ELTA: EVENT-LEVEL TEXT-TO-AUDIO GENERATION WITH LABELS AND DESCRIPTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Sound design for films and games requires control over individual sounds, their acoustic details, and their timing. Coarse event labels leave those details unspecified, while requests involving several sounds also require each description to be associated with its intended event. We introduce *Event-Level Text-to-Audio* (ELTA), which uses explicit sound labels together with detailed natural-language descriptions within user-defined event intervals. ELTA encodes each sound label separately while retaining the full description of its event, then learns how these conditions contribute to one jointly generated scene. Multiple sounds can share an event, and separate events can overlap without merging their requests. For supervision, we curate and annotate 373,733 source audio clips with sound labels and detailed acoustic captions, then compose training scenes while preserving each recording's labels, description, and placement interval. Our results show improved sound coverage and description matching over temporal-generation baselines.

1 INTRODUCTION

Sound design for films and games requires control over individual sounds, their acoustic details, and when they occur. A request for slow, heavy footsteps on gravel is only partly satisfied by generic footsteps: the pace and surface texture matter to the scene. When a request contains several sounds, reproducing only the most prominent one also leaves it incomplete. We therefore study generation from detailed event requests, including multiple sounds within one interval (Figure 1) as well as overlapping events.

Temporal generation methods already support timed labels, local descriptions, and acoustic controls. Some combine active label or description vectors into a shared timeline feature (Zheng et al., 2025; Zhu et al., 2026), while others retain description tokens or process local descriptions separately (Wang et al., 2024; Jiang et al., 2025). Labels make the requested sounds explicit but leave their acoustic details unspecified; descriptions express those details and the context shared by sounds in an event. We study these complementary conditions by encoding each label separately while retaining its event's full description and interval.

We introduce *Event-Level Text-to-Audio* (ELTA), with a *Short branch* that encodes each sound label separately within its event interval and a *Long branch* that uses the complete description to condition on acoustic details and the context shared by the event's sounds. The model learns how these conditions contribute to one jointly generated scene. Labels within an event share its description and interval, while overlapping events retain their own labels and descriptions (Figure 3).

To train this interface, we need recordings paired with sound labels and descriptions of their audible characteristics. We curate and annotate 373,733 source audio clips, including recordings with several sounds, and pair each clip with explicit labels and a detailed acoustic caption. We vary each recording's placement and overlap with other recordings, retain its labels and full caption, and record the assigned interval in each composed scene. These compositions provide supervision for local sound identities and acoustic details without requiring source separation or internal sound-boundary annotations.

Our experiments show that ELTA outperforms the compared temporal generation systems in sound coverage and description matching. Listening examples also demonstrate contrasts in surface texture, repetition rate, and timbre that match distinct descriptions of the same labeled sound in separate

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.

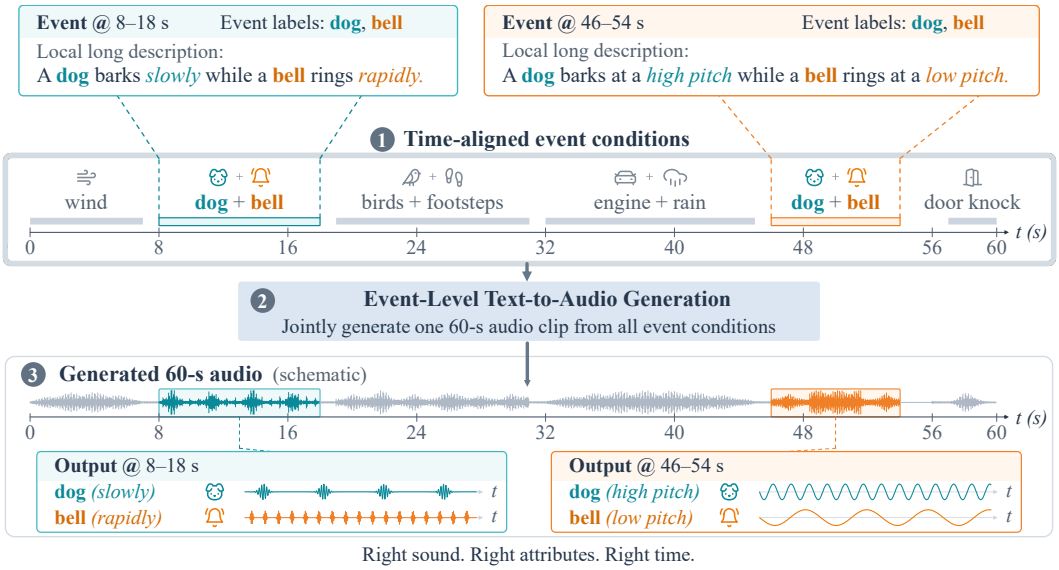

Figure 1: Overview of event-level audio generation. Each event pairs sound labels (Short) with a local description (Long) and a placement interval. The two highlighted events share dog and bell labels but specify different rhythm or pitch attributes. All events condition one generated scene. The waveforms and attribute traces illustrate the requested behavior schematically.

event windows (Figure 4). When we disable either conditioning branch at inference, both semantic measures decrease, showing that the trained model uses the label and description paths together.

Our contributions are:

- We introduce ELTA, a conditioning architecture that combines time-aligned label features with attention to each event's complete description. It computes these features separately and aggregates their window-masked contributions within a shared generation process, including when event intervals overlap.
- We develop a pipeline that curates and automatically annotates 373,733 source audio clips with sound labels and detailed acoustic descriptions. The clips form a reusable bank for timeline composition, where each clip is an event-level unit whose labels and description share a placement interval. This provides supervision from multi-sound recordings without internal onset/offset annotations.
- We compare ELTA with three temporal generation systems and observe higher Identity QA and Long-CLAP scores, while qualitative examples illustrate distinct acoustic realizations of the same sound label in event windows with different descriptions.

2 Related Work

Local descriptions and temporal control. Make-An-Audio 2 combines natural language with structured event-order descriptions (Huang et al., 2023), while AudioComposer retains description tokens for cross-attention and uses text specifying event times, pitch, and energy (Wang et al., 2024). PicoAudio2 expands local-description vectors over time and sums active vectors alongside coarse text conditioning (Zheng et al., 2025). FreeAudio first divides overlapping requests into non-overlapping windows and rewrites their combined descriptions for each window. It then computes independent cross-attention for local descriptions and aggregates the outputs without training (Jiang et al., 2025). We retain the original label sets, full descriptions, and overlapping intervals in ELTA, using separate trainable label and description branches.

Structured and adaptive conditions. PicoAudio combines event semantics with a category-based timestamp matrix (Xie et al., 2024), while Audio ControlNet sums active label embeddings on an

event timeline and also provides loudness and pitch controls (Zhu et al., 2026). ControlAudio adds timing and phoneme conditions (Jiang et al., 2026), and DegDiT models semantic and temporal relations between event nodes with a graph transformer (Liu et al., 2026b). For melody-conditioned song generation, SongEcho refines condition features using the generator's hidden state and injects them through element-wise modulation (Li et al., 2026). In ELTA, we use the audio hidden state to gate separately encoded label features before combining them with event-wise description features.

