound will be generated at that position.

The matched comparison tests whether this prediction holds. The window helps only in one part of the comparison (Table 3). MMAudio performs best when it is trained with a local window but can see the full video again at inference. Keeping the window at inference removes the gain, and results also vary across the other architectures (Appendix Table 17). Nearby frames provide timing, while the wider scene helps identify the object and action. The model needs both, so hiding distant frames is not a reliable fix. A separate step must make each event change only the matching part of the audio. The complete attention, probe, input reversal, and training results appear in Appendices E.1 and F.1.

Table 3: Temporal attention window. EA, PR, and JS under CLAP for MMAudio in a matched 2×2 comparison of full and local visual access. Local training works best when full access is restored at inference.

Training access	Inference access	EA	PR	JS
All video times	All video times	18.2	58.5	8.8
All video times	±1 s window	17.4	56.3	5.8
±1 s window	All video times	**20.7**	**65.0**	**10.3**
±1 s window	±1 s window	18.2	58.7	6.6

4.2 Regional Flow Uses Each Event at Its Own Time

Local windows change which frames an audio position can read, but they do not decide when the information from each event changes the audio. Regional Flow first predicts from the full video and then from one temporal cell at a time. A mask allows the difference between these predictions to affect only the audio covered by that cell. We use this intervention during inference for continuous latent generators (Lipman et al., 2023).

Let $v_g = v_\theta(z_t, t, h_g)$ be the vector field predicted from the full video. We divide the timeline into K cells and let $v_k = v_\theta(z_t, t, h_k)$ be the field predicted from cell k at the same latent state. The difference $v_k - v_g$ measures how that cell changes the prediction. A map ρ pairs each cell with an audio mask $m_{\rho(k)}$:

$$\widetilde{v}_t^{(\rho)} = v_g + \alpha_t \sum_{k=1}^{K} m_{\rho(k)} \odot \big(v_k - v_g\big). \tag{1}$$

Regional Flow uses $\rho(k) = k$, so each difference affects the audio at the same time as its cell. Appendix E.2 gives the masks, sampling weight, and the control that uses each cell during another

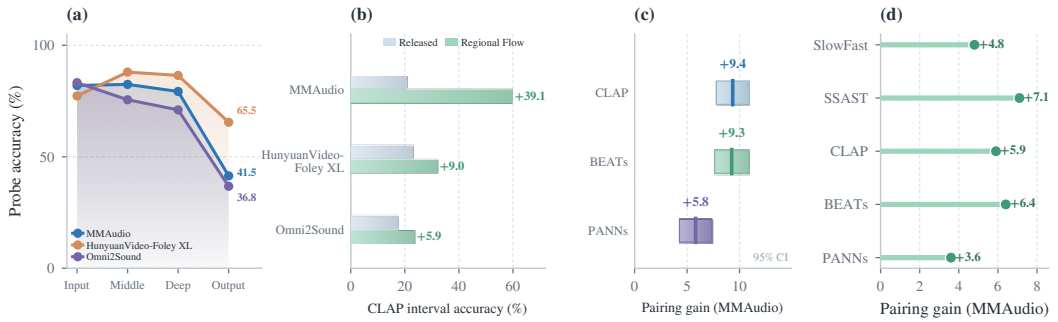

Figure 5: Sound identity remains readable, and using it at the right time improves the output. (a) Linear probes recover the intended sound from the input, intermediate features, and model output in three generators. (b) Regional Flow improves interval accuracy for the same three generators. (c) Aligned MMAudio training improves interval accuracy over shuffled training under all three evaluators. (d) On continuous video, annotated sound and time pairs improve MMAudio interval accuracy over video alone.

event. Unlike Window Stitch, Regional Flow does not sample each cell as a separate clip. It applies the cell differences during one full video sampling path.

We test whether the event time is what matters. The control uses the same predictions and sampling procedure, but lets the visual information from one event change the audio during another event. Regional Flow performs better across all three models, showing that the visual information must be used when its event occurs. On unedited sequences, Regional Flow raises CLAP interval accuracy by 5.9 to 39.1 points across the three generators (Figure 5(b)).

Table 4: Regional Flow improves paired edits. Δ = Regional Flow − released sampling under CLAP (points), with bars (,) proportional to Δ. Full scores and controls are in Appendix Table 19.

Model	ΔEA	ΔPR	ΔJS
MMAudio	+45.4	+32.3	+39.1
HunyuanVideo-Foley XL	+10.1	+7.3	+8.6
Omni2Sound	+9.5	−3.7	+8.9

Paired edits ask whether the same method also supports local revision (Table 4). Regional Flow raises EA and JS for all three models. It also raises PR for MMAudio and HunyuanVideo-Foley XL. For Omni2Sound, EA rises by 9.5 points while PR falls by 3.7 points. Regional Flow is useful when the model keeps the full video as context rather than generating each shot independently. Window Stitch applies when hard cuts separate shots that can be generated independently. Appendix Table 19 gives the full scores and controls. Appendices E.2 and E.4 describe the method and cost. The gain across three generators shows that using each event at its own time is not tied to one architecture. We next ask whether training can teach the same behavior.

4.3 Learning When Each Sound Should Occur

Regional Flow uses each event at its own time during inference. We next ask whether training can learn the same behavior. We train MMAudio with each sound at its event time and compare it with the released checkpoint. Figure 5(c) and Appendix F.2 report a matched run with shuffled times.

Table 5: Aligned training improves MMAudio. EA, PR, and JS for the released checkpoint and aligned targets (%).

	Released MMAudio			Aligned targets		
Evaluator	EA	PR	JS	EA	PR	JS
CLAP	17.6	50.4	5.4	25.6	61.5	12.6
BEATs	17.8	52.7	5.1	26.5	58.5	10.9
PANNs	13.4	57.9	4.8	19.4	67.1	11.4

For MMAudio, aligned targets improve EA, PR, and JS under all three evaluators (Table 5). Under CLAP, JS rises from 5.4% to 12.6%. Whole clip quality scores move in both directions. Omni2Sound and V-AURA show no clear gain, so this training result is specific to MMAudio. Appendix F gives the training setup and full measurements.

The constructed sequences contain hard cuts. We therefore repeat the test on continuous EPIC-SOUNDS clips. The paired input gives MMAudio a sound label and the interval where that sound should occur. This tells the model what sound to generate and when to generate it. Pairing

each annotated sound with its interval improves all five evaluators (Figure 5(d)). Boundaries alone change interval accuracy by about two points. Cycling the sound labels or reversing the interval order removes the gain (Appendix Table 25). The model needs both the sound identity and its time.

We next predict both pieces from silent video. On the same outputs, MMAudio improves all nine interval and whole clip scores, HunyuanVideo-Foley XL improves eight, and Omni2Sound improves seven. V-AURA varies. Interval scores check each event, while whole clip scores judge the output as a whole. Appendix Tables 28 and 29 give the complete results. The continuous test supports the controlled finding. The full video helps identify the event, while its time tells the model when to generate the sound. Annotated pairs test whether a generator can use a known sound and time. Predicted pairs also test whether both can be found from silent video. Regional Flow uses event time during inference across three generators, while aligned targets teach MMAudio to use it during training.

5 Related Work

V2A generation now includes visually guided token prediction, diffusion, flow matching, and multimodal autoregression (Iashin & Rahtu, 2021; Luo et al., 2023; Cheng et al., 2025; Shan et al., 2025; Dai et al., 2026; Viertola et al., 2025; Hadjeres et al., 2026; Tian et al., 2026a;b). Recent systems add text, reference audio, motion, or event controls. These systems use explicit timing, masked audiovisual alignment, multimodal references, or controls over the source and its time (Zhang et al., 2026a; Lee et al., 2025; Mo & Song, 2025; Chen et al., 2025; Hayakawa et al., 2026; Simon et al., 2026). Other systems target scenes with several events (Benita et al., 2025; Yang et al., 2026a; Li et al., 2026c;a; Lee et al., 2026; Liu et al., 2025; 2026; Yang et al., 2026b; Oh et al., 2026). They improve a soundtrack or give the user more control. Our question is different: when a local visual edit changes one or more events, can the model update the requested sounds while leaving the other sounds unchanged? Existing benchmarks cover visible sound sources, broad audio categories, stories with several shots, synchronization, and controlled physical factors (Dixit et al., 2026; Zhang et al., 2026b; Wei et al., 2026; Zhou et al., 2026; Li et al., 2026b). FlatSounds also uses counterfactual videos and asks whether the changed sound follows one physical factor. Our paired test asks whether the requested sound changes and whether every sound that was correct before the edit remains correct afterward. Audio editors such as AUDIT modify a supplied waveform and learn to preserve regions outside an instruction (Wang et al., 2023). Audiovisual editors transform both modalities, edit a sounding object, or regenerate audio from an edited video and its original soundtrack (Lin et al., 2026; Fu et al., 2025; Ishii et al., 2025). AVED-Bench, AVE-Compass, and AVENUE then measure instruction following and preservation in joint editing (Wen et al., 2026; Kim et al., 2026). In LocalSoundBench, both inputs are silent videos. There is no source soundtrack to copy. The original generation identifies which sounds were correct before the edit. EA, PR, and JS then measure whether the V2A mapping responds locally.

6 Conclusion

LocalSoundBench shows that a visual edit can update the requested sound while disturbing sounds at unchanged events. Controlled tests show that many affected sounds remain recognizable in isolation and in the model's intermediate features. Sampling variation, a new sound class, and hard cuts alone do not explain the failure. The model still responds to timing, but it does not reliably generate each sound when its event occurs. This explains why whole clip scores can remain high even when the sounds are in the wrong intervals. Evaluation should also report overall audio quality, which does not always improve with event accuracy. Regional Flow keeps the full video available and uses each event at its own time, improving all three tested continuous generators. Aligned training improves MMAudio in the same direction, although this training benefit does not extend to all tested models. Window Stitch is useful when shots can be generated independently. Regional Flow retains scene context during one generation. The continuous video results also show that boundaries alone are insufficient: the model needs both sound identity and event time. Learning these together from video is therefore the next step. A useful system must infer what sound to produce and when, while keeping the other sounds correct. Overlapping sounds remain an important test: changing one event may require separating its sound from another that must stay unchanged. Appendix I discusses the scope and settings that remain open.

AI USE STATEMENT

In this work, we used generative AI tools to polish author-written text, translate short passages, suggest clearer wording and organization, assist with literature search, and make minor stylistic edits to author-written code. The tools did not draft sections of the paper, create scientific figures, propose or refine research questions or hypotheses, design or run experiments, implement methods, generate or clean data, formulate or prove mathematical claims, analyze qualitative data, or interpret results. The plots were generated directly from experimental data with Python. The authors selected the cited work, checked each reference against the original publication, reviewed every accepted AI-assisted edit, and tested the affected code. We take responsibility for the final content of this work, including its text, references, claims, code, and artifacts.

REPRODUCIBILITY STATEMENT

Appendices A through H provide the data, inference, evaluation, uncertainty, and reproducibility details. The supplementary material also includes an anonymous code archive with the benchmark evaluation, Regional Flow, aligned MMAudio training, a runnable audio example, and tests. We will release the full project after review.

REFERENCES

Roi Benita, Michael Finkelson, Tavi Halperin, Gleb Sterkin, and Yossi Adi. CAFA: A controllable automatic Foley artist. In *IEEE/CVF International Conference on Computer Vision*, pp. 15917–15926, 2025.

Sanyuan Chen, Yu Wu, Chengyi Wang, Shujie Liu, Daniel Tompkins, Zhuo Chen, Wanxiang Che, Xiangzhan Yu, and Furu Wei. BEATs: Audio pre-training with acoustic tokenizers. In *International Conference on Machine Learning*, pp. 5178–5193, 2023.

Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. VGGSound: A large-scale audio-visual dataset. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 721–725, 2020.

Ziyang Chen, Prem Seetharaman, Bryan Russell, Oriol Nieto, David Bourgin, Andrew Owens, and Justin Salamon. Video-guided Foley sound generation with multimodal controls. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18770–18781, 2025.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28901–28911, 2025.

Yusheng Dai, Zehua Chen, Yuxuan Jiang, Baolong Gao, Qiuhong Ke, Jianfei Cai, and Jun Zhu. Omni2Sound: Towards unified video-text-to-audio generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1661–1671, 2026.

Satvik Dixit, Koichi Saito, Zhi Zhong, Yuki Mitsufuji, and Chris Donahue. FoleyBench: A benchmark for video-to-audio models. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 14512–14516, 2026.

Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. CLAP: Learning audio concepts from natural language supervision. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1–5, 2023.

Youquan Fu, Ruiyang Si, Hongfa Wang, Dongzhan Zhou, Jiacheng Sun, Ping Luo, Di Hu, Hongyuan Zhang, and Xuelong Li. Object-AVEdit: An object-level audio-visual editing model. *arXiv preprint arXiv:2510.00050*, 2025.

Akio Hayakawa, Masato Ishii, Takashi Shibuya, and Yuki Mitsufuji. Step-by-step video-to-audio synthesis via negative audio guidance. In *European Conference on Computer Vision*, pp. 681–698, 2026.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. ImageBind: One embedding space to bind them all. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15180–15190, 2023.

Yuan Gong, Cheng-I Jeff Lai, Yu-An Chung, and James Glass. SSAST: Self-supervised audio spectrogram transformer. In *AAAI Conference on Artificial Intelligence*, pp. 10699–10709, 2022.

Gaëtan Hadjeres, Marc Ferras, Khaled Koutini, Benno Weck, Alexandre Bittar, Thomas Hummel, Zineb Lahrichi, Hakim Missoum, Joan Serrà, and Yuki Mitsufuji. Woosh: A sound effects foundation model. *arXiv preprint arXiv:2604.01929*, 2026.

Jaesung Huh, Jacob Chalk, Evangelos Kazakos, Dima Damen, and Andrew Zisserman. EPIC-SOUNDS: A large-scale dataset of actions that sound. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1–5, 2023.

Vladimir Iashin, Weidi Xie, Esa Rahtu, and Andrew Zisserman. Synchformer: Efficient synchronization from sparse cues. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5325–5329, 2024.

Vladimir Iashin and Esa Rahtu. Taming visually guided sound generation. In *British Machine Vision Conference*, 2021.

Masato Ishii, Akio Hayakawa, Takashi Shibuya, and Yuki Mitsufuji. Coherent audio-visual editing via conditional audio generation following video edits. *arXiv preprint arXiv:2512.07209*, 2025.

Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Slow-Fast auditory streams for audio recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 855–859, 2021.

Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms. In *Interspeech*, pp. 2350–2354, 2019.

Hayeon Kim, Yoojin Jang, and Jaejun Yoo. AVENUE: Audio-video EditiNg Understanding and Evaluation. *arXiv preprint arXiv:2609.04253*, 2026.

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

Khaled Koutini, Jan Schlüter, Hamid Eghbal-zadeh, and Gerhard Widmer. Efficient training of audio transformers with Patchout. In *Interspeech*, pp. 2753–2757, 2022.

