REASONAVEDIT: EXPLICIT MULTIMODAL REASONING FOR INSTRUCTION-GUIDED AUDIO-VISUAL EDITING

Anonymous authors
Paper under double-blind review



Figure 1: **Joint audio-visual editing results of ReasonAVEdit.** The model takes the source audio-visual input, one instruction, and an optional reference image. It needs no user-provided mask or target audio. Seven examples cover long videos, single-person and multi-character scenes, local and full-clip edits of lines and lip motion with the appearance kept, singing, and sounding objects.

ABSTRACT

Existing audio-visual editing methods face a key trade-off between precise control and a unified instruction interface. Explicitly conditioned methods provide precise control, but they rely on task-specific priors such as spatial masks and target audio. Instruction-guided methods need no extra control, but they lack an explicit multimodal reasoning representation and therefore localize edit targets imprecisely. To resolve this tension, we propose ReasonAVEdit, a Perceive–Reason–Edit framework. ReasonAVEdit requires an audio-visual diffusion model to predict multimodal Reasoning Tokens while generating the target audio-visual tokens. The Reasoning Tokens represent the visual edit region, the acoustic target, and the edit time span shared by both modalities. The model thus forms a complete editing decision on its own, without user-provided masks or target audio. The Temporal Query Bridge (TQB) passes the edit time span from the acoustic Reasoning Tokens to the video Temporal Queries, so the edited lips follow only the target's own speech. The updated queries and a single-frame visual region jointly guide video editing, so video-side reasoning no longer grows multiplicatively with video length and resolution. We build ReasonAVEdit-Data with 200K samples and ReasonAVEdit-Bench with 1,200 samples. Across 33 metrics on our benchmark and two public benchmarks, ReasonAVEdit achieves the best result on 32. Relative to the strongest baseline on each metric, it improves audio-visual alignment by 57.4% and reduces the audio-visual temporal offset by 44.5%. Single-frame visual reasoning also reduces inference latency by 51.4% compared with a full-length mask sequence.

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.

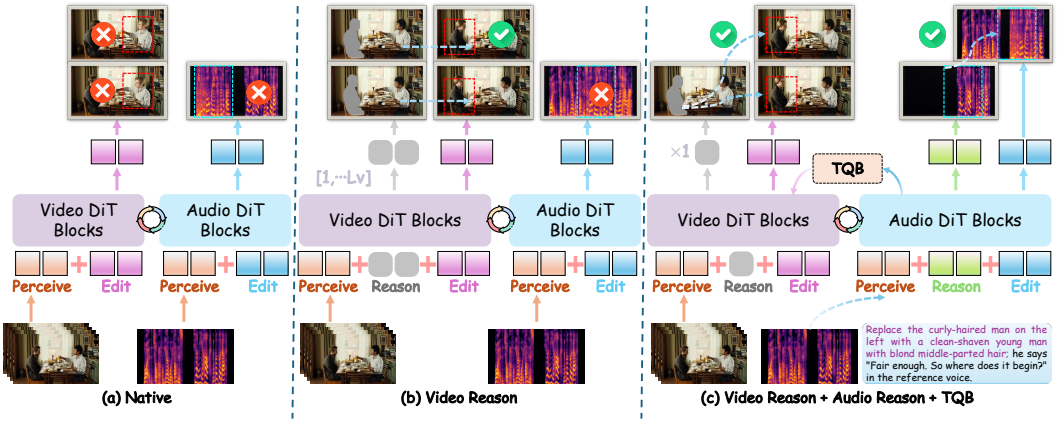

Figure 2: **Motivation of ReasonAVEdit.** (a) Native editing without reasoning. (b) Video-only reasoning with per-frame masks. (c) Single-frame visual reasoning, acoustic reasoning, and TQB.

1 Introduction

Diffusion-based video editing has made remarkable progress in modifying complex visual content (Qin et al., 2024; Jiang et al., 2025; Yang et al., 2025). Audio editing now also supports changes to speech content and sound events (Wang et al., 2023; Han et al., 2023; Liang et al., 2025; Lan et al., 2025; Tian et al., 2026). These capabilities show great potential in content creation and visual effects. However, existing video and audio editors usually handle their own modality in isolation. This limitation motivates joint audio-visual editing. Its goal is to modify a user-specified object in both modalities and to produce a coherent audio-visual result. To achieve this goal, a model must determine the visual region to modify, the acoustic target, and the edit time span shared by both modalities.

Existing joint audio-visual editing methods face a key trade-off between precise control and a unified instruction interface. Explicitly conditioned methods, represented by SpongeBob (Liang et al., 2026), specify the edit region with a target subject mask. They also use separated base audio to supply the non-target ambient sound. These inputs give joint editing a clear spatial region and acoustic context. However, a user or an external tool must first segment the target and separate the audio. In contrast, instruction-guided methods such as InstructAV2AV (Zheng et al., 2026b) take only a source audio-visual sample and a text instruction. They jointly modify the specified visual target and its soundtrack. They also preserve the background, unrelated objects, and non-target ambient sound. These methods need no user-provided spatial mask. However, they leave target selection implicit in end-to-end generation. They lack explicit reasoning about the visual region, the acoustic target, and the shared edit time span, and they place no explicit constraint on them. As a result, they map instructions to edit targets weakly and localize the targets imprecisely. In short, explicitly conditioned methods are precise but depend on task-specific priors. Instruction-guided methods are unified but lack explicit target localization. *Can natural-language instructions alone drive diverse audio-visual edits with the localization precision of explicitly conditioned methods?*

To answer this question, we propose ReasonAVEdit, a joint audio-visual editing framework in the Perceive–Reason–Edit paradigm. Instead of mapping inputs directly to edited outputs, ReasonAVEdit predicts multimodal Reasoning Tokens from the source audio-visual input and the text instruction. These tokens determine the visual region, the acoustic target, and the shared time span to edit. They guide precise audio-visual editing of the selected target in the same denoising process. The model predicts its editing decisions on its own. Users therefore need not provide masks, target audio, or separated acoustic context. Figure 1 shows editing results on long videos, single-person and multi-character scenes, local and full-clip edits, singing, and general sounding objects. All of these results are driven only by natural-language instructions.

As shown in Figure 2(b), a straightforward solution uses only spatial reasoning. The visual branch predicts a target mask for every frame, and the audio branch has no explicit reasoning. This design has two limitations. First, the video branch infers the edit time span without knowing which sound in the mixture belongs to the target. The speech of another speaker may then also drive the mouth of the edited person. Second, per-frame masks form a dense spatio-temporal representation. The cost of visual reasoning then grows multiplicatively with video length and per-frame spatial resolution.

To this end, as shown in Figure 2(c), ReasonAVEdit factorizes the editing decision by the modality in which the evidence lies. Single-frame visual Reasoning Tokens localize the visual region. Acoustic Reasoning Tokens over the full timeline identify the target sound source and determine the shared edit time span. The Temporal Query Bridge (TQB) then passes this edit time span to the video Temporal Queries. Together with the single-frame visual localization, the updated queries guide video editing. Even when the whole appearance is edited, the lips follow only the target's own speech, which suppresses cross-speaker lip-sync leakage. It also turns visual reasoning from a dense spatio-temporal representation into an additive spatial and temporal representation, so its cost no longer grows multiplicatively with video length.

Predicting Reasoning Tokens explicitly requires intermediate supervision beyond conventional source-target media pairs. We therefore build ReasonAVEdit-Data with 200K samples. It provides joint annotations of the visual region, the acoustic target, and the shared edit time span. We also build ReasonAVEdit-Bench with 1,200 samples. It evaluates final editing quality, multimodal reasoning accuracy, cross-modal temporal consistency, and computational efficiency. We further test cross-benchmark generalization on InsAVE-Eval (Zheng et al., 2026b) and JAVEditBench (Chen et al., 2026). JAVEditBench follows an evaluation protocol defined entirely by a third party. Across 33 metrics on the three benchmarks, ReasonAVEdit achieves the best result on 32. Relative to the strongest baseline on each metric, it improves audio-visual alignment by 57.4% and reduces the audio-visual temporal offset by 44.5%. It uses only $1/L_v$ of the visual reasoning tokens of a full-length mask sequence and reduces inference latency by 51.4%.

Our contributions are summarized as follows.

- We propose ReasonAVEdit. Its Perceive–Reason–Edit framework explicitly predicts multimodal Reasoning Tokens while generating the target audio and video. It keeps a unified instruction interface while forming visual, acoustic, and temporal editing decisions on its own.
- We propose a modality-factorized reasoning representation and TQB. Single-frame visual reasoning handles spatial localization. Full-timeline acoustic reasoning handles the acoustic target and the shared edit time span. TQB passes this edit time span to video editing.
- We build ReasonAVEdit-Data with 200K samples and ReasonAVEdit-Bench with 1,200 samples. They support joint evaluation of final edits, multimodal reasoning, and computational efficiency.

2 Related Work

Instruction-only joint audio-visual editing. These methods take only the source audio-visual input and a natural-language instruction. InstructAV2AV (Zheng et al., 2026b) builds InsAVE-80K. It adapts a dual-stream generator with source-instruction gated attention and two-stage training. JAVEdit (Chen et al., 2026) builds JAVEdit-100k, which contains about 103K human-centric editing triplets, and JAVEditBench, which contains 150 source videos. It fine-tunes LTX-2.3 on JAVEdit-100k with LoRA to obtain a joint audio-visual editing baseline. Both methods map the source audio-visual input and the text instruction directly to the edited result. They offer a unified natural-language interface. However, they do not explicitly output the visual region, the acoustic target, or the shared edit time span. Appendix H reviews joint audio-visual generation and explicitly conditioned editing.

3 Method

To address these issues, we propose ReasonAVEdit, a joint audio-visual editing framework in the Perceive–Reason–Edit paradigm. We first define the problem (Section 3.1). We then introduce modality-factorized reasoning. Single-frame visual Reasoning Tokens localize the visual region. Full-timeline acoustic Reasoning Tokens identify the target sound source and determine the shared edit time span (Section 3.2). Next, TQB passes this edit time span to the video Temporal Queries. Together with the single-frame visual localization, the updated queries guide video editing (Section 3.3). Finally, we describe the dual-stream architecture and causal visibility (Section 3.4), as well as the training and sampling methods (Section 3.5).

