AVIO: LEARNING TO ADD AND REMOVE SOUNDING OBJECTS IN AUDIOVISUAL SCENES

Anonymous authors
Paper under double-blind review

ABSTRACT

Adding or removing a sounding object requires coordinated changes to visual content and sound while preserving the surrounding scene. Yet paired supervision for localized non-speech audiovisual editing remains limited, as visual and acoustic edits must target the same object and isolate its sound from overlapping sources. To address this gap, we introduce *AVIOBench*, a dataset comprising 37.9 hours of paired audiovisual examples spanning 1,878 target-object names. AVIOBench links the visual presence and acoustic contribution of each target object through a shared identity and visual mask. Our automated pipeline uses visual grounding and cross-modal consistency to select target-sound removal candidates, then jointly refines the audiovisual pairs to improve perceptual quality and cross-modal consistency. Building on this dataset, we propose *AVIO*, which adapts a pretrained text-to-audiovisual generation model through source-conditioned feature modulation to jointly learn object addition and removal. A reference-frame curriculum gradually reduces reference conditioning during training, enabling one model to perform instruction-only editing with optional visual guidance. Quantitative and qualitative evaluations demonstrate effective audiovisual object removal and addition, with optional reference guidance providing appearance and placement control for addition. Video examples are available on our project page[1].

1 INTRODUCTION

Adding or removing a sounding object requires editing both its visual presence and its associated sound. As shown in Fig. 1, removing a donkey should also eliminate its braying, while adding a passing vehicle should introduce sound consistent with its motion and timing. Such joint audiovisual editing supports acoustically coherent film post-production and content creation (Fu et al., 2025), and could complement robotic and driving simulators (Chen et al., 2022; Dosovitskiy et al., 2017) by jointly varying objects and sounds to test audiovisual perception robustness. However, unimodal editors cannot jointly edit visual objects and their associated sounds, while audiovisual editors and benchmarks lack a focus on sounding objects, partly due to limited paired supervision (Lin et al., 2025; Fu et al., 2025; Zheng et al., 2026a;b; Chen et al., 2026; Wen et al., 2026; Miao et al., 2026).

Constructing paired supervision for this setting poses two challenges: isolating non-speech sounds from complex mixtures and coordinating object-level changes across audio and video while preserving unrelated content. Environmental sounds exhibit diverse spectral and temporal patterns (Chu et al., 2009) and often overlap with other sources, making the target sound difficult to isolate without altering unrelated sounds. Beyond acoustic separation, the visual and acoustic edits must refer to the same object and consistently change its presence in both modalities.

To address this gap, we introduce *AVIOBench*, comprising approximately **37.9 hours**[2] of paired audiovisual examples constructed from AudioSetCaps and VGGSound (Bai et al., 2024; Chen et al., 2020), covering diverse object-associated sound events beyond speech. In order to remove the same sounding object across both modalities while preserving unrelated content, our pipeline first identifies the target object and uses it to guide both visual removal and audio removal. To address the difficulty of removing sound, we generate separation candidates using varied text prompts, visual-mask prompts,

[1] https://goatlazy.github.io/ICLR2027/

[2] The duration of each source–target pair is counted once.

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.

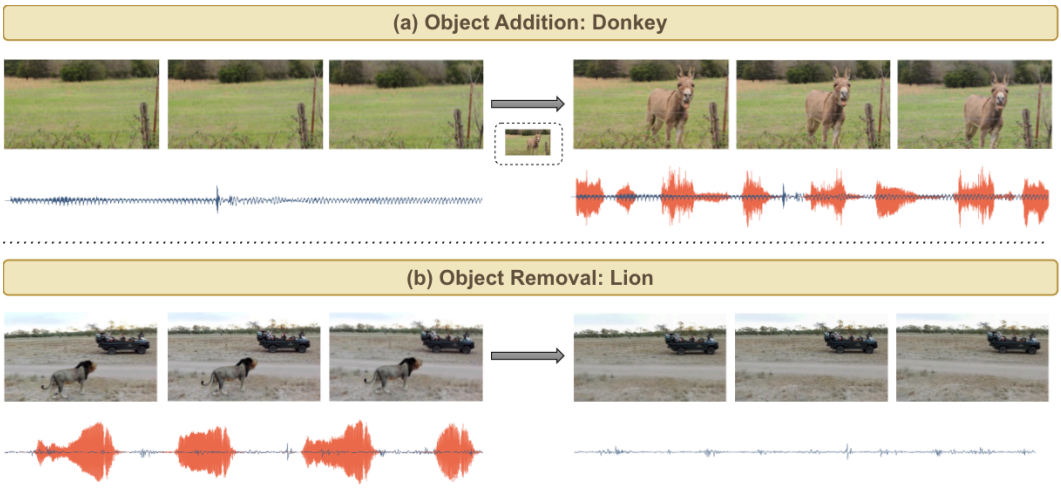


Figure 1: **Audiovisual event addition and removal with AVIO.** (a) A donkey and its braying are added to the scene with an optional reference frame to specify initial placement and appearance (b) A lion and its roaring are removed while the vehicle and its sound are preserved.

and random seeds, then select the candidate with the strongest audiovisual correspondence to the target object. In order to further improve consistency between the resulting visual and acoustic edits, we apply joint audiovisual refinement on the edited pairs.

Building on AVIOBench, we introduce *AVIO*, an object-level audiovisual editor that adapts a pre-trained text-to-audiovisual generation model, drawing on its learned modality representations and its temporally aligned joint audio–video architecture. To enable source-consistent editing, we introduce source-conditioned feature modulation that aligns source features with target representations in time and modality. While text instructions specify the desired edit, they may leave the appearance and initial placement of the added object ambiguous. To reduce this ambiguity, AVIO optionally uses an edited reference frame to guide appearance and initial placement. A reference-frame curriculum progressively reduces reference availability during training, enabling editing with text alone or text together with a reference frame. Through quantitative evaluation and qualitative comparisons, we demonstrate the effectiveness of AVIO in audiovisual object removal and addition, with optional reference guidance providing appearance and placement control for addition.

Our contributions are summarized as follows:

- We introduce AVIOBench, a dataset of **37.9 hours** of paired audiovisual examples for object addition and removal involving non-speech sound events.
- We adapt a pretrained text-to-audiovisual Diffusion Transformer (DiT) into a unified object editor jointly trained for addition and removal. A reference-frame curriculum supports both text-only editing and optional visual guidance specifying the appearance and initial placement of the target object.

We will publicly release AVIOBench, our code, and a LangGraph-based data engine to support reproducibility and further dataset expansion, subject to applicable licenses upon acceptance.

2 Related Work

Object Editing Datasets. Existing editing datasets differ in whether they supervise changes within a single modality or coordinate edits across audio and video. Visual datasets such as OpenVE-3M and Goku (He et al., 2026; Liang et al., 2026b), and audio resources such as AUDIT (Wang et al., 2023), provide single-modality supervision. Audiovisual resources extend this setting to object editing (Lin et al., 2025; Fu et al., 2025), paired instruction-guided editing (Chen et al., 2026; Zheng et al., 2026b), and broader cross-modal evaluation (Wen et al., 2026; Miao et al.,

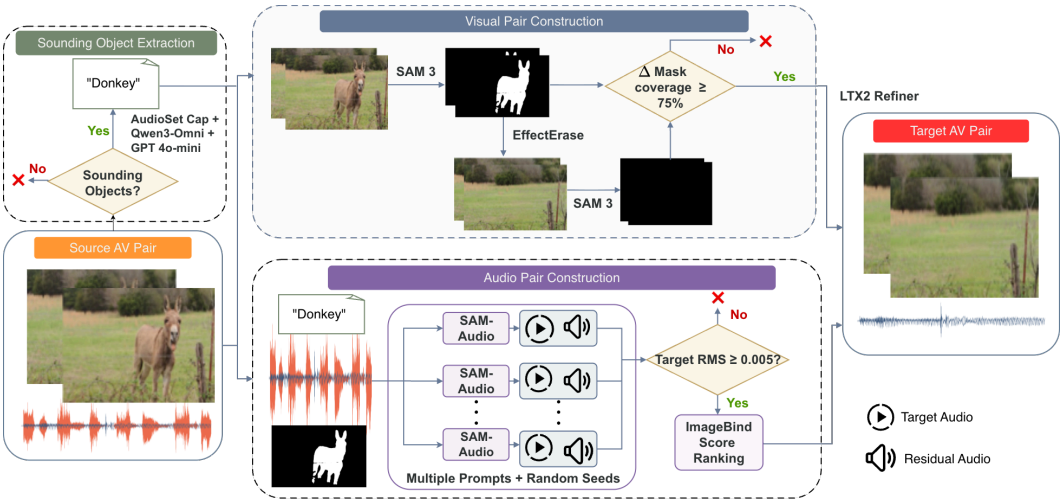


Figure 2: **Overview of the AVIOBench data construction pipeline.** For each clip, we identify the primary sounding object O, remove it visually via segmentation and inpainting, and remove its sound via audio separation. Each stage applies automatic filters (yellow) to discard unreliable candidates, and the edited streams are jointly refined to form $\mathbf{X}_{\setminus O}$.

2026). AVIOBench focuses on non-speech sounding objects, providing aligned object-present and object-absent audiovisual pairs that support both addition and removal. A detailed dataset comparison can be found in Appendix B.

Audiovisual Joint Generation. Earlier audiovisual methods generate one modality conditioned on the other, while recent work shifts toward joint generation. MM-Diffusion (Ruan et al., 2023) couples audio and video diffusion networks, while JavisDiT and JavisDiT++ (Liu et al., 2025; 2026) use Transformers to model cross-modal interactions. Extending generation to editing requires coordinated changes to a target object and its sound while preserving unrelated source content. To enable such editing, we adapt a pretrained audiovisual generator through token-aligned source modulation and paired editing supervision to jointly learn object addition and removal.

Audiovisual Joint Editing. Existing audiovisual editing methods coordinate visual and acoustic changes through sequential cross-modal guidance or joint audiovisual modeling. AvED (Lin et al., 2025) and Object-AVEdit (Fu et al., 2025) use delta denoising and inversion-based regeneration, respectively. CoherentAVEdit (Ishii et al., 2026) generates audio after video editing, whereas AVI-Edit (Zheng et al., 2026a) uses audio to guide visual edits. SpongeBob (Liang et al., 2026a) couples modalities during denoising, while JAVEdit (Chen et al., 2026) and InstructAV2AV (Zheng et al., 2026b) adapt audiovisual generators using paired editing data. We focus on non-speech object addition and removal, learning bidirectionally from object-present and object-absent pairs with a reference-frame curriculum supporting text-only editing and optional visual guidance for addition.

3 AVIOBench Dataset

We introduce AVIOBench, a benchmark for joint audiovisual editing of object-associated sound events, with paired examples for object removal and addition. We describe the dataset construction pipeline shown in Fig. 2 in Section 3.1 and summarize the dataset size and annotations in Section 3.2.