Text–audio alignment and acoustic control. TangoFlux uses CLAP-ranked preferences to improve text–audio alignment (Hung et al., 2026). AC-Foley transfers acoustic characteristics from reference audio for video-to-audio generation (Fang et al., 2026), while LatentFT controls musical structure at selected timescales through latent Fourier components (Wang & Huang, 2026). PrismAudio separates semantic, temporal, aesthetic, and spatial objectives in video-to-audio generation (Liu et al., 2026a). Recomposer edits input audio using timed action-class instructions (Ellis et al., 2025). We use labels and language to specify sound identities and acoustic details within user-defined event windows.

Audio supervision. AudioX builds large-scale structured captions for multimodal audio generation (Tian et al., 2026), and Aurelius pairs event and relation corpora for relation-aware generation (He et al., 2026). For supervision, we preserve each recording's labels and acoustic descriptions under one placement interval, without requiring internal event boundaries or systematic relation labels. Appendix C.2 specifies the evaluated systems' inputs.

3 Task formulation

We generate 60-second stereo audio at 44.1 kHz from a collection of user-specified events. Each event combines sound labels, a description, and a shared time interval, giving the scene request $\mathcal{C}$:

$$\mathcal{C} = \{(\mathcal{S}_i, L_i, I_i)\}_{i=1}^{E}, \qquad I_i = [a_i, b_i). \tag{1}$$

Here E is the number of event requests, $\mathcal{S}_i$ is a nonempty unordered set of sound labels (*Short labels*), L_i is a natural-language description (*Long description*), and I_i is their shared window. We specify its endpoints a_i, b_i in seconds, with $0 \leq a_i < b_i \leq 60$. Short labels name the requested sounds, while the Long description gives their attributes, actions, and relationships. The description can also include accompanying sounds: for example, labels may name horse hooves while the description adds a distant crow. We use Short and Long to distinguish these input roles, not to impose word-count thresholds.

Several physical sources can share an event interval, and separate events can overlap. A placement window allows pauses and intermittent activity and does not specify activity frame by frame. When a sound needs independent timing, we assign it a separate event. This also lets us distinguish instances of the same class; repeating a label within one event does not assign an instance identifier.

4 Dataset construction

To train local control, we need examples that associate sounds with acoustic details and intervals. We therefore annotate source clips and then vary their placement and surrounding sounds, preserving each clip's labels and full description throughout composition (Figure 2).

4.1 Recording-specific labels and descriptions

We first use Qwen3-Omni to caption each source recording (Xu et al., 2025), then extract a structured list of the described sounds and acoustic properties in a text-only stage. We normalize the labels to separate sound categories from pitch, timbre, rhythm, pace, and texture. When rule-based checks flag malformed or unsuitable text, we rewrite the caption from the extracted content without adding events. The audio captioning stage thus supplies the acoustic evidence, while text refinement organizes it into labels and descriptions. Appendix B gives the annotation provenance and checks.

We retain clips with a nonempty caption, which becomes the Long description, and one to 16 Short labels. Of these, 330,877 clips (88.53%) have multiple labels. We encode labels and descriptions separately without truncation.

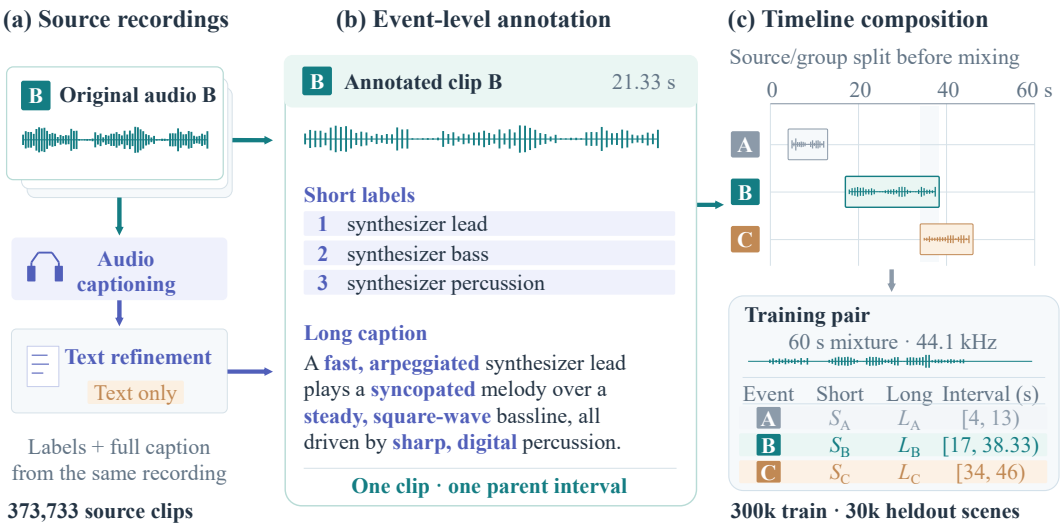

Figure 2: Data construction with clip-level annotations. Audio captioning and text refinement produce sound labels and a description for each recording. All labels in a clip share its placement interval. Source/group splitting precedes waveform mixing at 44.1 kHz. Waveforms and timeline positions are schematic; the clip's annotation is automatically generated.

When a recording contains several sounds, we keep them together in one clip so that its labels and caption share a placement window (Figure 2). We do not systematically annotate the sounds' internal timing or foreground/background relations.

4.2 Timeline composition and source splits

We compose scenes from one to six complete 1–60-second clips in proportions of 25/20/20/20/10/5%. During selection, we enforce $\sum_{i=1}^{E} d_i \leq 60$ seconds, where d_i is the duration of clip i, and then vary gaps and partial overlaps without cropping, looping, or stretching the clips. We convert source masters directly to 44.1 kHz before mixing them. If the mixture's peak exceeds -1 dBFS, we apply a single attenuation factor to the summed waveform.

We screen proposed overlaps with activity and relative-energy checks, then check the resulting mixture. If a layout fails, the total-duration constraint lets us place the same complete clips without overlap. In either case, we record the final intervals.

Across five datasets, we retain 373,733 source clips totaling 2,015.63 hours before reuse, covering sound effects, instruments, music, and vocal material (Table 4). Before composition, we split the bank by source/group into 336,773 training clips and 36,960 held-out clips, from which we build 300,000 and 30,000 scenes, respectively. Every retained clip appears in its assigned split. The training scenes contain 855,000 placements, corresponding to mean reuse of 2.54 and maximum reuse of three.

We verify that registered sources, groups, and exact waveforms have no cross-split conflicts. The held-out split includes development material; Appendix B reports source-level statistics, annotation provenance, and the final layout distribution.

Finally, we pair each composed scene with the event requests in Equation 1, using each placed clip's label set, full caption, and placement interval.