3.1 Problem Formulation

Given a source video V_s, a source audio A_s, and a text instruction T, the task is to generate an edited video V_e and an edited audio A_e. The two outputs must stay synchronized. Content outside the request of the instruction must remain unchanged. The model additionally receives an optional

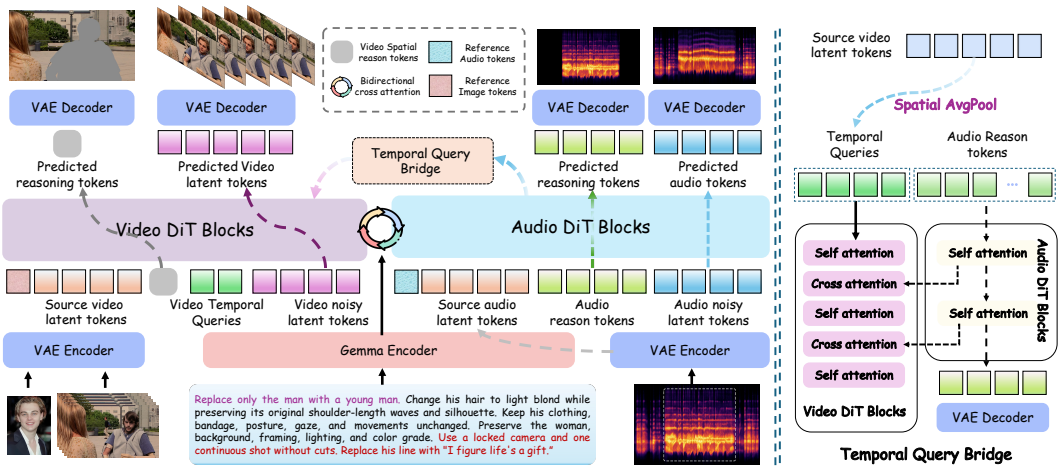

Figure 3: **Overview of ReasonAVEdit.** Left: the dual-stream DiT predicts Reasoning Tokens and edits the video and audio. Right: TQB passes the edit time span to the Temporal Queries.

reference image I only when the target appearance must be specified. We disable this optional condition in all main experiments except the reference-guided editing category.

This task requires the model to determine three items. They are the visual region R_v to modify, the acoustic target R_a to replace, and the edit time span R_t shared by both modalities. Explicitly conditioned methods require users to provide them. Instruction-only methods leave them implicit in end-to-end generation. ReasonAVEdit instead externalizes all three as decodable multimodal Reasoning Tokens. It generates them jointly with the final media:

$$(R_v, R_a, R_t, V_e, A_e) = F_\theta(V_s, A_s, T; I). \tag{1}$$

3.2 Modality-Factorized Reasoning

R_v can be determined only in the frames. R_a can be distinguished only in the soundtrack. R_t is shared by both. As shown in Figure 2(b), a straightforward solution uses only spatial reasoning. The visual branch predicts a target mask for every frame, which also implicitly determines R_t. Without acoustic reasoning, it cannot tell which sound in the mixture belongs to the target, so the speech of another speaker may drive the mouth of the edited person. Per-frame masks are also a dense spatio-temporal representation. The cost of visual reasoning thus grows multiplicatively with video length and spatial resolution. This computational demand is unacceptable.

We therefore factorize the editing decision by the modality in which the evidence lies. The visual side predicts R_v only on the first frame of the edit time span given by acoustic reasoning. This prediction locks the identity and the spatial location of the visible target. It does not predict the duration of the visual edit. The acoustic side predicts R_a and R_t together over the full timeline. The acoustic side carries the edit time span because the onset and offset of an event are often ambiguous in the frames. Gradual mouth movements, occluded targets, and off-screen sources all make visual boundaries hard to define. In contrast, the onset and offset in the soundtrack are clear. The edit time span thus comes only from the target sound. The visual side also no longer uses a dense spatio-temporal representation.

Single-frame visual Reasoning Tokens. Visual reasoning operates only on the first frame F_s of the edit time span given by acoustic reasoning. Our model marks the target subject region on this frame by covering it with a gray mask. It outputs an RGB image R_v at the resolution of the source frame. This single-frame mask locks the target identity and spatial location. It does not limit the duration of the visual edit. If no visible target exists in this frame, the model outputs a Null Anchor. The Null Anchor indicates an off-screen acoustic target. The video branch then performs no corresponding target edit.

Full-timeline acoustic Reasoning Tokens. The acoustic side outputs a waveform of the same length as A_s. The waveform is silent outside the edit interval. Inside the interval, it keeps only the source sound to be replaced. For example, suppose the instruction replaces "eating" with "swimming" in a sentence. The waveform keeps "eating" at its original timestamps and is silent elsewhere. It does not contain the new word "swimming". As another example, consider a dialogue between a man and a woman in which the instruction rewrites only the speech of the man. The waveform keeps the original speech of the man and zeroes out the voice of the woman and all other content.

Even where both speak at once, only the voice of the man remains. This waveform therefore carries two decisions. Its content gives R_a, and its non-silent interval gives R_t. The same set of acoustic Reasoning Tokens carries both. For speech edits, R_t is the speech interval of the target speaker.

3.3 Temporal Query Bridge (TQB)

The acoustic Reasoning Tokens give the shared edit time span over the full timeline. This span marks when the target's sound and its synchronized motion, such as lip movement, change together, and it does not bound appearance edits, which cover the whole clip. TQB passes it to the video Temporal Queries, so the lips follow only the target's own speech even when the appearance changes throughout the clip. With the single-frame visual localization, the updated queries guide video editing.

Let $z_{v,s} \in \mathbb{R}^{L_v \times h_v \times w_v \times d_v}$ denote the spatio-temporal token latent of the source video. Here L_v, h_v, and w_v are the temporal, height, and width sizes, and d_v is the feature dimension. We average-pool the spatial dimensions at each temporal position to obtain the Temporal Queries q_τ. We denote the acoustic Reasoning Token sequence by r_a, where L_a and d_a are its length and feature dimension:

$$q_\tau = \mathrm{AvgPool}_{h_v, w_v}(z_{v,s}) \in \mathbb{R}^{L_v \times d_v}, \qquad r_a \in \mathbb{R}^{L_a \times d_a}. \tag{2}$$

Spatial average pooling keeps the temporal dimension unchanged. The i-th token of q_τ is the mean of all spatial tokens of the source video at the i-th temporal position. It therefore carries no spatial index. Then q_τ acts as the query that reads the acoustic Reasoning Token sequence r_a, which yields $\tilde{q}_\tau$. This update passes the edit time span determined by acoustic reasoning to the video side. The Video Edit Tokens use it in subsequent self-attention:

$$\tilde{q}_\tau = q_\tau + g \cdot \mathrm{CrossAttn}(q_\tau, r_a), \qquad g_{\text{init}} = 0. \tag{3}$$

Here g is a learnable gate initialized to zero.

3.4 Dual-Stream Architecture and Causal Visibility

The above representations follow the Perceive–Reason–Edit paradigm proposed in this paper. In the information flow, the Reasoning Tokens first form the editing decision. They then guide the generation of the Edit Tokens. They do not explain an already generated result after the fact.

A video encoder converts the source video V_s and the reference image I into s_v and i_v. We denote the visual Reasoning Tokens by r_v and the Video Edit Token sequence of the edited video V_e by e_v. The video stream concatenates i_v, s_v, r_v, q_τ, and e_v into the video input sequence x_v:

$$x_v = [\, i_v,\ s_v,\ r_v,\ q_\tau,\ e_v \,]. \tag{4}$$

An audio encoder converts the source audio A_s into s_a, and an optional reference audio clip for the target voice into tokens placed before s_a. We denote the Audio Edit Token sequence of A_e by e_a. The audio stream concatenates s_a, r_a, and e_a into the audio input sequence x_a:

$$x_a = [\, s_a,\ r_a,\ e_a \,]. \tag{5}$$

To prevent generated edit states from leaking into reasoning, we set a directed visibility in the attention modules. Information flows from the given conditions to Reasoning and from Reasoning to Edit. The reference and source tokens i_v, s_v, and s_a only provide conditions. They do not read any Reasoning or Edit state under generation. The Reasoning Tokens can read the given conditions and existing Reasoning states. They cannot read any Edit state. The Edit Tokens can read the given conditions and the Reasoning states.

In the video stream, e_v can read r_v and $\tilde{q}_\tau$. In the audio stream, e_a can read r_a. TQB lets q_τ read r_a. Bidirectional cross-modal interaction lets e_v and e_a exchange current edit states. The reverse path from Edit Tokens to Reasoning Tokens is always masked. It only fixes the reading direction and needs no separate sampling passes. Appendix A.4 gives the full visibility matrix.

3.5 Training and Sampling

We train the network with a weighted sum of four flow matching losses, one for each Reasoning and Edit Token sequence of the two streams (Appendix A.1). The Temporal Queries receive no explicit supervision, and TQB is learned end to end through the flow losses, without a dedicated loss (Appendix A.2). We first train video and audio reasoning separately and then train both branches jointly (Appendix A.3). At inference, reasoning-conditioned joint denoising updates the Reasoning and Edit Tokens together along one flow trajectory, and it recomputes the TQB output from the current acoustic reasoning state at each step (Appendix A.5).

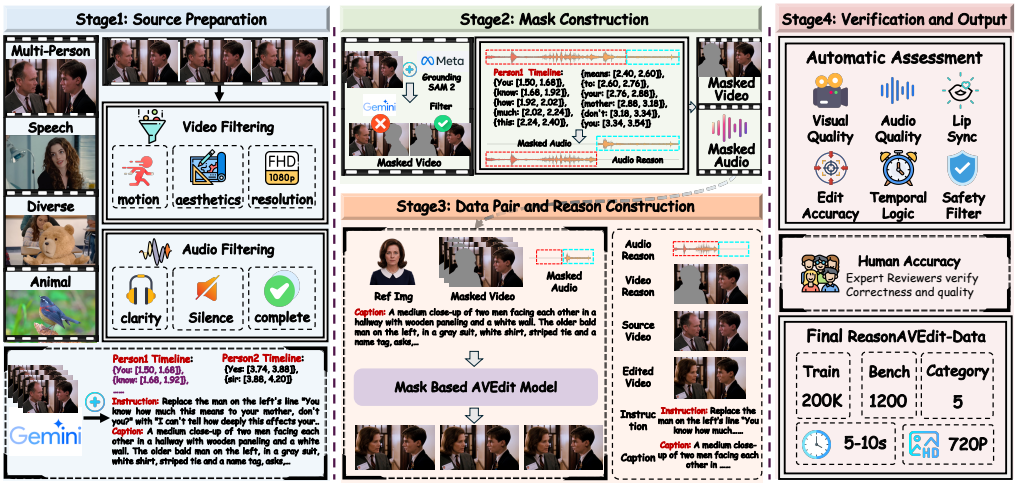

Figure 4: **Construction pipeline of ReasonAVEdit-Data.** The same set of masks is used both to generate the target audio and video and to produce the aligned Reasoning supervision.