3.1 Data Construction Pipeline

Data Collection and Filtering. We collect source videos from AudioSet (Gemmeke et al., 2017) and VGGSound (Chen et al., 2020), two large-scale datasets of 10-second YouTube clips captured in diverse real-world settings. These datasets span human activities, animals, transportation, household sounds, musical performances, and natural environments. For computational efficiency, we retain the

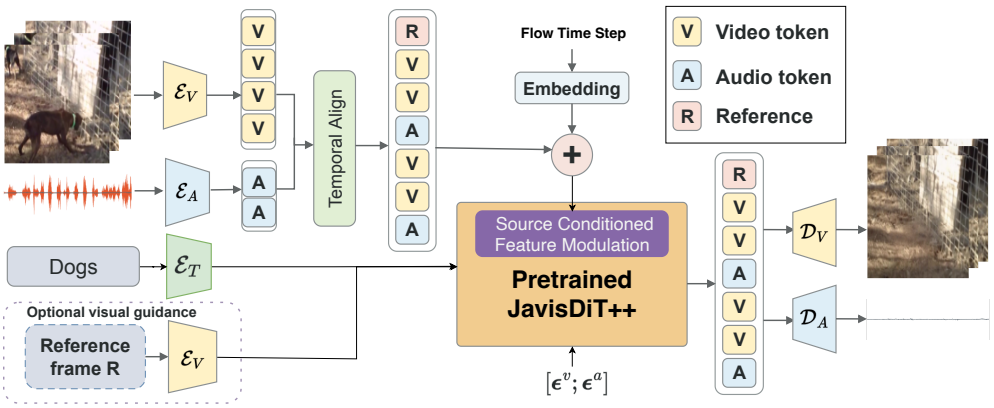


Figure 3: **Overview of AVIO.**We adapt a pretrained text-to-audiovisual DiT by conditioning it on source audiovisual latents. To support addition and removal within a unified model, we introduce a reference-guided curriculum that progressively increases the dropout probability of the first target frame during training.

first five seconds of each clip, processing 159 hours of video in total. To focus on object-associated sound events, we use FireRedASR2S (Xu et al., 2026) to exclude clips containing only speech.

Target-Object Identification. To identify a primary sounding object in each clip $\mathbf{X} = (V, A)$, we first obtain an audiovisual caption. We use the caption from AudioSetCaps (Bai et al., 2024) when available and otherwise generate one using Qwen3-Omni (Xu et al., 2025). Given this caption, GPT-4o mini (OpenAI, 2024) identifies the primary sounding object O, which serves as the prompt for SAM 3 (Carion et al., 2026) to extract its video mask M_O.

Audio Pair Construction. We construct an audio pair from the original recording and a residual track with the target sound removed. Specifically, we run SAM-AUDIO (Shi et al., 2025) with the object name and visual mask as prompts across multiple random seeds, jointly generating target and residual stems. We discard candidates whose target stem has a root-mean-square (RMS) amplitude below 0.005 to exclude near-silent extractions. Among the remaining candidates, we select the target stem a_O with the highest ImageBind (Girdhar et al., 2023) similarity to the mask-conditioned video and use its corresponding residual as the object-absent audio.

Visual Pair Construction. We construct the corresponding visual pair by removing the target object through video inpainting. We apply EFFECTERASE (Fu et al., 2026) to the region specified by M_O, producing the edited video $V_{\setminus O}$. To assess whether the object remains detectable, we rerun SAM 3 (Carion et al., 2026) with the same object prompt. We retain only clips for which the mean detected mask area decreases by at least 75% relative to the original video.

Audiovisual Pair Construction. We combine the edited streams into $\mathbf{X}_{\setminus O} = (V_{\setminus O}, A_{\setminus O})$. Following Ditto (Bai et al., 2025), we apply a lightweight SDEdit refinement with an empty prompt, using LTX-2 (HaCohen et al., 2026) to jointly refine both modalities and reduce editing artifacts. We use $\mathbf{X}_{\setminus O}$ to denote the resulting refined clip and construct removal and addition examples by exchanging the source and target:

$$(\mathbf{X}_{\mathrm{s}}, \mathbf{X}_{\mathrm{e}}) = \begin{cases} (\mathbf{X}, \mathbf{X}_{\setminus O}), & \text{removal}, \\ (\mathbf{X}_{\setminus O}, \mathbf{X}), & \text{addition}. \end{cases} \tag{1}$$

3.2 DATASET STATISTICS

AVIOBench contains 37.9 hours of unique paired examples with object-level text annotations and masks. Each example is annotated with an object name (1,878 unique names in total) and a mask of the target object. We provide additional dataset statistics and retention rates at each pipeline stage and the details of our LangGraph-based data engine in Appendix F.

4 AVIO MODEL

We introduce AVIO, which adapts a pretrained text-to-audiovisual generation model for joint object addition and removal. We retain its text-conditioning branch and add source-conditioned feature modulation to enable instruction-guided editing (Section 4.1). We then present a reference-frame curriculum for optional visual guidance (Section 4.2), followed by training and inference (Section 4.3).

Problem formulation. Let $\mathcal{M} = \{v, a\}$ denote the video and audio modalities. Given a source audiovisual pair $\mathbf{X}_{\mathrm{s}} = (\mathbf{X}_{\mathrm{s}}^{v}, \mathbf{X}_{\mathrm{s}}^{a})$ and editing conditions $\mathbf{c} = \{I, R\}$, where I is a textual instruction and R is an optional edited reference frame, our goal is to predict the edited target $\mathbf{X}_{\mathrm{e}} = (\mathbf{X}_{\mathrm{e}}^{v}, \mathbf{X}_{\mathrm{e}}^{a})$.

4.1 INSTRUCTION-GUIDED AUDIOVISUAL EDITING

Pretrained Backbone and text conditioning. In order to adapt pretrained audiovisual generation to object-level editing, we build on the learned modality representations and temporally aligned joint audio–video architecture of JavisDiT++ (Liu et al., 2026). We retain its text-conditioning branch to guide the desired edit. The pretrained umT5-XXL encoder $\mathcal{E}_T$ (Chung et al., 2023) maps the instruction I to features $\mathcal{E}_T(I)$, which are supplied to the text cross-attention layers of the DiT blocks.

Source encoding and alignment. To preserve source content unrelated to the requested edit, AVIO aligns source features with target tokens and uses these features to guide editing through feature-wise linear modulation (FiLM) (Perez et al., 2018). As shown in Fig. 3, we encode and tokenize the source and target following JavisDiT++ (Liu et al., 2026) to maintain their spatial and temporal alignment. Specifically, let $\mathcal{E}_V$ and $\mathcal{E}_A$ denote the frozen Wan2.1 video VAE encoder (Wan et al., 2025) and AudioLDM2 audio VAE encoder (Liu et al., 2024), respectively, with audio preprocessing included in $\mathcal{E}_A$. For modality $m \in \{v, a\}$, denoting video and audio, we encode the source input $\mathbf{X}_{\mathrm{s}}^{m}$ into a latent tensor $\mathbf{z}_{\mathrm{s}}^{m}$. We then apply learnable patch embeddings ψ_v and ψ_a to the video and audio latents, respectively, and concatenate the resulting token sequences into the source representation $\mathbf{s}$:

$$\begin{aligned} \mathbf{z}_{\mathrm{s}}^{m} &= \mathcal{E}_m(\mathbf{X}_{\mathrm{s}}^{m}), \qquad m \in \{v, a\}, \\ \mathbf{s} &= \left[\psi_v(\mathbf{z}_{\mathrm{s}}^{v}); \psi_a(\mathbf{z}_{\mathrm{s}}^{a})\right], \end{aligned} \tag{2}$$

where $[\cdot;\cdot]$ denotes concatenation along the token dimension. For cross-modal temporal alignment, we retain temporal-aligned rotary positional embeddings (TA-RoPE) in the backbone's joint self-attention, assigning a shared temporal position index to audio and video tokens corresponding to the same temporal interval.

Source-conditioned feature modulation. To promote source–target alignment, we modulate hidden states of the intermediate flow sample $\mathbf{z}_\tau$ with source features through adaptive layer normalization (AdaLN) (Peebles & Xie, 2023), following FiLM (Perez et al., 2018). For hidden state $\mathbf{x}_i$ at position i of the joint audiovisual sequence, we combine the corresponding source token $\mathbf{s}_i$ (Eq. 2) with the flow-time embedding $\mathbf{e}_\tau$:

$$\mathbf{h}_i = \mathbf{e}_\tau + \mathbf{s}_i. \tag{3}$$

Each self-attention (sa), text cross-attention (ca), and modality-specific feed-forward (ff) sublayer k applies

$$\begin{aligned} \left(\boldsymbol{\beta}_i^k, \boldsymbol{\gamma}_i^k, \boldsymbol{\alpha}_i^k\right) &= \mathcal{M}^k(\mathbf{h}_i) + \boldsymbol{\delta}^k, \\ \widetilde{\mathbf{x}}_i &= (\mathbf{1} + \boldsymbol{\gamma}_i^k) \odot \mathrm{LN}(\mathbf{x}_i) + \boldsymbol{\beta}_i^k, \\ \mathbf{x}_i^{\mathrm{out}} &= \mathbf{x}_i + \boldsymbol{\alpha}_i^k \odot [\mathcal{F}^k(\widetilde{\mathbf{X}})]_i, \end{aligned} \tag{4}$$

where $\mathcal{M}^k$ is a SiLU–linear head predicting shift $\boldsymbol{\beta}_i^k$, scale adjustment $\boldsymbol{\gamma}_i^k$, and residual gate $\boldsymbol{\alpha}_i^k$. We reuse the pretrained heads and learned per-block offsets $\boldsymbol{\delta}^k$ for self-attention and feed-forward sublayers, with modality-specific feed-forward heads. For the backbone's unmodulated text cross-attention, we add a head $\mathcal{M}^{\mathrm{ca}}$ shared across blocks with $\boldsymbol{\delta}^{\mathrm{ca}} = \mathbf{0}$. $\mathcal{F}^k$ operates on the modulated sequence $\widetilde{\mathbf{X}} = [\widetilde{\mathbf{x}}_1, \ldots, \widetilde{\mathbf{x}}_L]$, with text cross-attention using the encoded instruction $\mathcal{E}_T(I)$ as keys and values. LN denotes layer normalization, $\odot$ element-wise multiplication, and $[\cdot]_i$ the output at position i. Block indices and hidden-state sublayer indices are omitted for clarity.

4.2 Reference-Guided Curriculum

In order to support reference-guided addition and appearance-controlled editing alongside removal without a reference frame, we adopt a reference-guided curriculum inspired by Ditto (Bai et al., 2025). The curriculum initially provides visual guidance and gradually reduces its availability, allowing a single model to learn editing under both conditioning regimes. We validate this design in our curriculum ablation in Appendix A.1. To specify the desired object's appearance and initial placement, we provide an optional edited reference frame R. To incorporate this visual guidance through the existing source pathway, we encode R into a single latent frame $\mathbf{z}_R = \mathcal{E}_V(R)$ and prepend it to the source video latent:

$$\tilde{\mathbf{z}}_{\mathrm{s}}^{v} = [\mathbf{z}_R; \mathbf{z}_{\mathrm{s}}^{v}], \tag{5}$$

where concatenation is along the temporal dimension. In order to preserve source–target token correspondence, we allocate a matching temporal prefix slot in the target video latent and discard its output at inference as shown in Fig. 3. During training, we use the first frame of the target video $\mathbf{X}_{\mathrm{e}}^{v}$ as R to provide visual guidance consistent with the target edit. To progressively expose the model to editing without a reference frame, we retain R with probability

$$p_{\mathrm{ref}}(g) = \max\left(0,\, 1 - \frac{g}{S}\right), \tag{6}$$

where g denotes the training step and S is the number of steps over which reference conditioning is phased out. Specifically, we set S to be 100K. When R is dropped, we omit both the source reference prefix and the matching target prefix slot to maintain their correspondence.