5 Method

We condition a pretrained audio generative backbone through two parallel branches (Figure 3). In the Short branch, we process each label as a separate time-aligned feature sequence, while the Long branch attends to each event's complete description tokens. Both branches use the same audio

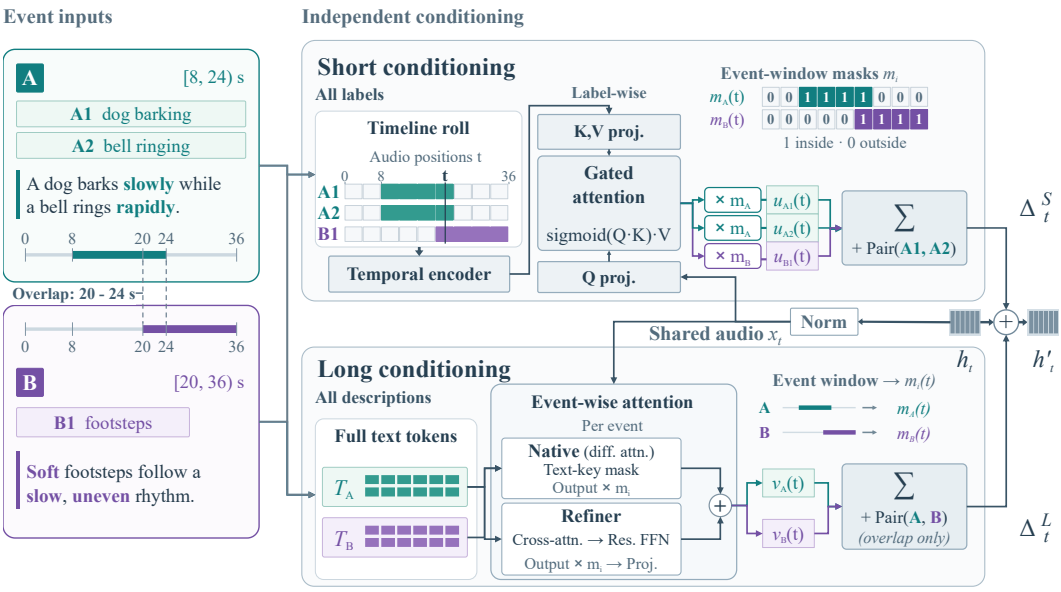

Figure 3: ELTA conditioning block. Each attention module derives its own query from $x_t = \mathrm{RMSNorm}(h_t)$. In the gated-attention schematic, Q, K, V denote a head's query, key, and value; dot-product scaling is described in the text. The Short temporal encoder produces separate label features $r_{ik}(t)$. Gated outputs are summed and projected by W_S; Pair_S uses the individual pre-projection features and has its own output projection. For each event i, the same complete token sequence T_i enters both pretrained attention (Native) and the cross-attention/FFN refiner in parallel. Attention is normalized separately for each event, and the two outputs add to form $v_i(t)$. Window masks restrict direct conditioning. Short interactions pair A1 and A2 within A; Long interactions pair A and B during their overlap at $[20, 24)$ s. Both branches combine additive and interaction terms in one residual update (Equation 2).

hidden state, and we combine their outputs in a residual update:

$$h'_t = h_t + \Delta^S_t + \Delta^L_t. \tag{2}$$

Here h_t and h'_t are the audio hidden states before and after conditioning, and Δ^S_t and Δ^L_t are the Short- and Long-branch contributions. Both contributions have the backbone's hidden width. We use the backbone's cross-attention normalization to obtain $x_t = \mathrm{RMSNorm}(h_t)$, then apply each attention module's own query projection to this shared state.

We index events by $i = 1, \ldots, E$ and their labels by $k = 1, \ldots, K_i$, where $K_i = |\mathcal{S}_i|$ and the label order is arbitrary. Audio time t denotes a latent-frame center in seconds, distinct from the diffusion step. The window mask $m_i(t) = \mathbf{1}[t \in I_i]$ restricts each event's direct contribution: it is one inside event i and zero outside. Layer and diffusion-step indices are omitted for readability.

5.1 Short conditioning

We first embed each label separately with a frozen text encoder. After projection, we repeat its embedding at audio latent frames within the event window and place zeros outside. A shared temporal encoder then processes each time-aligned sequence, reapplying the window mask after every convolution to obtain $r_{ik}(t)$. The label sequences remain separate even when their windows coincide.

At each frame t, we form the gated attention query from x_t and the keys and values from each label's feature at that same frame:

$$u_{ik}(t) = m_i(t)\ \mathrm{GatedAttn}(x_t, r_{ik}(t))\,. \tag{3}$$

Within $\mathrm{GatedAttn}$, each head scales its query–key dot product by the inverse square root of the head width and applies a sigmoid to gate the value. Because these gates are independent across labels, several labels can contribute at once without a shared softmax. We concatenate the gated head

outputs to obtain $u_{ik}(t)$ before output projection, then pass these separate features to an additive path and a within-event pairwise interaction module.

5.2 Long conditioning

The Long branch uses one complete description for each event. For example, the two labels of event A in Figure 3 share a description of slow barking and rapid ringing. We encode each description L_i with a frozen text encoder to obtain its token sequence T_i, whose positions index the text. Each active audio frame can attend to this full sequence. For each event, we feed the same T_i to the backbone's pretrained cross-attention and a cross-attention refiner in parallel, then add their outputs before aggregating across events:

$$v_i(t) = m_i(t)\left[\mathrm{PretrainedAttn}(x_t, T_i) + \mathrm{Refiner}(x_t, T_i)\right]. \tag{4}$$

Here $v_i(t)$ is event i's Long feature, and both operators return vectors with the backbone's hidden width. We reuse the backbone's text cross-attention as $\mathrm{PretrainedAttn}$. In the evaluated configuration, this is differential attention, which subtracts two softmax-weighted sums of the same values. Its context contains the event's projected text tokens and a token encoding the total scene duration. Alongside this path, $\mathrm{Refiner}$ independently conditions on the same description using standard cross-attention followed by a residual feed-forward network (FFN). The notation leaves the event window implicit in both operators and the duration token implicit in $\mathrm{PretrainedAttn}$.

The two modules apply the window at different points. In pretrained attention, we mask event-text keys outside I_i while retaining the duration key. The refiner instead applies the mask after attention and feed-forward processing, then uses a bias-free projection to the backbone dimension. Both direct outputs are zero outside the window, and we add them to form $v_i(t)$. Within each layer, each module shares its parameters across events but normalizes attention separately for each event.

5.3 Feature aggregation and residual update

We aggregate the individual contributions in each branch by adding their sum to a learned pairwise interaction term (Figure 3). Short interactions connect labels within the same event, while Long interactions connect events whose windows contain the current frame:

$$\begin{aligned}
\Delta_t^S &= W_S \sum_{i=1}^{E}\sum_{k=1}^{K_i} u_{ik}(t) + \sum_{i=1}^{E} m_i(t)\,\mathrm{Pair}_S\Big((u_{ik}(t))_{k=1}^{K_i}\Big), \\
\Delta_t^L &= \sum_{i=1}^{E} v_i(t) + \mathrm{Pair}_L\big((v_i(t))_{i:m_i(t)=1}\big).
\end{aligned} \tag{5}$$

We use W_S to project the summed Short features into the backbone dimension. Pair_S instead receives the individual label features before this projection and uses its own output projection, while Pair_L receives each event's combined pretrained-attention and refiner contribution. We sum these features without dividing by the number of active labels or events. A single active event still contributes through the additive path, while cross-event interactions require overlap. Each interaction module is permutation-invariant and returns zero with fewer than two active members, and we initialize its output projection to zero. Appendix A.1 specifies the computation.