4 REASONAVEDIT-DATA

Explicit Reasoning Tokens require training data that describe both the final edit and the intermediate editing decisions. We therefore build ReasonAVEdit-Data with 200K samples. Each sample contains the source audio and video (V_s, A_s), a text instruction T, the target audio and video (V_e, A_e), the visual and acoustic Reasoning targets $(R_v^\star, R_a^\star)$, and an optional reference image I. As shown in Figure 4, construction has four stages: source media preparation, mask construction, edit pair and Reasoning supervision construction, and quality control and splitting. Appendix E details each stage.

The source clips come from the open-source OpenHumanVid (Li et al., 2025) and Koala-36M (Wang et al., 2025). They cover human speech, multi-person interaction, animals, and general sound events. We train a Mask-based AVEdit model that generates synchronized target audio and video under a video mask and an audio mask (Appendix F). This model is used only for data construction, and ReasonAVEdit needs no mask or target audio at inference.

Reasoning supervision. The same masks become the Reasoning supervision. $R_v^\star$ is rendered from the first frame of the edit time span and its subject mask. The Null Anchor replaces it when the target is off-screen. $R_a^\star$ has the same length as A_s. Within the audio mask, it keeps the source sound to be replaced, separated by SAM Audio (Shi et al., 2025), and it is zero elsewhere. Edit targets and Reasoning supervision are thus consistent by construction.

Quality control and splitting. We filter candidates by instruction fidelity, audio-visual quality, non-target preservation, audio-visual synchronization, and Reasoning reliability, and we manually verify 4,000 samples. Before target generation, we deduplicate and split the data by source video, speaker, subject identity, and near-duplicate clips. The training and validation sets contain 200,000 and 2,000 samples. A disjoint set of 1,200 test samples forms ReasonAVEdit-Bench.

5 EXPERIMENTS

5.1 IMPLEMENTATION DETAILS

We adopt LTX-2.3-22B (HaCohen et al., 2026) as the base audio-visual generation model. We train all of its parameters on ReasonAVEdit-Data from Section 4. Training uses 5–10 s clips at 720p resolution. The audio side follows the LTX-2.3 configuration, with a 16 kHz input sampling rate and a 48 kHz output waveform. Following Section 3.5, we first train video and audio reasoning separately for 5K steps each. Both branches use a global batch size of 128 and a learning rate of 1×10^{-5}. We then train both branches jointly for 10K steps with a global batch size of 256 and the same learning rate. Appendix B gives the inference and remaining configurations.

5.2 EXPERIMENTAL SETUP

Metrics. ReasonAVEdit-Bench and InsAVE-Eval use the same video, audio, and audio-visual metrics, and JAVEditBench uses its six official metrics. Appendix C.1 defines each metric and its computation, and Appendix C.3 gives the diagnostic protocol for explicit reasoning.

Table 1: **Joint audio-visual editing results on ReasonAVEdit-Bench and InsAVE-Eval.** All values are recomputed with the same evaluator. We do not quote the reported results of InstructAV2AV. InsAVE-Eval has no edit masks, so PSNR is omitted (–) and SSIM uses the whole frame. Bold marks the best result in each column of each benchmark.

Method	Video			Background		Audio				Audio-Visual				
	FVD↓	TV-A↑	TC↑	PSNR↑	SSIM↑	FAD↓	TA-A↑	LPAPS↓	PQ↑	AV-A↑	PEAVS↑	Sync-C↑	Sync-D↓	DeSync↓
ReasonAVEdit-Bench														
AvED	267.90	0.180	0.961	18.0	0.707	1.949	0.385	6.014	5.505	0.152	2.001	0.604	12.600	0.808
CoherentAVEdit	**22.94**	0.175	0.990	14.9	0.573	1.479	0.383	5.784	5.322	0.162	2.811	1.169	11.200	0.656
AVI-Edit	210.50	0.187	0.991	24.6	0.801	2.173	0.331	5.084	6.481	0.145	2.211	3.324	9.951	0.382
InstructAV2AV	210.90	0.184	0.990	22.1	0.778	1.075	0.311	5.489	6.140	0.202	3.213	2.200	10.300	0.420
ReasonAVEdit	49.98	**0.193**	**0.994**	**26.3**	**0.830**	**0.770**	**0.400**	**4.329**	**6.800**	**0.318**	**3.217**	**3.713**	**9.649**	**0.212**
InsAVE-Eval														
AvED	38.500	0.203	0.968	–	0.639	2.257	0.326	4.354	5.427	0.170	1.751	0.637	12.000	0.756
CoherentAVEdit	44.181	0.194	0.992	–	0.727	2.319	0.347	5.922	5.187	0.136	2.572	2.307	10.100	0.361
AVI-Edit	53.180	0.201	**0.994**	–	0.747	2.158	0.316	3.947	6.427	0.144	1.954	2.587	10.200	0.402
InstructAV2AV	43.714	0.195	0.993	–	0.430	1.775	0.353	4.368	5.810	0.198	3.255	3.689	9.231	0.261
ReasonAVEdit	**32.500**	**0.204**	**0.994**	–	**0.750**	**1.546**	**0.369**	**3.840**	**6.934**	**0.300**	**3.475**	**6.299**	**7.563**	**0.251**

Table 2: **Joint audio-visual editing results on JAVEditBench.** We follow its official evaluation protocol. Bold marks the best result in each column.

Method	VTSS↑	UTMOS↑	SyncNet↑	Instruction↑	Fidelity↑	AV quality↑
AvED	0.055	1.214	0.312	3.000	3.275	2.127
CoherentAVEdit	0.059	1.596	0.339	2.507	4.088	3.547
AVI-Edit	0.057	2.540	0.360	3.592	3.493	3.346
InstructAV2AV	0.058	1.321	0.355	2.885	3.803	3.700
ReasonAVEdit	**0.060**	**2.637**	**0.447**	**4.181**	**4.114**	**3.820**

Baselines. We compare with AvED (Lin et al., 2026), CoherentAVEdit (Ishii et al., 2025), AVI-Edit (Zheng et al., 2026a), and InstructAV2AV (Zheng et al., 2026b), all with official weights and default settings. CoherentAVEdit needs an edited video, so we run the cascade VACE-14B (Jiang et al., 2025) + CoherentAVEdit and abbreviate it as CoherentAVEdit. Following the established protocols, VACE-14B and AVI-Edit receive ground-truth instance masks, and AVI-Edit also receives the final target audio. Appendix C.4 details each baseline, and we do not retrain any of them.

Evaluation data. To evaluate both final editing quality and explicit editing decisions, we build ReasonAVEdit-Bench. It contains 1,200 samples from non-overlapping source videos. It is disjoint from the training set in source videos and subject identities. Each sample provides the source and target audio and video, a text instruction, and visual and acoustic Reasoning annotations. We divide it into five disjoint subsets: Where-only, Which-only, When-only, Combined Reasoning, and Reference Editing. The first four subsets form the main evaluation of instruction-only editing. We report Reference Editing separately, and Appendix D.4 gives its results. Appendix C.2 describes the subset composition. Appendix E.3 describes the annotation verification process. We also perform cross-benchmark evaluation on InsAVE-Eval and JAVEditBench.

5.3 Comparison with Existing Methods

Quantitative results. On ReasonAVEdit-Bench (top of Table 1), ReasonAVEdit is best on 13 of the 14 metrics. Relative to the strongest baseline on each metric, it improves AV-A by 57.4% and reduces DeSync by 44.5%. Explicit localization thus improves both instruction following and cross-modal synchronization. ReasonAVEdit also leads in audio quality and background preservation, even though CoherentAVEdit and AVI-Edit receive ground-truth instance masks and ReasonAVEdit does not. The only exception is FVD, where CoherentAVEdit is lowest. FVD measures only the distance to the distribution of real videos. This baseline also has the lowest TV-A and background preservation, so its natural-looking frames neither follow the instruction nor preserve the source. Section 5.4 controls for training data with the same backbone, data, and optimization.

Cross-benchmark evaluation. ReasonAVEdit is best or tied for best on all 13 metrics of InsAVE-Eval (bottom of Table 1; tied with AVI-Edit on TC). It also leads on all six official metrics of JAVEditBench (Table 2). Both benchmarks differ from ours in data sources, edit categories, and metrics, and JAVEditBench uses a third-party protocol. The gains from explicit reasoning therefore do not come from overfitting to our data or metrics.

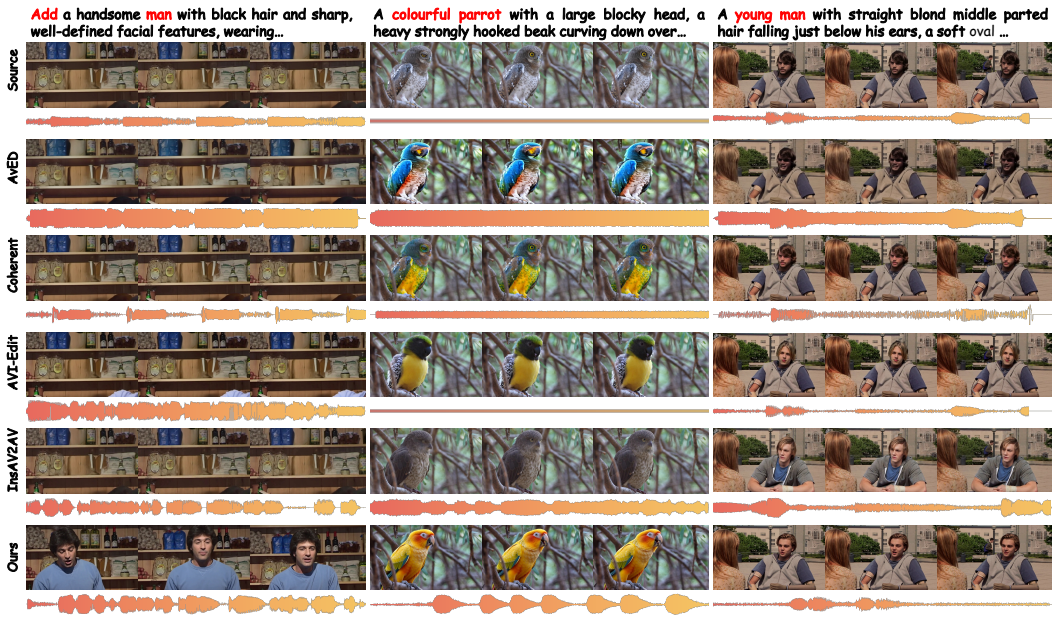

Figure 5: **Qualitative comparison with existing methods.** The columns show instance addition, appearance replacement of a sounding object, and human appearance editing, with the instructions on top. Each row shows one method, with frames above the audio waveform. The first row is the source. Coherent and InsAV2AV denote CoherentAVEdit and InstructAV2AV.