Junwon Lee, Jaekwon Im, Dabin Kim, and Juhan Nam. Video-Foley: Two-stage video-to-sound generation via temporal event condition for Foley sound. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 33:3813–3825, 2025.

Junwon Lee, Juhan Nam, and Jiyoung Lee. Hear what matters! Text-conditioned selective video-to-audio generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36680–36690, 2026.

Bingxuan Li, Yiming Cui, Yicheng He, Yiwei Wang, Shu Zhang, Longyin Wen, and Yulei Niu. EchoFoley: Event-centric hierarchical control for video grounded creative sound generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27229–27238, 2026.

Tingle Li, Siddharth Gururani, Kevin J. Shih, Gantavya Bhatt, Sang-gil Lee, Zhifeng Kong, Arushi Goel, Gopala Anumanchipalli, and Ming-Yu Liu. Benchmarking single-factor physical video-to-audio generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1939–1949, 2026.

You Li, Dewei Zhou, Fan Ma, Fu Li, Dongliang He, and Yi Yang. FoleyDirector: Fine-grained temporal steering for video-to-audio generation via structured scripts. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 29254–29264, 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Yan-Bo Lin, Kevin Lin, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Chung-Ching Lin, Xiaofei Wang, Gedas Bertasius, and Lijuan Wang. Zero-shot audio-visual editing via cross-modal delta denoising. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 7344–7354, 2026.

Huadai Liu, Kaicheng Luo, Jialei Wang, 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 *Advances in Neural Information Processing Systems*, volume 38, pp. 87244–87270, 2025.

Huadai Liu, Kaicheng Luo, Wen Wang, Qian Chen, Peiwen Sun, Rongjie Huang, Xiangang Li, Jieping Ye, and Wei Xue. PrismAudio: Decomposed chain-of-thoughts and multi-dimensional rewards for video-to-audio generation. In *International Conference on Learning Representations*, 2026.

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

Shentong Mo and Yibing Song. Foley-Flow: Coordinated video-to-audio generation with masked audio-visual alignment and dynamic conditional flows. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28912–28921, 2025.

Andrew Owens, Phillip Isola, Josh McDermott, Antonio Torralba, Edward H. Adelson, and William T. Freeman. Visually indicated sounds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2405–2413, 2016.

Hyun-Bin Oh, Yuhta Takida, Toshimitsu Uesaka, Tae-Hyun Oh, and Yuki Mitsufuji. PAVAS: Physics-aware video-to-audio synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14481–14491, 2026.

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.

Christian Simon, Masato Ishii, Wei-Yao Wang, Koichi Saito, Akio Hayakawa, Dongseok Shim, Zhi Zhong, Shuyang Cui, Takashi Shibuya, Shusuke Takahashi, and Yuki Mitsufuji. Echoes over time: Unlocking length generalization in video-to-audio generation models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15840–15849, 2026.

Zeyue Tian, Zhaoyang Liu, Yizhu Jin, Ruibin Yuan, Liumeng Xue, Xu Tan, Qifeng Chen, Wei Xue, and Yike Guo. AudioX: A unified framework for anything-to-audio generation. In *International Conference on Learning Representations*, 2026.

Zeyue Tian, Lei Ke, Zhaoyang Liu, Ruibin Yuan, Liumeng Xue, Yujiu Yang, Weijia Chen, Xu Tan, Qifeng Chen, Wei Xue, and Yike Guo. AudioX-Turbo: A unified framework for efficient anything-to-audio generation. *arXiv preprint arXiv:2606.12555*, 2026.

Ilpo Viertola, Vladimir Iashin, and Esa Rahtu. Temporally aligned audio for video with autoregression. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1–5, 2025.

Yuancheng Wang, Zeqian Ju, Xu Tan, Lei He, Zhizheng Wu, Jiang Bian, and Sheng Zhao. AUDIT: Audio editing by following instructions with latent diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 71340–71357, 2023.

Yuqing Wen, Yukai Huang, Qianqian Xie, Jiangtao Wu, Yibin Lin, Yikai Gu, Jialu Chen, Yuanxing Zhang, and Jiaheng Liu. AVE-Compass: Towards holistic evaluation for audio-video editing abilities. *arXiv preprint arXiv:2607.24821*, 2026.

Yujie Wei, Yujin Han, Zhekai Chen, Yongming Li, Kaixun Jiang, Zhihang Liu, Quanhao Li, Zhiwu Qing, Xiang Wang, Zhen Xing, Ruihang Chu, Lingyi Hong, Yefei He, Junjie Zhou, Junqiu Yu, Yang Shi, Difan Zou, Kai Zhu, Shiwei Zhang, Yingya Zhang, Yu Liu, Xihui Liu, and Hongming Shan. MSAVBench: Towards comprehensive and reliable evaluation of multi-shot audio-video generation. *arXiv preprint arXiv:2605.20183*, 2026.

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 *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1–5, 2023.

Jianxuan Yang, Xinyue Guo, Zhi Cheng, Kai Wang, Lipan Zhang, Jinjie Hu, Qiang Ji, Yihua Cao, Yihao Meng, Zhaoyue Cui, Mengmei Liu, Meng Meng, and Jian Luan. ControlFoley: Unified and controllable video-to-audio generation with cross-modal conflict handling. In *ACM International Conference on Multimedia*, 2026.

Jianxuan Yang, Xiaoran Yang, Lipan Zhang, Xinyue Guo, Zhao Wang, and Gongping Huang. MultiSoundGen: Video-to-audio generation for multi-event scenarios via SlowFast contrastive audio-visual pretraining and direct preference optimization. In *ACM International Conference on Multimedia Retrieval*, pp. 1918–1926, 2026.

Yiming Zhang, Yicheng Gu, Yanhong Zeng, Zhening Xing, Yuancheng Wang, Zhizheng Wu, Bin Liu, and Kai Chen. FoleyCrafter: Bring silent videos to life with lifelike and synchronized sounds. *International Journal of Computer Vision*, 134(1):46, 2026.

Qian Zhang, Yuqin Cao, Yixuan Gao, and Xiongkuo Min. VidAudio-Bench: Benchmarking V2A and VT2A generation across four audio categories. *arXiv preprint arXiv:2604.10542*, 2026.

Tianhong Zhou, Mingyang Han, Boyu Li, Yuxuan Jiang, Jiaxin Ye, Dongxiao Wang, Haoxiang Shi, Kunpeng Wang, Jun Song, Cheng Yu, and Bo Zheng. AV-SyncBench: Decoupled benchmarking of temporal and semantic audio-visual synchronization. *arXiv preprint arXiv:2607.00726*, 2026.

Appendix Table of Contents

A Benchmark Construction
- A.1 Source Selection 15
- A.2 Test Construction 15
- A.3 Source Quality Checks 15

B Evaluation Protocol
- B.1 Data Splits 16
- B.2 Inference Inputs 16
- B.3 Sound Classification by Interval 17
- B.4 EA, PR, and JS Definitions 17
- B.5 Confidence Intervals 18

C Controlled Results
- C.1 Full Benchmark 18
- C.2 Additional Edit Tests 18
- C.3 Source Composition and Shot Order 19
- C.4 Window Stitch 21

D Metric Validation
- D.1 Evaluator Agreement 22
- D.2 Audio Only Evaluation 22
- D.3 Human Listening Audit 23
- D.4 Whole Clip and Synchronization Metrics 23

E Temporal Evidence and Interventions
- E.1 Attention, Probes, and Input Swaps 24
- E.2 Using Each Event at Its Own Time 27
- E.3 Inference Procedure 29
- E.4 Inference Cost and Boundary Quality 30
- E.5 Results Across Generator Architectures 31

F Learning Sound and Time Alignment
- F.1 Training with a Temporal Attention Window 31
- F.2 Training with Paired and Shuffled Sounds 32

G Tests on Continuous Video
- G.1 Data and Statistical Units 33
- G.2 Context Controls 33
- G.3 Annotated Sound and Time Pairs 34
- G.4 Event Boundaries from Silent Video 34
- G.5 Sound and Time Pairs from Silent Video 34
- G.6 Audio Quality 35

H Statistics and Reproducibility
- H.1 Statistical Units 36
- H.2 Reproducibility 36

I Discussion 36

A Benchmark Construction

A.1 Source Selection

We construct the main benchmark from the official VGGSound test pool. Media checks first remove missing or unreadable files, nearly silent audio, and clips dominated by speech or music. Audio and visual checks then remove class mismatches, sources outside the frame, and sounds that do not occur with the visible event. Manual review checks source visibility, the action and label, reference audio quality, and timing. Source selection uses the original media and annotations. We also review the class inventory so that each retained Foley sound has a distinct evaluation target.

The resulting inventory contains 1,024 clips from 64 classes and yields 2,048 paired edits. Source clips do not cross evaluation groups. Smaller diagnostic experiments use separate sources, and the evaluation size analysis uses subsets of the main benchmark.

Additional edit and continuous video tests use separate source clips. Greatest Hits (Owens et al., 2016) supplies visible impacts for insertion and deletion, with both source pairs and event onsets reviewed before rendering. EPIC-SOUNDS (Huh et al., 2023) supplies labels and time spans for continuous video. We apply the media, annotation, visibility, and class filters before generation. Appendix G.1 gives the resulting splits.

Token attention and the metric identity control use a smaller set because they require many forward passes per example. This set contains 64 clips from eight balanced classes and 16 groups without repeated source clips. It includes impacts, repeated events, and sustained sounds. Source composition, shot order, unequal duration, and multiple replacement tests use all 256 groups. Every split and bootstrap draw keeps each group of four intact.

A.2 Test Construction

Every rendered example lasts eight seconds at 25 fps on a 384×384 canvas. Within a group, source clips a, b, c, d remain fixed while their order and switching rate change. Table 6 lists the resulting tests.

Table 6: Controlled video constructions. Isolated clips test source recognizability, alternating schedules vary the number of sound switches, sequences with four source clips vary composition and order, and sequences with one repeated source clip retain the cuts while keeping one sound identity. The table describes the 16-group diagnostic set. The main replacement benchmark uses 256 groups.

Test	Source order	Videos/group	Videos	Intervals/video	Scored intervals
Source shown alone	a, b, c, d	4	64	1	64
4 s switch	ab, ba	2	32	2	64
2 s alternation	$abab$, $baba$	2	32	4	128
1 s alternation	$ababab$, $bababa$	2	32	8	256
Four source clips	four orders of a, b, c, d	4	64	4	256
Repeated source clip	cuts every 2 s and 1 s	2	32	4 / 8	192
Total		**16**	**256**		**960**

Each construction changes one factor. Isolated source clips remove competing events. Alternating schedules change how often sound identity switches. Schedules with four source clips test whether each shot retains its own sound. Schedules that repeat one source clip keep the cuts and remove identity changes. Together, the 256 videos yield 960 scored intervals.

Every source clip is also rendered alone for eight seconds and generated once. We classify the complete output and the crop aligned with the shot later used in a sequence. If that shot begins at offset o_k and occupies $I_k = [s_k, e_k)$ in the sequence, the isolated crop is $[o_k, o_k + e_k - s_k)$. This construction compares waveforms aligned to the same shot without repeating, stretching, or padding a two-second input.

The direct prompt is exactly "Generate the audio corresponding to this video." It names neither the sound class nor an event time. ThinkSound and PrismAudio use this prompt in one setting and their visual reasoning prompt in another. Original VGGSound prompts remain in the metadata and are never substituted for the direct prompt.

A.3 Source Quality Checks

For each source, we verify the clip, event time, label, and review decision. Qwen3-VL-8B inspects uniformly sampled frames without seeing the candidate label. Qwen2.5-Omni-7B separately captions the source audio without seeing that label. An audio semantic gate then averages calibrated CLAP 630k AudioSet fusion and LanguageBind Audio FT scores for the VGGSound label. We require this mean to be at least 0.85, the calibrated CLAP component to be at least 0.80, and the fused audio and visual confidence to be at least 0.65. These thresholds remove weak or conflicting candidates. We then inspect every retained candidate and its sampled frames. The checks keep each target tied to a visible event and a clear reference sound.

B Evaluation Protocol

B.1 Data Splits

All generators use their released inference paths on shared splits and paired inputs. The source group is the unit for splitting and resampling. Isolated generations, replacements, reversed orders, and aligned crops made from the same four clips therefore stay together. The main benchmark and its VGGSound extensions contain 256 groups from 1,024 clips. The Greatest Hits insertion and deletion test uses 128 groups. Token attention and the metric identity control use 16 groups from 64 clips, and the attention figures average 64 rendered sequences from these groups. Tables and figures state the unit when it differs from the main benchmark.

Paired variants share the prompt, output duration, sampling configuration, and seed whenever the released model exposes one. Each model retains its published sample rate and codec. Evaluation resampling is deterministic, and silent output, missing files, or decode failures count as errors. Tables report both requested and valid sample counts when they differ.

B.2 Inference Inputs

Each model retains its published inputs. MMAudio combines video, text, and a synchronization stream, while Omni2Sound and HunyuanVideo-Foley XL use their native video and text paths. ControlFoley, FoleyCrafter, and Video-Foley keep their released temporal, semantic, or event features, and V-AURA uses video alone. ThinkSound and PrismAudio are evaluated with both the direct prompt and their visual reasoning prompt. SelVA, Woosh, CAFA, Diff-Foley, AudioX, and AudioX-Turbo use their released video inputs. Table 7 records the actual sampler and guidance values used for every evaluation setting.

Table 7: Evaluation settings. Checkpoints, model inputs, samplers, sampling steps, and classifier-free guidance values for the controlled benchmark settings. ThinkSound and PrismAudio are each tested with a direct prompt and their visual reasoning prompt. Benchmark class labels and interval boundaries are withheld from every generator.

Model	Input	Sampler	CFG
Flow matching			
MMAudio	video + direct prompt	Euler (25)	4.5
ControlFoley	video + direct prompt	25 steps	4.5
ThinkSound	video + direct prompt	24 steps	5.0
PrismAudio	video + direct prompt	24 steps	5.0
ThinkSound	video + visual reasoning	24 steps	5.0
PrismAudio	video + visual reasoning	24 steps	5.0
SelVA	video + direct prompt	25 steps	4.5
Woosh-VFlow	video + direct prompt	Dopri5 (10^{-3} tolerance)	4.5
Woosh-DVFlow	video + direct prompt	4 steps	3.0
Diffusion			
Omni2Sound	video + direct prompt	100 steps	2.5
HunyuanVideo-Foley XL	video + direct prompt	50 steps	4.5
FoleyCrafter	video + direct prompt	25 steps	1.0 / 0.2
Video-Foley	video + direct prompt	DDIM (200)	3.5
CAFA	video + direct prompt	DPM++ 3M SDE (50)	7.0 / 0.5
Diff-Foley	video only	DPM-Solver (25)	4.5 / 50
AudioX	video + direct prompt	DPM++ 3M SDE (250)	7.0
AudioX-Turbo	video + direct prompt	DMD (4)	None
AudioX-MAF	video + direct prompt	DPM++ 3M SDE (250)	7.0
AudioX-MAF-MMDiT	video + direct prompt	DPM++ 3M SDE (250)	7.0
Autoregressive			
V-AURA	video only	temperature .95, top-k 128	6.0
SpecVQGAN	video only	temperature 1.0, top-k 50%	None

Within each model, paired variants use the same seed. HunyuanVideo-Foley XL is deterministic with seed 1. Video-Foley keeps its released seed policy for each sample. FoleyCrafter reports separate scales for its semantic adapter and temporal ControlNet. AudioX-Turbo uses its distilled DMD sampler with four steps and no classifier-free guidance branch. SpecVQGAN samples autoregressive codebook tokens with temperature and top-k and has no CFG branch. Their CFG entries are marked "None." Woosh-VFlow uses relative and absolute ODE tolerances of 10^{-3}. CAFA uses classifier-free guidance 7.0 and asymmetric guidance 0.5. Diff-Foley uses its released CAVP encoder, latent diffusion checkpoint, and double guidance classifier. Its classifier-free scale is 4.5 and its classifier scale is 50. All synchronization, motion, and event features come from the input video. The generator never receives benchmark class labels, interval annotations, or event scripts.