We then add both branch contributions through the residual update in Equation 2. Their masks restrict direct conditioning to the requested frames, although shared self-attention can still carry sound information outside those windows. We retain pretrained attention in every backbone block and place the Short branch, refiner, and interaction modules in alternating blocks. Appendix A.1 summarizes the evaluated configuration.

5.4 Training

We use the frozen audio autoencoder to encode each waveform mixture into a clean audio latent, then train the model to predict flow velocity from a noisy latent. The loss combines velocity MSE on valid frames, with interval and boundary weighting, and an event-averaged term of weight 0.5 (Appendix A.4).

We train pretrained Long attention at a constant learning rate of 10^{-5} and the new modules at 10^{-4}, while keeping the text/audio encoders and remaining backbone weights frozen. Appendix A.3 specifies the training budget, data mixture, and condition sampling.

6 EXPERIMENTS

We compare sound coverage and description matching across systems, examine how event descriptions control acoustic attributes, and test the contributions of the two conditioning branches.

6.1 EXPERIMENTAL SETUP

We use Stable Audio 3 Medium Base (Evans et al., 2026) as the pretrained generative backbone, retaining its duration conditioning without a global text prompt. At inference, *Full* enables both branches, whereas *Short-only* disables the Long branch while using the same checkpoint. We sample with 50 Euler steps and CFG 1.5. Appendix A.3 gives the training budget, data mixture, and condition sampling.

We compare with PicoAudio2 (Zheng et al., 2025), Audio ControlNet (Zhu et al., 2026), and AudioComposer (Wang et al., 2024) on 1,979 source-disjoint, reference-supported 60-second scenes with one to three events spanning sound effects, instruments, and music. Reference checks use no generated outputs (Appendix C.1). For each method, we use two seeds per scene and provide complete labels, descriptions, and intervals through its interface. External systems jointly generate events in each active 10-second block, whereas ELTA generates the complete scene. Appendix C.2 gives adaptations, sampling profiles, and the comparison scope.

6.2 QUANTITATIVE COMPARISON

We evaluate local request following through sound coverage and description matching. For Identity QA, we ask Qwen3-Omni (Xu et al., 2025) whether each Short label is audible in its event window, so each requested label receives a separate check. We pair this with Long-CLAP, which uses CLAP (Wu et al., 2023) to compare the same window with its complete event description. Long-CLAP assesses overall similarity rather than checking every sound or attribute; sounds mentioned only in the description are not separately counted by Identity QA.

As an auxiliary diagnostic, we report the mean detector response inside the event window as FlexSED in (Hai et al., 2025). Our windows allow pauses, so we do not compute segment F1 against masks that mark the entire window as active: these would penalize valid gaps. FlexSED in is also sensitive to how long detector activity persists and does not measure boundary accuracy. Appendix C.4 specifies these distinctions and the aggregation.

We complement these condition-based measures with predicted production quality (PQ) from Audiobox Aesthetics (Tjandra et al., 2025) and Fréchet audio distance (Kilgour et al., 2019) using L2-normalized CLAP audio embeddings (FAD-CLAP-L2). Both use complete event windows. PQ assesses predicted production quality, while FAD compares generated and reference feature distributions without matching individual outputs to their requests.

ELTA leads Identity QA and Long-CLAP across the compared systems (Table 1), with paired comparisons supporting both gains over every baseline (Appendix C.1). Together, these results indicate better coverage of the requested labels and closer matching to their complete descriptions.

Audio ControlNet has the highest average FlexSED response. Since this score combines response strength with the duration of detector activity, its ranking does not establish which method better follows intermittent descriptions or places event boundaries.

The quality ranking is mixed: PicoAudio2 leads PQ, Audio ControlNet has the lowest FAD-CLAP-L2, and ELTA ranks second on both. Stronger request following therefore does not imply the best predicted production quality or the closest reference distribution.

Table 1: Quantitative comparison on 1,979 reference-supported scenes, with 3,958 outputs per generator. Identity QA, Long-CLAP, FlexSED in, and PQ are event-weighted means over two seeds; FAD compares feature distributions. Bold and underlining mark the best and second-best generative means; reference mixtures are excluded from ranking. FlexSED is uncalibrated; FAD uses L2-normalized CLAP embeddings.

Method	Identity QA ↑	Long-CLAP ↑	FlexSED in ↑	PQ ↑	FAD-CLAP-L2 ↓
PicoAudio2	0.7859	<u>0.3503</u>	<u>0.1746</u>	**6.0215**	0.2212
Audio ControlNet	<u>0.7943</u>	0.3411	**0.1983**	5.7131	**0.1877**
AudioComposer	0.7516	0.2986	0.1418	5.7899	0.3315
ELTA	**0.8490**	**0.3668**	0.1737	<u>5.9768</u>	<u>0.2134</u>
Reference mixtures	0.9931	0.3941	0.1932	6.5066	—

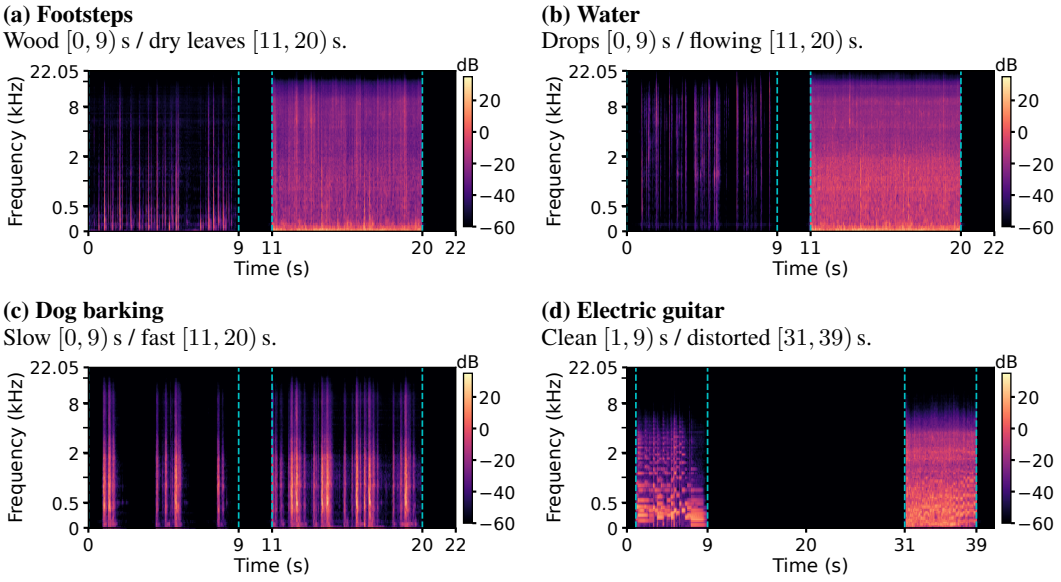

Figure 4: Attribute control through Long descriptions. Each panel shows part of one jointly generated 60-second track with the same Short label and contrasting descriptions in the two windows: (a) surface texture, (b) discrete versus continuous sound, (c) repetition rate, and (d) timbre. Panel headings summarize the inputs; Appendix D gives the complete text and settings. Dashed lines mark requested windows. Panels (a–c) show 0–22 s; panel (d) shows 0–41 s. The mel-frequency and power scales are shared; color shows mel power in dB (reference 1).