Table 3: **Ablation of Reasoning Tokens and TQB.** ✓/✗: on/off. GT: benchmark annotations at inference. PSNR/SSIM: background. Bold: best excluding GT.

r_v	r_a	TQB	TV-A↑	TA-A↑	LPAPS↓	AV-A↑	PEAVS↑	PSNR↑	SSIM↑
✗	✗	✗	0.185	0.341	5.216	0.214	3.021	22.7	0.781
✓	✗	✗	0.190	0.345	5.148	0.243	3.062	25.4	0.818
✗	✓	✗	0.186	0.389	4.573	0.256	3.108	22.9	0.784
✓	✓	✗	0.191	0.394	4.412	0.291	3.145	25.8	0.822
✓	✓	✓	**0.193**	**0.400**	**4.329**	**0.318**	**3.217**	**26.3**	**0.830**
GT	GT	✓	0.206	0.427	3.914	0.361	3.418	28.1	0.857

Table 4: **Acoustic reasoning representations.** T+Sil./T+Noise/Base: target + silence, target + noise, or source audio outside the span.

Repr.	TA-A↑	LPAPS↓	V-T IoU↑	PEAVS↑
T+Sil.	**0.400**	4.329	**0.763**	**3.217**
T+Noise	0.381	4.688	0.742	3.154
Base	0.362	**4.212**	0.681	3.049

Qualitative results. Figure 5 shows three representative cases. AvED and AVI-Edit often edit beyond the target and alter the background. The cascaded CoherentAVEdit rewrites the audio, but its waveform follows the visual events weakly. InstructAV2AV finds roughly the right subject but misses fine-grained attributes. ReasonAVEdit confines each edit to the instructed target, and its waveform onsets and offsets align with the visual changes.

5.4 Ablation of Core Components

Table 3 adds the visual Reasoning Tokens, the acoustic Reasoning Tokens, and TQB in turn under the same backbone, data, and optimization. The first row predicts no Reasoning Tokens, which also controls for training data in Section 5.3. The last row replaces the predictions with benchmark annotations at inference. Appendix D.1 gives the full Reasoning diagnostics.

Visual Reasoning Tokens mainly improve visual alignment and background preservation. Acoustic Reasoning Tokens mainly improve audio alignment and fidelity. Each kind thus acts on its own modality. Using both further improves AV-A and PEAVS. Adding TQB passes the shared edit time span to the video side and further improves PEAVS. Benchmark annotations still bring gains, so more accurate reasoning would help further.

5.5 Analysis

Representation of acoustic reasoning. Table 4 compares three supervision targets with the same edit time span. *T+Sil.* keeps the target voice within the span and silences the rest. *T+Noise* replaces the silence with Gaussian noise of equal RMS. *Base* only removes the target voice within the span



Figure 6: **Qualitative ablation of TQB.** Rows show the source, w/o TQB, and the full model, with frames above the audio waveform. The cyan boxes mark the speaker on the left, and the red boxes mark the target on the right, who should not speak at that moment.

Table 5: **Visual reasoning frames: performance and efficiency.** Mask sequences keep the first L_v, $L_v/2$, or $L_v/4$ frames. The last row is the single frame of ReasonAVEdit. PSNR: background. Lat./Mem.: per-sample inference latency and peak GPU memory at 736×1280.

Frames	TV-A↑	PSNR↑	PEAVS↑	Tokens↓	Lat.↓	Mem.↓
L_v	0.192	26.1	3.198	$L_v h_v w_v$	379.4 s	57.4 GB
$L_v/2$	0.190	25.8	3.176	$L_v h_v w_v/2$	285.2 s	47.7 GB
$L_v/4$	0.187	25.2	3.141	$L_v h_v w_v/4$	232.7 s	46.7 GB
1	**0.193**	**26.3**	**3.217**	$h_v w_v$	**184.2 s**	**44.7 GB**

from the source audio. *T+Sil.* is best on TA-A, V-T IoU (the temporal IoU of the video edit), and PEAVS. Representing the edit target directly on a silent background is thus easiest for subsequent editing to use. *Base* leads only on LPAPS, because it keeps the source audio outside the span.

Number of visual reasoning frames. Table 5 keeps L_v, $L_v/2$, or $L_v/4$ frames from the front of a full-length mask sequence and compares them with the single frame of ReasonAVEdit. The edit time span always comes from acoustic reasoning through TQB. Fewer mask frames reduce the token count, latency, and peak memory with only a slight performance loss. The single frame has the lowest cost and the best performance. It reduces inference latency by 51.4% compared with the full-length mask sequence. Appendix D.2 gives the remaining results.

Qualitative ablation of TQB. Figure 6 compares the full model with *w/o TQB* in a Multi-Character dialogue. Both use the same source input, instruction, Reasoning Tokens, and edit noise. The instruction replaces only the woman on the right and redubs her lines. The cyan boxes mark the left speaker, and the red boxes mark the target on the right, who should stay silent at these moments. Without TQB, the video side receives no constraint from the target's speech intervals. The left speaker's voice then also drives the target's lips, which we call cross-speaker lip-sync leakage. With TQB, the target's lips move only while the target speaks. The acoustic Reasoning Tokens keep only the target's speech, and TQB passes this target-specific temporal prior to the video side. TQB therefore provides speaker attribution for video editing and suppresses cross-speaker lip-sync leakage.

6 Conclusion

We have presented ReasonAVEdit, a Perceive–Reason–Edit framework for audio-visual editing. The model localizes the visual region with single-frame visual Reasoning Tokens. Full-timeline acoustic Reasoning Tokens determine the acoustic target and the shared edit time span. TQB passes this edit time span to the video side and suppresses cross-speaker lip-sync leakage. The factorization also replaces per-frame visual Reasoning Tokens with a single frame. We also build ReasonAVEdit-Data with 200K samples and ReasonAVEdit-Bench with 1,200 samples. ReasonAVEdit achieves the best result on 32 of the 33 metrics across ReasonAVEdit-Bench, InsAVE-Eval, and JAVEditBench. It compresses the visual reasoning tokens to $1/L_v$ of a full-length mask sequence and reduces inference latency by 51.4%. Appendix G discusses the limitations.

ETHICS STATEMENT

Joint audio-visual editing can lower the barrier to content creation and improve accessibility. The same capability can also produce misleading media. It can also imitate the appearance and voice of a specific person without consent. We address this risk in three aspects.

Data. The source clips of ReasonAVEdit-Data must comply with their source licenses. Samples that involve people must satisfy the conditions for using their likeness and voice. The data split deduplicates by source video, speaker, and subject identity. No identity therefore spans training and testing. The annotators in manual verification receive informed instructions and compensation. The verified content contains no sensitive identities or harmful instructions.

Release. Public examples will avoid identifiable sensitive identities and instructions that may cause harm. Any future release of the data will document source licenses, filtering rules, and a takedown procedure. Any future release of the model will come with terms of use and a recommendation to keep provenance metadata in generated outputs.

Capability boundary. The explicit reasoning design of this paper improves the localization precision of editing. This improves the controllability of legitimate edits. It may also increase the credibility of misused outputs. We believe that release policies and provenance mechanisms should mitigate this risk. Lowering the precision of the method itself is not the right remedy. Any future release will follow this position.

REPRODUCIBILITY STATEMENT

Sections 3.1 to 3.5 define the method. Appendix A gives the full flow matching objective and explains why the Temporal Queries receive no explicit supervision. It also gives the division of labor in two-stage training, the full attention visibility matrix, and the sampling algorithm. Section 5.1 and Appendix B give the training and inference configurations. Table B2 gives the shape of each token sequence. All ablation variants and efficiency comparisons use the same inference configuration in Table B1. Appendix C.4 describes the input conversion and output alignment of the external baselines.

Section 4 describes the data construction pipeline. Appendix E details the four stages, the manual verification process, and the task distribution. Appendix F describes the Mask-based AVEdit model used for data construction and its training configuration. Appendix C gives the computation of the quality and diagnostic metrics. It also gives the subset composition of ReasonAVEdit-Bench and the rules for applying each metric to each subset. For cross-benchmark evaluation, we recompute InsAVE-Eval with the same evaluator as ReasonAVEdit-Bench. JAVEditBench strictly uses its official protocol and implementation without any modification.

AI USE STATEMENT

We used large language models to aid and polish the writing of this paper, including improving grammar, wording, and clarity. We also used them to refine code in the early stage of the project. The authors take full responsibility for all content of this paper.

REFERENCES

Kaito Baba, Wataru Nakata, Yuki Saito, and Hiroshi Saruwatari. The t05 system for the voicemos challenge 2024: Transfer learning from deep image classifier to naturalness mos prediction of high-quality synthetic speech. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, pp. 818–824. IEEE, 2024.

Yinan Chen, Chuming Lin, Zhennan Chen, Yuxiang Zeng, Junwei Zhu, Yali Bi, Xijie Huang, Chengming Xu, Donghao Luo, Zhucun Xue, et al. Javedit: Joint audio-visual instruction-guided video editing with agentic data curation. *arXiv preprint arXiv:2606.03168*, 2026.

Joon Son Chung and Andrew Zisserman. Out of time: automated lip sync in the wild. In *Asian conference on computer vision*, pp. 251–263. Springer, 2016.

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

Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. Clap learning audio concepts from natural language supervision. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

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

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

Lucas Goncalves, Prashant Mathur, Chandrashekhar Lavania, Metehan Cekic, Marcello Federico, and Kyu J Han. Perceptual evaluation of audio-visual synchrony grounded in viewers' opinion scores. In *European Conference on Computer Vision*, pp. 288–305. Springer, 2024.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, et al. Ltx-2: Efficient joint audio-visual foundation model. *arXiv preprint arXiv:2601.03233*, 2026.

Bing Han, Junyu Dai, Weituo Hao, Xinyan He, Dong Guo, Jitong Chen, Yuxuan Wang, Yanmin Qian, and Xuchen Song. Instructme: An instruction guided music edit and remix framework with latent diffusion models. *arXiv preprint arXiv:2308.14360*, 2023.

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

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

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

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17191–17202. IEEE, 2025.

Zitong Lan, Yiduo Hao, and Mingmin Zhao. Guiding audio editing with audio language model. *arXiv preprint arXiv:2509.21625*, 2025.