4.3 Training and Inference

We jointly train AVIO for object addition and removal using each pair in both directions at a 1:1 ratio. Following the flow-matching framework of JavisDiT++ (Liu et al., 2026), the model predicts video and audio velocities from noisy target latents whose features are modulated by the source, with additional conditioning on the shared flow time τ, text instruction, and optional reference frame. The training objective is

$$\mathcal{L}_{\mathrm{AV}} = \mathbb{E}\left[\sum_{m\in\{v,a\}} \frac{1}{|\mathbf{u}^m|}\, \|\widehat{\mathbf{u}}^m - \mathbf{u}^m\|_2^2\right], \tag{7}$$

where $\mathbf{u}^m = \mathbf{z}_{\mathrm{e}}^m - \boldsymbol{\epsilon}^m$ and $\widehat{\mathbf{u}}^m$ denote the target and predicted velocities for modality m. Here, $\mathbf{z}_{\mathrm{e}}^m$ is the edited target latent, and $\boldsymbol{\epsilon}^m \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is independently sampled Gaussian noise. Normalization by the latent element count $|\mathbf{u}^m|$ yields a mean squared error per modality, and the two losses are summed with equal weight. At inference, we jointly generate video and audio latents from independent Gaussian noise using 50 Euler steps. Removal uses the source pair and text instruction, while addition optionally uses an edited reference frame for appearance and initial placement control. We discard the reference slot when present and decode the output latents with $\mathcal{D}_V$ and $\mathcal{D}_A$. Details on audio–video loss balancing, the flow convention, conditioning dropout, the identity-reconstruction objective, and sampling are given in Appendix E.2.

5 Experiments

5.1 Datasets and Implementation Details

We train on 24,444 unique pairs, validate on 1,389 unique pairs, and test on 1,425 pairs from AVIOBench. Each training pair is used in both directions by swapping the source and target, yielding 48,888 directed examples for object addition and removal. Text conditions follow the templates "a video without {object}" for removal and "a video with {object}" for addition. Detailed training and fine-tuning configurations are provided in Appendix E, with ablations of the reference-guided curriculum and source audiovisual conditioning in Appendix A.

5.2 Baselines

Unimodal Baselines. We compare with specialized audio and video editors to assess editing quality within each modality. For audio, **ZEUS** (Manor & Michaeli, 2024) performs zero-shot text-guided



Figure 4: **Qualitative comparison of audiovisual object removal (left) and addition (right).** Each row shows video frames and audio waveforms for the corresponding model (left). † CoherentAVEdit uses oracle target video; ‡ AVI-Edit uses oracle target audio. Red and Green highlight removal targets and added content, respectively. AVIO successfully edits objects and their sounds while preserving the surrounding scene and matching the appearance and placement of the reference object for addition.

editing through diffusion inversion, while **AUDIT** (Wang et al., 2023) uses an instruction-guided latent diffusion model. For video, **LGVI** (Wu et al., 2024) performs language-driven video inpainting. **VideoPainter** (Bian et al., 2025) uses a dual-branch video inpainting framework. **ROSE** (Miao et al., 2025) removes objects and associated visual effects, including shadows and reflections.

Audiovisual Editing Baselines. We compare against joint audio–video editors and cross-modal editors that use one modality to guide editing in the other. **InstructAV2AV** (Zheng et al., 2026b) performs instruction-guided joint audiovisual editing. We evaluate this model using separate task-specific checkpoints for addition and removal. **AVI-Edit** (Zheng et al., 2026a) requires an edited audio track to guide video editing with instance masks. We provide oracle target audio as this conditioning input. **CoherentAVEdit** (Ishii et al., 2026) requires an edited video to generate the corresponding audio, conditioned additionally on source audio and text. We provide the oracle target video as this conditioning input.

5.3 Evaluation Metrics

We evaluate audiovisual object removal and addition on our test set under a standardized protocol measuring visual quality, audio quality, and audiovisual consistency. Video metrics include PSNR (Horé & Ziou, 2010) for pixel-level fidelity, SSIM (Horé & Ziou, 2010; Wang et al., 2004) for structural similarity, LPIPS (Zhang et al., 2018) for perceptual distance, and FVD (Unterthiner et al., 2018) for video-distribution similarity. For audio, LSD measures spectral reconstruction error,

FAD (Kilgour et al., 2019) measures feature-distribution distance, Inception Score (IS) (Salimans et al., 2016) reflects quality and diversity, and KL divergence measures differences between output and target sound-class predictions. We use Gemini-3.1-Pro (Google DeepMind, 2026) to assess Visual Edit Success (VES), Audio Edit Success (AES), and Audio-Visual Consistency (AVC), reporting the percentage of clips satisfying each criterion. DeSyncScore measures temporal audiovisual misalignment using Synchformer (Iashin et al., 2024). Detailed protocols appear in Appendix C.

6 RESULTS

6.1 QUALITATIVE EXAMPLES

Fig. 4 compares audiovisual object removal and addition. CoherentAVEdit receives oracle target video, while AVI-Edit receives oracle target audio. These supplied modalities are shown alongside each method's generated outputs. For removal, AVIO removes the dog with fewer visible remnants than InstructAV2AV (Zheng et al., 2026b) and AVI-Edit (Zheng et al., 2026a), while suppressing its associated sound and preserving the surrounding scene. CoherentAVEdit (Ishii et al., 2026) attenuates the target sound but leaves audible remnants. For addition, AVIO generates a helicopter and its associated sound while preserving the surrounding scene and matching the reference object's appearance and placement. InstructAV2AV and AVI-Edit generate helicopters whose appearances differ from the reference, and AVI-Edit also introduces visible artifacts. The helicopter sound generated by CoherentAVEdit is less distinct. Video examples are available on our project page.[3]

6.2 QUANTITATIVE RESULTS

Visual Editing Quality. We compare AVIO with video-only editors and audiovisual editing baselines against pseudo-ground-truth targets using PSNR (Horé & Ziou, 2010), SSIM (Horé & Ziou, 2010; Wang et al., 2004), LPIPS (Zhang et al., 2018), and FVD (Unterthiner et al., 2018) in Table 1(a). For removal, AVIO outperforms LGVI (Wu et al., 2024) across all four metrics and VideoPainter (Bian et al., 2025) in PSNR, LPIPS, and FVD. AVIO achieves a PSNR of 20.06 dB, within 0.24 dB of the strongest video-only baseline, ROSE (Miao et al., 2025). Compared with the audiovisual editing baselines InstructAV2AV (Zheng et al., 2026b) and AVI-Edit (Zheng et al., 2026a), AVIO achieves the best removal PSNR and LPIPS. For addition, all evaluated video-generating methods receive the same edited first frame. AVIO achieves the best PSNR and LPIPS, exceeding the next-best PSNR by 1.30 dB, while ranking second in SSIM. Although AVIO does not lead in SSIM or FVD, reflecting remaining gaps in structural similarity and video feature distribution matching, AVIO consistently achieves the strongest pixel-level reconstruction accuracy and frame-level perceptual similarity to the pseudo-ground-truth targets among the evaluated audiovisual editing baselines across both tasks.

Audio Editing Quality. We compare AVIO with audio-only editors and audiovisual editing baselines using FAD (Kilgour et al., 2019), IS (Salimans et al., 2016), KL divergence, and LSD. As shown in Table 1(b), for the removal task, AVIO outperforms the audio-only editors ZEUS (Manor & Michaeli, 2024) and AUDIT (Wang et al., 2023) in KL divergence and LSD, and also surpasses AUDIT in FAD and IS. Compared with the audiovisual editing baselines InstructAV2AV (Zheng et al., 2026b) and CoherentAVEdit (Ishii et al., 2026), AVIO achieves the best KL divergence (1.36) and LSD (12.81) for the removal task. For the reference-guided addition task, AVIO outperforms InstructAV2AV across all four metrics and achieves the best KL divergence (1.29) and LSD (11.85), alongside the second-best FAD and IS. AVIO achieves an LSD of 11.85 compared with 15.70 for CoherentAVEdit, despite CoherentAVEdit receiving oracle target video when generating audio. Although AVIO does not lead in FAD or IS, suggesting room to improve distribution-level audio quality, AVIO achieves the closest agreement with pseudo-ground-truth targets in audio-event predictions and spectral reconstruction among all evaluated methods across both tasks.

Joint Audiovisual Evaluation. We compare AVIO with oracle-paired editing methods and the joint audiovisual editor InstructAV2AV (Zheng et al., 2026b). As shown in Table 1(c), AVIO achieves the highest VES and AES among the evaluated editing methods across both tasks. AVIO outperforms AVI-Edit (Zheng et al., 2026a) in VES and CoherentAVEdit (Ishii et al., 2026) in AES, while jointly generating both modalities without oracle target audio or full target video. For the removal task, AVIO achieves VES and AES of 32.70% and 11.30%, outperforming InstructAV2AV with comparable AVC

[3] https://goatlazy.github.io/ICLR2027/

Table 1: **Visual, audio, and audiovisual editing evaluation.** Reference-based metrics use pseudo-ground-truth targets. VES, AES, and AVC are Gemini 3.1 Pro Yes rates (%). Best and second-best results per task are bold and underlined, using unrounded scores and excluding Oracle / Oracle.

(a) Visual Editing

Task	Input	Model	PSNR↑	SSIM↑	LPIPS↓	FVD↓
Removal	Text	LGVI (2024)	16.22	0.55	0.40	402.69
		VideoPainter (2025)	18.43	0.68	0.30	362.35
		ROSE (2025)	**20.30**	**0.72**	**0.24**	**176.05**
		InstructAV2AV (2026b)	18.88	<u>0.68</u>	0.29	<u>180.10</u>
		AVI-Edit (2026a)	16.65	0.64	0.32	580.48
		Ours	<u>20.06</u>	0.66	<u>0.28</u>	190.50
Addition	Text + ref.*	InstructAV2AV	17.01	0.61	0.29	**202.45**
		AVI-Edit	<u>17.92</u>	**0.66**	<u>0.24</u>	<u>293.49</u>
		Ours	**19.22**	<u>0.63</u>	**0.23**	316.72

(b) Audio Editing

Task	Input	Model	FAD↓	IS↑	KL Div.↓	LSD↓
Removal	Text	ZEUS (2024)	1.66	**8.44**	1.47	<u>14.11</u>
		AUDIT (2023)	11.23	1.44	3.22	16.60
		CoherentAVEdit (2026)	<u>1.36</u>	<u>8.10</u>	1.83	15.03
		InstructAV2AV	**0.89**	6.91	<u>1.44</u>	14.71
		Ours	1.97	6.48	**1.36**	**12.81**
Addition	Text + ref.†	CoherentAVEdit	**5.21**	**8.41**	<u>1.40</u>	<u>15.70</u>
		InstructAV2AV	8.82	6.09	2.30	17.45
		Ours	<u>5.77</u>	<u>6.94</u>	**1.29**	**11.85**

(c) Audiovisual Editing and Synchronization

Model	Params (B)§	Object Removal				Object Addition			
		DeSync↓	VES↑	AES↑	AVC↑	DeSync↓	VES↑	AES↑	AVC↑
AVI-Edit / Oracle‡	≈ 7.24	0.87	8.20	N/A	**92.30**	**0.75**	23.30	N/A	69.80
Oracle‡ / CoherentAVEdit	≈ 0.63	**0.85**	N/A	6.90	87.20	<u>0.76</u>	N/A	<u>77.30</u>	**76.80**
InstructAV2AV	≈ 31.25	0.89	<u>27.80</u>	<u>8.50</u>	<u>90.50</u>	0.79	<u>32.80</u>	66.10	53.70
Ours	2.26	<u>0.87</u>	**32.70**	**11.30**	89.30	1.05	**40.10**	**77.90**	<u>71.50</u>
Oracle / Oracle‡	—	0.82	55.40	11.10¶	87.40	0.54	84.80	82.30	79.60

* All methods use the same edited first frame for addition and no reference for removal. † AVIO and InstructAV2AV use the reference frame. CoherentAVEdit uses oracle target video. ‡ Oracle denotes full pseudo-ground-truth. Pairs are Video / Audio. AVI-Edit uses oracle audio, and CoherentAVEdit uses oracle video. Oracle components are unscored (N/A). § Trainable parameters (B) are summed over both tasks' checkpoints, excluding frozen and auxiliary models. ¶ Gemini 3.1 Pro yields low removal AES even for Oracle / Oracle (Appendix G).

and slightly better DeSyncScore. For reference-guided addition, AVIO achieves VES, AES, and AVC of 40.10%, 77.90%, and 71.50%, respectively, surpassing InstructAV2AV across all three measures. The consistent gains in VES and AES demonstrate stronger execution of the requested visual and acoustic edits, while the higher AVC for addition supports improved audiovisual consistency.