6.3 Fine-grained control with event descriptions

The quantitative comparison measures overall content matching; in Figure 4, we examine specific acoustic attributes controlled through Long descriptions. Each example is one jointly generated 60-second track containing two events with the same Short label and contrasting descriptions in their respective windows. We generate all four tracks from noise as complete scenes using the model evaluated in Table 1.

In our qualitative listening, we hear the requested contrasts: footsteps on wood and dry leaves, separated water drops and flowing water, slow and rapid barking, and clean and distorted electric guitar. The mel spectrograms show corresponding changes in temporal activity and spectral texture. For example, the water changes from isolated impacts to sustained activity, while the faster barking contains more closely spaced bursts. Long descriptions thus specify different acoustic forms of the same labeled sound in these examples.

Table 2: Inference-time branch ablation on 100 scenes with two seeds and event-weighted means. All conditions use the checkpoint from Table 1. Bold and underlining mark the best and second-best means in each column. Appendix C.3 gives the subset and paired confidence intervals.

Condition	Identity QA ↑	Long-CLAP ↑
ELTA (Full)	**0.8325**	**0.3681**
w/o Short branch	0.6255	<u>0.2405</u>
w/o Long branch (Short-only)	<u>0.6874</u>	0.2237

The first three examples place the windows two seconds apart; the guitar uses more widely separated windows. Appendix D gives all eight complete descriptions, their intervals, and the shared generation settings.

6.4 Contributions of Short and Long conditioning

To measure the trained model's use of each branch, we disable it at inference and evaluate the same checkpoint on 100 metadata-stratified scenes with two shared generation seeds and identical sampling settings. Removing Short disables its label and interaction contributions, while removing Long disables pretrained description attention and the added Long modules. We keep event intervals and evaluation queries unchanged.

Full outperforms both branch-removal conditions on Identity QA and Long-CLAP (Table 2), with paired intervals supporting all comparisons (Appendix C.3). Removing Short reduces sound coverage even when complete Long descriptions remain available. Conversely, retaining Short without Long reduces both sound coverage and description matching. The trained model therefore uses the explicit label inventory and the complete descriptions together to realize local requests.

7 Limitations

Our Short and Long branches update a shared audio state, and secondary sounds can still be omitted or receive the wrong attributes. The event interface also assigns one window to all sounds within an event, so independent timing requires separate events. Repeating a label within one event does not identify separate same-class sources.

Our automatic captions can miss details, and mixing can mask quiet sounds. Although we create new layouts by reusing source clips, this does not add new recordings. We split the data by source/group, but this alone does not establish unseen attribute combinations, and overlap with backbone pretraining data remains unknown.

We also rely on automatic judges whose responses depend on the sound family and question format, and window-level detection cannot establish rhythm or continuous simultaneous audibility. Our qualitative examples cover selected sounds and attributes, leaving broader reliability to be measured.

8 Conclusion

We introduced ELTA, which combines time-aligned sound labels and event descriptions in one generation process, giving users an event-level interface for specifying what sounds to generate, how they sound, and when they occur. The model keeps label and event-description features separate until aggregation, even when intervals overlap, and learns their combined contribution to the audio hidden state. Our data pipeline supplies these inputs by annotating source recordings and composing them into timed mixtures. On 1,979 reference-supported scenes, ELTA achieves higher Identity QA and Long-CLAP scores than all three compared systems. Same-checkpoint ablations show that removing either branch reduces sound coverage and description matching, while qualitative examples show distinct acoustic versions of the same labeled sound in windows with different Long descriptions.

AI USE STATEMENT

We used generative AI tools to assist with method design and formulation, experiment planning, code implementation, dataset annotation and text processing, and result analysis, as well as literature search and manuscript drafting and revision. We specify the annotation and evaluation models in Appendices B and C.4. To verify the work, we ran software tests, checked the work against experiment records and cited sources, and listened to selected audio examples. The dataset annotations remain automatic and have not been exhaustively human-verified. We are responsible for the final text, methods, claims, and supporting materials.

ETHICS STATEMENT

Controllable audio generation can support sound design and accessibility, but it can also produce misleading recordings. Our attribute-control examples are non-vocal, although the source bank and broader comparison include vocal material. We report our own qualitative listening observations, not results from a recruited participant study. We do not redistribute the source audio bank with this submission, and the original recordings remain subject to their source terms. Generated audio should be identified as synthetic when shared.

REPRODUCIBILITY STATEMENT

We define the inputs, dataset construction, and architecture in Sections 3, 4, and 5, then give the conditioning modules, training settings, and objective in Appendix A. For evaluation, Appendix C specifies the comparison scenes, baseline adaptations, metrics, and supplementary results, while Appendix D records the illustrated outputs' prompts and generation settings.

REFERENCES

Daniel P. W. Ellis, Eduardo Fonseca, Ron J. Weiss, Kevin Wilson, Scott Wisdom, Hakan Erdogan, John R. Hershey, Aren Jansen, R. Channing Moore, and Manoj Plakal. Recomposer: Event-roll-guided generative audio editing, 2025. URL https://arxiv.org/abs/2509.05256.

Zach Evans, Julian D. Parker, Matthew Rice, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Stable Audio 3, 2026. URL https://arxiv.org/abs/2605.17991.

Pengjun Fang, Yingqing He, Yazhou Xing, Qifeng Chen, Ser-Nam Lim, and Harry Yang. AC-Foley: Reference-audio-guided video-to-audio synthesis with acoustic transfer. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=URPXhnWdBF.

Eduardo Fonseca, Xavier Favory, Jordi Pons, Frederic Font, and Xavier Serra. FSD50K: An open dataset of human-labeled sound events. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:829–852, 2022. doi: 10.1109/TASLP.2021.3133208. URL https://arxiv.org/abs/2010.00475.

Jiarui Hai, Helin Wang, Weizhe Guo, and Mounya Elhilali. FlexSED: Towards open-vocabulary sound event detection. *arXiv preprint arXiv:2509.18606*, 2025. URL https://arxiv.org/abs/2509.18606.

Yuhang He, He Liang, Yash Jain, Andrew Markham, and Vibhav Vineet. Aurelius: Relation aware text-to-audio generation at scale. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/2bdaea4883792791c652314967a8fa89-Abstract-Conference.html.

Shawn Hershey, Daniel P W Ellis, Eduardo Fonseca, Aren Jansen, Caroline Liu, R Channing Moore, and Manoj Plakal. The benefit of temporally-strong labels in audio event classification, 2021. URL https://arxiv.org/abs/2105.07031.

Jiawei Huang, Yi Ren, Rongjie Huang, Dongchao Yang, Zhenhui Ye, Chen Zhang, Jinglin Liu, Xiang Yin, Zejun Ma, and Zhou Zhao. Make-An-Audio 2: Temporal-enhanced text-to-audio generation, 2023. URL https://arxiv.org/abs/2305.18474.

