SOUNDSCAPEGEN: SMART SPATIAL SOUNDSCAPE CONSTRUCTION FROM HIGH-LEVEL DESCRIPTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Generating spatial soundscapes from high-level natural-language descriptions requires not only realistic audio synthesis but also reliable control over what sound events occur, when they occur, and how they move in three-dimensional space. Existing end-to-end and compositional approaches often rely on the implicit knowledge of large language models (LLMs) for scene planning, and provide limited guarantees that the generated audio follows the intended semantic, temporal, and spatial structure. We introduce **SoundscapeGen**, a framework that represents a soundscape as a set of associated controllable sound objects and constructs it through a *Plan* $\rightarrow$ *Generate* $\rightarrow$ *Verify* $\rightarrow$ *Compose* pipeline. An acoustic scene knowledge graph (ASKG) provides explicit priors for event selection, temporal relations, source properties, and motion, while a fine-tuned physics-guided First-Order Ambisonics generator synthesizes each sound object with explicit trajectory control. Object-level semantic, temporal, and spatial verification further filters inconsistent generations before composition. Experiments show that SoundscapeGen improves event coverage and temporal consistency over both end-to-end text-to-audio and compositional baselines, while providing explicit control over individual source trajectories. Its object-based representation further supports localized interactive editing without regenerating the entire soundscape. Demos are available at `https://soundscapegeneration.github.io/SoundscapeGen/`.

1 INTRODUCTION

Spatial audio is essential for immersive applications such as virtual reality (VR), augmented reality (AR), and 3D games, where multiple sound sources distributed and moving in a three-dimensional space create a vivid soundscape around the listener. Such soundscapes typically consist of multiple acoustic events, each with its own timing, loudness, and motion, and constructing one requires substantial manual effort in sound selection, temporal coordination, and spatial arrangement. A high-level description such as "a city park in the morning" conveys the overall scene but leaves many acoustic details unspecified. Automatically translating high-level descriptions into complete spatial soundscapes can substantially reduce the efforts required for immersive audio creation.

Existing approaches to soundscape generation broadly fall into two categories: end-to-end generation and compositional generation. End-to-end text-to-audio (TTA) models (Liu et al., 2023; 2024; Ghosal et al., 2023; Majumder et al., 2024; Huang et al., 2023b;a) directly synthesize monaural audio from text, and recent spatial TTA models extend this capability to spatial audio: ImmerseDiffusion (Heydari et al., 2025) generates First-Order Ambisonics (FOA) audio for static sources, and SonicMotion (Templin et al., 2026) extends generation to moving sources. However, as scene complexity increases, end-to-end models may fail to generate all requested events or preserve their temporal and semantic relationships (He et al., 2026). Compositional approaches instead decompose a scene into individual events, generate them separately, and arrange them on a shared timeline. WavJourney (Liu et al., 2025), WavCraft (Liang et al., 2024), and SoundscapeAgent (Zhang et al., 2026) use structured planning to coordinate audio generation and composition, and TTMBA (He et al., 2025) further assigns temporal and spatial attributes to individual events for binaural soundscape generation. However, these methods largely rely on the LLM planner itself to infer scene structure. This knowledge is only implicitly encoded in the model and may not reliably reflect real-world acoustic relationships.

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.

We therefore formulate spatial soundscape generation as *relational soundscape generation*, where a scene is represented as a set of controllable sound objects, each with explicit semantic, temporal, loudness, and spatial trajectory attributes jointly determined by the acoustic relations among the objects. Creating a coherent spatial soundscape requires both a plausible arrangement of events in time and space and spatial audio that faithfully realizes the planned source behaviors.

Such soundscape generation has two key challenges. **First, how can a high-level instruction be translated into a specified soundscape plan?** A high-level instruction specifies the overall scene but leaves unresolved which sound events should occur, how they relate in time, what acoustic properties they should have, and how their sources should move. Producing a coherent plan requires reasoning over acoustic relationships among scenes, events, and source properties. For example, footsteps and speech from the same moving person should follow a consistent trajectory, and related events may follow characteristic temporal orders. Existing LLM-based planning pipelines do not represent these relationships explicitly, which makes it difficult to incorporate them consistently. The challenge is therefore to provide structured acoustic knowledge that grounds event selection, temporal organization, source characteristics, and motion planning. **Second, how can we ensure that the generated spatial audio satisfies all planned attributes?** Even when a sound object is specified with target semantics, timing, and trajectory, generation may still miss the intended event, produce incorrect temporal activity, or deviate from the planned spatial motion. Physical constraints on direction and distance can mitigate spatial deviation, but stochastic generation does not guarantee that every sampled object satisfies all planned attributes. Reliable soundscape generation therefore requires explicit verification of each generated object before final composition.