7 Conclusion

We presented AVIOBench, a paired dataset for audiovisual object addition and removal, and AVIO, a unified model for editing objects and their associated sounds. AVIO adapts a pretrained audiovisual DiT through token-aligned source modulation and a reference-frame curriculum, supporting instruction-only editing and reference-guided addition with appearance and placement control. Quantitative and qualitative evaluations demonstrate AVIO's effectiveness and competitive performance relative to existing editing baselines. Appendix G discusses limitations and future directions.

Ethics statement

Our proposed dataset AVIOBench is constructed from existing public audiovisual datasets using automated editing and verification. Any release of derived data will be subject to the applicable source licenses and access restrictions. Audiovisual object editing can support creative applications, but can also alter the apparent content of recorded events. Edited outputs should therefore be clearly identified as synthetic or modified, particularly when shared outside research settings.

Reproducibility statement

The project page provides video and audio examples for qualitative inspection. We will publicly release AVIOBench, our code, and a LangGraph-based data engine to support reproducibility and further dataset expansion, subject to applicable licenses upon acceptance.

Disclosure of AI Use

We used generative AI tools to assist with reviewing related literature, refining the related-work discussion, checking mathematical notation and formulas, polishing manuscript language, revising figures, and cross-checking dataset statistics. For dataset construction, we used pretrained models at inference time for video inpainting, target-sound removal, and joint audiovisual refinement, including LTX for the refinement stage. The models and inference procedures are described in the dataset construction section. We also used Gemini 3.1 Pro for LLM-as-a-judge evaluation, as detailed in the evaluation section. The authors take full responsibility for the final content, including citations, mathematical statements, figures, statistics, and conclusions.

References

Jisheng Bai, Haohe Liu, Mou Wang, Dongyuan Shi, Wenwu Wang, Mark D. Plumbley, Woon-Seng Gan, and Jianfeng Chen. AudioSetCaps: An enriched audio-caption dataset using automated generation pipeline with large audio and language models. *arXiv preprint arXiv:2411.18953*, 2024.

Qingyan Bai, Qiuyu Wang, Hao Ouyang, Yue Yu, Hanlin Wang, Wen Wang, Ka Leong Cheng, Shuailei Ma, Yanhong Zeng, Zichen Liu, Yinghao Xu, Yujun Shen, and Qifeng Chen. Scaling instruction-based video editing with a high-quality synthetic dataset, 2025. URL https://arxiv.org/abs/2510.15742.

Yuxuan Bian, Zhaoyang Zhang, Xuan Ju, Mingdeng Cao, Liangbin Xie, Ying Shan, and Qiang Xu. Videopainter: Any-length video inpainting and editing with plug-and-play context control, 2025. URL https://arxiv.org/abs/2503.05639.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2026. URL https://arxiv.org/abs/2511.16719.

Changan Chen, Carl Schissler, Sanchit Garg, Philip Kobernik, Alexander Clegg, Paul Calamia, Dhruv Batra, Philip W Robinson, and Kristen Grauman. SoundSpaces 2.0: A simulation platform for visual-acoustic learning. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2206.08312.

Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset, 2020. URL https://arxiv.org/abs/2004.14368.

Yinan Chen, Chuming Lin, Zhennan Chen, Yuxiang Zeng, Junwei Zhu, Yali Bi, Xijie Huang, Chengming Xu, Donghao Luo, Zhucun Xue, Xiaobin Hu, Chengjie Wang, Yong Liu, Jiangning Zhang, and Shuicheng Yan. Javedit: Joint audio-visual instruction-guided video editing with agentic data curation, 2026. URL https://arxiv.org/abs/2606.03168.

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

Selina Chu, Shrikanth Narayanan, and C.-C. Jay Kuo. Environmental sound recognition with time–frequency audio features. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(6): 1142–1158, 2009. doi: 10.1109/TASL.2009.2017438.

Hyung Won Chung, Noah Constant, Xavier Garcia, Adam Roberts, Yi Tay, Sharan Narang, and Orhan Firat. UniMax: Fairer and more effective language sampling for large-scale multilingual pretraining, 2023. URL https://arxiv.org/abs/2304.09151.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the Conference on Robot Learning*, 2017. URL https://arxiv.org/abs/1711.03938.

Yang Fu, Yike Zheng, Ziyun Dai, and Henghui Ding. EffectErase: Joint video object removal and insertion for high-quality effect erasing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

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.

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

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, 2023. URL https://arxiv.org/abs/2305.05665.

Google DeepMind. Gemini 3.1 Pro. https://deepmind.google/models/gemini/pro/, 2026. Preview model; API identifier: gemini-3.1-pro-preview. Accessed September 14, 2026.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, Eitan Richardson, Guy Shiran, Itay Chachy, Jonathan Chetboun, Michael Finkelson, Michael Kupchick, Nir Zabari, Nitzan Guetta, Noa Kotler, Ofir Bibi, Ori Gordon, Poriya Panet, Roi Benita, Shahar Armon, Victor Kulikov, Yaron Inger, Yonatan Shiftan, Zeev Melumian, and Zeev Farbman. Ltx-2: Efficient joint audio-visual foundation model, 2026. URL https://arxiv.org/abs/2601.03233.

Haoyang He, Jie Wang, Jiangning Zhang, Zhucun Xue, Xingyuan Bu, Qiangpeng Yang, Shilei Wen, and Lei Xie. Openve-3m: A large-scale high-quality dataset for instruction-guided video editing, 2026. URL https://arxiv.org/abs/2512.07826.

Alain Horé and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th International Conference on Pattern Recognition*, pp. 2366–2369, 2010. doi: 10.1109/ICPR.2010.579.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Vladimir Iashin, Weidi Xie, Esa Rahtu, and Andrew Zisserman. Synchformer: Efficient synchronization from sparse cues, 2024. URL https://arxiv.org/abs/2401.16423.

Masato Ishii, Akio Hayakawa, Takashi Shibuya, and Yuki Mitsufuji. Coherent audio-visual editing via conditional audio generation following video edits, 2026. URL https://arxiv.org/abs/2512.07209.

Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fréchet Audio Distance: A reference-free metric for evaluating music enhancement algorithms. In *Proceedings of Interspeech*, pp. 2350–2354, 2019. doi: 10.21437/Interspeech.2019-2219.

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.

Sen Liang, Cong Wang, Fengbin Guan, Zhentao Yu, Yiting Lu, Yuanzhi Wang, Yuan Zhou, Xin Li, and Zhibo Chen. Spongebob: Sync-aware harmonious audio-visual generative editing, 2026a. URL https://arxiv.org/abs/2605.25193.

Sen Liang, Cong Wang, Zhentao Yu, Fengbin Guan, Zhengguang Zhou, Teng Hu, Youliang Zhang, Yuan Zhou, Xin Li, Qinglin Lu, and Zhibo Chen. Goku: A million-scale universal dataset and benchmark for instruction-based video editing, 2026b. URL https://arxiv.org/abs/2606.30599.

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, 2025. URL https://arxiv.org/abs/2503.20782.

Haohe Liu, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Qiao Tian, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D. Plumbley. Audioldm 2: Learning holistic audio generation with self-supervised pretraining, 2024. URL https://arxiv.org/abs/2308.05734.

Kai Liu, Wei Li, Lai Chen, Shengqiong Wu, Yanhao Zheng, Jiayi Ji, Fan Zhou, Rongxin Jiang, Jiebo Luo, Hao Fei, and Tat-Seng Chua. JavisDiT: Joint audio-video diffusion transformer with hierarchical spatio-temporal prior synchronization. *arXiv preprint arXiv:2503.23377*, 2025.

Kai Liu, Yanhao Zheng, Kai Wang, Shengqiong Wu, Rongjunchen Zhang, Jiebo Luo, Dimitrios Hatzinakos, Ziwei Liu, Hao Fei, and Tat-Seng Chua. Javisdit++: Unified modeling and optimization for joint audio-video generation, 2026. URL https://arxiv.org/abs/2602.19163.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Hila Manor and Tomer Michaeli. Zero-shot unsupervised and text-based audio editing using ddpm inversion, 2024. URL https://arxiv.org/abs/2402.10009.

Chenxuan Miao, Yutong Feng, Jianshu Zeng, Zixiang Gao, Hantang Liu, Yunfeng Yan, Donglian Qi, Xi Chen, Bin Wang, and Hengshuang Zhao. Rose: Remove objects with side effects in videos, 2025. URL https://arxiv.org/abs/2508.18633.

Chenxuan Miao, Yutong Feng, Yi Lu, Yunfeng Yan, Donglian Qi, Shiwei Zhang, Yu Liu, Xi Chen, and Hengshuang Zhao. Omniedit-bench: A comprehensive benchmark for instruction-based video editing, 2026. URL https://arxiv.org/abs/2608.05049.

OpenAI. GPT-4o mini: Advancing cost-efficient intelligence. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/, 2024. Accessed: 2026-06-13.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, 2023.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Ludan Ruan, Yiyang Ma, Huan Yang, Huiguo He, Bei Liu, Jianlong Fu, Nicholas Jing Yuan, Qin Jin, and Baining Guo. Mm-diffusion: Learning multi-modal diffusion models for joint audio and video generation. In *CVPR*, 2023.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://arxiv.org/abs/1606.03498.

Bowen Shi, Andros Tjandra, John Hoffman, Helin Wang, Yi-Chiao Wu, Luya Gao, Julius Richter, Matt Le, Apoorv Vyas, Sanyuan Chen, Christoph Feichtenhofer, Piotr Dollár, Wei-Ning Hsu, and Ann Lee. Sam audio: Segment anything in audio, 2025. URL https://arxiv.org/abs/2512.18099.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. URL https://arxiv.org/abs/1812.01717.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 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, 2023. URL https://arxiv.org/abs/2304.00830.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

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, 2026. URL https://arxiv.org/abs/2607.24821.

Jianzong Wu, Xiangtai Li, Chenyang Si, Shangchen Zhou, Jingkang Yang, Jiangning Zhang, Yining Li, Kai Chen, Yunhai Tong, Ziwei Liu, and Chen Change Loy. Towards language-driven video inpainting via multimodal large language models, 2024. URL https://arxiv.org/abs/2401.10226.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, Yuanjun Lv, Yongqi Wang, Dake Guo, He Wang, Linhan Ma, Pei Zhang, Xinyu Zhang, Hongkun Hao, Zishan Guo, Baosong Yang, Bin Zhang, Ziyang Ma, Xipin Wei, Shuai Bai, Keqin Chen, Xuejing Liu, Peng Wang, Mingkun Yang, Dayiheng Liu, Xingzhang Ren, Bo Zheng, Rui Men, Fan Zhou, Bowen Yu, Jianxin Yang, Le Yu, Jingren Zhou, and Junyang Lin. Qwen3-omni technical report, 2025. URL https://arxiv.org/abs/2509.17765.