All 21 settings enter the main replacement, isolated source comparison, error breakdown, and continuous video baseline. The other controls use representative models that expose the operation being tested. Appendix H reports the coverage and sample count for each experiment.

B.3 Interval Classification

We build each class prototype from reference groups outside the group being evaluated. For encoder ϕ, let $e(r) = \phi(r)/\|\phi(r)\|_2$ be the normalized embedding of reference audio excerpt r. For class c and test group g, we average the embeddings from all other groups and normalize again. These references form $\mathcal{R}_c^{(-g)}$.

$$\bar{e}_c^{(-g)} = \frac{1}{|\mathcal{R}_c^{(-g)}|} \sum_{r \in \mathcal{R}_c^{(-g)}} e(r), \qquad q_c^{(-g)} = \frac{\bar{e}_c^{(-g)}}{\|\bar{e}_c^{(-g)}\|_2}. \tag{2}$$

For generated interval $\hat{a}_{g,k}$, we define $e_{g,k} = \phi(\hat{a}_{g,k})/\|\phi(\hat{a}_{g,k})\|_2$. The nearest prototype gives its class:

$$\hat{c}_{g,k} = \arg\max_{c \in \mathcal{C}} e_{g,k}^\top q_c^{(-g)}. \tag{3}$$

A class enters $\mathcal{C}$ only if at least two source groups contain it. After classifying all intervals, sequence accuracy counts a sequence only if every interval is correct.

The error breakdown also separates a recognized class substitution from audio that is too weak or unclear to classify reliably. For each encoder, an interval is treated as recognizable only when its largest prototype similarity is at least the fifth percentile of target similarities on reference audio and its RMS is no lower than 12 dB below the first percentile of reference RMS values. An interval below either threshold enters the weak or unrecognized category. These thresholds affect only the diagnostic breakdown, not EA, PR, JS, or the nearest prototype labels in the main tables.

B.4 EA, PR, and JS Definitions

Let $p \in \mathcal{P}$ denote an edit pair, j_p its edited interval, and $\mathcal{U}_p$ its unchanged intervals. The original target in interval i is $c_{p,i}$, and the new target in the edited interval is c'_{p,j_p}. We write $\hat{c}_{p,i}$ and $\hat{c}'_{p,i}$ for the predictions before and after the edit. Only unchanged intervals that are correct before the edit can test preservation. Their set is

$$\mathcal{S}_p = \{i \in \mathcal{U}_p : \hat{c}_{p,i} = c_{p,i}\}, \qquad \mathcal{P}_+ = \{p \in \mathcal{P} : |\mathcal{S}_p| > 0\}. \tag{4}$$

The three measures are

$$\textbf{EA} = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \mathbf{1}[\hat{c}'_{p,j_p} = c'_{p,j_p}], \tag{5}$$

$$\textbf{PR} = \frac{\sum_{p \in \mathcal{P}_+} \sum_{i \in \mathcal{S}_p} \mathbf{1}[\hat{c}'_{p,i} = c_{p,i}]}{\sum_{p \in \mathcal{P}_+} |\mathcal{S}_p|}, \tag{6}$$

$$\textbf{JS} = \frac{1}{|\mathcal{P}_+|} \sum_{p \in \mathcal{P}_+} \mathbf{1}[\hat{c}'_{p,j_p} = c'_{p,j_p}] \prod_{i \in \mathcal{S}_p} \mathbf{1}[\hat{c}'_{p,i} = c_{p,i}]. \tag{7}$$

PR gives credit for each eligible unchanged interval that remains correct, whereas **JS** gives credit only when the edited interval and all eligible unchanged intervals are correct in the same pair. A pair with no initially correct unchanged interval can contribute to **EA**, but it is excluded from **PR** and **JS**. Eligibility depends on the model and evaluator because it comes from the original prediction. Missing or failed outputs remain incorrect. PR is therefore reported with its eligible count. It measures stability among sounds that the same model produced correctly before the edit. Table 8 reports the CLAP denominators used with Table 1.

Table 8: Effective sample counts for the CLAP preservation measures. EA uses all 2,048 edit pairs. PR uses unchanged intervals that were correct before the edit, and JS uses pairs containing at least one such interval. The counts differ because each model produces a different number of correct sounds before editing.

Model	PR intervals	JS pairs	Model	PR intervals	JS pairs
MMAudio	1,254	1,076	FoleyCrafter	333	327
ControlFoley	1,497	1,200	Video-Foley	174	174
ThinkSound†	1,044	915	CAFA	606	539
PrismAudio†	897	782	Diff-Foley	819	727
ThinkSound‡	1,260	1,055	AudioX	588	571
PrismAudio‡	1,227	1,019	AudioX-Turbo	870	813
SelVA	1,410	1,107	AudioX-MAF	816	735
Woosh-VFlow	357	337	AudioX-MAF-MMDiT	822	756
Woosh-DVFlow	588	552	V-AURA	2,103	1,485
Omni2Sound	1,065	935	SpecVQGAN	408	394
HunyuanVideo-Foley XL	1,395	1,125			

The eligible counts vary by more than an order of magnitude across settings. The count shows how often a model produces an unchanged sound before the edit. PR then shows how often that sound survives the edit.

EA, PR, and JS require a base soundtrack and a locally edited counterpart. Complete sequence tests have no edited interval. Feature probes score decodability, and quality tables score the whole waveform, so those experiments use their own measures.

B.5 Confidence Intervals

Point estimates use every eligible observation. Confidence intervals account for shared sources. Each bootstrap draw resamples source groups and carries all variants of a chosen group together. Paired differences are computed before resampling. The number of draws for each experiment is listed in Table 30. References of one second are harder to classify, so conclusions at that duration must also agree in direction across interval evaluators.

C Controlled Results

The local revision problem remains across evaluators and controlled changes to edit type, temporal layout, source composition, and shot order.

C.1 Full Benchmark

The complete benchmark contains 64 sound classes, 1,024 clips, and 256 independent source groups. Figure 6 repeats the isolated source comparison on random subsets to determine whether its gap is sustained as the evaluation grows.

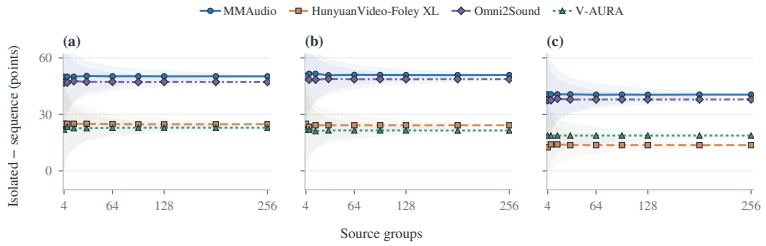

Figure 6: Evaluation size stability. Panels (a), (b), and (c) show accuracy when a source clip is generated alone minus its accuracy in sequence under CLAP, BEATs, and PANNs across random source group subsets. Bands give the central 95% range. The positive, stable gap persists as more source groups are added.

Table 9 gives one matrix for EA, PR, and JS measures under CLAP, BEATs, and PANNs. Absolute rates change with the evaluator, but the diagnosis does not: preservation is much easier than completing the update and preservation together. The isolation and sequence scores remain in the main table because they show whether a missing sound can be generated outside the sequence.

The matched controls test class novelty and seed variation (Table 10). A donor from the same class raises PR and JS for MMAudio, Omni2Sound, HunyuanVideo-Foley XL, and V-AURA, yet JS remains at most 24.0%. SpecVQGAN does not improve. Collateral changes therefore also occur without a new class. A second seed usually moves interval accuracy on complete sequences by less than 1.5 points. HunyuanVideo-Foley XL is the exception, changing by 3.3 to 6.2 points. Sampling matters for that model, but does not remove the broader difficulty of updating one sound while preserving the others.

Figure 7 shows where those predictions go: many become another recognized class rather than merely weaker audio. The unchanged interval often still contains a recognizable sound, but it is no longer the sound associated with that event.

C.2 Additional Edit Tests

Replacement changes the expected sound while keeping event count fixed. Insertion and deletion change event count. Unequal intervals change the temporal layout. Simultaneous replacements enlarge the edit. These tests ask whether the same update and preservation problem remains under broader changes.

Insertion and deletion use five systems and reviewed source pairs from 128 groups. Event onsets are also checked before rendering, so their targets remain explicit despite the change in event count. A calibrated onset envelope gives a binary present or absent decision around the center of each interval. Insertion EA requires an onset in the edited interval. Deletion EA requires no onset there, so a remaining impact above the threshold is an error. The unchanged intervals use the same eligibility rule as the replacement test.

The variable layout test keeps total duration at eight seconds but samples unequal boundaries for two to five shots, then replaces each available position in turn across all 256 groups. The simultaneous edit test replaces one, two, or three intervals in the same sequence. Its update measure requires every edited interval to contain its new class, preservation requires every eligible unchanged interval to remain correct, and joint success requires both. Confidence intervals retain complete source groups.

Table 9: Local edit results under three interval evaluators. EA, PR, and JS are reported for all 21 settings under CLAP, BEATs, and PANNs (%). Across evaluators, PR generally exceeds EA and JS remains low.

	CLAP			BEATs			PANNs		
Model	EA	PR	JS	EA	PR	JS	EA	PR	JS
Flow matching									
MMAudio	17.6	50.4	5.4	17.8	52.7	5.1	13.4	57.9	4.8
ControlFoley	27.2	67.5	15.0	29.0	70.1	16.7	20.3	73.2	13.6
ThinkSound†	14.9	57.8	7.0	14.6	57.5	5.1	9.4	66.0	7.4
PrismAudio†	14.0	58.4	7.5	16.2	57.5	6.5	11.7	64.3	6.4
ThinkSound‡	20.1	58.1	8.2	20.5	57.1	9.0	12.5	65.4	8.9
PrismAudio‡	18.5	57.2	7.1	21.8	51.7	8.6	15.6	60.1	8.7
SelVA	21.1	67.9	13.2	22.6	69.4	14.8	15.3	74.1	9.4
Woosh-VFlow	6.0	49.0	1.2	5.5	43.4	0.7	4.5	65.5	0.3
Woosh-DVFlow	9.2	57.5	3.1	9.9	56.6	3.0	5.8	64.5	1.9
Diffusion									
Omni2Sound	16.8	50.9	4.2	19.9	52.1	5.7	14.1	59.5	5.2
HunyuanVideo-Foley XL	21.2	69.5	11.6	22.4	71.1	12.9	16.6	71.3	10.0
FoleyCrafter	4.1	52.0	0.6	4.7	48.7	1.6	5.4	62.4	2.3
Video-Foley	2.0	81.0	0.6	1.5	60.8	0.6	4.0	81.8	0.0
CAFA	10.4	54.5	3.3	9.0	51.1	3.1	6.8	62.1	2.8
Diff-Foley	12.1	52.6	4.5	15.2	51.8	5.8	10.6	55.3	3.9
AudioX	6.7	51.4	1.1	7.2	49.4	0.7	5.4	55.2	0.8
AudioX-Turbo	15.3	49.8	3.4	13.9	45.8	2.6	11.2	50.0	3.1
AudioX-MAF	13.3	48.3	3.4	13.9	48.5	1.6	10.7	46.9	2.2
AudioX-MAF-MMDiT	12.3	46.8	2.8	12.6	45.1	3.1	10.7	51.0	2.6
Autoregressive									
V-AURA	**33.6**	78.2	**22.6**	**36.5**	78.8	**23.5**	**27.7**	80.2	**18.2**
SpecVQGAN	3.9	29.2	0.0	5.8	21.4	0.3	4.5	54.5	0.0

†Direct prompt. ‡Visual reasoning prompt. PR and JS use eligible sets for each model listed in Table 8.

Table 10: The replacement class and generation seed have different effects. Results (%) compare donors from another class and the same class on 128 source groups, followed by the base sequences under two seeds. Same class donors usually raise PR and JS, while seed changes are small.

		Replacement from different class			Replacement from same class			Base sequence	
Model	Evaluator	EA	PR	JS	EA	PR	JS	**Seed 0**	**Seed 1**
MMAudio	CLAP	17.4	45.7	5.5	16.7	56.1	8.4	16.6	17.4
	BEATs	17.9	54.5	6.4	17.8	62.3	8.4	17.0	15.8
	PANNs	12.0	57.4	2.1	12.6	64.5	6.5	11.8	11.0
Omni2Sound	CLAP	16.9	52.7	4.4	15.5	62.1	5.3	15.0	14.9
	BEATs	19.0	51.9	5.8	16.6	63.6	6.0	17.2	16.6
	PANNs	13.2	58.8	5.9	12.1	65.9	6.0	13.0	11.9
HunyuanVideo-Foley XL	CLAP	18.7	71.1	11.1	23.5	77.9	13.4	22.9	17.2
	BEATs	20.8	74.5	13.4	22.0	81.0	15.2	20.7	14.6
	PANNs	15.0	67.8	8.7	17.2	75.5	11.6	17.7	14.4
V-AURA	CLAP	32.6	77.9	21.9	34.3	80.6	23.3	35.3	35.4
	BEATs	35.8	79.2	23.3	36.2	82.2	24.0	37.4	36.5
	PANNs	23.7	76.8	14.1	25.8	81.7	18.4	26.4	27.8
SpecVQGAN	CLAP	4.3	26.1	0.0	5.1	25.0	0.0	4.3	4.8
	BEATs	6.0	19.9	0.0	6.2	20.8	0.0	5.2	6.2
	PANNs	5.4	69.4	0.0	3.6	83.3	0.0	3.5	3.7

The replacement blocks use the same source groups, donor groups, prompt, sampler, and seed. Only the donor class relation changes. Seed columns report interval accuracy for complete sequences from the same 128 source groups.