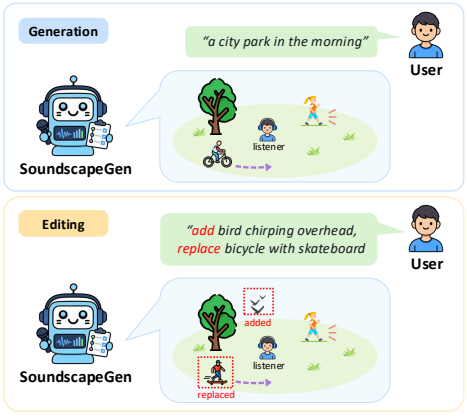


Figure 1: SoundscapeGen enables soundscape generation and interactive editing through natural-language descriptions.

To address these challenges, we introduce **SoundscapeGen**, a framework for relational spatial soundscape generation from high-level scene descriptions. SoundscapeGen follows a *Plan* → *Generate* → *Verify* → *Compose* pipeline in which sound objects remain separate until final composition. For scene planning, we construct an acoustic scene knowledge graph (ASKG) that provides structured acoustic priors over scene-event associations, inter-event relations, source characteristics, and typical motion patterns. These priors ground an LLM-based planner as it translates a high-level description into an executable collection of sound objects. For acoustic realization, each planned object is independently synthesized as a controllable First-Order Ambisonics (FOA) signal under explicit physical consistency constraints, and then verified against its planned semantics, timing, and trajectory. Objects that fail verification are regenerated before scene composition. This object-level separation naturally supports localized editing, including adding or removing events, modifying their timing, loudness, or trajectories, and regenerating individual sources without reconstructing the entire scene. Figure 1 illustrates an example of both generation and editing.

In summary, our main contributions are:

- **Relational spatial soundscape generation.** We formulate spatial soundscape generation as *relational soundscape generation*, representing a scene as controllable sound objects whose attributes are coordinated by their acoustic relations. Our *Plan* → *Generate* → *Verify* → *Compose* framework, SoundscapeGen, outperforms TTA and compositional baselines in semantic, temporal, and spatial consistency while supporting localized editing.
- **Knowledge-grounded scene planning.** We construct an Acoustic Scene Knowledge Graph (ASKG) covering 50 scenes and 140 sound events that provides structured acoustic priors for event selection, temporal relationships, source characteristics, and motion patterns.
- **Physics-guided generation with explicit verification.** We build a large-scale FOA dataset spanning 140 event categories and five motion types for controllable single-source generation under explicit physical consistency constraints, and introduce a verification module that checks generated objects against their planned semantics, timing, and trajectory before composition.

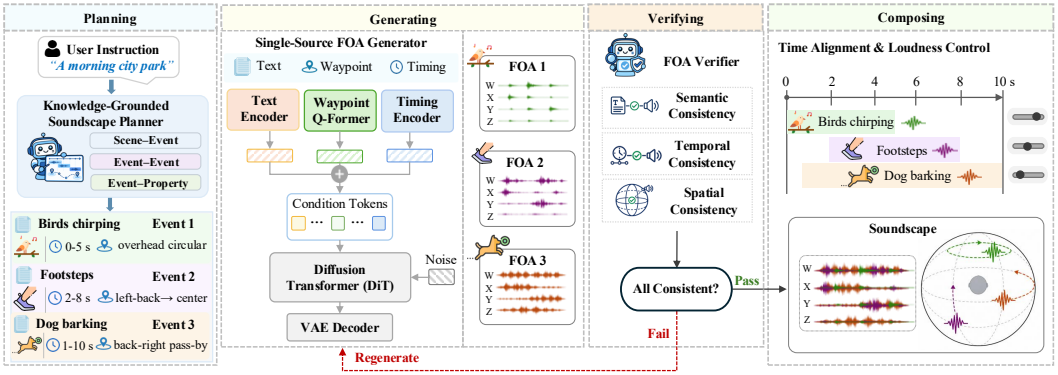


Figure 2: Overview of SoundscapeGen. An ASKG-grounded planner converts the instruction into sound objects, each of which is generated as a single-source FOA signal, verified for semantic, temporal, and spatial consistency (regenerated upon failure), and composed into the final soundscape.