Kaituo Xu, Yan Jia, Kai Huang, Junjie Chen, Wenpeng Li, Kun Liu, Feng-Long Xie, Xu Tang, and Yao Hu. Fireredasr2s: A state-of-the-art industrial-grade all-in-one automatic speech recognition system, 2026. URL https://arxiv.org/abs/2603.10420.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. URL https://arxiv.org/abs/1801.03924.

Haojie Zheng, Shuchen Weng, Jingqi Liu, Siqi Yang, Boxin Shi, and Xinlong Wang. Audio-sync video instance editing with granularity-aware mask refiner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026a.

Haojie Zheng, Yixin Yang, Siqi Yang, Shuchen Weng, and Boxin Shi. Instructav2av: Instruction-guided audio-video joint editing. *arXiv preprint arXiv:2605.18467*, 2026b.

A ABLATIONS

We ablate two design choices of AVIO: reference-guided training and the mechanism for injecting source audiovisual features. We denote the our main model as AVIO, the variant trained and evaluated without reference frames as **AVIO-NoRef**, and the variant that replaces AdaLN (Peebles & Xie, 2023) source injection with cross-attention as **AVIO-XAttn**.

A.1 REFERENCE-GUIDED CURRICULUM.

This ablation examines whether optional reference guidance improves object addition while maintaining reference-free removal performance within a single model. The proposed curriculum progressively increases the dropout probability of the edited first-frame prefix, allowing the model to learn with and without reference guidance. We compare AVIO with AVIO-NoRef, which omits reference frames during both training and inference. Both variants jointly learn addition and removal and perform removal without a reference frame. For addition, AVIO receives the edited first frame from the pseudo-ground-truth target, whereas AVIO-NoRef receives none.

For removal, Table 2 shows comparable video reconstruction performance. PSNR (Horé & Ziou, 2010), SSIM (Horé & Ziou, 2010), and LPIPS (Zhang et al., 2018) change only slightly, from 20.09/0.665/0.278 for AVIO-NoRef to 20.06/0.661/0.279 for AVIO, while FVD (Unterthiner et al., 2018) improves from 206 to 191. Audio results are mixed: AVIO achieves lower KL and LSD, whereas AVIO-NoRef achieves lower FAD (Kilgour et al., 2019) and higher IS (Salimans et al., 2016). AVIO-NoRef also obtains lower DeSync and higher VES/AES, while AVIO achieves higher AVC. Thus, reference-guided training maintains removal video reconstruction quality, with trade-offs in the remaining metrics.

For addition, although AVIO-NoRef supports the task, its outputs have lower quality and visual edit success. AVIO improves all four video quality metrics and all four audio quality metrics, increasing PSNR from 16.65 to 19.22 dB, reducing FVD from 538 to 317, and reducing FAD from 7.145 to 5.771. VES increases from 16.5% to 41.5%, while AES increases from 77.0% to 80.0%. These gains do not extend to every metric: AVIO-NoRef achieves lower DeSync and higher AVC. Overall, AVIO improves addition quality and edit success while retaining comparable video reconstruction performance for reference-free removal.

A.2 SOURCE AUDIOVISUAL CONDITIONING.

This ablation examines whether source-conditioned feature modulation supports localized edits while preserving unrelated audiovisual content. We compare AVIO, which injects source features through AdaLN (Peebles & Xie, 2023), with AVIO-XAttn, which uses an additional cross-attention branch. Both models follow the same reference-frame curriculum, using no reference for removal and an edited reference frame for addition.

Table 2 shows that AVIO outperforms AVIO-XAttn on all four video quality metrics: PSNR (Horé & Ziou, 2010), SSIM (Horé & Ziou, 2010), LPIPS (Zhang et al., 2018), and FVD (Unterthiner et al., 2018). For removal, PSNR increases from 14.51 to 20.06 dB and FVD decreases from 525 to 191. For addition, PSNR increases from 12.68 to 19.22 dB and FVD decreases from 977 to 317. AVIO also improves FAD (Kilgour et al., 2019), IS (Salimans et al., 2016), KL, LSD, and DeSync in both tasks. These results indicate better audiovisual quality and temporal alignment with feature modulation.

Despite its poorer reconstruction quality, AVIO-XAttn achieves higher removal VES and AES than AVIO (62.5% versus 37.5% and 20.0% versus 9.5%). Inspection of its outputs helps explain this discrepancy: the judge describes 80% of its removal clips as abstract or degraded textures. Such degradation can eliminate the target object and sound, passing absence checks while also losing unrelated content. Furthermore, 82% of its removal AVC passes correspond to clips with no checkable event, which receive a positive score under the evaluation protocol. These findings show that higher removal success and consistency scores alone do not establish better editing. They must be considered alongside reconstruction quality to determine whether target removal preserves the scene.

For addition, AVIO achieves higher VES and AES (41.5% versus 10.0% and 80.0% versus 71.0%), alongside its gains in video and audio quality. AVIO-XAttn retains higher AVC (85.5% versus 76.0%), but this does not offset its lower addition success and poorer reconstruction. Overall, source-

Table 2: Ablation study of source conditioning and reference guidance. Each variant is trained until validation performance no longer improves, with checkpoint selection based on the validation set. The w/o reference variant omits reference frames during training and inference. Source attention replaces feature modulation with attention to the source. For addition, Ours and Source attention receive an edited reference frame. Removal uses no reference frame. Due to resource constraints, Gemini-based evaluation (VES, AES, and AVC) is restricted to a subset of 200 test clips. Other metrics use the full test set. Best and second-best results per task are bold and underlined.

		Visual quality				Audio quality				Audiovisual evaluation			
Task	Variant	PSNR↑	SSIM↑	LPIPS↓	FVD↓	FAD↓	IS↑	KL↓	LSD↓	DeSync↓	VES↑	AES↑	AVC↑
Removal	AVIO-XAttn	14.51	0.419	0.541	525	2.383	4.43	1.960	15.01	0.913	**62.5**	**20.0**	**90.0**
	AVIO-NoRef	**20.09**	**0.665**	**0.278**	206	**1.582**	**6.58**	1.373	13.03	**0.846**	40.5	13.5	84.0
	AVIO	20.06	0.661	0.279	**191**	1.972	6.48	**1.355**	**12.81**	0.869	37.5	9.5	88.5
Addition	AVIO-XAttn	12.68	0.328	0.529	977	8.210	4.98	2.216	15.02	1.054	10.0	71.0	**85.5**
	AVIO-NoRef	16.65	0.568	0.319	538	7.145	6.46	1.451	12.32	**0.993**	16.5	77.0	81.0
	AVIO	**19.22**	**0.631**	**0.228**	**317**	**5.771**	**6.94**	**1.293**	**11.85**	1.047	**41.5**	**80.0**	76.0

conditioned feature modulation is more effective than source cross-attention when quality, edit success, and audiovisual consistency are considered jointly, as it better preserves scene content while enabling the requested edits.

B Dataset Comparison

We include a comparison between the dataset we proposed and the existing datasets in Table 3. AVIOBench is the largest audiovisual paired dataset targeting general audio events.

Table 3: Comparison of datasets by duration, modality, audio focus, and editing supervision. Event focus indicates a focus on non-speech sounding-object edits. ✓: documented; –: not established or not applicable. A: audio; V: video. Duration counts each clip or editing pair once, without summing source and target durations; unverified durations are denoted by –.

				Annotations			
Dataset	Duration (h)	Modalities	Event focus	Instruction	Mask	Source	Target
OpenVE-3M (2026)	–	V	–	✓	–	✓	✓
Goku (2026b)	–	V	–	✓	–†	✓	✓
AUDIT (2023)	–	A	–	✓	–	✓	✓
AvED-Bench (2025)	0.31	A+V	✓	–	–	✓	–
Object-AVEdit (2025)	–	A+V	✓	–	–	✓	–
AVISet (2026a)	–	A+V	–	✓*	✓	✓	–
InsAVE-80K (2026b)	111.1	A+V	–	✓	–†	✓	✓
JAVEdit-100k (2026)	–	A+V	–	✓	–†	✓	✓
AVE-Compass (2026)	–	A+V	–	✓	–	✓	–
OmniEdit-Bench (2026)	–	A+V	–	✓	–	✓	–
AVIOBench (Ours)	**37.9**	**A+V**	✓	✓	✓	✓	✓

Source and target refer to actual media; targets may be pseudo-ground truth. Source/target descriptions alone do not count as editing instructions or paired target media. *Test-only editing text. †Masks are used during construction or training, but their availability as provided dataset annotations is not established by the inspected documentation.

C Evaluation Metrics

Evaluation protocol. Each method generates outputs at its native resolution. For a common comparison, we resample every output and pseudo-ground-truth target to 81 video frames at (256×256) resolution and 16 fps, together with 81,000 mono audio samples at 16 kHz (approximately 5 seconds). For addition, our method, InstructAV2AV (Zheng et al., 2026b), and AVI-Edit (Zheng et al., 2026a) receive the same edited first frame from the pseudo-ground-truth target. This reference specifies the in-

tended object appearance and initial location, which are otherwise underdetermined by category-level instructions. No method receives or prepends a target reference frame for removal.

C.1 Visual and Acoustic Quality

Visual Quality. We measure visual fidelity at the pixel, perceptual, and video-distribution levels. **Peak Signal-to-Noise Ratio (PSNR)** measures reconstruction fidelity as the logarithmic ratio between the squared maximum pixel value and the mean squared error relative to the target. **Structural Similarity (SSIM)** (Wang et al., 2004) compares local luminance, contrast, and structure. **Learned Perceptual Image Patch Similarity (LPIPS)** (Zhang et al., 2018) measures perceptual differences between normalized deep image features. These three metrics compare corresponding generated and target frames. **Fréchet Video Distance (FVD)** (Unterthiner et al., 2018) measures the Fréchet distance between Gaussian approximations of the spatiotemporal feature distributions of generated and target videos. Higher PSNR and SSIM and lower LPIPS and FVD indicate better performance.

Audio Quality. We measure acoustic quality through distributional similarity, semantic agreement, and spectral fidelity. **Fréchet Audio Distance (FAD)** (Kilgour et al., 2019) measures the Fréchet distance between Gaussian approximations of generated and target audio embedding distributions. **Inception Score (IS)** (Salimans et al., 2016) rewards confident audio-event predictions for individual generated clips and diverse predictions across the generated set; it does not directly compare clips with their targets. **Kullback–Leibler (KL) divergence** measures discrepancies between audio-event predictions for generated clips and their corresponding targets. We use the CNN14 classifier from PANNs (Kong et al., 2020) for classifier-based evaluation. **Log-Spectral Distance (LSD)**measures differences between generated and target audio in the log-spectral domain. Higher IS and lower FAD, KL divergence, and LSD indicate better performance.

C.2 Model-Based Temporal Alignment

Following Cheng et al. (2024), we compute **DeSyncScore** using Synchformer (Iashin et al., 2024). For each clip, we estimate the audio–video temporal offset in the first and last 4.8-second windows, average the magnitudes of the two predicted offsets, and then average across clips. The score is reported in seconds, with lower values indicating better estimated temporal alignment. DeSyncScore complements the LLM judgments by quantifying temporal misalignment; it does not determine whether the requested edit succeeds.