The update and preservation gap remains when the operation and timeline change (Table 11). Each block changes a different part of the task: whether a sound is present, how the timeline is divided, or how many targets must be updated at once. Deletion can be easier than insertion, while shorter intervals and simultaneous edits usually reduce joint success. The size of the change varies by model. Deletion EA tests whether a sound is absent, while insertion and replacement require the intended sound to be present. A higher deletion score therefore does not imply better sound generation. JS also checks the sounds that should remain unchanged.

C.3 Source Composition and Shot Order

The source composition control changes sound diversity while holding sequence length and cut count fixed. Each setting generates either one repeated source or four distinct sources from matched groups. Their difference measures the cost of separating several sound identities (Figure 3(a)).

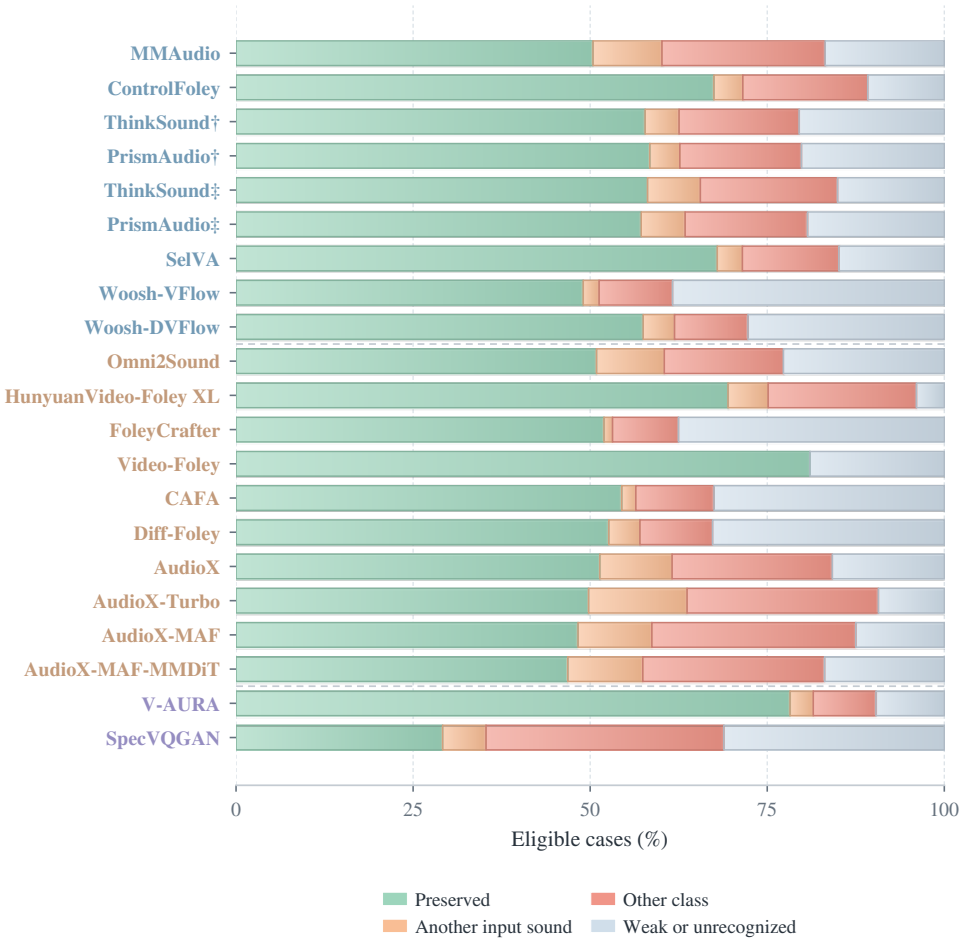

Figure 7: Where unchanged sounds go after a local replacement. Initially correct unchanged sounds are separated into preserved predictions, substitutions by another sound shown in the input, predictions outside the input sound set, and weak or unrecognized audio. † marks the direct prompt, and ‡ marks the native visual reasoning prompt. Substitutions and other recognized classes account for much of the loss, so a local edit often changes sound identity rather than merely weakening the output.

Table 11: Local revision remains difficult as the edit operation and timeline change. CLAP results (%) cover insertion and deletion, two to five unequal intervals, and one to three simultaneous replacements. JS falls as sequences and edits grow.

	Insertion			Deletion			JS by interval count				JS by edited intervals		
Model	EA	PR	JS	EA	PR	JS	$K=2$	$K=3$	$K=4$	$K=5$	One	Two	Three
MMAudio	59.8	88.1	47.0	71.9	83.0	51.0	11.2	6.4	3.8	3.5	6.5	0.6	0.0
HunyuanVideo-Foley XL	68.6	82.8	46.6	57.4	89.3	44.0	20.2	11.7	9.7	9.0	12.7	1.5	0.6
Omni2Sound	41.2	86.2	25.1	77.7	68.8	40.7	13.6	9.2	4.5	2.6	4.9	0.3	0.0
V-AURA	49.2	88.3	36.1	79.1	78.8	53.6	40.7	25.0	20.3	17.3	22.0	4.4	1.1
SpecVQGAN	16.0	90.5	9.5	78.1	52.8	31.8	0.0	1.0	1.0	1.4	0.8	0.0	0.0

For simultaneous edits, JS requires every edited interval to contain its new target and every eligible unchanged interval to remain correct. Point estimates use all 256 source groups for layout and replacement tests. insertion and deletion use 128 groups.

The order test renders all 24 permutations of each four shot group. MMAudio, Omni2Sound, V-AURA, and SpecVQGAN use two seeds, while HunyuanVideo-Foley XL retains its fixed seed. We average over orders and available seeds within each of the 256 source groups before resampling (Table 12).

The same footage produces different sounds across positions. V-AURA has the largest order range and reversal rate, while even the least sensitive model changes predictions when the shot order is reversed. Sequence context therefore changes which sound a model assigns to a shot. The change is larger than a shift in confidence for one fixed label.

Table 12: Sensitivity to shot order. Interval and sequence accuracy are evaluated across all 24 orders of the same four shots for five models and 256 source groups (%). Order range is the best minus worst result, Reversal is the share of predictions changed by reversing the order, and First−fourth is the first position minus the fourth.

Model	Evaluator	Interval acc. ↑	Sequence acc. ↑	Order range	Reversal	First−fourth
MMAudio	CLAP	19.2	0.0	45.9	19.1	-2.3
	BEATs	19.7	0.0	46.5	20.1	-1.6
	PANNs	15.6	0.0	37.4	14.6	-1.5
HunyuanVideo-Foley XL	CLAP	22.3	0.2	52.8	26.0	+11.8
	BEATs	21.7	0.1	48.0	23.4	+10.4
	PANNs	16.9	0.1	39.6	17.5	+7.1
Omni2Sound	CLAP	16.6	0.0	40.9	17.6	+8.3
	BEATs	19.0	0.0	43.5	20.1	+11.1
	PANNs	14.3	0.0	36.4	15.4	+6.4
V-AURA	CLAP	35.0	1.0	61.0	35.1	+29.2
	BEATs	36.8	1.0	61.5	35.9	+29.2
	PANNs	30.0	0.7	51.8	27.6	+24.7
SpecVQGAN	CLAP	6.4	0.0	25.9	8.9	-1.9
	BEATs	7.0	0.0	28.5	10.6	-2.6
	PANNs	5.1	0.0	16.3	4.5	-0.8

C.4 Window Stitch

The isolated source result suggests a direct baseline: generate each source separately and join the aligned crops. Window Stitch measures both interval accuracy and the waveform change at each join.

Window Stitch raises CLAP interval accuracy for most of the 21 settings (Table 13). For MMAudio, a 50 ms crossfade retains 67.7% interval accuracy, reduces the normalized sample jump from 0.153 in joint generation to 0.074, and keeps spectral flux close to the joint generation value (0.315 versus 0.299). A short crossfade therefore recovers sound identity without a hard join. Window Stitch works well when hard cuts separate shots that can be generated independently.

Table 13: Window Stitch across settings. CLAP interval accuracy is reported for all 21 settings using a hard join and two crossfade widths (%). Window Stitch improves most settings, and short crossfades retain nearly all of the gain.

Setting	Full video	Hard join	20 ms	50 ms
MMAudio	20.4	70.6	68.8	67.7
ControlFoley	24.4	42.6	41.9	42.1
ThinkSound†	17.0	45.1	43.2	41.4
ThinkSound‡	20.5	59.1	55.4	53.8
PrismAudio†	14.6	31.0	30.6	29.5
PrismAudio‡	20.0	46.6	45.9	44.1
SelVA	22.9	35.3	35.4	33.9
Woosh-VFlow	5.8	13.6	13.7	14.0
Woosh-DVFlow	9.6	26.9	26.9	25.0
Omni2Sound	17.3	64.6	63.8	61.4
HunyuanVideo-Foley XL	22.7	47.5	46.3	43.9
FoleyCrafter	5.4	5.4	5.4	4.8
Video-Foley	2.8	3.0	2.9	2.9
CAFA	9.9	14.4	14.8	14.2
Diff-Foley	13.3	31.8	30.2	28.6
AudioX	9.6	37.9	36.8	34.8
AudioX-Turbo	14.2	55.8	54.5	52.6
AudioX-MAF	13.3	54.7	53.5	51.2
AudioX-MAF-MMDiT	13.4	50.2	48.3	46.9
V-AURA	34.2	57.1	56.1	55.4
SpecVQGAN	6.6	18.2	17.1	16.4

†Direct prompt. ‡Visual reasoning prompt.

What Window Stitch shows. Window Stitch is not a failed alternative to Regional Flow. Its strong result helps explain the failure. After a sequence is split into separate clips, the generators can produce many sounds that they place at the wrong events when given the full video. Window Stitch supplies the placement outside the model: it generates each source clip separately and puts the resulting crop in the known interval. Regional Flow

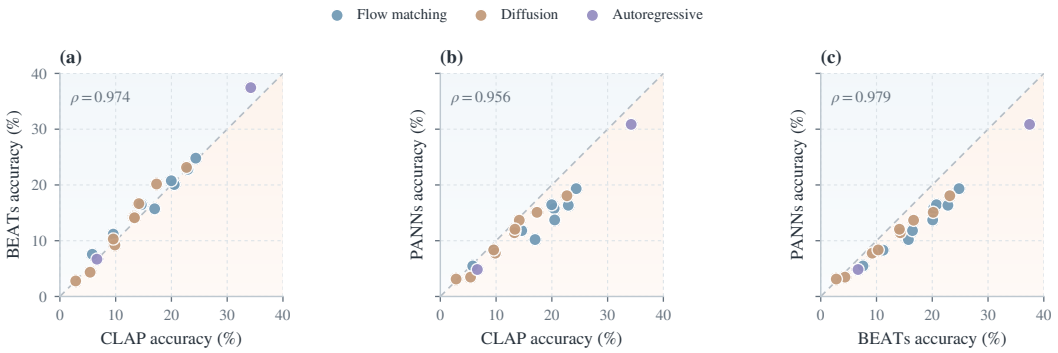

Figure 8: Interval evaluators rank the primary settings consistently. Each point is one evaluation setting scored on complete sequences. Dashed lines mark equal accuracy, and ρ denotes Spearman rank correlation.

Table 14: Audio-only evaluator check. Qwen2.5-Omni classifies 64 blinded two-second intervals for each model. It chooses among the four sound classes in each sequence and gives the same model order as CLAP. Brackets show 95% confidence intervals.

Audio source	Qwen2.5-Omni	CLAP	BEATs
MMAudio	40.6 [31.2, 50.0]	32.8	39.1
Omni2Sound	43.8 [35.9, 51.6]	35.9	37.5
HunyuanVideo-Foley XL	54.7 [42.2, 67.2]	42.2	45.3
V-AURA	73.4 [65.6, 81.2]	67.2	76.6
Reference audio	96.9 [92.2, 100.0]	100.0	98.4

instead keeps the full video and uses the information from each cell at that cell's time. The matched control in Table 19 uses the same information at another event's time and removes much of the gain. This result shows that when the model uses each event matters.

When each method applies. Window Stitch and Regional Flow apply to different cases. Window Stitch works well when each source clip can be generated independently. Regional Flow keeps the full video during one sampling path and uses each event at its own time. Surrounding frames remain available to help identify the object, action, or scene. On continuous video, cropping around one event lowers MMAudio accuracy under SSAST, CLAP, and PANNs (Appendix G). Both methods require usable event intervals in our study. A stitcher could retain more context with overlapping windows or nearby frames, but we do not test those designs. We therefore do not claim that Regional Flow always outperforms Window Stitch. Dialogue, ambience, music, overlapping events, and sound bridges remain open questions.

D Metric Validation

D.1 Evaluator Agreement

On the reference inventory used for calibration, CLAP and BEATs each reach 80.1% accuracy. Both exceed 98% on the smaller set of eight classes. These checks evaluate reference recognition rather than generated audio quality.

The three interval evaluators also rank the expanded set consistently. Figure 8 reports pairwise Spearman correlations for interval accuracy on complete sequences after pooling all 21 primary settings. PANNs has a lower reference ceiling, so CLAP and BEATs remain the primary metrics.

D.2 Audio Only Evaluation

Qwen2.5-Omni provides a separate audio only check. It receives 320 blinded audio intervals of two seconds and chooses among the four shuffled classes in the soundtrack. Its reference accuracy is 96.9%, close to CLAP and BEATs on the same subset. Agreement on individual intervals is moderate, but Qwen and CLAP give the same model ranking (Table 14). Qwen agrees with CLAP on 64.1% of intervals and with BEATs on 65.6%. The isolated source clip ranking is the same under the embedding classifiers and the audio judge.

D.3 Human Listening Audit

The paired edit audit uses a predefined trial bank chosen without access to model scores. Trials are assigned anonymously. Each of the 15 listeners completes eight scored trials and two hidden content checks, giving 120 scored responses across MMAudio, HunyuanVideo-Foley XL, Omni2Sound, and V-AURA. Each model has 30 ratings, and each of the 20 trials has four to nine ratings. All submitted sessions satisfy the playback and study protocol. EA and JS use the same 97 complete decisive trials. Their pooled counts are 45/97 and 41/97, respectively. PR is measured over eligible unchanged intervals and preserves 196 of 217. For MMAudio, HunyuanVideo-Foley XL, Omni2Sound, and V-AURA, the common EA and JS denominators are 21, 28, 25, and 23 trials. The corresponding PR denominators are 43, 55, 52, and 67 eligible intervals. Seventeen target judgments are unsure. Agreement on known binary judgments is 86.6%, with nominal Krippendorff's $\alpha = 0.712$. Table 2 reports model and pooled rates.

The Window Stitch audit uses a separate page with the same design. Fifteen listeners provide 120 scored comparisons across 20 source groups. The source group mean preference for the stitched output is 96.7% for event matching, 93.2% for continuity, and 97.6% overall. The corresponding source group bootstrap intervals are 93.1 to 99.4%, 86.4 to 98.0%, and 93.7 to 100.0%. Other responses are ties or unsure. Table 15 summarizes the rates. Both audits ask whether the sound matches the visible event and whether unchanged sounds remain correct. Listeners can hear timbre, texture, weak sounds, and mixtures that a nearest prototype classifier reduces to one label. Unsure responses remain outside the decisive rate. Hidden examples check content consistency.