Hui Li, Mingwang Xu, Yun Zhan, Shan Mu, Jiaye Li, Kaihui Cheng, Yuxuan Chen, Tan Chen, Mao Ye, Jingdong Wang, et al. Openhumanvid: A large-scale high-quality dataset for enhancing human-centric video generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7752–7762. IEEE, 2025.

Jinhua Liang, Yuanzhe Chen, Yi Yuan, Dongya Jia, Xiaobin Zhuang, Zhuo Chen, Yuping Wang, and Yuxuan Wang. Audiomorphix: Training-free audio editing with diffusion probabilistic models. *arXiv preprint arXiv:2505.16076*, 2025.

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. *arXiv preprint arXiv:2605.25193*, 2026.

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

Chetwin Low, Weimin Wang, and Calder Katyal. Ovi: Twin backbone cross-modal fusion for audio-video generation. *arXiv preprint arXiv:2510.01284*, 2025.

Bosheng Qin, Juncheng Li, Siliang Tang, Tat-Seng Chua, and Yueting Zhuang. Instructvid2vid: Controllable video editing with natural language instructions. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE, 2024.

Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks, 2024.

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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10219–10228. IEEE, 2023.

Bowen Shi, Andros Tjandra, John Hoffman, Helin Wang, Yi-Chiao Wu, Luya Gao, Julius Richter, Matt Le, Apoorv Vyas, Sanyuan Chen, et al. Sam audio: Segment anything in audio. *arXiv preprint arXiv:2512.18099*, 2025.

Zeyue Tian, Binxin Yang, Zhaoyang Liu, Jiexuan Zhang, Ruibin Yuan, Hubery Yin, Qifeng Chen, Chen Li, Jing Lyu, Wei Xue, et al. Audio-omni: Extending multi-modal understanding to versatile audio generation and editing. *arXiv preprint arXiv:2604.10708*, 2026.

Andros Tjandra, Yi-Chiao Wu, Baishan Guo, John Hoffman, Brian Ellis, Apoorv Vyas, Bowen Shi, Sanyuan Chen, Matt Le, Nick Zacharov, et al. Meta audiobox aesthetics: Unified automatic quality assessment for speech, music, and sound. *arXiv preprint arXiv:2502.05139*, 2025.

Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Fvd: A new metric for video generation. 2019.

Qiuheng Wang, Yukai Shi, Jiarong Ou, Rui Chen, Ke Lin, Jiahao Wang, Boyuan Jiang, Haotian Yang, Mingwu Zheng, Xin Tao, et al. Koala-36m: A large-scale video dataset improving consistency between fine-grained conditions and video content. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8428–8437. IEEE, 2025.

Yuancheng Wang, Zeqian Ju, Xu Tan, Lei He, Zhizheng Wu, Jiang Bian, et al. Audit: Audio editing by following instructions with latent diffusion models. *Advances in Neural Information Processing Systems*, 36:71340–71357, 2023.

Xiangpeng Yang, Ji Xie, Yiyuan Yang, Yan Huang, Min Xu, and Qiang Wu. Unified video editing with temporal reasoner. *arXiv e-prints*, pp. arXiv–2512, 2025.

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*, pp. 23150–23160, 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 METHOD DETAILS

This section supplements the method details of Section 3. Appendix A.1 gives the full flow matching objective. Appendix A.2 explains why the Temporal Queries receive no explicit supervision. Appendix A.3 describes the division of labor in two-stage training. Appendix A.4 gives the block-level attention visibility. Appendix A.5 gives the full sampling procedure of reasoning-conditioned joint denoising.

A.1 FLOW MATCHING TRAINING OBJECTIVE

We mark supervision targets with a superscript $\star$. The visual and acoustic Reasoning Tokens and the corresponding Edit Tokens are encoded from their supervision targets:

$$r_v^\star = \mathcal{E}_v(R_v^\star), \quad e_v^\star = \mathcal{E}_v(V_e), \quad r_a^\star = \mathcal{E}_a(R_a^\star), \quad e_a^\star = \mathcal{E}_a(A_e). \tag{6}$$

Given a flow time $t \in [0,1]$ and four independent standard Gaussian noises, we construct the noisy Reasoning and Edit states:

$$\begin{aligned} r_{v,t} &= (1-t)\,\epsilon_{v,r} + t\,r_v^\star, & e_{v,t} &= (1-t)\,\epsilon_{v,e} + t\,e_v^\star, \\ r_{a,t} &= (1-t)\,\epsilon_{a,r} + t\,r_a^\star, & e_{a,t} &= (1-t)\,\epsilon_{a,e} + t\,e_a^\star, \end{aligned} \tag{7}$$

where $\epsilon_{v,r}, \epsilon_{v,e}, \epsilon_{a,r}, \epsilon_{a,e} \sim \mathcal{N}(0, \mathbf{I})$ are independent. The corresponding target velocities are

$$u_{v,r}^\star = r_v^\star - \epsilon_{v,r}, \quad u_{v,e}^\star = e_v^\star - \epsilon_{v,e}, \quad u_{a,r}^\star = r_a^\star - \epsilon_{a,r}, \quad u_{a,e}^\star = e_a^\star - \epsilon_{a,e}. \tag{8}$$

Let C denote the text and reference conditions. The dual-stream network jointly predicts four velocity fields:

$$(\hat{u}_{v,r,t}, \hat{u}_{v,e,t}, \hat{u}_{a,r,t}, \hat{u}_{a,e,t}) = v_\theta(t, x_v^t, x_a^t, C). \tag{9}$$

The training objective is a weighted sum of four flow matching losses:

$$\begin{aligned} \mathcal{L} = \mathbb{E}\big[&\lambda_{v,r}\|\hat{u}_{v,r,t} - u_{v,r}^\star\|_2^2 + \lambda_{v,e}\|\hat{u}_{v,e,t} - u_{v,e}^\star\|_2^2 \\ &+ \lambda_{a,r}\|\hat{u}_{a,r,t} - u_{a,r}^\star\|_2^2 + \lambda_{a,e}\|\hat{u}_{a,e,t} - u_{a,e}^\star\|_2^2\big]. \end{aligned} \tag{10}$$

A.2 WHY THE TEMPORAL QUERIES RECEIVE NO EXPLICIT SUPERVISION

The Temporal Queries are not generated outputs. They therefore receive no flow loss and no explicit supervision. TQB has no dedicated loss, and its parameters are optimized end to end through the flow losses. The form of $\tilde{q}_\tau$ emerges under this objective. The edit time span is already supervised through r_a on the acoustic side. Supervising it again here would let the Temporal Queries bypass TQB and predict the span on their own.

A.3 MOTIVATION OF TWO-STAGE TRAINING

Training the visual and acoustic Reasoning Tokens jointly from the start requires the model to learn visual region localization, acoustic target identification, and the shared edit time span at once. This makes joint optimization harder. We therefore adopt a two-stage strategy. Each kind of Reasoning Tokens first learns the editing decisions in its own modality.

Modality-specific reasoning training. The first stage trains the visual and acoustic Reasoning Tokens separately. The video branch learns visual region localization. The audio branch learns the acoustic target and the shared edit time span. Each branch is optimized with its own flow matching objective.

Joint audio-visual training. The second stage initializes from the first-stage parameters and trains the video and audio branches jointly. Both kinds of Reasoning Tokens then take part in the subsequent audio-visual editing within one model.

A.4 ATTENTION VISIBILITY

Table A1 gives the block-level matrix of the visibility rules in Section 3.4. Rows are readers, and columns are the blocks being read. ✓ allows reading, and ✗ masks it. The rules of Section 3.4 apply within and across the two streams. The given conditions read only the given conditions. Reasoning reads the given conditions and Reasoning, and it never reads Edit. Edit reads the given conditions, Reasoning, and Edit. TQB adds two restrictions. The video side reads acoustic reasoning only through TQB, so r_v and e_v do not read r_a directly. r_v reads $\tilde{q}_\tau$ instead, which gives it the edit time span required in Section 3.2. The audio side does not read $\tilde{q}_\tau$, which serves only the video side. The row of $\tilde{q}_\tau$ follows its definition $\tilde{q}_\tau = f_q(q_\tau, r_a)$, where q_τ is spatially average-pooled from s_v.

Table A1: **Block-level attention visibility matrix.** Rows read columns.

Reader	i_v	s_v	r_v	$\tilde{q}_\tau$	e_v	s_a	r_a	e_a
i_v (reference)	✓	✓	✗	✗	✗	✓	✗	✗
s_v (source video)	✓	✓	✗	✗	✗	✓	✗	✗
r_v (visual Reasoning)	✓	✓	✓	✓	✗	✓	✗	✗
$\tilde{q}_\tau$ (TQB output)	✗	✓	✗	✗	✗	✗	✓	✗
e_v (Video Edit)	✓	✓	✓	✓	✓	✓	✗	✓
s_a (source audio)	✓	✓	✗	✗	✗	✓	✗	✗
r_a (acoustic Reasoning)	✓	✓	✓	✗	✗	✓	✓	✗
e_a (Audio Edit)	✓	✓	✓	✗	✓	✓	✓	✓

Algorithm A1 Reasoning-conditioned joint denoising

Input: source video V_s, source audio A_s, instruction T, optional reference image I; number of solver steps N, guidance scale w

$s_v \leftarrow \mathcal{E}_v(V_s),\ i_v \leftarrow \mathcal{E}_v(I),\ s_a \leftarrow \mathcal{E}_a(A_s),\ C \leftarrow (T, I)$
$q_\tau \leftarrow \mathrm{AvgPool}_{h_v, w_v}(s_v)$ ▷ computed once
sample $r_v, e_v, r_a, e_a \sim \mathcal{N}(0, \mathbf{I}),\ \Delta t \leftarrow 1/N$
for $k = 0, 1, \ldots, N-1$ **do**
  $t \leftarrow k\,\Delta t$
  $x_v \leftarrow [\, i_v, s_v, r_v, q_\tau, e_v \,],\ x_v^{\varnothing} \leftarrow [\, s_v, r_v, q_\tau, e_v \,],\ x_a \leftarrow [\, s_a, r_a, e_a \,]$
  $\hat{u} \leftarrow v_\theta(t, x_v, x_a, C),\ \hat{u}_{\varnothing} \leftarrow v_\theta(t, x_v^{\varnothing}, x_a, \varnothing)$ ▷ TQB inside v_θ; visibility of Table A1
  $\hat{u} \leftarrow \hat{u}_{\varnothing} + w\,(\hat{u} - \hat{u}_{\varnothing})$ ▷ classifier-free guidance
  $(r_v, e_v, r_a, e_a) \leftarrow (r_v, e_v, r_a, e_a) + \Delta t\,\hat{u}$ ▷ Euler step
end for
$R_v \leftarrow \mathcal{D}_v(r_v),\ R_a \leftarrow \mathcal{D}_a(r_a),\ V_e \leftarrow \mathcal{D}_v(e_v),\ A_e \leftarrow \mathcal{D}_a(e_a)$