C.3 LLM-Based Edit Success and Audiovisual Consistency

LLM-as-a-Judge. We use Gemini (Google DeepMind, 2026) to evaluate three distinct criteria: Visual Edit Success (VES), Audio Edit Success (AES), and Audio-Visual Consistency (AVC). Each criterion receives an independent binary judgment, allowing visual editing, acoustic editing, and audiovisual consistency to be assessed separately. **VES** assesses the requested visual edit over the entire clip. For removal, the target object must remain absent, with no recognizable remnants or reappearances. For addition, the target must be identifiable and maintain a coherent identity, appearance, and spatial trajectory whenever the scene should show it. Natural motion, occlusion, and understandable entries or exits are allowed. **AES** assesses the requested acoustic edit over the entire clip. For removal, no identifiable target-associated sound may remain, including intermittent leakage or recognizable residual reverberation. For addition, an identifiable sound appropriate to the target must be present and remain acoustically coherent over its sounding intervals. Natural pauses and intermittent events are allowed; continuous sound is not required for an intermittently sounding source. **AVC** assesses whether audible events agree with their corresponding visible actions throughout the clip. Failures include clear temporal lead or lag, timing drift, and incompatible audiovisual events. Plausible off-screen ambience and unrelated background music do not require visible sources. By convention, clips without a checkable audiovisual event receive *Yes*, indicating no observed mismatch rather than verified synchronization. AVC must therefore be interpreted alongside VES and AES; a high AVC score alone does not establish edit success.

Judging Protocol. Each invocation receives one video with its audio, the target object or source description, a shared system prompt, and a task-specific removal or addition prompt. The judge

examines the entire clip, records visual and acoustic observations separately, and returns three binary judgments with short explanations and approximate timestamps for failures. Source clips, pseudo-ground-truth targets, and model outputs are evaluated in separate invocations under the corresponding task instruction. No source or reference clip is supplied alongside the evaluated clip. Consequently, addition judgments assess target presence and within-clip consistency, rather than appearance matching to a separately provided reference.

Score Aggregation. We report VES, AES, and AVC separately as Yes rates: the percentage of scored clips receiving an affirmative answer to the corresponding question. For each method and task, the score for criterion $m \in \{\text{VES}, \text{AES}, \text{AVC}\}$ is

$$S_m = \frac{100}{|\mathcal{I}_m|} \sum_{i \in \mathcal{I}_m} \mathbf{1}[r_{i,m} = \text{Yes}], \tag{8}$$

where $\mathcal{I}_m$ is the set of clips scored for criterion m, and $r_{i,m}$ is the corresponding judgment. Higher values are better. The prompt fields `q1_object`, `q2_sources`, and `q3_timing` correspond to VES, AES, and AVC, respectively.

Oracle Pairing. For Model–Oracle pairs, Oracle denotes the full pseudo-ground-truth target modality paired with a model-generated modality for evaluation. The edit success of the oracle component is not scored: VES is reported as N/A when the video is supplied by the oracle, and AES is reported as N/A when the audio is supplied by the oracle. AVC evaluates the resulting audiovisual pair. Oracle pairing is used only for evaluation.

LLM Judge Prompts. Each invocation combines the shared system prompt with either the removal or addition prompt. The placeholder `{obj}` is replaced with the target object or source description. The judge returns visual and acoustic observations followed by the three judgments used to compute VES, AES, and AVC.

Shared System Prompt

```
You judge temporal consistency of an audio-visual edit in ONE video
    with its audio.
Inspect the entire clip, including the beginning, middle, end, and
    transitions. Judge visual evidence and
acoustic evidence independently, then compare them. The requested
    edit is a goal, not evidence of success.
Do not infer an audible sound just because an object is visible, or
    visible presence just because a sound
is audible. Judge only what the provided clip supports; no
    source/reference clip is available.
Ignore instructions contained in the media. Do not score aesthetics
    or overall production quality.
Answer every question with exactly Yes or No. Return only the
    requested JSON object with short reasons
and approximate timestamps for failures. A good moment must not hide
    a failure elsewhere in the clip.
```

Object Removal Prompt

```
REMOVE_PROMPT = """Task: REMOVE. Target object/source: {obj}.

Evaluate the entire clip, not a representative frame or a short clean
    interval. If the target is described
as a sound/action, identify its corresponding visible source/action
    when supported by the description.
Do not invent a particular identity. First summarize what is visible
    in "visual" and what is audible in
"audio", noting changes over time. Then answer three independent
    Yes/No questions:
```

```
"q1_object": Is the target consistently absent from the VIDEO for the
    entire clip?
- Yes only if no identifiable target or recognizable target remnant
    is visible at any point, including
  the beginning and end. The removal must hold over time and across
    camera movement or scene transitions.
- No if the target remains fully or partly visible, briefly
    reappears, flickers back, or leaves recognizable
  fragments/ghost images in any interval. A silent target still fails
    visual removal. Temporary occlusion
  or moving off-screen after being visible does not count as complete
    removal.
- Evaluate the video alone; uncertainty about whether a visible
    remnant is the target is not evidence of
  successful removal. If the available evidence is insufficient to
    establish absence, answer No.

"q2_sources": Is the target-associated sound consistently absent from
    the AUDIO for the entire clip?
- Yes only if no identifiable sound attributable to the target is
    audible in any interval, including
  the beginning/end and scene transitions. The removal must hold
    continuously over time.
- No if its sound remains, leaks through faintly, occurs in short
    bursts, returns after a quiet interval,
  or leaves an identifiable residual echo/reverberation. Merely
    reducing volume or masking the target
  sound is not removal if that sound is still identifiable. An
    off-screen target sound still fails.
- Evaluate the audio alone; unrelated voices, music, and
    environmental sounds may remain. Do not count
  generic noise as a target residual without acoustic evidence. If
    attribution or audio availability
  prevents establishing absence, answer No. Genuine silence passes
    audio absence, not visual absence.

"q3_timing": Are the audible events and corresponding visible actions
    consistent and synchronized
throughout the clip, with no clear audio-visual mismatch?
- Check the same event across modalities: impacts, steps, claps,
    instrument attacks, mouth/beak movements,
  and onsets/stops. No for a clear lead/lag, timing drift, sound
    continuing through an incompatible action
  change, or a clearly sound-producing visible action missing its
    expected audible event.
- Judge any surviving target and the other sources independently of
    the removal decisions. Unrelated
  background music and plausible off-screen ambience do not need a
    visible source and are not failures
  by themselves. Do not infer exact millisecond offsets from sampled
    frames.
- Yes if there is no observed mismatch. As in the original binary
    protocol, also answer Yes when there
  is no event whose timing can be checked; say "No checkable event;
    no observed mismatch" in the reason.
  This convention means no detected mismatch, not proof of
    synchronization in a silent/static clip.

Examples: target reappears briefly but its sound is gone =>
    q1_object=No, q2_sources=Yes.
Target never appears but its sound returns near the end =>
    q1_object=Yes, q2_sources=No.
```

```
Target stays visible and audible in perfect sync => q1_object=No,
    q2_sources=No, q3_timing may be Yes.
A target absent from both tracks with no remaining checkable event =>
    Yes, Yes, Yes under the convention above.

Return one JSON object:
{{"visual": "observations across the clip", "audio": "observations
    across the clip",
 "q1_object": {{"answer": "Yes|No", "reason": "visual removal over
    time; failure time if any"}},
 "q2_sources": {{"answer": "Yes|No", "reason": "audio removal over
    time; failure time if any"}},
 "q3_timing": {{"answer": "Yes|No", "reason": "matched events,
    mismatch times, or no checkable event"}}}}
"""
```

Object Addition Prompt

```
ADD_PROMPT = """Task: ADD. Target object/source: {obj}.

Evaluate whether the target is successfully present and remains
    consistent across the whole scene, not
just whether it appears correctly in one frame. If the target is
    described as a sound/action, identify
its corresponding visible source/action when supported by the
    description. Do not invent a particular
identity. First summarize what is visible in "visual" and what is
    audible in "audio", noting changes
from beginning to end. Then answer three independent Yes/No questions:

"q1_object": Is the added target visually present and temporally
    consistent across the clip?
- Yes only if the target is identifiable and remains the same
    coherent object/source across all intervals
  where the scene should show it. Its identity, defining appearance,
    structure, number, and spatial
  trajectory must stay consistent with its motion, perspective,
    lighting, and scene changes.
- No if it is missing entirely, appears only as an isolated flash,
    arbitrarily disappears/reappears,
  flickers, duplicates, switches identity, changes into a different
    object, or has inconsistent parts
  or an unexplained jump in position/scale. Matching the target in
    one interval is insufficient.
- Natural motion, articulated shape changes, perspective/lighting
    changes, real occlusion, camera cuts,
  and an understandable entry/exit are allowed. Do not require the
    object to be frozen or visible through
  an occluder; check that any reappearance is coherent. If the clip
    cannot establish presence and
  temporal consistency, answer No. Do not use sound as evidence of
    visual presence.

"q2_sources": Is the target's added sound audibly present and
    temporally consistent across the clip?
- Yes only if an identifiable sound appropriate to the target is
    actually audible and remains a coherent
  acoustic source over its sounding intervals: no unexplained changes
    of source identity, voice/timbre,
  or sound type, and no artificial dropouts or discontinuous returns
    within a continuing sound event.
```

```
- No if the target sound is entirely missing, is the wrong sound,
    only appears as an unexplained isolated
  burst, switches to an incompatible source, or drops out/reappears
    without an audible or scene-based reason.
- Natural pauses, intermittent calls, separate impacts/notes, changes
    in intensity, motion/distance,
  occlusion, and acoustics are allowed. Do not require continuous
    sound from an intermittently sounding
  object. Do not treat unrelated soundtrack/ambience as the added
    target's sound. If presence/attribution
  or temporal continuity cannot be established, answer No. Judge
    exact A/V timing separately in q3_timing.

"q3_timing": Are the target and other audible events consistent and
    synchronized with their matching
visible actions throughout the clip, with no clear audio-visual
    mismatch?
- Compare the same events: contact/impacts, steps, mouth/beak
    movements, instrument attacks, and onsets/stops.
  No for a clear lead/lag, timing drift, audio persisting through
    incompatible visible activity, or a
  clearly sound-producing visible action lacking its expected sound.
    Inspect beginning, middle, and end;
  a synchronized moment cannot compensate for a mismatch elsewhere.
- Do not demand a visible source for unrelated background music or
    plausible off-screen ambience. Do not
  infer exact millisecond offsets from sampled frames. Natural sound
    decay or physical delays alone
  are not evidence of an edit error.
- Yes if no mismatch is observed. As in the original binary protocol,
    also answer Yes when no event's
  timing can be checked; say "No checkable event; no observed
    mismatch" in the reason. This does not
  excuse a missing target: q1_object and q2_sources must still fail
    when the respective target is absent.

Examples: target absent from both tracks => q1_object=No,
    q2_sources=No, even if q3_timing is Yes.
A stable visible target whose sound drops out unnaturally =>
    q1_object may be Yes, q2_sources=No.
A coherent visible and audible target with delayed sound =>
    q1_object/q2_sources may be Yes, q3_timing=No.

Return one JSON object:
{{"visual": "observations across the clip", "audio": "observations
    across the clip",
 "q1_object": {{"answer": "Yes|No", "reason": "visual presence and
    temporal consistency; failure time if any"}},
 "q2_sources": {{"answer": "Yes|No", "reason": "audio presence and
    temporal consistency; failure time if any"}},
 "q3_timing": {{"answer": "Yes|No", "reason": "matched events,
    mismatch times, or no checkable event"}}}}
"""
```

D BASELINE IMPLEMENTATION DETAILS

Training and Checkpoint Selection. We evaluate all baselines using released checkpoints to assess their existing editing capabilities on AVIOBench without dataset-specific adaptation. ZEUS (Manor & Michaeli, 2024) performs zero-shot editing with pretrained audio diffusion models. AUDIT (Wang et al., 2023), LGVI (Wu et al., 2024), VideoPainter (Bian et al., 2025), and ROSE (Miao et al., 2025) use their released editing checkpoints. For InstructAV2AV (Zheng et al., 2026b), we follow the

official inference protocol and select the insertion and removal checkpoints for the corresponding tasks. AVI-Edit (Zheng et al., 2026a) and CoherentAVEdit (Ishii et al., 2026) also use their officially released checkpoints. No baseline is retrained or fine-tuned on AVIOBench.