Table 15: Human comparison of Window Stitch and joint generation. Preference rates use decisive judgments for event matching, boundary continuity, and overall quality.

Criterion	Window Stitch preferred (%)
Event matching	96.7
Boundary continuity	93.2
Overall quality	97.6

Some drum and piano examples also remain natural when their shots are generated separately. A short crossfade can preserve a shared musical or reverberant layer across the join. This helps explain why listeners can prefer the stitched result even for sustained sounds.

D.4 Whole Clip and Synchronization Metrics

The first metric control cyclically permutes reference intervals. It preserves the original audio segments and unordered sound set while placing every sound at the wrong time. Figure 3(c) shows which interval, clip level, and synchronization measures respond to this change. A stricter control changes identity while retaining the temporal envelope. Every altered interval keeps the Hilbert amplitude envelope of its original sound but uses the phase of a cyclically permuted donor. Video, sound set, event times, and duration remain fixed, while each identity moves to a wrong interval. The original and altered RMS envelopes have a mean correlation of 0.973.

We score both the processed controls and identity swaps with the three interval evaluators, semantic metrics over the whole clip, and DeSync. Confidence intervals use 10,000 source group bootstrap draws. Table 16 compares the processed and swapped soundtracks, with degradation measured in each metric's worse direction.

Table 16: Sound identity and temporal envelope. The identity swap preserves each interval's amplitude envelope while replacing its sound with a donor from another interval. Degradation is reported in the worse direction of each metric, with 95% confidence intervals. Large semantic losses despite the preserved envelopes show that temporal energy alone does not retain interval sound identity.

Test	**Interval CLAP**	**Interval BEATs**	**Interval PANNs**
Control	99.6	98.4	79.7
Identity swap	23.8	21.1	13.3
Degradation (95% CI)	75.8 [69.1, 82.4]	77.3 [69.5, 85.2]	66.4 [58.6, 74.2]

Test	**DeSync (s)**	**ImageBind**	**LAION-CLAP**	**MS-CLAP**	**Whole clip CLAP**
Control	.006	.204	.249	.488	.997
Identity swap	.833	.126	.106	.403	.337
Degradation (95% CI)	.827 [.594, 1.064]	.078 [.052, .105]	.144 [.096, .193]	.085 [.054, .118]	.660 [.608, .708]

The controls expose different blind spots. Global measures can reward the correct unordered sound set, while synchronization can follow an amplitude envelope without identifying the action that produced it. Checking which sound occurs in each interval is therefore needed alongside both metric families.

These controls test what a metric can detect; they do not establish that the altered audio has unchanged perceptual quality. A soundtrack can contain the expected sounds and still place them at the wrong events. Conversely, correct interval labels do not guarantee natural timbre or smooth transitions. The interval scores and whole clip measures therefore answer different questions.

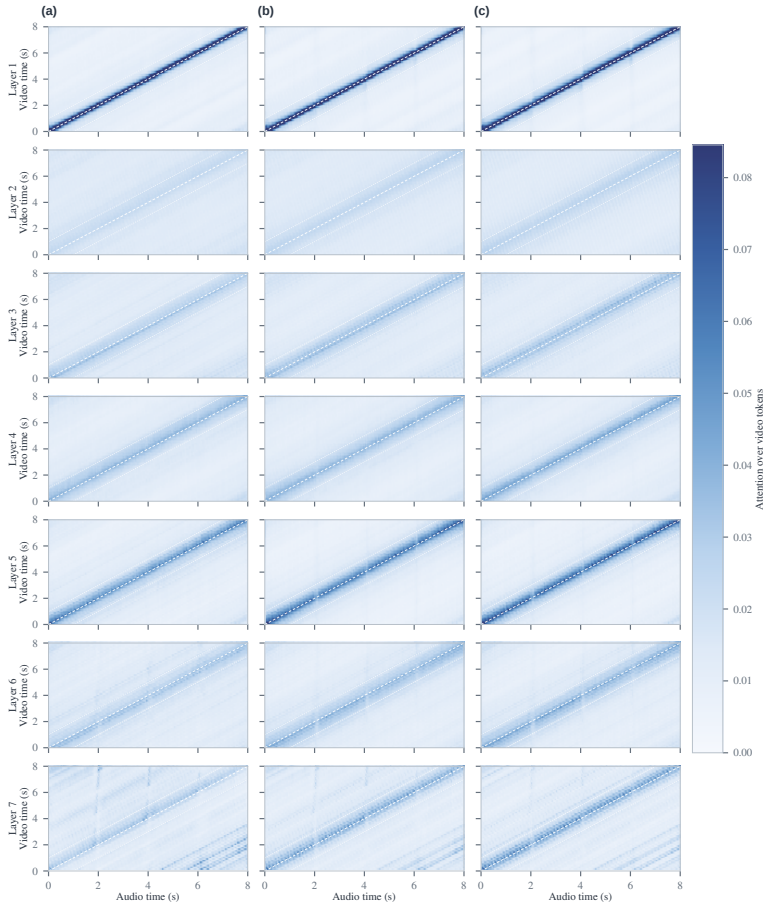

Figure 9: Attention before interval pooling. Panels (a), (b), and (c) show flow times 0.1, 0.5, and 0.9, respectively, while rows show the seven MMAudio joint attention layers. Each pixel measures attention from one audio token to one video token after averaging heads and source sequences and normalizing over video keys. Flow time runs from initial noise at zero toward generated audio at one. The center line marks equal time, and the flanking lines are one second away.

E Temporal Evidence and Interventions

Attention maps and window tests ask whether limiting visual access keeps each sound at the right time. Feature probes and input swaps test whether sound and timing still shape the output. Regional Flow tests whether each event must affect the audio when it occurs.

E.1 Attention, Probes, and Input Swaps

Temporal attention. Across 6,720 maps from 320 outputs, 41.6% of the visual attention comes from the same interval after grouping audio and video tokens into four intervals. At token resolution, 42.5% falls within one second of the audio position. The rest is split almost equally between earlier and later video (Figure 4).

Figure 9 shows the same result before tokens are grouped. Each pixel connects one audio token to one video token. Values are averaged over heads and sequences and normalized within each audio row. The center line marks equal time, and the outer lines mark one second on either side. Attention is strongest near equal time, but every layer and flow stage also reads more distant video.

The diagonal remains visible at every flow time, so MMAudio assigns the most attention near the matching video time. Structure away from the diagonal also appears near the start and remains until the end. Distant frames contribute throughout generation. The maps show where information can travel. The later Regional Flow test changes when each event can affect the audio.

The other architectures show different patterns at early, middle, and late generation stages. Similar interval averages can hide this variation. A model may read nearby evidence first and mix it with the scene later. Another model may reuse the same visual evidence across the full soundtrack.

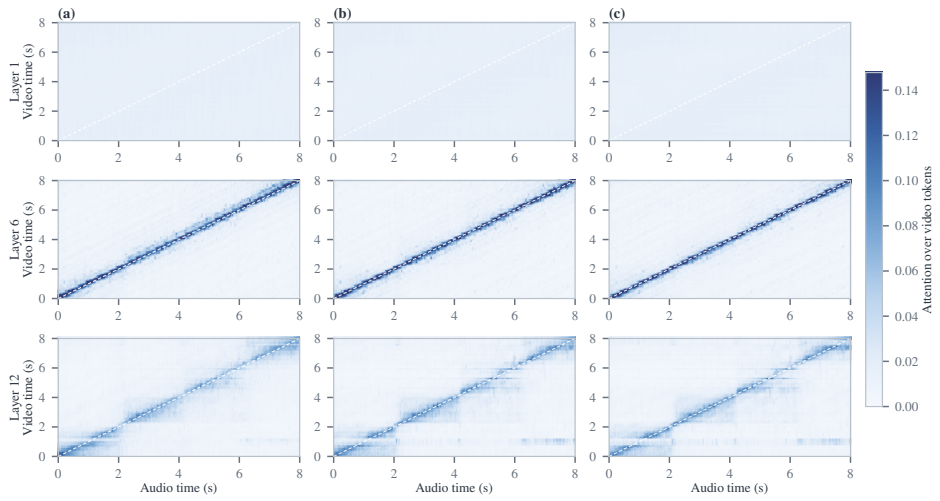

Figure 10: HunyuanVideo-Foley XL attention before interval pooling. Panels (a), (b), and (c) show early, middle, and late generation. Rows show layers 1, 6, and 12. Each pixel measures attention from one audio token to one video token after averaging 64 sequences and normalizing over video keys. The middle layer forms a sharp temporal diagonal, while the final layer retains a broader event dependent pattern.

HunyuanVideo-Foley XL changes with depth (Figure 10). The first layer reads widely, the middle layer forms a sharp diagonal, and the final layer broadens again. Even the middle and final blocks place 34.4% and 40.8% of their attention outside the corresponding interval. Local evidence becomes clearer in the middle and mixes with the wider scene later. A centered window can help, while the same window at the wrong time sharply lowers interval accuracy.

Omni2Sound shows horizontal bands across the audio axis (Figure 11). Many audio positions therefore read the same visual moments. About 25% of the attention at each observed depth comes from video at the same time, close to equal access across four intervals. No observed layer consistently prefers video from the same time. A temporal window changes the available frames, but it does not determine which visual event affects each output time.

V-AURA combines visual features aligned to frames with audio embeddings before causal self attention (Figure 12). A later edit cannot change audio that has already been generated, but an earlier edit can affect later positions. Under CLAP, 5.2% of predictions before the edit change, compared with 45.1% after it. The other models are more symmetric, and BEATs and PANNs show the same direction. Causal generation limits an edit to the audio that follows it. SpecVQGAN behaves differently, showing that autoregression can still produce another temporal pattern.

The same carry-over can be heard in sustained sources. A drum roll or piano phrase can continue into the next shot and mix with the new event. Later audio is generated from a history that already contains the earlier sound. The model must therefore decide both when a sound begins and when it ends.

Figure 13 places these three structures on the same interval scale. It also compares the direction in which predictions change around an edit. The models spread information in three different ways. HunyuanVideo-Foley XL becomes local and then mixes wider context. Omni2Sound shares visual evidence across time. V-AURA blocks future evidence but lets earlier events affect later audio. Their attention patterns differ, yet each model can put sounds at the wrong times. A temporal window removes some paths but does not fix the error.

Temporal attention windows. We next limit only the edges from audio queries to video keys, leaving the model and all other generation settings fixed. Table 17 compares the three architectures in one place. For MMAudio, a 0.5 second window raises PR by 5.2 points and JS by 0.3 points while reducing EA by 1.2 points. HunyuanVideo-Foley XL improves with one selected one-second window and drops sharply when the same window is shifted to a wrong time. Omni2Sound changes little. The useful width and window location depend on the architecture, layer, and sampling stage.

We also compare successful updates with and without complete preservation. The attention difference changes sign across layers and sampling stages. Attention outside the window has a per example Spearman correlation of -0.088 with collateral change, with a 95% interval of $[-0.329, 0.159]$. Its mass does not predict the outcome of one edit.

MMAudio shows why reducing visual access does not determine when a sound is produced. The 0.5-second window preserves more unchanged sounds, but it also makes the edited target less likely to appear. Moving the

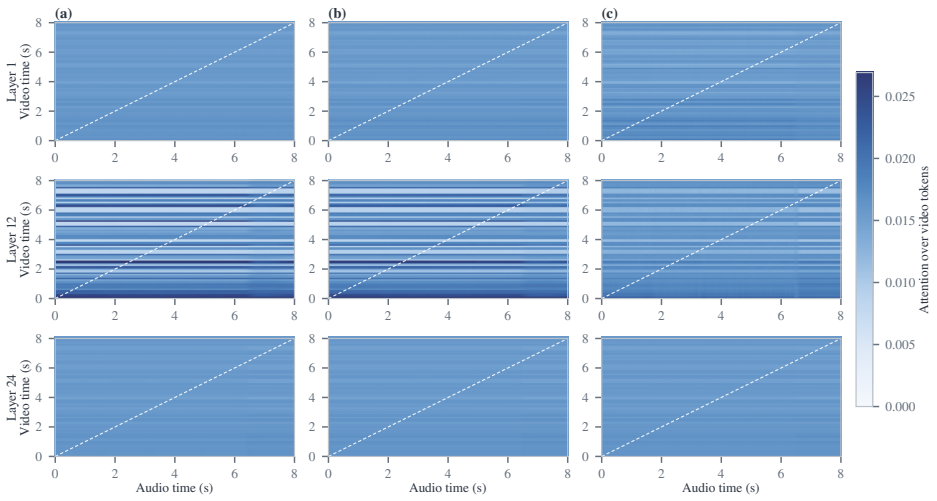

Figure 11: Omni2Sound attention before interval pooling. Panels (a), (b), and (c) show early, middle, and late generation. Rows show layers 1, 12, and 24. Each pixel measures attention from one audio token to one video token after averaging 64 sequences and normalizing over video keys. The horizontal structure shows that many audio positions read similar visual tokens instead of forming a stable time aligned diagonal.

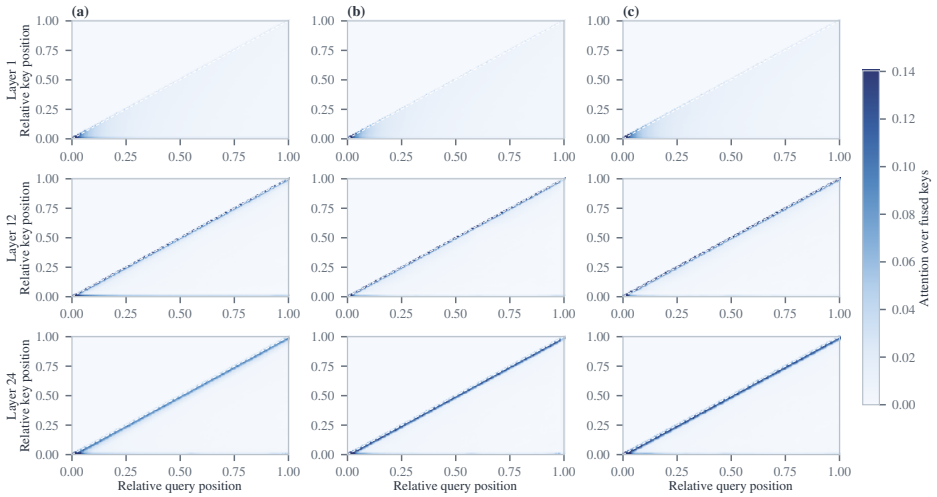

Figure 12: V-AURA fused attention before interval pooling. Panels (a), (b), and (c) show early, middle, and late autoregressive generation. Rows show layers 1, 12, and 24. V-AURA exposes causal self attention after visual and audio features have been fused, so the axes are relative positions within each decoding window rather than separate audio and video times. Each map averages 64 sequences. The empty region follows from the causal mask, while deeper layers concentrate near the current position.

window to a wrong time or selecting equally many video keys at random reduces joint success more sharply. Nearby frames therefore carry useful timing evidence, while wider context still helps identify the object and action. A fixed window makes the same compromise in every layer and example. Before controlling when each sound appears, we test whether the model still retains and uses the two signals that must be matched.