Output: edited result (V_e, A_e) and decodable Reasoning (R_v, R_a)

A.5 Sampling Procedure of Reasoning-Conditioned Joint Denoising

Algorithm A1 gives the full procedure of reasoning-conditioned joint denoising in Section 3.5. Inference uses a single flow trajectory. Reasoning and Edit are integrated synchronously in the same solver steps. The Temporal Queries are not integration variables. At each step, TQB inside the video DiT blocks recomputes the updated queries from the current acoustic reasoning state. The flow time follows Appendix A.1, where $t = 0$ is noise and $t = 1$ is data. Let s, r, and e denote the source, Reasoning, and Edit tokens of both streams. The visibility rules give the following dependencies:

$$\dot{r} = f_r(s, r, t, C), \qquad \tilde{q}_\tau = f_q(q_\tau, r_a), \qquad \dot{e} = f_e(s, r, \tilde{q}_\tau, e, t, C). \tag{11}$$

Reasoning therefore lies upstream in the information flow. Yet it is updated synchronously with editing in the numerical integration. We call this process reasoning-conditioned joint denoising. In Algorithm A1, $\mathcal{D}_v$ and $\mathcal{D}_a$ are the decoders paired with $\mathcal{E}_v$ and $\mathcal{E}_a$, and s_v is $z_{v,s}$ in Section 3.3.

B Training and Inference Configuration

This section summarizes the implementation settings not detailed in Section 5.1. The external baselines use their official default configurations (Appendix C.4).

B.1 Configuration Summary

Table B1 lists the training and inference configuration. Section 5.1 already gives the batch size, the learning rate, and the training steps, so the table does not repeat them. All ablation variants and efficiency comparisons use the same inference configuration in this table.

B.2 Tensor Shapes and Token Counts

Table B2 lists the shape and length of each token sequence for a 5 s input at 720p. It is used to derive the Tokens column of Table 5 and the compression ratio of visual reasoning tokens in the introduction and the conclusion. For longer clips, the token count of a full-length mask sequence grows with the duration. The token count of single-frame visual reasoning stays constant. The video values assume 121 frames at 25 FPS and a 736×1280 input (height $\times$ width), with $8\times$ temporal and $32\times$ spatial compression in the video VAE. The audio lengths follow the temporal resolution of the audio VAE. Under these values, single-frame visual reasoning uses 1/16 of the tokens of a full-length mask sequence, which is the $1/L_v$ in the main text.

Table B1: **Training and inference configuration.** Latency and peak memory are measured at 736×1280.

Item	Stage 1	Stage 2	Inference
Optimizer	AdamW (β_1=0.9, β_2=0.95)	AdamW (β_1=0.9, β_2=0.95)	–
Learning rate schedule	Cosine decay	Cosine decay	–
Warmup steps	500	1,000	–
Gradient clipping	1.0	1.0	–
Loss weight λ	1.0	1.0	–
Precision	bf16	bf16	bf16
Hardware	32 × 141GB GPUs	64 × 141GB GPUs	1 × 141GB GPU
Training time	18 h	46 h	–
Solver	–	–	Flow matching Euler
Sampling steps	–	–	30
Classifier-free guidance scale	–	–	5.0
Per-sample latency	–	–	184.2 s
Peak memory	–	–	44.7 GB

Table B2: **Shape and length of each token sequence.**

Token sequence	Symbol	Shape	Value	Tokens
Source video	s_v	$L_v \times h_v \times w_v$	$16 \times 23 \times 40$	14,720
Reference image (optional)	i_v	$1 \times h_v \times w_v$	$1 \times 23 \times 40$	920
Visual Reasoning (single frame)	r_v	$1 \times h_v \times w_v$	$1 \times 23 \times 40$	**920**
Comparison: full-length mask sequence	–	$L_v \times h_v \times w_v$	$16 \times 23 \times 40$	14,720
Temporal Queries / TQB output	q_τ / $\tilde{q}_\tau$	L_v	16	16
Video Edit	e_v	$L_v \times h_v \times w_v$	$16 \times 23 \times 40$	14,720
Source audio	s_a	L_a	126	126
Acoustic Reasoning	r_a	L_a	126	126
Audio Edit	e_a	L_a	126	126

C Evaluation Metrics and Benchmark Protocols

C.1 Computation of Quality Metrics

This section gives the computation and implementation of the metrics in Section 5.2. JAVEditBench strictly uses its official implementation. All values on ReasonAVEdit-Bench and InsAVE-Eval are recomputed with the same evaluator and the same weight versions. We do not quote the results reported in the original papers. Table C1 lists each metric.

BG-PSNR and BG-SSIM are computed only on pixels outside the GT edit mask. Sync-C and Sync-D are computed only on the visible-speaker subset. DeSync is used only for non-speech events. For samples with a separated track, LPAPS is computed on the Base Audio that should be preserved. FVD and FAD are computed over the whole evaluation set, and the other metrics are averaged over samples.

JAVEditBench (Table 2) strictly follows its official protocol and six metrics. VTSS measures visual quality. UTMOS, predicted by UTMOSv2 (Baba et al., 2024), measures audio quality. SyncNet measures audio-visual synchronization. Gemini (Comanici et al., 2025) scores instruction following, visual fidelity, and audio-visual quality on a 5-point scale. All methods are evaluated on the same 150 source videos.

C.2 Composition of ReasonAVEdit-Bench

This section details the benchmark split in Section 5.2. The five subsets are disjoint, and each source video belongs to only one subset. Table C2 lists the subsets and their categories. Each Which-only sample contains one clear acoustic target and one main distractor source. The subset excludes scenes with three or more speakers, multiple distractors, or highly similar spectra. A failure in target identification can then be attributed on its own.

C.3 Reasoning Diagnostic Metrics and Protocol

The diagnostic metrics are Visual IoU (V-IoU), Small-Object V-IoU, A-T IoU (Acoustic Temporal IoU), A-TF IoU (Acoustic Time-Frequency IoU), Which-Acc (Target Source Identification Accuracy), V-T IoU (Video Temporal IoU), and A/V Edit-range Offset. The first five act on the decoded Reasoning (R_v and R_a). They apply only to methods that output Reasoning Tokens. The last two act on the final audio and video and apply to any method. Table C3 gives their computation, and Table C4 gives the rules for each subset. V-T IoU is computed from the outputs. Table D1 therefore

Table C1: **Computation and models of the quality metrics.**

Metric	Computation	Model or implementation
FVD↓ (Fréchet Video Distance) (Unterthiner et al., 2019)	Fréchet distance between generated and real videos in feature space, over the whole evaluation set	VideoMAE-v2 features
TV-A↑ (Text-Video Alignment)	Cosine similarity between the target text and the generated video	VideoCLIP-XL
TC↑ (Temporal Consistency)	Mean cosine similarity between image embeddings of adjacent frames	CLIP ViT-L/14
BG-PSNR↑ (Background Preservation)	PSNR between generated and source frames outside the GT edit mask	Pixel domain, averaged over frames
BG-SSIM↑ (Background Preservation)	Structural similarity between generated and source frames outside the GT edit mask	SSIM, averaged over frames
FAD↓ (Fréchet Audio Distance)	Fréchet distance between generated and real audio in feature space, over the whole evaluation set	CLAP audio encoder (Elizalde et al., 2023)
TA-A↑ (Text-Audio Alignment)	Cosine similarity between the target text and the generated audio	CLAP (Elizalde et al., 2023)
LPAPS↓ (Learned Perceptual Audio Patch Similarity)	Learned perceptual patch similarity between generated and reference audio	LPAPS (Iashin & Rahtu, 2021)
PQ↑ (Production Quality)	No-reference score of audio production quality	Audiobox-Aesthetics (Tjandra et al., 2025)
AV-A↑ (Audio-Video Alignment)	Cosine similarity between the generated video and audio in a shared embedding space	ImageBind (Girdhar et al., 2023)
PEAVS↑ (Perceptual Evaluation of Audio-Visual Synchrony)	Perceptual score of audio-visual synchrony	PEAVS (Goncalves et al., 2024)
Sync-C↑ (SyncNet Confidence)	Lip-sync confidence	SyncNet (Chung & Zisserman, 2016)
Sync-D↓ (SyncNet Distance)	Distance between lip and audio features	SyncNet (Chung & Zisserman, 2016)
DeSync↓ (Synchformer-predicted Temporal Offset)	Absolute predicted audio-visual offset in seconds	Synchformer (Iashin et al., 2024)

Table C2: **Subsets and categories of ReasonAVEdit-Bench.**

Subset	Category	Samples
Where-only	Two people, animals, or objects of the same kind	100
	Spatial references such as left and right, front and back, or size	80
	Small or partially occluded targets	70
Which-only	Two people speaking at once	120
	Conflict between speech and ambient sound	100
	Off-screen speech or off-screen sound sources	80
When-only	Word or phrase replacement	200
	Partial sentence rewriting	100
Combined Reasoning	Local speech editing while two people speak at once	100
	Local editing under a conflict between speech and ambient sound	50
	Local editing of animals or general sounding objects	40
	Audio-visual instance addition with spatial requirements	50
	Instance removal in complex scenes	10
Reference Editing	Reference for human identity or appearance	50
	Reference for animal and object appearance	30
	Instance addition guided by a reference image	20
Total	–	**1,200**

reports V-T IoU for the variant without Reasoning Tokens, although that variant has no Reasoning diagnostics.

All subsets report the final audio-visual quality metrics. The diagnostic metrics only verify whether the reasoning holds, and they do not enter the method ranking. When-only contains only speech edits, while DeSync applies only to non-speech events (Appendix C.1). This subset therefore measures lip sync with Sync-C and Sync-D.

C.4 Baseline Implementation Details

The four external baselines use their officially released pretrained weights and default inference configurations. We do not retrain them on ReasonAVEdit-Data. Their input formats differ, so we convert each ReasonAVEdit-Bench sample into the input that each method needs.