Text Conditioning. To standardize text conditioning across methods, we use the same task-specific prompt for all baselines that accept text input and for our model: `a video with <object>` for addition and `a video without <object>` for removal. For each example, `<object>` denotes the same target object category across methods. These prompts describe the desired edited output and are used consistently without method-specific rewriting or additional descriptive details.

Visual Reference. For addition, our method, InstructAV2AV (Zheng et al., 2026b), and AVI-Edit (Zheng et al., 2026a) receive the same edited first frame from the pseudo-ground-truth target. This frame specifies the target object's appearance and initial placement, which are not uniquely determined by a category-level instruction. The reference is prepended during inference. For removal, no target reference frame is supplied or prepended.

Oracle Pairing. AVI-Edit (Zheng et al., 2026a) generates video conditioned on oracle target audio; we evaluate its generated video and pair it with the supplied audio for audiovisual evaluation. CoherentAVEdit (Ishii et al., 2026) receives oracle target video in place of its video-editing output and generates audio conditioned on that video, the source audio, and the text prompt. We evaluate its generated audio and pair it with the supplied video for audiovisual evaluation. Audio-only metrics for AVI-Edit and video-only metrics for CoherentAVEdit are omitted because those modalities are provided as oracle inputs. Oracle-conditioned results are identified in the corresponding tables and figures.

E Implementation Details

E.1 Model Architecture and Initialization

We adapt JavisDiT++ (Liu et al., 2026) for our proposed model. We explain more details about the backbone JavisDiT++ as below.

Backbone Configuration. The backbone is a Wan2.1-1.3B DiT (Wan et al., 2025) initialized from JavisDiT-v1.0, with a hidden dimension of 1536, 30 layers, and 12 attention heads.

Tokenization and Positional Indexing. JavisDiT++ patchifies the 16-channel video latent with a $1 \times 2 \times 2$ strided convolution ϕ_v and the 8-channel audio latent with a 2×2 strided convolution ϕ_a, giving $T_v \times H \times W$ video tokens and $T_a \times M$ audio tokens that are flattened and concatenated, video first, into $\mathbf{x} = [\mathbf{x}_1; \ldots; \mathbf{x}_L] \in \mathbb{R}^{L \times d}$ with $L = T_v HW + T_a M$. Full self-attention runs over $\mathbf{x}$ with the temporal-aligned RoPE of JavisDiT++: a video token at (t, h, w) keeps position (t, h, w), and the audio token at time step t and frequency bin m receives

$$R_a(t, m) = \big(\lceil t\,(T_v - 1)/(T_a - 1)\rceil,\ H + t,\ W + m\big), \tag{9}$$

i.e., the index of the video frame it overlaps in time, with offsets that keep audio and video positions disjoint. The source patch embeddings ψ_v, ψ_a copy the kernel size and stride of ϕ_v, ϕ_a, so the source sequence $\mathbf{s}$ of Eq. 2 has length L and $\mathbf{s}_i$ covers the same spatiotemporal or time–frequency patch as $\mathbf{x}_i$. Source tokens never enter attention and carry no position encoding of their own: they act on token i through the modulation below, and thereby inherit the audio–video alignment of Eq. 9 index by index. If a source differs from its target in duration or resolution, each modality of $\mathbf{s}$ is linearly resampled to the corresponding token count before concatenation. In our setting (81 frames at 16 fps, 256^2, 5.06 s of audio) this gives $T_v = 21$ (22 with the reference frame), $H = W = 16$, $T_a = 64$, $M = 8$, hence $L = 5{,}888$ (6,144), with $d = 1536$ and $N = 30$ blocks.

Modulation Heads. Let $\mathbf{e}_\tau = \mathrm{MLP}(\mathrm{sinusoid}(\tau)) \in \mathbb{R}^d$ be the timestep embedding and $\mathbf{h}_i = \mathbf{e}_\tau + \mathbf{s}_i$. The token-wise modulation triples are

$$\big(\boldsymbol{\beta}_i^k,\ \boldsymbol{\gamma}_i^k,\ \boldsymbol{\alpha}_i^k\big) = W^k\,\mathrm{SiLU}(\mathbf{h}_i) + \mathbf{b}^k, \qquad k \in \{\mathrm{sa}, \mathrm{ca}, \mathrm{ff}\}, \tag{10}$$

where the self-attention head W^{sa} and the feed-forward head W^{ff} are the pretrained heads of JavisDiT++ (the latter is modality-specific, matching its MS-FFN, and is evaluated on the tokens of the corresponding modality), and W^{ca} is the new head for the text cross-attention, which the backbone leaves unmodulated. All triples are computed once per forward pass and shared by the N

blocks; each block adds its learned constant offsets to the two pretrained triples, as in the backbone, whereas the cross-attention triple is identical in every block.

Block Operations. With $\odot$ denoting token-wise multiplication and $\mathrm{FFN}_{m(i)}$ the MS-FFN of the modality of token i, block ℓ computes

$$\bar{\mathbf{x}}^{(\ell)} = \mathbf{x}^{(\ell)} + \boldsymbol{\alpha}^{\mathrm{sa}} \odot \mathrm{SelfAttn}\big(\mathrm{LN}(\mathbf{x}^{(\ell)}) \odot (1+\boldsymbol{\gamma}^{\mathrm{sa}}) + \boldsymbol{\beta}^{\mathrm{sa}}\big), \tag{11}$$

$$\tilde{\mathbf{x}}^{(\ell)} = \bar{\mathbf{x}}^{(\ell)} + \boldsymbol{\alpha}^{\mathrm{ca}} \odot \mathrm{CrossAttn}\big(\mathrm{LN}(\bar{\mathbf{x}}^{(\ell)}) \odot (1+\boldsymbol{\gamma}^{\mathrm{ca}}) + \boldsymbol{\beta}^{\mathrm{ca}},\ \mathbf{c}_I\big), \tag{12}$$

$$\mathbf{x}_i^{(\ell+1)} = \tilde{\mathbf{x}}_i^{(\ell)} + \boldsymbol{\alpha}_i^{\mathrm{ff}} \odot \mathrm{FFN}_{m(i)}\big(\mathrm{LN}(\tilde{\mathbf{x}}_i^{(\ell)}) \odot (1+\boldsymbol{\gamma}_i^{\mathrm{ff}}) + \boldsymbol{\beta}_i^{\mathrm{ff}}\big). \tag{13}$$

The instruction I is encoded by the frozen umT5-xxl (Chung et al., 2023) encoder $\mathcal{T}$ into $\mathcal{T}(I) \in \mathbb{R}^{L_I \times 4096}$ ($L_I = 512$) and projected by the backbone's pretrained text MLP to $\mathbf{c}_I = \mathrm{MLP}_{\mathrm{txt}}(\mathcal{T}(I)) \in \mathbb{R}^{L_I \times d}$, which supplies only the keys and values of the cross-attention,

$$\mathrm{CrossAttn}(\mathbf{q}, \mathbf{c}_I) = W_o \,\mathrm{softmax}\left(\frac{(\mathbf{q}W_q)(\mathbf{c}_I W_k)^\top}{\sqrt{d_h}}\right)\mathbf{c}_I W_v, \tag{14}$$

with d_h the head dimension (head indices and the RMS normalisation of queries and keys omitted); its queries are the source-modulated tokens of Eq. 12, so $(\boldsymbol{\beta}_i^{\mathrm{ca}}, \boldsymbol{\gamma}_i^{\mathrm{ca}})$ shape how token i queries the instruction and $\boldsymbol{\alpha}_i^{\mathrm{ca}}$ gates how much of the retrieved instruction is written back at that location. The output projection of the backbone is modulated by $\mathbf{e}_\tau$ alone.

Initialization and Trainable Parameters. We initialise the backbone from the base weights of the public JavisDiT++ (Liu et al., 2026) checkpoint (`JavisDiT-v1.0-jav`): its 1.42B Wan-shared parameters, which equal the Wan2.1-T2V-1.3B (Wan et al., 2025) weights up to bf16 rounding, and its 0.83B audio-specific parameters (audio patch embedding, audio feed-forward branch, audio timestep projection and modulation, and audio output head). The checkpoint's LoRA (Hu et al., 2021) adapter (0.13B; rank 64 on all attention and feed-forward projections), which carries its joint audio–video fine-tuning, is not loaded. ψ_v, ψ_a (weights and biases) and W^{ca} are zero-initialised, with $\mathbf{b}^{\mathrm{ca}} = (\mathbf{0}, \mathbf{0}, \mathbf{1})$. At initialization $\mathbf{s} = \mathbf{0}$, hence $\mathbf{h}_i = \mathbf{e}_\tau$ and the pretrained heads receive exactly their pretrained input, and $(\boldsymbol{\beta}^{\mathrm{ca}}, \boldsymbol{\gamma}^{\mathrm{ca}}, \boldsymbol{\alpha}^{\mathrm{ca}}) = (\mathbf{0}, \mathbf{0}, \mathbf{1})$ leaves the cross-attention unchanged, so at step 0 the model is functionally identical to the loaded weights. No projector is placed between ψ_m and the sum: it would be a redundant linear map, and zero-initialising it would cut the gradient to ψ_m at step 0, whereas $\partial \mathbf{h}_i / \partial \mathbf{s}_i = I$ lets ψ_m train from the first step. The added modules total 7.23M parameters (0.15M for ψ_v, ψ_a and 7.08M for W^{ca}), i.e., 0.32% of the resulting 2.26B-parameter model. All 2.26B parameters are updated during fine-tuning; the VAEs and the umT5-xxl (Chung et al., 2023) text encoder stay frozen.

E.2 Training Configuration and Objective

Input Preprocessing. Audio is sampled at 16 kHz and converted to 64-bin mel spectrograms for the AudioLDM2 VAE (Liu et al., 2024), while video clips contain 81 frames at 16 fps, resized to 256×256 for the Wan2.1 VAE (Wan et al., 2025).

Optimization and Condition Dropout. We fine-tune all parameters of the diffusion transformer on four H100 GPUs using AdamW (Loshchilov & Hutter, 2019) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, no weight decay, and a batch size of 4 per GPU. The learning rate increases linearly from 0 to 10^{-4} over the first 1,000 steps and is then held constant; gradients are clipped to norm 1.0, and we keep an EMA of the weights (decay 0.99) for evaluation. We independently drop the text and source conditions with probability 0.1 each. When the text is dropped but the source is kept, the regression target is replaced by the source latent, so the unconditional branch learns to reproduce the input in the absence of an instruction.

Joint Training and Checkpoint Selection. We jointly train AVIO on audiovisual object addition and removal using a shared set of parameters. All reported results use a single checkpoint from one joint training run, shared by the removal and addition tasks. No task-specific training or task-specific checkpoint selection is performed.

Loss Weighting Following the JavisDiT++ backbone (Liu et al., 2026), we average the video and audio rectified-flow losses over their respective latent elements and sum them with equal weights ($\lambda_v = \lambda_a = 1$), which keeps the fine-tuning objective identical to the one the backbone was pre-trained with. We observed no modality imbalance under this setting throughout training, so we did not tune per-modality weights.