Feature probes. Linear probes test whether sound identity remains present during sequence generation. At each model's midpoint noise level, we record input projections, intermediate blocks, and the model output for 2,048 intervals. We average features within each video interval. A ridge classifier ($\lambda = 10$) predicts among 64 classes while leaving out one source group at a time. Shuffling labels within each sequence tests whether the decoded class is still attached to the right interval.

Input interventions. A probe shows that information is present, but not that the generator uses it. We test use with matched input swaps. We keep the weights, prompt, noise, solver, and seed fixed and reverse semantic features, timing features, or both. MMAudio reverses its CLIP and Synchformer sequences. HunyuanVideo-Foley XL reverses SigLIP2 and Synchformer tokens. Omni2Sound reverses semantic or temporal tokens but keeps positional embeddings. V-AURA receives the original or reversed video under matched seeds.

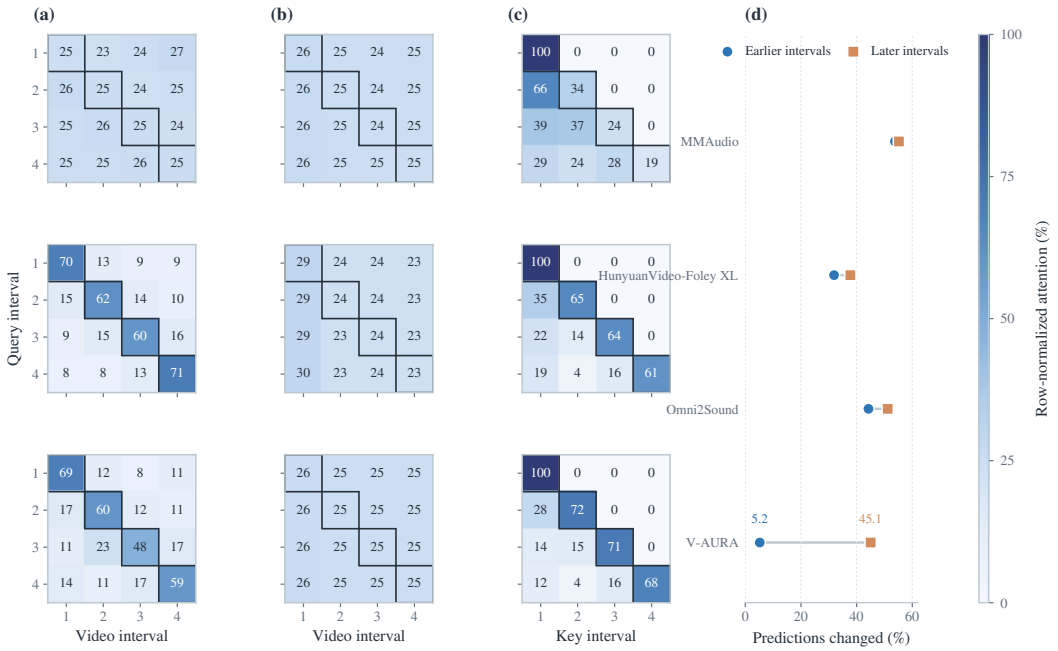

Figure 13: Temporal structure is associated with where an edit propagates. (a) HunyuanVideo-Foley XL develops a diagonal pattern but retains attention across intervals. (b) Omni2Sound remains close to uniform over video intervals. (c) V-AURA uses fused causal self attention, which excludes future keys and becomes more local with depth. Rows show early, middle, and late layers, corresponding to layers 1, 6, and 12 in (a), and layers 1, 12, and 24 in (b) and (c). Panel (c) therefore visualizes a different attention operator from (a) and (b). Each heat map averages 64 evaluation sequences, and boxes mark matching intervals. (d) V-AURA alone shows a strong directional asymmetry in CLAP prediction changes.

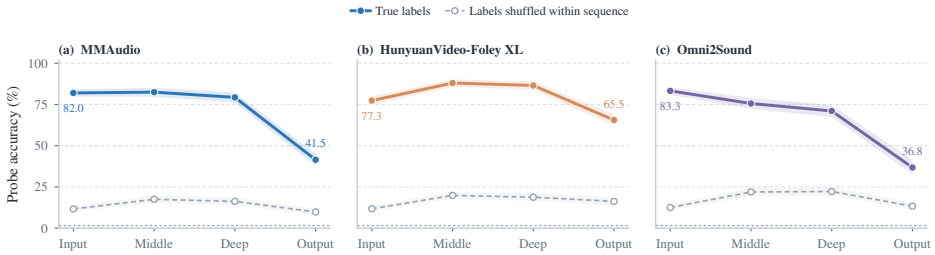

Figure 14: Sound identity remains available along the generation path. Each curve follows selected readouts from model input to output. Solid curves use the true interval labels. Gray dashed curves rotate labels within each sequence. Shaded bands give 95% confidence intervals from a source group bootstrap. The output remains predictive in all three generators, although less so than their earlier states.

Figure 14 reports the feature probes. Sound identity remains decodable in their deep states. Rotating labels among intervals sharply lowers probe accuracy, which shows that the signal carries some interval information. The matched interventions in Table 18 give complementary evidence. Reversing timing features moves MMAudio toward the reversed sound order under BEATs. Reversing both HunyuanVideo-Foley XL feature streams and reversing the V-AURA video move the output toward the reversed order under both evaluators. Sound and timing signals can therefore remain active even when the final interval is wrong.

E.2 Using Each Event at Its Own Time

Regional Flow. At each sampling step, Regional Flow evaluates the full video and each temporal cell at the same latent state. It subtracts the field predicted from the full video from the field predicted from one cell. A mask allows this difference to affect only the audio at the time covered by that cell. In the control, the same difference affects the audio during another event. Each run then continues along its own latent path.

Vector field interpretation. For a flow matching sampler, the latent state follows $dz_t/dt = v_\theta(z_t, t, h)$ (Lipman et al., 2023). Regional Flow leaves the training objective unchanged and modifies the field

Table 17: Temporal attention restrictions across three generator architectures. EA, PR, and JS under CLAP (%) when only the accessible video time changes. A one second window helps HunyuanVideo-Foley XL, but no setting improves all three architectures.

Model	Attention setting	EA	PR	JS
MMAudio	Released	16.8	50.3	6.4
	All layers, 0.25 s	13.3	54.8	5.7
	All layers, 0.5 s	15.6	55.5	6.7
	All layers, 1 s	15.2	50.3	5.1
	All layers, 2 s	15.2	47.4	3.7
	Layer 1, 1 s	15.6	49.7	6.5
	Final sampling third, 1 s	17.6	50.6	5.8
	Shifted window†	11.3	49.6	2.7
	Random window‡	12.5	50.4	3.4
HunyuanVideo-Foley XL	Released	23.8	69.3	15.1
	Unrestricted control	23.4	70.8	14.3
	All layers, 0.5 s	22.3	63.9	14.2
	All layers, 1 s	27.0	74.6	17.9
	All layers, 2 s	24.2	71.6	11.9
	Early layers, 1 s	26.2	71.9	13.9
	Middle layers, 1 s	21.1	71.4	13.2
	Late layers, 1 s	28.9	70.9	15.7
	Shifted window†	3.9	48.9	0.0
Omni2Sound	Released	17.2	56.7	4.2
	All layers, 0.5 s	17.6	57.4	4.2
	All layers, 1 s	17.6	57.4	4.2
	All layers, 2 s	17.6	57.4	4.2
	Early layers, 1 s	18.4	58.2	5.0
	Middle layers, 1 s	17.6	56.6	4.1
	Late layers, 1 s	17.6	59.1	3.4
	Shifted window†	17.6	57.4	4.2

A width of w lets each audio query read video keys within w seconds. For HunyuanVideo-Foley XL, the unrestricted control uses the same PyTorch attention path required by the windowed rows but applies no temporal mask. †The window of one second is moved to the wrong time. ‡Video keys are sampled randomly with matched sparsity. Omni2Sound's restriction changes semantic cross attention. Its separate timing path is unchanged.

Table 18: Reversing model inputs changes the generated sound order. Reversed order minus original order interval accuracy after the stated input reversal (points), with 95% confidence intervals from a source group bootstrap. Positive values mean that the output follows the reversed input.

Model	Reversed input	ΔCLAP			ΔBEATs			Groups
MMAudio	timing features	+7.8 [	-1.6,	18.8]	+17.2 [	4.7,	29.7]	16
HunyuanVideo-Foley XL	semantic and timing features	+14.8 [	8.6,	21.9]	+11.7 [	5.5,	18.0]	32
V-AURA	video order	+56.2 [	42.2,	70.3]	+57.8 [	45.3,	70.3]	16

only during sampling. Define the difference between the field for cell k and the field for the full video:

$$\Delta v_{k,t}(z) = v_\theta(z, t, h_k) - v_\theta(z, t, h_{\mathrm{g}}). \tag{8}$$

This quantity is evaluated at the same latent state. It is a difference between two predictions, rather than a derivative with respect to the input. Let $q_n = \sum_k m_{\rho(k),n}$. Equation 1 can then be written at audio position n as

$$\widetilde{v}_t^{(\rho)}[n] = (1 - \alpha_t q_n) v_{\mathrm{g}}[n] + \alpha_t \sum_{k=1}^{K} m_{\rho(k),n} v_k[n]. \tag{9}$$

The masks are normalized where they overlap, so $q_n = 1$ on covered positions. The equation therefore blends the full video field with the local field for that position. If $q_n = 0$, Regional Flow adds no local correction at that position.

Classifier-free guidance (Ho and Salimans, 2022) has the same decomposition. With unconditional field v_{u} and guidance scale γ, Equation 14 expands to

$$\widetilde{v}_t^{\mathrm{CFG}} = \underbrace{v_{\mathrm{u}} + \gamma(v_{\mathrm{g}} - v_{\mathrm{u}})}_{\text{full video guidance}} + \gamma\alpha_t \sum_{k=1}^{K} m_{\rho(k)} \odot \Delta v_{k,t}. \tag{10}$$

Thus Regional Flow keeps the released full video guidance and adds the difference caused by each temporal cell only at that cell's time. The mask limits where this difference can change the field. It does not restrict what the model may read when computing it.

This support statement concerns one update evaluated at one latent state. It does not imply that the final waveform outside a mask is mathematically unchanged. Let z_t^{g} follow the full video guided field b_t, and let z_t^{r} follow

$b_t + r_t$, where r_t is the masked term in Equation 10. If b_t is L-Lipschitz in z, the standard ODE perturbation bound gives

$$\|z_t^{\mathrm{r}} - z_t^{\mathrm{g}}\| \le \int_0^t e^{L(t-s)} \|r_s(z_s^{\mathrm{r}})\|\, ds. \tag{11}$$

The mask and α_t limit the added difference, but later fields are recomputed on a different latent path and can couple audio positions. Preservation and boundary quality must therefore be measured. Appendix E.4 reports both.

MMAudio, HunyuanVideo-Foley XL, and Omni2Sound expose compatible continuous latent updates, so the same intervention applies to all three. V-AURA generates autoregressively. We therefore test it with the input swaps above rather than applying Regional Flow to a different generation process.

Temporal cells and masks. Equation 1 computes the field for every temporal cell at the same latent state. We first encode the complete edited video of eight seconds. For each temporal cell, we take its visual and synchronization features and repeat them to the input length expected by the model under the same direct prompt. MMAudio uses eight cells lasting one second, whereas HunyuanVideo-Foley XL and Omni2Sound use four cells lasting two seconds. A target interval of two seconds therefore covers two MMAudio cells but one cell in the other models. Regional Flow and the control reuse the same cell features, so their comparison changes only when each cell affects the audio. Latent cell n has center time

$$\tau_n = (n + \tfrac{1}{2})D/N, \qquad m_{k,n} = \mathbf{1}\left\{t_k^{\mathrm{start}} \le \tau_n < t_k^{\mathrm{end}}\right\}, \tag{12}$$

Here, $D = 8$ seconds and N is the latent length for that model. We use hard masks with intervals closed on the left and open on the right. Overlapping masks are normalized to sum to one at each point.

Weight and event time. We next define the weight of the masked difference at each sampling step. For MMAudio, 25 Euler steps integrate from $t = 0$ (initial noise) to $t = 1$ (data), and the weight is

$$\alpha_t = \sigma\left(\frac{t_\star - t}{\beta}\right). \tag{13}$$

We use $t_\star = 0.70$ and $\beta = 0.08$. HunyuanVideo-Foley XL runs 50 scheduler steps, with sampling progress $p_s = s/(S-1)$ and weight $\alpha_s = \sigma((t_\star - p_s)/\beta)$. Omni2Sound runs 100 DPM++ 2M SDE steps. For its native noise level t_s, we set $p_s = 1 - \sin(\pi t_s/2)$ and use the same $t_\star$ and β. In all three models, the weight crosses 0.5 at 70% of sampling progress and then decays smoothly. Here, $t_\star$ sets the midpoint of the decay, while β controls its width. We set $t_\star = 0.70$ and $\beta = 0.08$. The resulting smooth handoff is used for every test set and model. Guidance remains 4.5 for MMAudio and HunyuanVideo-Foley XL and 2.5 for Omni2Sound. The fields in Equation 1 are conditional predictions. MMAudio and HunyuanVideo-Foley XL compute one unconditional field v_{u} and share it across all local differences:

$$\widetilde{v}_t^{\mathrm{CFG}} = v_{\mathrm{u}} + \gamma\big(\widetilde{v}_t^{(\rho)} - v_{\mathrm{u}}\big). \tag{14}$$

Omni2Sound uses the same algebra through its released batched CFG path, which repeats the unconditional input for each cell. The control uses the prediction from each temporal cell during a different event. The model, initial noise, temporal cells, masks, solver, and number of model evaluations remain matched. After the first changed step, the latent trajectories diverge as a consequence of that one controlled change.

For the control, MMAudio and HunyuanVideo-Foley XL use the temporal cells in reverse order. Omni2Sound shifts them by one cell:

$$\rho(k) = K + 1 - k \quad \text{or} \quad \rho(k) = 1 + (k \bmod K), \tag{15}$$

The temporal cells stay fixed, but each one affects the audio during another event. Regional Flow and the control differ only in when each cell is used. Their comparison tests whether the event time matters. Comparing Regional Flow with released sampling measures the resulting gain. Isolated generation checks whether the model can produce the sound alone.