AvED. AvED (Lin et al., 2026) describes the content before and after editing with a source prompt and a target prompt. We use Gemini to rewrite each instruction into a pair of source and target

Table C3: **Computation of the diagnostic metrics.**

Metric	Target	Computation
Visual IoU↑	Decoded R_v	We extract the gray-covered region of R_v as the predicted mask and compute its IoU with the GT subject mask on the first frame of the edit time span. When the GT is the Null Anchor, a Null Anchor prediction scores 1 and any other prediction scores 0.
Small-Object V-IoU↑	Decoded R_v	Same as Visual IoU, over the samples whose GT subject covers less than 5% of the frame.
A-T IoU↑	Decoded R_a	We extract the non-silent intervals of R_a with a short-time energy threshold of -40 dBFS and compute their temporal IoU with the GT edit time span.
A-TF IoU↑	Decoded R_a	We binarize the STFT magnitude of R_a with a threshold to obtain a predicted time-frequency mask and compute its IoU with the time-frequency mask of the GT target source.
Which-Acc↑	Decoded R_a	We compare the source-embedding cosine similarity between R_a and the GT target source with that between R_a and the main distractor. A higher similarity to the target counts as correct, and we report the accuracy. The embeddings are speaker embeddings for speech and CLAP audio embeddings for non-speech.
V-T IoU↑	Output V_e	We compute the per-frame difference between V_e and V_s within the GT subject region. Frames above a threshold form the actual edit interval of the video, and we compute its temporal IoU with the GT edit time span. We compute it only on samples with a visible target.
A/V Edit-range Offset↓	Outputs V_e and A_e	We extract the actual edit intervals of the audio and the video and report the mean absolute deviation of their start and end times in seconds. We compute it only on samples with a visible target.

Table C4: **Metrics reported for each subset.**

Subset	Metrics reported
Where-only	Visual IoU, BG-PSNR, BG-SSIM
Which-only	Which-Acc, A-TF IoU, LPAPS
When-only	A-T IoU, V-T IoU, Sync-C, Sync-D, A/V Edit-range Offset
Combined Reasoning	All three groups above, plus DeSync on non-speech samples
Reference Editing	Reported separately and excluded from the main ranking (Appendix D.4)

descriptions. The rewriting keeps the target object, the attributes, and the temporal references of the instruction. We manually spot-check the rewritten pairs.

CoherentAVEdit. CoherentAVEdit (Ishii et al., 2025) needs an edited video as its condition. VACE-14B (Jiang et al., 2025) first generates the edited video from the GT instance mask and the target text. CoherentAVEdit then takes this video, the source audio, and the target text to generate the edited audio. The two results form the output of this cascaded baseline.

AVI-Edit. AVI-Edit (Zheng et al., 2026a) needs a spatial mask and a target audio. The spatial mask is the GT instance mask. The target audio is generated from the target text. We mix the target audio with the residual audio after the target source is removed. The mixture forms the audio condition of AVI-Edit. The residual audio comes from the source separation in data construction (Appendix E.2).

InstructAV2AV. InstructAV2AV (Zheng et al., 2026b) has the same interface as ReasonAVEdit. It takes the source audio-visual input and the instruction directly.

Output alignment and evaluation. Each method runs at its native resolution and duration. Its outputs are aligned to the duration of the source clip and resampled to 720p before the same evaluator. VACE-14B and AVI-Edit use GT instance masks, following the evaluation protocols of their original papers. They therefore receive extra spatial information that ReasonAVEdit does not have. All methods on JAVEditBench use its official evaluation code.

D Additional Ablations and Causal Diagnostics

This section gives the full results of Sections 5.4 and 5.5 (Appendices D.1 and D.2). It reports per-subset diagnostics (Appendix D.3) and separate results on the Reference Editing subset (Appendix D.4). It also tests through inference-time interventions whether the final edits actually use the reasoning (Appendix D.5).

D.1 Full Diagnostics of Reasoning Tokens

Table D1 reports the Reasoning diagnostics and the additional output metrics of the variants in Table 3.

D.2 Additional Metrics for Different Numbers of Visual Reasoning Frames

Table D2 reports the localization accuracy and the additional output metrics of the variants in Table 5.

Table D1: **Reasoning diagnostics and additional output metrics of the variants in Table 3.** Bold: best excluding GT.

Variant	V-IoU↑	A-T IoU↑	A-TF IoU↑	Which-Acc↑	V-T IoU↑	PQ↑	DeSync↓
w/o Reasoning Tokens	–	–	–	–	0.612	6.204	0.381
Visual Reasoning Tokens only	0.724	–	–	–	0.638	6.301	0.352
Acoustic Reasoning Tokens only	–	0.781	0.694	0.862	0.702	6.427	0.287
Visual + acoustic Reasoning Tokens w/o TQB	0.731	0.786	0.701	0.871	0.719	6.612	0.253
Full ReasonAVEdit	**0.738**	**0.792**	**0.708**	**0.879**	**0.763**	**6.800**	**0.212**
GT Reasoning Tokens	–	–	–	–	0.841	6.947	0.168

Table D2: **Localization accuracy and additional output metrics of the variants in Table 5.**

Method	V-IoU↑	Small-Object V-IoU↑	TV-A↑	FVD↓	AV-A↑	PEAVS↑	V-T IoU↑	A/V Edit-range Offset↓	BG-PSNR↑	DeSync↓
Mask sequence (L_v)	**0.742**	**0.651**	0.192	51.30	0.314	3.198	0.758	0.094	26.1	0.219
Mask sequence ($L_v/2$)	0.736	0.638	0.190	52.10	0.309	3.176	0.752	0.101	25.8	0.226
Mask sequence ($L_v/4$)	0.727	0.612	0.187	53.40	0.301	3.141	0.744	0.112	25.2	0.238
ReasonAVEdit (1 frame)	0.738	0.629	**0.193**	**49.98**	**0.318**	**3.217**	**0.763**	**0.087**	**26.3**	**0.212**

D.3 Per-Subset Diagnostics

Table D3 reports the diagnostics of ReasonAVEdit on the four instruction-only subsets, following the rules in Table C4. Each subset reports only the metrics for its reasoning need, and the other cells are marked "–". The three single-reasoning subsets test the visual region, the acoustic target, and the edit time span, respectively. Combined Reasoning tests all three. Offset is the A/V Edit-range Offset in seconds. For Combined Reasoning, Sync-C and Sync-D are computed on its speech samples, and DeSync on its non-speech samples.

D.4 Reference Editing Results

The Reference Editing subset contains 100 samples whose target appearance is given by a reference image (Table C2). None of the four external baselines accepts a reference image. This subset is therefore excluded from the main ranking. We only compare ReasonAVEdit with the reference-image condition on and off. When it is off, the model generates the target appearance from the instruction text alone. Besides the quality metrics in Appendix C.1, we report Ref-Sim. It crops the edited frames with the GT edit region and computes their cosine similarity with the reference image in the CLIP image embedding space, averaged over frames. Table D4 reports the results. With the reference image, Ref-Sim rises from 0.642 to 0.781, and the other metrics stay about the same. The reference image thus mainly constrains the target appearance. It does not change target localization or audio-visual synchronization.

D.5 Reasoning Intervention Experiments

Accurate reasoning predictions do not prove that the final edits actually use them. We therefore intervene on the Reasoning Tokens directly at inference and check whether the edits change accordingly. All conditions use the same trained model and the same Edit noise. They differ only in the value of the reasoning. Under an intervention, the reasoning is clamped at every flow time step to the noisy version of the intervention target R^{int}:

$$r_t = (1 - t)\,\epsilon_r + t\,\mathcal{E}(R^{\text{int}}). \tag{12}$$

The other tokens are integrated as in Algorithm A1.

We compare seven conditions. *Predicted* uses the reasoning predicted by the model. *Oracle* replaces both kinds of reasoning with benchmark annotations, which is the GT row of Table 3. *Zero* uses the Null Anchor and a fully silent acoustic trace. *Shuffled* uses the reasoning of another sample matched by task category, duration, mask area, and acoustic energy. Three single-factor interventions each change only one decision. *Swapped anchor* replaces the target in R_v with another subject in the same frame. *Swapped source* keeps the temporal envelope of R_a and replaces its waveform content with the main distractor source. *Shifted time* keeps the source content and shifts the non-silent interval of R_a by 1 s, in the direction that keeps the interval inside the clip. These interventions keep the chosen marginal statistics and break only the intended spatial, source, or temporal decision. Each single-factor intervention is evaluated only on the samples where it applies. Swapped anchor uses samples with several subjects of the same kind in the frame. Swapped source uses samples with a main distractor source. Shifted time uses local editing samples.

Table D3: **Diagnostics of ReasonAVEdit on each instruction-only subset.**

Subset	Visual region			Acoustic target			Edit time span					
	V-IoU↑	BG-PSNR↑	BG-SSIM↑	Which-Acc↑	A-TF IoU↑	LPAPS↓	A-T IoU↑	V-T IoU↑	Sync-C↑	Sync-D↓	Offset↓	DeSync↓
Where-only	0.746	26.9	0.841	–	–	–	–	–	–	–	–	–
Which-only	–	–	–	0.884	0.716	4.187	–	–	–	–	–	–
When-only	–	–	–	–	–	–	0.811	0.779	3.902	9.418	0.079	–
Combined Reasoning	0.730	25.7	0.819	0.873	0.699	4.472	0.770	0.744	3.487	9.902	0.096	0.231

Table D4: **Reference Editing with the reference-image condition on and off.**

Setting	Ref-Sim↑	TV-A↑	BG-PSNR↑	AV-A↑	PEAVS↑
Reference image off (instruction only)	0.642	0.189	25.9	0.305	3.176
Reference image on	**0.781**	**0.191**	**26.1**	**0.311**	**3.192**

Table D5 reports the results from two views. All quality metrics are computed against the original GT. If the edits use the reasoning, Oracle should bring gains and wrong decisions should cause drops. The follow rate measures whether the edit executes the intervened decision. For Swapped anchor, it is the fraction of samples whose actual video edit region has an IoU above 0.5 with the mask of the new subject. For Swapped source, it is the fraction of samples whose edited source is the swapped source, judged as in Which-Acc. For Shifted time, it is the fraction of samples whose actual audio and video edit intervals both have a temporal IoU above 0.5 with the shifted interval.

The results show three patterns. First, the correctness of the reasoning determines the editing quality. Oracle outperforms Predicted on all quality metrics. Shuffled lowers V-T IoU from 0.763 to 0.214 and BG-PSNR from 26.3 to 21.4. Second, the effect of each intervention is modality specific. Swapped anchor lowers V-T IoU to 0.331, while TA-A barely changes (0.398). Swapped source lowers TA-A to 0.336, while the video-side V-T IoU barely changes (0.752). Third, Shifted time lowers V-T IoU against the original interval to 0.402. However, AV-A and PEAVS stay close to Predicted. In 84.6% of the samples, the audio and video edits both move to the new interval. Changing only the acoustic reasoning moves the edit time of both modalities together. This is direct evidence that TQB passes the shared edit time span. Under Zero, the model barely edits, and V-T IoU drops to 0.093. BG-PSNR is high because the frames are barely changed. This agrees with the meaning of the Null Anchor, under which the video branch performs no target edit.