Chia-Yu Hung, Navonil Majumder, Zhifeng Kong, Ambuj Mehrish, Amir Zadeh, Chuan Li, Rafael Valle, Bryan Catanzaro, and Soujanya Poria. TangoFlux: Super fast and faithful text to audio generation with flow matching and CLAP-ranked preference optimization. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/f475c4cf43826411ae61c94bab360a0d-Abstract-Conference.html.

Yuxuan Jiang, Zehua Chen, Zeqian Ju, Chang Li, Weibei Dou, and Jun Zhu. FreeAudio: Training-free timing planning for controllable long-form text-to-audio generation. In *Proceedings of the 2025 ACM International Conference on Multimedia (ACM MM)*, 2025. URL https://freeaudio.github.io/FreeAudio/.

Yuxuan Jiang, Zehua Chen, Zeqian Ju, Yusheng Dai, Weibei Dou, and Jun Zhu. ControlAudio: Tackling text-guided, timing-indicated and intelligible audio generation via progressive diffusion modeling. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1394–1413, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.62. URL https://aclanthology.org/2026.acl-long.62/.

Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fréchet Audio Distance: A Reference-Free Metric for Evaluating Music Enhancement Algorithms. In *Interspeech 2019*, pp. 2350–2354, 2019. doi: 10.21437/Interspeech.2019-2219.

Chun-Yi Kuan, Kai-Wei Chang, and Hung-yi Lee. AQAScore: Evaluating semantic alignment in text-to-audio generation via audio question answering. *arXiv preprint arXiv:2601.14728*, 2026. URL https://arxiv.org/abs/2601.14728.

Sifei Li, Yang Li, Zizhou Wang, Yuxin Zhang, Fuzhang Wu, Oliver Deussen, Tong-Yee Lee, and Weiming Dong. SongEcho: Towards cover song generation via instance-adaptive element-wise linear modulation. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/1abc87c67cc400a67b869358e627fe37-Abstract-Conference.html.

Huadai Liu, Kaicheng Luo, Wen Wang, Qian Chen, Peiwen Sun, Rongjie Huang, Xiangang Li, Jieping Ye, and Wei Xue. PrismAudio: Decomposed chain-of-thoughts and multi-dimensional rewards for video-to-audio generation. In *International Conference on Learning Representations*, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/2aca4acf8af2a649baaefb633c26a75f-Abstract-Conference.html.

Yisu Liu, Chenxing Li, Wanqian Zhang, Wenfu Wang, Meng Yu, Ruibo Fu, Zheng Lin, Weiping Wang, and Dong Yu. DegDiT: Controllable audio generation with dynamic event graph guided diffusion transformer. *IEEE Transactions on Audio, Speech and Language Processing*, 34:1300–1311, 2026b. doi: 10.1109/TASLPRO.2026.3663920.

Xinhao Mei, Chutong Meng, Haohe Liu, Qiuqiang Kong, Tom Ko, Chengqi Zhao, Mark D. Plumbley, Yuexian Zou, and Wenwu Wang. WavCaps: A ChatGPT-assisted weakly-labelled audio captioning dataset for audio-language multimodal research. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024. URL https://arxiv.org/abs/2303.17395.

Paul Primus, Florian Schmid, and Gerhard Widmer. TACOS: Temporally-aligned audio CaptiOnS for language-audio pretraining, 2025. URL https://arxiv.org/abs/2505.07609.

Zeyue Tian, Zhaoyang Liu, Yizhu Jin, Ruibin Yuan, Liumeng Xue, Xu Tan, Qifeng Chen, Wei Xue, and Yike Guo. AudioX: A unified framework for anything-to-audio generation. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=qjJWxK3yWo.

Andros Tjandra, Yi-Chiao Wu, Baishan Guo, John Hoffman, Brian Ellis, Apoorv Vyas, Bowen Shi, Sanyuan Chen, Matt Le, Nick Zacharov, Carleigh Wood, Ann Lee, and Wei-Ning Hsu. Meta Audiobox aesthetics: Unified automatic quality assessment for speech, music, and sound. *arXiv preprint arXiv:2502.05139*, 2025. URL https://arxiv.org/abs/2502.05139.

Mason L. Wang and Cheng-Zhi Anna Huang. Latent Fourier transform. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/3ae02df8061edeba2acc6f79294fa8e0-Abstract-Conference.html.

Yuanyuan Wang, Hangting Chen, Dongchao Yang, Zhiyong Wu, Helen Meng, and Xixin Wu. AudioComposer: Towards fine-grained audio generation with natural language descriptions, 2024. URL https://arxiv.org/abs/2409.12560v1.

Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2023. URL https://github.com/LAION-AI/CLAP.

Zeyu Xie, Xuenan Xu, Zhizheng Wu, and Mengyue Wu. PicoAudio: Enabling precise timestamp and frequency controllability of audio events in text-to-audio generation, 2024. URL https://arxiv.org/abs/2407.02869.

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. *arXiv preprint arXiv:2509.17765*, 2025. URL https://arxiv.org/abs/2509.17765.

Zihao Zheng, Zeyu Xie, Xuenan Xu, Wen Wu, Chao Zhang, and Mengyue Wu. PicoAudio2: Temporal controllable text-to-audio generation with natural language description, 2025. URL https://arxiv.org/abs/2509.00683v2.

Haina Zhu, Yao Xiao, Xiquan Li, Ziyang Ma, Jianwei Yu, Bowen Zhang, Mingqi Yang, and Xie Chen. Audio ControlNet for fine-grained audio generation and editing. *arXiv preprint arXiv:2602.04680*, 2026. URL https://arxiv.org/abs/2602.04680.

A Model and training settings

A.1 Conditioning modules

For the configuration in Table 3, we initialize Long cross-attention from the pretrained generative backbone, while the added modules are newly initialized. All branches receive the same normalized audio hidden state and contribute to the residual update in Equation 2.

Table 3: Conditioning architecture in the evaluated configuration. Parameter counts refer to joint training.

Component	Configuration
Short branch	Separate label embeddings; a shared three-layer temporal encoder of width 128; eight-head sigmoid-gated attention
Pretrained Long attention	Backbone differential attention over each complete event description and the scene-duration token; separate normalization per event
Long refiner	Width 512; eight-head cross-attention and a residual feed-forward layer
Pairwise interactions	Width 64; Short pairs within an event and Long pairs across active events
Layer placement	Pretrained Long attention in all 24 backbone blocks; added modules in alternating blocks
Temporal support	Event masks at audio latent-frame centers; memory positions excluded
Trainable parameters	Short: 11.8M; pretrained Long: 339.7M; added Long: 55.9M; total: 407.5M

Pairwise interactions. For an indexed collection $\mathcal{F} = (f_p)_{p=1}^{n}$ of n active feature vectors, each interaction module computes

$$\mathrm{Pair}(\mathcal{F}) = \begin{cases} W_{\mathrm{pair}}\,\mathrm{LN}\left(\sum_{1\le p<q\le n} (Af_p)\odot(Af_q)\right), & n \ge 2, \\ \mathbf{0}, & n < 2. \end{cases} \tag{6}$$

Here p, q index distinct members; A projects each input to a shared interaction space, $\odot$ is elementwise multiplication, and LN is layer normalization over that space. The output projection W_{pair} maps back to the backbone's hidden width and is initialized to zero; $\mathbf{0}$ has that same width. Short and Long use separate parameters. The pair sum is permutation-invariant and is not divided by the number of active members. Short interactions use label features before their output projection; Long interactions use the combined pretrained-attention and refiner features of each event.