E.3 Inference Procedure

Using each event at its own time is the common factor behind the Regional Flow gains. Table 19 groups the complete sequence and paired edit tests. The control uses the visual information from one event during another event but otherwise uses the same computation as Regional Flow. Regional Flow raises interval accuracy, EA, and JS across all three models. PR also rises for MMAudio and HunyuanVideo-Foley XL, but not for Omni2Sound, which uses one shared sound prompt across intervals. EA uses the same 2,048 edits for every sampler. PR and JS eligibility is recomputed from the corresponding output, as stated below the table.

Algorithm 1 Regional Flow inference

Input: noise z_0, full video features h_g, temporal cell features $\{h_k\}_{k=1}^{K}$, masks $\{m_k\}_{k=1}^{K}$, interval map ρ, and solver steps S.

for solver step $s = 0, \ldots, S-1$ **do**
  Evaluate the update from the complete sequence, $v_g \leftarrow v_\theta(z_s, t_s, h_g)$.
  for temporal cell $k = 1, \ldots, K$ **do**
    $v_k \leftarrow v_\theta(z_s, t_s, h_k)$.
    $\Delta v_k \leftarrow v_k - v_g$.
  end for
  $\tilde{v}_s \leftarrow v_g + \alpha_{t_s} \sum_{k=1}^{K} m_{\rho(k)} \odot \Delta v_k$.
  $z_{s+1} \leftarrow \mathrm{SolverStep}(z_s, \tilde{v}_s, t_s)$.
end for
return $\mathrm{Decode}(z_S)$.

Steps 2 to 6 are evaluated as a batch in all three implementations. CFG is applied after the masked differences are added as in Equation 14.

Table 19: Regional Flow improves only when each event is used at its own time. (a) Interval accuracy (%) compares released sampling, Regional Flow, a control that uses each event at another time, and generation from each event alone. (b) Paired edits compare the first three versions.

(a) Interval accuracy on complete sequences

Model	Evaluator	Released	Regional Flow	Wrong event time	Each event alone
MMAudio	CLAP	20.4	**59.5**	3.1	70.6
	BEATs	20.2	**57.7**	3.3	71.1
	PANNs	15.8	**49.5**	2.5	56.2
HunyuanVideo-Foley XL	CLAP	22.7	**31.7**	3.9	47.5
	BEATs	23.1	**30.9**	4.8	47.4
	PANNs	18.1	**24.3**	3.0	31.7
Omni2Sound	CLAP	17.3	**23.2**	7.2	64.6
	BEATs	20.2	**28.3**	7.9	68.8
	PANNs	15.1	**20.7**	5.7	53.0

(b) Paired edits

		Released			Regional Flow			Wrong event time		
Model	Evaluator	EA	PR	JS	EA	PR	JS	EA	PR	JS
MMAudio	CLAP	17.6	50.4	5.4	**63.0**	**82.7**	**44.5**	2.5	36.5	2.1
	BEATs	17.8	52.7	5.1	**67.4**	**84.6**	**48.9**	2.9	25.9	1.0
	PANNs	13.4	57.9	4.8	**54.9**	**83.7**	**40.9**	1.9	35.3	0.7
HunyuanVideo-Foley XL	CLAP	21.2	69.5	11.6	**31.3**	**76.8**	**20.3**	4.1	37.9	3.0
	BEATs	22.4	71.1	12.9	**31.4**	**79.1**	**22.1**	4.0	41.2	0.4
	PANNs	16.6	71.3	10.0	**23.3**	**78.5**	**17.0**	4.2	44.8	0.6
Omni2Sound	CLAP	16.8	**50.9**	4.2	**26.4**	47.2	**13.1**	5.4	22.8	0.4
	BEATs	19.9	**52.1**	5.7	**31.2**	48.2	**14.0**	7.5	18.9	1.1
	PANNs	14.1	**59.5**	5.2	**21.0**	42.9	**8.5**	4.7	27.2	0.7

EA uses the same 2,048 edits in each paired version. PR eligibility is recomputed from the output of each sampler before editing. JS uses pairs with at least one eligible unchanged interval. Regional Flow and the control use the same visual inputs, masks, model calls, and sampling schedule.

E.4 Inference Cost and Boundary Quality

Regional Flow evaluates the full video and K temporal cells at each step. The inputs are evaluated in one batch, so runtime and memory grow with the number of temporal cells. MMAudio and HunyuanVideo-Foley XL share one unconditional CFG prediction across the batch. Omni2Sound evaluates both sides of classifier-free guidance for each cell.

We measure released sampling and Regional Flow on the same hardware and inputs (Table 20). Regional Flow takes 1.52× as long for MMAudio and 1.56× for HunyuanVideo-Foley XL, with little change in peak process memory. Omni2Sound costs more because it expands both sides of classifier-free guidance for every temporal cell and holds the resulting batch in memory. This gives 3.14× the time and 2.57× the peak memory.

We also test whether hard latent masks create an abrupt acoustic change where two intervals meet. The test uses the same complete waveforms as the Regional Flow evaluation. Let b be a shot boundary and let R_b be the RMS in a 100 ms window on either side. The normalized sample jump is

$$J_b = \frac{|x_b - x_{b-1}|}{R_b + 10^{-8}}. \tag{16}$$

For spectral flux, we take the 40 ms samples immediately before and after b, apply a Hann window, and normalize each magnitude spectrum to sum to one. If the two spectra are p_b^- and p_b^+, then

$$F_b = \tfrac{1}{2}\|p_b^+ - p_b^-\|_1. \tag{17}$$

Both measures are lower when the boundary changes less.

Regional Flow returns one waveform. The jump between adjacent samples falls for MMAudio and HunyuanVideo-Foley XL and remains similar for Omni2Sound. Spectral flux rises for all three models. The waveform stays continuous, but its spectrum changes more sharply near some shot boundaries. Table 20 reports the resource and boundary checks together. Current implementations take longer, and the sound can change more sharply near a cut.

Table 20: Regional Flow cost and boundary quality. Runtime and peak memory are measured on matched eight-second outputs on an NVIDIA H20. For the boundary columns, Δ = Regional Flow − released sampling over 768 shot boundaries. Runtime rises by 1.52 to 3.14×. Sample jump falls, while spectral change rises.

	Runtime (s)			Peak memory (GiB)			Boundary change	
Model	Released	Regional Flow	Ratio	Released	Regional Flow	Ratio	Δ sample jump ↓	Δ spectral flux ↓
MMAudio	2.89	4.40	1.52×	11.7	12.2	1.04×	-0.038	+0.061
HunyuanVideo-Foley XL	7.57	11.83	1.56×	14.3	14.3	1.00×	-0.011	+0.029
Omni2Sound	7.91	24.80	3.14×	15.1	38.9	2.57×	-0.009	+0.032

E.5 Results Across Generator Architectures

The three models use the visual information from one event in different ways. MMAudio combines multimodal tokens with joint attention and aligned Synchformer features (Cheng et al., 2025). HunyuanVideo-Foley XL uses joint attention for audio and video and adds synchronization through adaptive normalization and gates (Shan et al., 2025). Omni2Sound separates semantic cross attention from a timing path (Dai et al., 2026). Regional Flow improves CLAP by 39.1, 9.0, and 5.9 points, respectively. It also improves Omni2Sound by 8.2 points under BEATs (Table 19). The size of the gain differs, but all three models improve.

F Learning Sound and Time Alignment

Both MMAudio studies start from the released large 44 kHz v2 checkpoint and update only the synchronization input projection and positional embedding. Training uses bfloat16 and AdamW with betas $(0.9, 0.95)$, learning rate 2×10^{-5}, weight decay 10^{-6}, gradient clipping at 1, 100 warmup steps, and cosine decay. Each of 3,000 updates contains 6 constructed, 4 natural audiovisual, and 4 audio text examples. Video and text inputs are dropped independently with probability 0.1 on replay streams. The two training seeds use the same data and schedule.

F.1 Training with a Temporal Attention Window

We test whether a window of one second is more useful during learning than during inference. Two MMAudio training runs use the same cached examples, seeds, flow matching loss, replay streams, updated parameters, and optimization schedule. Only audio-to-video attention differs.

Each frozen checkpoint is then evaluated with unrestricted and windowed inference, forming a two by two design. Source group resampling keeps both training seeds and every derived edit variant together. Training with the window and removing it at inference gives the highest mean scores. On shared eligible observations, windowed training improves EA, PR, and JS by 2.5, 2.7, and 2.0 points. Keeping the window at inference reduces all three. Run averages and paired effects use different eligible sets. The paired effects use observations eligible under both runs (Table 21). A local view can help the model learn, while broad visual access remains useful when it generates audio.

Whole clip quality moves differently from EA, PR, and JS (Table 21). Windowed training with unrestricted inference improves ImageBind and DeSync but worsens FD-PaSST (Kilgour et al., 2019; Koutini et al., 2022) and LAION-CLAP. Retaining the window at inference produces a different pattern. We therefore report EA, PR, and JS separately from whole clip quality.

Table 21: Local attention helps during training when inference restores the full video. MMAudio checkpoints are compared with EA, PR, JS, and four whole clip measures. Windowed training followed by full video inference gives the highest EA, PR, and JS. Keeping the window at inference removes most of the gain.

Training attention	Inference attention	EA	PR	JS	FD-PaSST ↓	ImageBind ↑	LAION-CLAP ↑	DeSync ↓
Released	Unrestricted	17.6	50.4	5.4	305.43	.365	.347	.380
Unrestricted	Unrestricted	18.2	58.5	8.8	**301.64**	.371	.351	**.331**
Unrestricted	Local window	17.4	56.3	5.8	307.13	.362	**.366**	.375
Local window	Unrestricted	**20.7**	**65.0**	**10.3**	309.32	**.372**	.342	.354
Local window	Local window	18.2	58.7	6.6	307.70	.366	.363	.360

Local attention restricts each audio query to video keys within one second. EA, PR, and JS use CLAP. The released EA, PR, and JS values come from the primary benchmark under the same unrestricted inference path.

F.2 Training with Paired and Shuffled Sounds

The aligned and shuffled study uses 128 examples with four shots from 32 classes. The aligned run places each waveform under its visible source. The shuffled run moves every waveform to a wrong interval within the same example. The two runs share videos, waveforms, durations, boundaries, losses, trainable parameters, optimization, and seeds. Only the relation between sound and time changes. Evaluation on 16 unseen classes tests whether the gain extends beyond the training classes.

Updated parameters. We update MMAudio's synchronization input projection and positional embedding, which connect timing features to audio generation (25.47M parameters, 2.47% of the model). Ablations place rank 16 LoRA adapters on the same projection. One LoRA run also matches the pretrained model on natural examples and within 150 ms of each constructed boundary. All settings train for 3,000 steps with two seeds.

For optimization, let x_1 be an audio latent, x_0 Gaussian noise, and t a sampled time. Training uses $x_t = (1-t)x_0 + tx_1$ and the velocity target $u_t = x_1 - x_0$.

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}\left[\|v_\theta(x_t, t, c) - u_t\|_2^2\right]. \tag{18}$$

Objective and replay. With natural replay, the objective is $\mathcal{L}_{\mathrm{comp}} + \mathcal{L}_{\mathrm{AV}} + \mathcal{L}_{\mathrm{AT}}$ with equal weights. The final LoRA run also matches teacher velocity on natural batches and within 150 ms of each constructed boundary.

Checkpoint selection. We select checkpoints on held-out clips using mean CLAP and BEATs interval accuracy, while requiring ImageBind and LAION-CLAP not to fall below the pretrained model. Final evaluation uses disjoint sources and 16 unseen classes.

Table 5 compares aligned training with the released MMAudio checkpoint. Table 22 gives the matched training control. Correct sound and time pairing improves all three measures under each evaluator, while shuffled targets remain close to the released checkpoint. The two comparisons answer different questions. The released checkpoint measures the benefit of adaptation, whereas the shuffled run tests whether correct sound and time pairing matters when the training data and optimization are held fixed.

Table 22: Aligned targets outperform shuffled targets. MMAudio EA, PR, and JS across two training seeds (%). Both runs use the same sounds and videos. Only their temporal pairing changes.

	Aligned targets			Shuffled targets		
Evaluator	EA	PR	JS	EA	PR	JS
CLAP	25.6	61.5	12.6	16.8	52.0	5.0
BEATs	26.5	58.5	10.9	17.2	49.4	4.3
PANNs	19.4	67.1	11.4	12.7	56.4	5.3

Other architectures. We next ask whether the same training change transfers to other architectures. MMAudio exposes a matched update of the timing path, while Omni2Sound and V-AURA use different training interfaces. Correctly paired Omni2Sound adaptation changes CLAP and BEATs by -0.2 and $+0.2$ points relative to its released model. For V-AURA, training with different rather than repeated sound classes changes the two scores by $+1.6$ and $+0.3$ points. These adaptations show no clear gain. HunyuanVideo-Foley XL is evaluated through the inference intervention supported by its released interface. Table 23 therefore uses MMAudio to compare controlled interval accuracy, reserved clips with natural transitions, and quality over the whole clip.

Table 23: Accuracy and quality after adaptation. Five MMAudio settings on controlled intervals, reserved clips with natural transitions, and four whole clip metrics. Updating the timing projection with replay gives the best controlled scores while meeting the ImageBind and LAION-CLAP selection constraints.

	Controlled (%)		Reserved clips (%)		FD-PaSST	ImageBind	LAION-CLAP	DeSync
Update	CLAP	BEATs	CLAP	BEATs	↓	↑	↑	↓
Pretrained	24.8	24.8	51.0	63.8	305.44	.366	.348	.379
Timing projection, paired only	35.8	33.6	47.3	62.1	328.03	.353	.339	.333
Timing projection + replay	31.8	31.1	49.4	63.4	309.34	.368	.353	.341
LoRA + replay	30.3	29.2	49.3	63.8	309.12	.367	.351	.341
LoRA + teacher match	27.3	25.9	50.6	64.0	306.15	.366	.351	.372

Under the ImageBind and LAION-CLAP selection constraints, updating the timing projection with replay gives the best controlled CLAP and BEATs scores. Stronger teacher matching keeps LoRA closer to the released model on reserved clips but reduces the controlled gain. Better retention comes with a smaller gain on the controlled test.

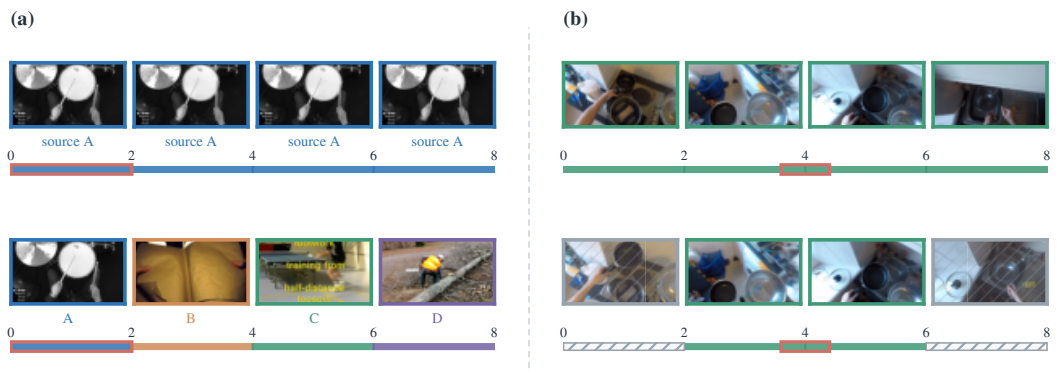

Figure 15: Source clip isolation and continuous video cropping answer different questions. (a) The top row shows the target source clip alone, and the bottom row places the same target beside three other source clips. This retains the target evidence while introducing competing events. (b) The top row shows the original continuous clip, and the bottom row shows a four second crop padded to the original duration. Red boxes mark the scored target interval, and hatched regions show cloned boundary frames.