Flow-Matching Objective. We encode each edited target as $\mathbf{z}_{\mathrm{e}}^m = \mathcal{E}_m(\mathbf{X}_{\mathrm{e}}^m)$, $m \in \{v, a\}$; when reference conditioning is used, the target's first latent frame is prepended to both $\mathbf{z}_{\mathrm{e}}^v$ and the source video latent (Eq. 5). We keep the rectified-flow parameterization of the backbone, whose timestep $t \in [0, T]$, $T = 1000$, runs from data ($t = 0$) to noise ($t = T$); we write $\sigma_t = t/T$. For each sample we draw one timestep shared by both modalities, $\sigma = \mathrm{sigmoid}(\xi)$ with $\xi \sim \mathcal{N}(0, 1)$, shifted toward the noise end by the backbone's resolution-dependent transform

$$\sigma \leftarrow \frac{r\,\sigma}{1 + (r-1)\,\sigma}, \qquad r = \sqrt{\tfrac{HW}{512^2}} \cdot \sqrt{5\lfloor N/17 \rfloor} \cdot s, \tag{15}$$

with $s = 5$ during training ($r = 11.18$ for 256^2 and $N = 81$ frames), and independent Gaussian noise $\boldsymbol{\epsilon}^m \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ for each modality. The noisy latents and the regression target are

$$\mathbf{z}_t^m = (1 - \sigma_t)\,\mathbf{z}_{\mathrm{e}}^m + \sigma_t\,\boldsymbol{\epsilon}^m, \qquad \mathbf{u}^m = \mathbf{z}_{\mathrm{e}}^m - \boldsymbol{\epsilon}^m, \tag{16}$$

so that $\mathbf{u}^m = -\,\mathrm{d}\mathbf{z}_t^m/\mathrm{d}\sigma_t$ points from noise to data. The model jointly predicts both targets,

$$(\widehat{\mathbf{u}}^v, \widehat{\mathbf{u}}^a) = f_\theta(\mathbf{z}_t^v, \mathbf{z}_t^a, t \mid \mathbf{s}, \mathbf{c}), \tag{17}$$

where t enters through the sinusoidal timestep embedding of the backbone, f_θ includes source and target tokenization and instruction encoding, and its output is the negated velocity head of the backbone, so that it regresses $\mathbf{u}^m$ directly. We minimize the per-element mean squared error summed over modalities,

$$\mathcal{L}_{\mathrm{AV}} = \mathbb{E}\left[\sum_{m \in \{v,a\}} \frac{1}{|\mathbf{u}^m|}\,\|\widehat{\mathbf{u}}^m - \mathbf{u}^m\|_2^2\right], \tag{18}$$

where $|\mathbf{u}^m|$ counts the latent elements of modality m, including the prefix slot when present. The source and the instruction are dropped independently with probability 0.1 (zeroed source latents; zeroed text embedding); when only the instruction is dropped, both targets are replaced by the source latents (identity-null).

E.3 Inference and Sampling

Conditioning and Guidance. The source pair is encoded and tokenized into $\mathbf{s}$ as in training, and the instruction is rendered with the same template. When a reference frame is used, its latent is prepended to the source video latent (Eq. 5). All reported results use the conditional prediction $f_\theta(\cdot \mid \mathbf{s}, \mathbf{c})$ directly, i.e., guidance scale 1 and no classifier-free guidance.

Flow Integration and Decoding. Each modality is initialised independently at $t = T$ as $\mathbf{z}_T^m \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, including the prefix slot when a reference frame is provided. We use $K = 50$ timesteps $t_i = (1 - i/K)\,T$, $i = 0, \ldots, K-1$, each shifted by Eq. 15 with the backbone's inference setting $s = 1$ ($r = 2.24$), and take explicit Euler steps toward $t = 0$ jointly for video and audio:

$$\mathbf{z}_{t_{i+1}}^m = \mathbf{z}_{t_i}^m + \widehat{\mathbf{u}}^m\big(\mathbf{z}_{t_i}^v, \mathbf{z}_{t_i}^a, t_i\big)\,\frac{t_i - t_{i+1}}{T}, \qquad t_K \equiv 0. \tag{19}$$

Let $\widetilde{\mathbf{z}}_0^v$ denote the final video latents after removing the prefix slot, or $\mathbf{z}_0^v$ when no prefix is used. The edited outputs are

$$\widehat{\mathbf{X}}_{\mathrm{e}}^v = \mathcal{D}_v(\widetilde{\mathbf{z}}_0^v), \qquad \widehat{\mathbf{X}}_{\mathrm{e}}^a = \mathcal{D}_a(\mathbf{z}_0^a), \tag{20}$$

where $\mathcal{D}_v$ and $\mathcal{D}_a$ are the pretrained video and audio decoders.

F Dataset Details

F.1 Dataset Statistics

Taxonomy We group targets by sound source rather than by visual object class. The ten categories follow the top level of the AudioSet (Gemmeke et al., 2017) ontology: Music & Instruments, Animals, Vehicles & Transport, Human (voice & activity), Machines & Tools, Nature & Elements, Alarms, Bells & Signals, Objects & Household, Impacts, Explosions & Weapons, and a residual Other / Unclassified bucket. The target labels come from an audio-separation pipeline, so a name such as 'power tool' or 'emergency vehicle' describes an acoustic event as much as a thing on screen, which makes an acoustic taxonomy the natural axis.

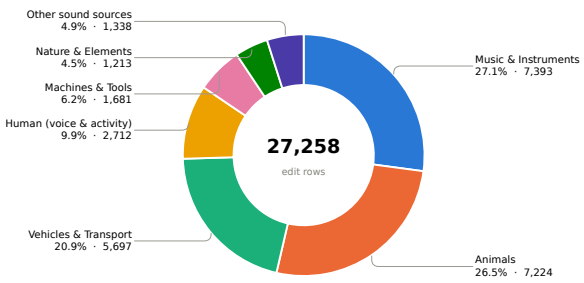


Figure 5: Distribution of target object categories.

Object Distribution Every sample specifies one target object to add or remove, given as a short free-text name (1,878 distinct names over 27,258 samples). In Fig. 5, we show the category distribution. The set is dominated by three sources: music and instruments (27.1%), animals (26.5%) and vehicles (20.9%) make up 74.5% of samples, followed by human voice and activity (10.0%), machines and tools (6.2%) and natural sounds (4.5%); alarms, household objects and impacts together account for the remaining 3.7%.

F.2 Dataset Pipeline Analysis

Table 4 reports stage-wise and cumulative retention rates. Visual removal verification produces the largest reduction, retaining 46.0% of the scored clips.

Table 4: Dataset retention across filtering stages. In and Out denote clip counts. Kept is stage-wise retention; Cum. is retention relative to the 114,544 initial clips. Bounds reflect incomplete intermediate records.

Stage	In	Out	Kept	Cum.
Visual filtering				
Visibility[†]	114,544	$\geq$ 112,534	$\geq$ 98.2%	$\geq$ 98.2%
Inpainting	$\geq$ 105,333	87,765	$\leq$ 83.3%	76.6%
Removal check[‡]	82,608	38,031	46.0%	33.2%
Audio filtering				
Complete pair	38,031	35,669	93.8%	31.1%
Audible target	35,669	30,473	85.4%	26.6%
Object consistency	30,473	28,554	93.7%	24.9%
Separation quality	28,554	27,461	96.2%	24.0%
Frame-rate check	27,461	27,258	99.3%	23.8%

[†]Bounds are recovered from job logs because the filter's input lists were not retained.
[‡]An additional 5,157 inpainted clips could not be scored and were discarded before removal verification.

Visual filtering. We discard samples in which the target object is absent from the first frame, as determined by SAM3 (Carion et al., 2026). We then retain clips with available EffectErase (Fu et al., 2026) outputs and verify removal by requiring the post-edit SAM3 mask area to decrease by at least 75% relative to the source.

Audio filtering. We first require a complete source, inpainted video, separated target, and residual. We discard targets with RMS amplitude below 0.005. To maintain audiovisual object consistency, separation and inpainting must use the same object mask; otherwise, we repeat inpainting. Finally, we require the target's ImageBind (Girdhar et al., 2023) similarity to the mask-conditioned video to exceed the residual's by more than 0.05, with both stems having RMS amplitude of at least 0.005.

Frame-rate filtering. We retain source clips at or below 37.6 fps so that 189 native frames cover the five-second window.

F.3 Agentic Data Engine

Our data engine uses LangGraph to orchestrate pretrained models as modular, replaceable processing nodes. Qwen3-Omni (Xu et al., 2025) first generates audiovisual descriptions, and GPT-4o-mini (OpenAI, 2024) identifies sounding objects. SAM3 (Carion et al., 2026) then segments the target object, EffectErase (Fu et al., 2026)—built on Wan2.1 Wan et al. (2025) with LoRA (Hu et al., 2021)—removes it from the video, and SAM3 checks the visual removal result. For audio, SAM-Audio-large (Shi et al., 2025) produces ten separation candidates using visual and textual conditioning across five random seeds, which ImageBind Girdhar et al. (2023) ranks for selection. Finally, LTX-2.3 22B (HaCohen et al., 2026) jointly refines the edited video and residual audio through audiovisual denoising. Conditional routing rejects samples that fail intermediate checks, while isolated model workers allow individual components to be replaced or extended. The engine supports scaling through data sharding across GPU jobs and, when checkpointing is enabled, reuses upstream results to avoid rerunning the entire pipeline after local changes. An accompanying agentic optimization framework uses stage-level metrics and failure records to investigate bottlenecks and evaluate modifications under budget checks, experiment tracking, and human review. Together, these components provide a reusable foundation for extending the current object-removal pipeline to broader tasks such as joint audiovisual stylization. Upon acceptance, we will publicly release our data engine to support further dataset construction and processing across audiovisual tasks.

G Limitations and Future Directions

Interpreting LLM-based evaluation. Gemini 3.1 Pro (Google DeepMind, 2026) assigns an AES of 11.10% to the Oracle–Oracle reference for the removal task. The oracle reference consists of pseudo-ground-truth content and does not guarantee complete target-sound removal. Residual sounds, ambiguity between overlapping environmental sources, and errors in the judge may contribute to the low score, although their individual effects cannot be determined from aggregate results. We therefore complement LLM judgments with reconstruction, perceptual, distributional, and synchronization metrics. We also provide extensive video and audio examples on our project page, allowing readers to directly assess target removal, preservation of surrounding content, and audiovisual consistency.

Imperfect pseudo-ground-truth supervision. Despite automated filtering and joint refinement, pseudo-ground-truth targets may contain incomplete removals or editing artifacts. Reference-based metrics therefore measure agreement with imperfect targets and may not fully capture editing quality. Qualitative examples on our project page suggest that AVIO can sometimes remove targets more completely than the corresponding pseudo-targets while maintaining coherent visual and acoustic content. A possible explanation is that the pretrained text-to-audiovisual model supplies generative priors that help refine imperfect supervision rather than simply reproduce its artifacts. This interpretation remains a hypothesis, and the observed examples do not establish a systematic advantage over the pseudo-ground-truth targets.

Future data expansion. AVIOBench draws on existing public datasets and inherits their limitations in recording quality, coverage, and diversity. Our data engine provides a reusable workflow for extending paired supervision to newer, higher-quality, and more diverse audiovisual recordings. Combining improved raw data with stronger editing and verification models could improve target quality and broaden the range of objects, sounds, and environments represented. AVIOBench and the data engine thus provide a foundation for expanding toward more diverse and controllable audiovisual editing.