E Construction Details and Statistics of ReasonAVEdit-Data

This section details the four-stage construction pipeline in Section 4. Appendix E.1 covers source media preparation and instruction generation. Appendix E.2 covers mask and Reasoning supervision construction. Appendix E.3 covers quality control and manual verification. Appendix E.4 gives the sample statistics of each task category.

E.1 Source Media Preparation and Instruction Generation

The source clips come from the open-source OpenHumanVid (Li et al., 2025) and Koala-36M (Wang et al., 2025). We first cut long videos into 5–10 s clips. We then filter the frames and the audio separately. The visual filter uses motion magnitude, aesthetic quality, and resolution. The audio filter uses audio quality, the silence ratio, and content completeness. We set the thresholds so that the filters keep about 70% of the clips. For the kept clips, Gemini (Comanici et al., 2025) generates editing instructions for the five task categories in Table E1 from the visual and audio content of each clip. Each instruction specifies the target object and the target attribute. For local edits, it refers to the edit time span through the speech content or an event description.

E.2 Mask and Reasoning Supervision Construction

For each instruction, we first determine the edit time span shared by both modalities. We then construct a video mask and an audio mask. The video mask is the per-frame segmentation of the target subject over the whole clip for appearance edits and within the edit time span for other edits. Grounding SAM 2 (Ren et al., 2024) generates it. The audio mask marks the edit time span along the time axis. SAM Audio (Shi et al., 2025) separates the source sound to be replaced from the source audio. The Mask-based AVEdit model (Appendix F) generates the target audio and video under these masks.

Table D5: **Effect of inference-time interventions on the Reasoning Tokens.** Quality metrics are computed against the original GT.

Condition	TV-A↑	TA-A↑	BG-PSNR↑	AV-A↑	PEAVS↑	V-T IoU↑	Follow rate↑
Predicted	0.193	0.400	26.3	0.318	3.217	0.763	–
Oracle	0.206	0.427	28.1	0.361	3.418	0.841	–
Zero	0.168	0.302	30.2	0.196	2.884	0.093	–
Shuffled	0.172	0.318	21.4	0.203	2.917	0.214	–
Swapped anchor	0.179	0.398	22.9	0.262	3.064	0.331	81.3%
Swapped source	0.191	0.336	26.0	0.271	3.052	0.752	76.8%
Shifted time	0.187	0.371	25.8	0.301	3.141	0.402	84.6%

Table E1: **Statistics of ReasonAVEdit-Data.**

Task category	Main reasoning need	Train	Validation	Test
Local speech editing	Edit time span	62,000	620	300
Multi-speaker target editing	Acoustic target	48,000	480	300
General sounding object editing	Visual region + edit time span	36,000	360	200
Audio-visual instance addition and removal	Visual region	38,000	380	300
Reference-guided editing	Visual region (with appearance reference)	16,000	160	100
Total	–	**200,000**	**2,000**	**1,200**

The same masks render the Reasoning supervision. The visual target $R_v^\star$ takes the first frame of the edit time span and covers the target subject region with gray. It is an RGB image at the resolution of the source frame. When the first frame has no visible target, we use the Null Anchor. The acoustic target $R_a^\star$ has the same length as A_s. Within the edit time span, it keeps the separated sound to be replaced, and it is zero elsewhere. For tasks such as instance addition, the source audio has no sound to be replaced, so $R_a^\star$ is silent throughout. For these tasks, $R_v^\star$ takes the first frame of the clip, and the edit time span covers the whole clip. The edit targets and the Reasoning supervision come from the same masks, so they are strictly aligned by construction.

E.3 Quality Control and Manual Verification

Candidate samples go through automatic filtering and then manual verification. Both cover the five dimensions in Section 4. Instruction fidelity checks whether the target audio and video execute the instruction. Audio-visual quality checks the frames and sounds for obvious artifacts. Non-target preservation checks whether the frames and sounds outside the edit region are changed. Audio-visual synchronization checks the temporal alignment of the edited audio and video. Reasoning reliability checks whether $R_v^\star$ and $R_a^\star$ match the target in the instruction. Automatic filtering keeps a sample only if it passes all five checks. Instruction fidelity requires a Gemini instruction-following score of at least 4 out of 5. Audio-visual quality requires a Gemini audio-visual quality score of at least 4 and a PQ of at least 6.0. Non-target preservation requires a BG-PSNR of at least 28 dB outside the video mask and a signal-to-difference ratio of at least 25 dB for the audio outside the edit time span. Audio-visual synchronization requires a Sync-C of at least 3.0 for visible speech and a DeSync below 0.3 s for other sounds. Reasoning reliability requires Gemini to confirm that $R_v^\star$ marks the instructed target and that $R_a^\star$ contains only the instructed sound.

We manually verify 4,000 training samples drawn from those that pass automatic filtering. All 1,200 ReasonAVEdit-Bench samples are verified in the same way. A pool of ten annotators performs the verification. Three annotators from the pool independently judge whether each sample passes the five dimensions. A sample is removed if the majority of them reject any dimension. The data split is done before target generation. It deduplicates by source video, speaker, subject identity, and near-duplicate clips. The training, validation, and test parts are therefore disjoint in these dimensions.

E.4 Data Statistics

Table E1 summarizes the training, validation, and test samples of each task category. The test split forms ReasonAVEdit-Bench.

F Mask-based AVEdit Model and Training Configuration

The Mask-based AVEdit model is used only to construct ReasonAVEdit-Data. It is conditioned on the source audio and video, the text instruction, a spatio-temporal video mask, and an audio mask. It is a joint audio-visual editing model that can edit the audio and the video at any masked location.

Table F1: **Training configuration of the Mask-based AVEdit model.**

Item	Configuration
Base model	LTX-2.3-22B
Training data	Mask-conditioned audio-visual edit pairs from OpenHumanVid and Koala-36M, about 120K; masks extracted by Grounding SAM 2 and SAM Audio
Clip duration and resolution	5–10 s / 720p
Global batch size	128
Optimizer and learning rate	AdamW, 1×10^{-5}
Training steps	20K
Precision and hardware	bf16, 32 × 141GB GPUs

The video branch takes the source video with the masked region filled with gray as its condition. The audio mask is a one-dimensional mask along time that marks the segments to edit and covers all frequencies. Table F1 lists its training configuration.

Edit pair generation. During data construction, the model receives the video and audio masks from Appendix E.2. It generates new content only within the video mask and the audio mask. It keeps the source audio and video elsewhere. Some instructions specify the new sound content, such as a new word in local speech editing. The model fills the masked audio segment with this content according to the instruction. The quality control in Appendix E.3 filters the generated pairs.

G Limitations and Failure Modes

The source of reasoning supervision bounds localization. In ReasonAVEdit-Data, both kinds of Reasoning supervision are converted from the same masks of the construction engine (Section 4). The Reasoning targets and the edit targets are therefore strictly consistent by construction. This keeps the training signal self-consistent. However, it also means that the model inherits the target partitioning habits of the construction engine rather than an independent localization ability. The target boundary implied by a real instruction may differ from that of the construction engine. In that case, the predictions shift systematically. One example is whether a person and a handheld object count as one edit target. Table 3 directly measures this bound. Replacing the model predictions with benchmark annotations still improves AV-A and PEAVS by 0.043 and 0.201. Reasoning prediction accuracy is thus not yet saturated. The current results are not the performance ceiling of the framework.

The single-frame visual anchor has a limited scope. Compressing visual reasoning into a single frame assumes that the target stays recognizable within the edit time span. Long occlusion, fast deformation, and shot cuts all break this assumption. Table D2 shows an observable cost. The full-length mask sequence variant reaches a Small-Object V-IoU of 0.651. This is higher than the 0.629 of the single-frame variant. The single-frame anchor thus localizes small and partially occluded targets less accurately. The gains in output quality and efficiency offset this loss. When the target is off-screen, the visual side falls back to the Null Anchor. Localization then relies entirely on the acoustic Reasoning Tokens and TQB.

Acoustic target separation and the evaluation scope limit the conclusions. Acoustic reasoning supervision requires that the target sound source can be separated from the source audio. The Which-only subset is therefore intentionally limited to one clear target and one main distractor source. It excludes scenes with three or more speakers, multiple distractors, or highly similar spectra (Appendix C.2). Our conclusions on acoustic target identification do not directly extend to heavily overlapping acoustic scenes. In addition, training and evaluation use 5–10 s clips at 720p resolution. Long videos are edited segment by segment, and the edited segments are concatenated. Each instruction is assumed to target a single source, and edits of several targets or events at different times are not covered. Reference Editing contains only 100 samples. We report it separately and exclude it from the main ranking.

H Additional Related Work

H.1 Joint Audio-Visual Generation

Joint audio-visual generation synthesizes video and sound together. It keeps the two modalities consistent in semantics and timing. MM-Diffusion (Ruan et al., 2023) is an early work that applies joint diffusion modeling to audio-visual generation. Ovi (Low et al., 2025) connects symmetric twin backbones through cross-modal fusion. LTX-2 (HaCohen et al., 2026) adopts an asymmetric dual-stream architecture. It continuously exchanges information between the video stream and the audio stream. These models provide strong multimodal priors for synchronized content generation.

ReasonAVEdit builds on LTX-2.3-22B, but it studies a different problem. Joint generation concerns what content to produce from the conditions. Joint editing must also preserve the source media. It must further determine the visual region, the acoustic target, and the shared edit time span. A generation backbone alone therefore cannot localize edit targets from instructions.

H.2 Explicitly Conditioned Joint Audio-Visual Editing

These methods restrict the edit target in advance through extra inputs or dedicated pipelines. AvED (Lin et al., 2026) performs zero-shot editing with source and target prompts and cross-modal delta denoising. CoherentAVEdit (Ishii et al., 2025) first obtains an edited video and then synthesizes audio conditioned on it. Object-AVEdit (Fu et al., 2025) targets object-level manipulation. AVI-Edit (Zheng et al., 2026a) uses spatial masks and target audio for instance control. SpongeBob (Liang et al., 2026) uses a target subject mask and separated base audio. It introduces bidirectional audio-visual interaction in unified dual-stream denoising. These methods provide clear spatial or acoustic constraints. However, they require extra conditions, external tools, or cascaded pipelines.