G Tests on Continuous Video

EPIC-SOUNDS (Huh et al., 2023) extends the analysis from constructed shot sequences to continuous scenes with annotated sounds and event times. Event count and event centered crops first test how scene context affects generation. We then keep the video fixed and change the time at which each sound is generated.

G.1 Data and Statistical Units

We build two sets from different EPIC-SOUNDS splits. On the validation split, media and annotation checks retain 414 eight second clips from 103 videos. Requiring every scored sound to overlap a compatible visible EPIC-KITCHENS action yields the 145 clips from 76 videos used for annotated comparisons. A separate training split supplies 107 clips from 75 videos for predicted pairs after stricter overlap and class filtering. Appendix G.5 gives its selection counts.

Selection uses the original video, waveform, timestamps, labels, and visible actions. A clip may retain one visible action while excluding another annotation whose source is outside the frame. The event count comparison retains 51 matched label and video pairs that occur with one and with multiple events. They come from 41 source videos and 19 participants. Auditory SlowFast (Kazakos et al., 2021), SSAST (Gong et al., 2022), and the generic interval evaluators have different valid subsets, so each result reports its own count.

G.2 Context Controls

Matching sound labels controls identity but not source difficulty: clips with one and multiple events still contain different objects, backgrounds, actions, and overlaps. Table 24 compares label and video pairs observed in both regimes under the two official EPIC-SOUNDS recognizers.

Because source difficulty remains mixed with event count, we also keep each clip fixed and remove its surrounding context. For each visible annotation, we pair the original video of eight seconds with a crop of four seconds centered on the event. Boundary frames pad the crop back to eight seconds. We score both outputs on the same interval and keep the model, prompt, seed, and target fixed. Confidence intervals resample the 76 source videos.

Both controls ask whether less context makes the target sound easier to generate, but they remove context in different ways and use different matched subsets (Table 24).

Table 24: Context controls on continuous video. Interval accuracy (%) for matched clips with one or several events, and for each full clip and a four second crop centered on the event. Effects vary across models, while cropping lowers three MMAudio scores.

	Matched event count				Crop centered on event				
	SlowFast		SSAST		SlowFast	SSAST	CLAP	BEATs	PANNs
Model	One event	Multiple events	One event	Multiple events	Original→crop	Original→crop	Original→crop	Original→crop	Original→crop
MMAudio	38.2	34.0	40.2	37.9	25.0→23.1	26.3→19.4	18.6→13.8	16.8→15.7	13.2→8.6
Omni2Sound	41.2	41.0	46.1	40.3	24.7→25.0	24.5→25.3	19.5→17.3	15.9→15.9	8.6→11.1
HunyuanVideo-Foley XL	26.8	47.2	28.1	47.1	30.6→29.6	30.6→30.6	20.5→16.8	17.8→18.6	12.7→11.9
V-AURA	48.7	49.7	51.0	45.5	28.0→30.9	27.4→30.9	13.2→12.4	9.7→12.2	7.8→6.8

The event count differences change direction across models and evaluators. HunyuanVideo-Foley XL, for example, performs better on clips with several events, so the clips with more events are not always harder.

Cropping has little effect on Omni2Sound and HunyuanVideo-Foley XL, gives V-AURA small gains under the EPIC-SOUNDS recognizers, and reduces MMAudio under SSAST, CLAP, and PANNs. For MMAudio, removing competing events also removes useful scene evidence, leaving a net loss.

The controls isolate different changes (Figure 15). Constructed isolation starts with a recognized target and adds competing source clips. Continuous cropping removes competitors but also removes evidence about the object, action, and scene. Less competition does not always make the sound easier to identify. Because these controls also change the visible context, they do not show why a sound appears at the wrong time. The next test keeps the video fixed and changes only when each sound is generated.

G.3 Annotated Sound and Time Pairs

The annotated test keeps each continuous clip fixed and changes only when each sound is generated. The annotated input uses label ℓ_k during interval I_k. Controls cycle the labels, reverse the interval order, provide only boundaries, or provide an unordered set of labels. All versions share model weights, sampling settings, seed, masks, and number of model calls.

Table 25: Using known sounds at their annotated times. Gains over video only, boundaries only, cycled labels, and reversed times (points), with bars ( ) proportional to the gain.

Evaluator	vs. video only	vs. boundaries	vs. cycled labels	vs. reversed times
Auditory SlowFast	**+4.8**	**+4.3**	**+8.9**	**+7.7**
SSAST	**+7.1**	**+6.9**	**+13.3**	**+10.5**
CLAP	**+5.9**	**+6.7**	**+4.1**	**+7.4**
BEATs	**+6.4**	**+4.4**	**+7.7**	**+5.9**
PANNs	**+3.6**	**+3.3**	**+5.1**	**+5.6**

Using each sound at its annotated time gives MMAudio its highest score under every evaluator. It improves over generation from video alone and boundaries alone. Boundaries or an unordered label set change little, while reversing times or cycling labels removes the gain. The gain appears only when each sound is used at its annotated time.

G.4 Event Boundaries from Silent Video

The annotated study assumes known boundaries, so we next predict them from silent video. Qwen3-VL-8B receives frames and timestamps without event names, EPIC labels, or audio and returns only the start and end of visible acoustic events. Its proposals reach a mean maximum IoU of 0.325 with the annotations, with 53.7% recall at IoU 0.25 and 25.0% at IoU 0.50.

We then compare released generation with Regional Flow using intervals predicted from silent video or supplied by annotations. Both variants use the same direct prompt. Boundaries alone do not supply sound identity, and neither set of intervals improves interval accuracy consistently (Table 26).

Table 26: Event boundaries without sound labels. Interval accuracy (%) for video only and Regional Flow using predicted or annotated boundaries with the same sound prompt throughout. Neither boundary source gives a consistent gain.

Evaluator	Video only	Predicted boundaries	Annotated boundaries
Auditory SlowFast	25.5	26.5	26.0
SSAST	26.0	25.8	26.3
CLAP	18.7	18.7	17.9
BEATs	19.0	20.5	21.0
PANNs	13.1	12.8	13.3

Whole clip metrics also move in different directions under predicted and annotated boundaries. Interval accuracy changes by about two points when the same sound prompt is used in every interval. A boundary says when an event occurs, but not what it should sound like. The next test predicts both pieces.

G.5 Sound and Time Pairs from Silent Video

Data selection. The predicted pair test uses EPIC-SOUNDS training videos that do not appear in the annotated intervention. We form nonoverlapping eight second clips around visible events and exclude unsupported or ambiguous sounds. The split is made by source video. The final set contains 107 clips from 75 videos: 78 contain one event and 29 contain two or more, for 150 annotated events in total.

Predicting label and time pairs. We group 26 EPIC-SOUNDS labels into 13 acoustic classes. Qwen3-VL-8B receives silent frames, timestamps, and this class list, then assigns at most one class to each two-second bin. Adjacent bins with the same class are merged. The 107 evaluation clips produce 191 predicted intervals. No V2A model is adapted on EPIC-SOUNDS.

The official EPIC-SOUNDS recognizers score all 150 annotated intervals. CLAP, BEATs, and PANNs score 149 because one interval has no independent reference for its class.

Table 27: Continuous video baseline. Interval accuracy (%) from video alone for all 21 settings under five evaluators. Later tests compare each added input with the same model and evaluator.

Model and input	SlowFast	SSAST	CLAP	BEATs	PANNs
Flow matching					
MMAudio	39.3	37.3	42.3	30.2	35.6
ControlFoley	34.0	29.3	37.6	35.6	33.6
ThinkSound†	31.3	24.0	35.6	36.2	29.5
PrismAudio†	18.0	20.7	17.4	24.8	33.6
ThinkSound‡	34.0	27.3	35.6	41.6	34.9
PrismAudio‡	33.3	36.0	32.2	34.9	39.6
SelVA	14.7	20.0	27.5	25.5	30.9
Woosh-VFlow	32.7	7.3	9.4	33.6	40.3
Woosh-DVFlow	18.0	16.0	22.1	22.1	32.2
Diffusion					
Omni2Sound	30.7	36.7	41.6	44.3	40.9
HunyuanVideo-Foley XL	43.3	37.3	39.6	42.3	34.2
FoleyCrafter	32.0	30.7	22.1	30.9	30.2
Video-Foley	25.3	29.3	34.2	36.9	36.2
CAFA	18.0	23.3	26.8	24.8	24.8
Diff-Foley	18.7	18.7	23.5	25.5	34.2
AudioX	24.0	23.3	29.5	24.8	33.6
AudioX-Turbo	22.7	24.7	27.5	24.2	31.5
AudioX-MAF	24.0	22.0	30.9	26.2	32.9
AudioX-MAF-MMDiT	26.0	23.3	30.9	28.2	33.6
Autoregressive					
V-AURA	38.7	38.7	36.2	20.8	34.9
SpecVQGAN	13.3	11.3	19.5	26.2	26.2

†Direct prompt. ‡Visual reasoning prompt.

No model leads all five evaluators. We therefore compare paired versions within each model. Each comparison asks whether sound and time information helps the same generator on the same clip.

Each model uses the predicted sound and time through inputs it already supports. For MMAudio, Omni2Sound, and HunyuanVideo-Foley XL, (ℓ_k, I_k) combines video features from interval I_k with a prompt for label ℓ_k. Regional Flow lets these features change the audio only during interval I_k while retaining synchronization features from the full video. V-AURA has no text path, so its timing control restarts generation at the predicted boundaries and joins the local waveforms. Each comparison uses matched clips and the model's original seed policy, with uncertainty clustered by source video.

The subset with several events tests whether a predicted pair remains useful when events compete. Table 28 places this subset beside all scored clips under the official and generic evaluators.

Table 28: Sound and time pairs predicted from silent video. Change in EPIC-SOUNDS interval accuracy over video only (points), for all clips and clips with multiple events.

	All clips: Δ from video only					Multiple events: Δ from video only				
Model	SlowFast	SSAST	CLAP	BEATs	PANNs	SlowFast	SSAST	CLAP	BEATs	PANNs
MMAudio	+6.9	+10.1	+10.9	+19.5	+9.1	+7.2	+5.8	+12.2	+22.9	+12.3
Omni2Sound	+2.4	+5.2	+4.4	+1.0	-1.2	+12.0	+15.2	-0.1	-12.1	-2.6
HunyuanVideo-Foley XL	-1.0	+2.9	+9.7	+2.7	+3.4	-2.6	+2.3	+12.8	-2.4	+7.0
V-AURA†	-3.9	-2.3	-3.6	+14.2	-0.5	-3.9	-4.1	-12.1	+3.5	+4.2

SlowFast and SSAST are the official EPIC-SOUNDS recognizers. CLAP, BEATs, and PANNs use prototypes built from other source groups. †V-AURA has no text input. Its timing interface restarts generation at predicted boundaries and joins the resulting waveforms.

MMAudio improves under all three general evaluators and remains positive on clips with several events. HunyuanVideo-Foley XL improves under CLAP, while Omni2Sound and V-AURA change direction across evaluators. V-AURA can restart at a predicted time, but it has no text path for the predicted sound. These results depend on both the pair prediction and how each generator uses the predicted sound and time.

G.6 Audio Quality

Table 29 compares interval accuracy and whole clip quality on the same outputs. It uses the same four quality measures for predicted and annotated pairs, so their tradeoffs can be read in one place.

Predicted pairs improve several whole clip measures for MMAudio and HunyuanVideo-Foley XL. Omni2Sound improves quality without a stable interval gain, while V-AURA's restart worsens synchronization and CLAP. Thus interval accuracy and overall quality can move separately. Annotated pairs test how the generator uses a known sound and time. Predicted pairs also test whether both can be recovered from silent video.

Table 29: Whole clip quality after adding sound and time information. Each cell shows video only → generation with predicted or annotated sound and time pairs on the same continuous video.

Sound and time source	Model	FD-PaSST ↓	ImageBind ↑	DeSync ↓	LAION-CLAP ↑
Predicted	MMAudio	435.1→407.5	.321→.336	.197→.196	.178→.198
	Omni2Sound	424.8→407.0	.331→.346	.335→.227	.195→.194
	HunyuanVideo-Foley XL	673.0→543.5	.277→.318	.365→.221	.098→.145
	V-AURA†	879.9→869.0	.268→.270	.892→.930	.142→.129
Annotated	MMAudio	635.3→646.8	.340→.350	.275→.206	.175→.172

Arrows in the headers give the preferred direction. † V-AURA has no text input. Its timing interface restarts generation at predicted boundaries and joins the resulting waveforms.

H Statistics and Reproducibility

H.1 Statistical Units

Table 30: Statistical units and uncertainty. Bootstraps preserve source dependence.

Test	Unit (n)	Comparison	Uncertainty
Controlled benchmark	128–256 groups	main / edit / sequence controls	group or paired bootstrap (5k–10k)
Window Stitch	256 groups / 1,536 joins	hard / 20 / 50 ms	5k group bootstrap
Model analysis	16–256 groups / 120 judgments	readouts / metric audit	bootstrap / aggregation / decisive rate
Regional Flow	8–256 groups / 768 joins	baseline / time / boundary	5k–10k paired bootstrap
Training	16–256 groups × 2 seeds	aligned / shuffled / pretrained	10k paired bootstrap
Continuous video	51–76 videos	controls / predicted / annotated	video bootstrap

Each experiment keeps all edits, intervals, seeds, and derived versions from the same source together. This prevents repeated views of one source from being counted as independent evidence. Table 30 groups tests that use the same unit. The corresponding result tables report the exact valid counts.

H.2 Reproducibility

The release records inputs, splits, prompts, seeds, model versions, sampling settings, valid counts, predictions, and bootstrap configurations. Reference and generated audio remain separate, and prototypes use only reference audio. Releases include their controls and source video clusters.

I Discussion

LocalSoundBench studies visible Foley events with identifiable sound regions. The main sequences use hard cuts for controlled edits, while EPIC-SOUNDS repeats the test on continuous video. Dialogue, ambience, music, overlapping events, and sound bridges remain open because they may span several events or lack a clear onset. Window Stitch suits independent shots with known cuts and constructs the result from separately sampled segments. Regional Flow instead changes sound placement inside full video generation, but it still needs usable event regions. The two methods apply in different settings. These results do not show that either is always better. Future systems could infer event regions from video and choose between the two strategies. Longer videos will test whether this remains reliable when sounds overlap.