A.2 Conditioning modes

In Full mode, we enable both branches; in Short-only and Long-only, we disable the other branch and its interaction modules. For the unconditional prediction, we disable both branches and their cross-attention context while retaining the backbone's duration conditioning elsewhere. We use standard classifier-free guidance with scale 1.5 and 50 Euler steps, without APG or CFG rescaling. For Table 1 and Figure 4, we generate each 60-second scene jointly without a global text prompt.

A.3 Training configuration

Starting from the pretrained generative backbone, we train ELTA for 20,000 Short-only updates with the Long modules frozen, followed by 180,000 joint updates. The initial stage uses 28,920 auxiliary training scenes with packed, partial-overlap, and high-overlap layouts. These are additional compositions of recordings from the main training source bank, without introducing new sources.

For joint training, we use a fixed mixture of 70% main training scenes and 30% auxiliary scenes, with 80% Full and 20% Short-only conditions in each pool, keeping both branches trainable throughout this stage.

We use AdamW with global batch eight and constant learning rates of 10^{-5} for pretrained Long attention and 10^{-4} for the added modules, while keeping text/audio encoders and all other backbone parameters frozen. Both stages use the mixture objective below, without per-source stem targets or detector losses.

A.4 Mixture training objective

For the clean latent z_0 of an encoded waveform mixture, standard Gaussian noise ϵ of the same shape, and diffusion time $\tau \in [0, 1]$, the noisy input and target velocity are

$$z_\tau = (1 - \tau)z_0 + \tau\epsilon, \qquad v^\star = \epsilon - z_0. \tag{7}$$

At generation time, we integrate the predicted velocity from $\tau = 1$ (noise) to $\tau = 0$ (audio). Diffusion time τ is distinct from audio-frame time t.

Let B be the global batch size and $\mathcal{J}_b$ the valid latent-frame indices of sample b. At frame $j \in \mathcal{J}_b$, ℓ_{bj} is the channel-averaged squared error between the predicted velocity and the corresponding target $v^\star$. Let $F_{bi} \subseteq \mathcal{J}_b$ contain the frames whose centers lie in event i's window, and let $\mathcal{V} = \{(b, i) : |F_{bi}| > 0\}$ collect all nonempty sample–event pairs in the batch. Both stages minimize

$$\mathcal{L} = \frac{1}{B}\sum_{b=1}^{B} \frac{\sum_{j\in\mathcal{J}_b} w_{bj}\ell_{bj}}{\sum_{j\in\mathcal{J}_b} w_{bj}} + \frac{0.5}{|\mathcal{V}|}\sum_{(b,i)\in\mathcal{V}} \frac{1}{|F_{bi}|}\sum_{j\in F_{bi}} \ell_{bj}. \tag{8}$$

The event term is defined as zero if $\mathcal{V}$ is empty. The frame weight w_{bj} is two at the first and last latent cells touched by any event interval, 1.5 at other frames whose centers lie inside a window, and one elsewhere. Boundary weighting takes precedence, and padding is excluded. Overlap does not increase the first term's frame weight; shared frames enter each relevant event average in the second term. Thus each nonempty event has equal weight in that term.

B Data sources and annotation

B.1 Sources and splits

Table 4 reports retained clips by source and split. The bank contains 290,677 sound-effect clips, 68,897 isolated-instrument clips, and 14,159 full-music clips; vocal material is included.

Table 4: Retained source clips by dataset and split. Source/group splitting precedes composition. Train and held-out clips total 1,815.52 and 200.11 hours before reuse and produce 300,000 and 30,000 60-second scenes, respectively. The held-out split includes development material. Scene duration includes reused clips, gaps, and overlaps.

Source	Train clips	Held-out clips
AudioSet-Strong (Hershey et al., 2021)	1,456	137
FSD50K (Fonseca et al., 2022)	30,127	3,203
TACOS (Primus et al., 2025)	7,552	809
WavCaps (Mei et al., 2024)	222,532	24,499
AudioX-IFcaps (Tian et al., 2026)	75,106	8,312
Total	336,773	36,960

AudioX-IFcaps is publicly hosted on Hugging Face.[1] We keep source, group, and exact-waveform identities disjoint across splits and reserve known AudioSet-Strong test identities. These checks do not establish the absence of perceptual near-duplicates or overlap with backbone pretraining.

[1] https://huggingface.co/datasets/HKUSTAudio/AudioX-IFcaps

B.2 Labels and descriptions

We caption audio with Qwen3-Omni-30B-A3B-Captioner and refine the text with Qwen3-Omni-30B-A3B-Instruct without another audio pass (Xu et al., 2025). These automatic annotations have not undergone collection-wide human verification.

B.3 Composition statistics

Across both splits, we compose 82,500 single-clip, 207,029 non-overlapping multi-clip, and 40,471 partially overlapping scenes, using the same event-count proportions for training and held-out scenes. These counts describe placement geometry, not continuous simultaneous activity. Low-bandwidth recordings retain that limitation after conversion to 44.1 kHz.

C Evaluation settings and supplementary results

C.1 Evaluation scenes and uncertainty

We evaluate 1,979 scenes retained from 2,000 metadata-stratified candidates. Audio-model checks require support for every original label in the isolated recording and, for overlapping events, in the reference mixture. We select scenes using reference audio only, without consulting generated outputs or removing individual labels. Reference scores therefore reflect this eligibility rule and the automatic annotations.

The resulting set contains 4,459 distinct source clips and 10,701 label occurrences, with 333/812/834 one-/two-/three-event scenes and 309 overlapping scenes. Source, group, and registered waveform identities are disjoint from project training data, although overlap with backbone pretraining is unknown. We restrict evaluation to complete 1–10-second clips that fit within individual 10-second blocks so that all external interfaces receive full labels and captions. This restriction does not change the training data's 1–60-second clip range.

After fixing checkpoints and sampling profiles, we run all four methods with two shared generation seeds, yielding 3,958 outputs and 8,918 event windows per method. We use one reference mixture per scene and obtain 21,402 Identity QA answers per generator, with no invalid answers or execution errors.

Paired differences. For ELTA minus PicoAudio2, Audio ControlNet, and AudioComposer, respectively, we obtain scene-clustered 95% intervals of $[0.0526, 0.0737]$, $[0.0464, 0.0630]$, and $[0.0876, 0.1072]$ for Identity QA, and $[0.0106, 0.0224]$, $[0.0210, 0.0303]$, and $[0.0634, 0.0731]$ for Long-CLAP. The corresponding PQ intervals are $[-0.0787, -0.0139]$, $[0.2336, 0.2925]$, and $[0.1531, 0.2200]$. These support the semantic gains and mixed quality ordering reported in Table 1.

C.2 External systems and comparison scope

In the main table, we compare time-conditioned systems with different backbones, training data, and inference workflows, using the fixed input and sampling profiles in Table 5. We do not compare with Generate-and-Render, which generates events separately before waveform mixing, or claim superiority over that workflow.

For each external request, we prefix the original Long description with the complete label list and also supply labels and intervals through the available temporal interface. Each active block is generated jointly, without reference-derived audio, pitch, or loudness controls. DegDiT and FreeAudio are related methods for which we do not report local quantitative comparisons.

Duration adaptations. We assemble the approximately 10-second blocks generated by the external configurations on a fixed 60-second grid, leaving empty blocks silent. The common panel retains requests supported by all interfaces. We record failures separately and do not replace them with silent outputs; slightly shorter Audio ControlNet and AudioComposer-S outputs are right-padded without time stretching.

Table 5: External-system inputs and sampling profiles. All profiles are fixed before generation; scores from other protocols are not imported.

System	Conditions and sampling
PicoAudio2 (Zheng et al., 2025)	Complete event labels and descriptions in local text, with intervals and a shared block caption. Full author implementation; 50 DDIM steps, guidance 7.5.
Audio ControlNet (Zhu et al., 2026)	Labels and times through the released events adapter; complete labels and descriptions in the block caption. Events adapter v1.0; 25 Euler steps, guidance 4.5.
AudioComposer (Wang et al., 2024)	Complete event labels and descriptions with textual start/end times. AudioComposer-S, non-EMA weights; 25 Euler time points (24 integration intervals), guidance 5.0.

PicoAudio2 retains its released temporal-mask behavior: a full-block interval contributes no local temporal-description feature, although the global caption still carries the description.

C.3 Branch ablation

We select the 100-scene subset by input metadata to match the benchmark's event counts, layouts, domains, label counts, and durations. It contains 225 distinct source clips, with 17/41/42 one-/two-/three-event scenes and 16 overlapping scenes; all scenes pass the reference checks without replacement. For each condition, we generate 200 outputs from two shared generation seeds using the main sampling settings.

For Table 2, we remove each branch and its interactions at inference while keeping initial noise, event intervals, text, and scoring queries fixed. Training includes Full and Short-only conditions, but not Long-only (Appendix A.3); this experiment measures the trained model's dependence on each branch.

For Full minus w/o Short, paired 95% intervals are $[0.1612, 0.2533]$ for Identity QA and $[0.1070, 0.1493]$ for Long-CLAP. For Full minus w/o Long, the intervals are $[0.1050, 0.1866]$ and $[0.1273, 0.1622]$, respectively. All four semantic differences favor Full.

C.4 Metrics and calibration

Identity QA. We pass each complete event window to Qwen3-Omni Instruct as 16 kHz mono audio without gain adjustment and ask a separate Yes/No question for each label using greedy decoding. We average equally over labels within an event, then events and generation seeds. Invalid answers remain in the denominator, but we do not score execution errors as No. This Yes rate measures automatic identity support, unlike AQAScore's normalized Yes/No likelihoods (Kuan et al., 2026); it is not calibrated accuracy or F1. A 30-question calibration found a missed hiss and a false positive on silence. For positive sound queries, the silence check assigns No to exact-zero controls and retains them in the denominator.

Long-CLAP. We use cosine similarity between normalized embeddings from LAION HTSAT-unfused CLAP (Wu et al., 2023), comparing the original Long description with a 48 kHz mono copy of the complete event window. We divide each window into 10-second blocks and repeat-pad the final block. Scores are weighted by real duration before event averaging, with pauses included. Long-CLAP measures overall description similarity, not the correctness of every sound or attribute.

FlexSED. We run the AudioSet-Strong FlexSED model (Hai et al., 2025) on 16 kHz mono, peak-normalized 10-second blocks with fixed identity queries. To compute inside responses, we average over frames within the requested interval, then labels and events. For similar responses on active frames, a larger fraction of detectable activity raises the window mean, so sustained sound can score higher than a valid intermittent realization. These uncalibrated responses are neither detection accuracy nor loudness, and the inside mean alone does not measure activity outside the requested interval.

Table 6: Exact inputs for the four attribute examples in Figure 4. Both events in each track use the same Short label. Intervals are specified separately from the Long text.

Panel / Short	Interval (s)	Complete Long description
(a) footsteps	$[0, 9)$	Steady footsteps on a wooden floor produce crisp, hollow taps with a short woody resonance.
	$[11, 20)$	Steady footsteps on dry leaves produce crisp crunches and papery rustling with each step.
(b) water	$[0, 9)$	Individual water drops fall into a small pool, making separated, rounded plinks with quiet gaps.
	$[11, 20)$	Water flows steadily in a small stream, making a continuous, gently splashing and gurgling sound.
(c) dog barking	$[0, 9)$	A dog gives repeated short barks at a slow, steady pace, with widely spaced pauses between barks.
	$[11, 20)$	A dog gives repeated short barks at a rapid, steady pace, with closely spaced pauses between barks.
(d) electric guitar	$[1, 9)$	An electric guitar plays with a clean tone.
	$[31, 39)$	An electric guitar plays with a distorted tone.

Placement windows and activity targets. Our intervals specify where a source clip is placed, without labeling each sound's internal activity. Treating the full interval as a positive segment-F1 target would count valid pauses as missed detections. Standard segment F1 remains appropriate when reliable active/inactive targets match the requested timing; we do not have those dense targets for this benchmark.

Production quality and audio distribution. We compute PQ with Audiobox Aesthetics (Tjandra et al., 2025), using its official 16 kHz mono preprocessing and duration-weighted window scores, averaged equally over events and then generation seeds. It predicts production quality rather than human MOS and does not assess frequencies above 8 kHz.

We compute FAD-CLAP-L2 by applying Fréchet distance (Kilgour et al., 2019) to 512-dimensional L2-normalized features from the same CLAP encoder. Each complete event window gives one feature from a 48 kHz mono copy, with repeat-padding to 10 seconds. For the generated and reference distributions, we use sample means and unbiased covariances, then average distances computed separately for each generation seed. Their scale is specific to this encoder. Both quality measures retain pauses and use no gain normalization or energy-based exclusion, while excluding empty regions outside event windows.

Coverage and uncertainty. The main comparison is complete for both generation seeds. To estimate paired 95% intervals, we draw 20,000 scene-cluster bootstrap samples for semantic metrics and 2,000 for PQ, keeping all events and both seeds of a scene together. We do not estimate a FAD confidence interval or replace missing or inapplicable scores with zeros. Appendix C.1 reports reference eligibility and the main comparison's paired intervals.

D Generation example settings

We generate the examples in Figure 4 with the 200k checkpoint, both branches active, and no global prompt. Each 60-second stereo track is generated jointly at 44.1 kHz from noise, using a shared fixed seed, Euler 50, CFG 1.5, and no APG or CFG rescaling. We apply no post-generation assembly, timeline shifts, or gain normalization. Table 6 gives the complete inputs behind the figure's shortened headings.

We plot the native audio with common color limits and no per-panel normalization, preserving the original tracks' absolute time: 0–22 s for (a–c) and 0–41 s for (d). Dashed lines mark input windows, not predicted boundaries.