2 Related Work

2.1 End-to-End Text-to-Audio Generation

Text-to-audio (TTA) models such as AudioGen (Kreuk et al., 2023), the AudioLDM series (Liu et al., 2023; 2024), the Make-An-Audio series (Huang et al., 2023b;a), the TANGO series (Ghosal et al., 2023; Majumder et al., 2024), and Stable Audio (Evans et al., 2025) synthesize realistic audio directly from natural-language descriptions, achieving strong semantic alignment and perceptual quality. Spatial TTA extends this paradigm to three-dimensional audio. Diff-SAGe (Kushwaha et al., 2025) and ImmerseDiffusion (Heydari et al., 2025) generate First-Order Ambisonics (FOA) audio with spatial conditioning, and SonicMotion (Templin et al., 2026) and PhysWave (Yao et al., 2026) further support controllable moving sources. However, these models produce the entire scene as a single signal. As scene complexity increases, they may miss requested events or fail to preserve the relationships among them (He et al., 2026). In contrast, SoundscapeGen represents and generates each sound event independently, enabling explicit control over individual events and their relationships within a complex soundscape.

2.2 Controllable Soundscape Generation

Compositional methods decompose complex audio creation into planning, generation, and composition. WavJourney (Liu et al., 2025) converts high-level instructions into structured audio scripts, and WavCraft (Liang et al., 2024) supports interactive creation and editing through modular audio operations. More recent systems, including AudioGenie (Rong et al., 2025), Audio-Oscar (Duan et al., 2026), and SoundscapeAgent (Zhang et al., 2026), coordinate complex audio creation through agent-based planning. However, these methods largely rely on the implicit knowledge of LLMs to infer scene structure and acoustic relations. They also do not verify each generated sound object against its planned semantic, temporal, and spatial attributes before composition. SoundscapeGen instead grounds scene planning with an Acoustic Scene Knowledge Graph, represents each event as a controllable FOA sound object with semantic, temporal, loudness, and trajectory attributes, and performs object-level verification before final composition.

3 SoundscapeGen

3.1 Overview

SoundscapeGen generates controllable multi-source spatial soundscapes from high-level natural-language instructions. Rather than synthesizing an entire scene jointly, we represent a soundscape as a collection of controllable sound objects, each characterized by its semantic, temporal, gain, and spatial attributes. As shown in Figure 2, SoundscapeGen follows a *Plan* $\rightarrow$ *Generate* $\rightarrow$ *Verify* $\rightarrow$

Compose pipeline. A knowledge-grounded LLM planner first converts the input description into a structured set of sound objects. Each object is then independently generated as a single-source First-Order Ambisonics (FOA) signal and verified against its planned attributes, and objects that fail verification are regenerated. Finally, the verified objects are composed into the target spatial soundscape. The object-based representation also supports localized editing without regenerating unaffected parts of the scene.

3.2 Object-Based Soundscape Representation

We represent a spatial soundscape of duration T as a set of N sound objects,

$$\mathcal{S} = \{e_i\}_{i=1}^{N}, \tag{1}$$

where each object is defined as

$$e_i = \big(c_i,\ t_i^{\mathrm{on}},\ t_i^{\mathrm{off}},\ g_i,\ \tau_i\big), \tag{2}$$

where c_i is the event category, $[t_i^{\mathrm{on}}, t_i^{\mathrm{off}}] \subseteq [0, T]$ is the active interval, $g_i \in \mathbb{R}_{>0}$ is the relative gain, and τ_i is the listener-relative spatial trajectory. We parameterize τ_i by M_i waypoints,

$$\tau_i = \big\{(t_{i,m}, \theta_{i,m}, \phi_{i,m}, r_{i,m})\big\}_{m=1}^{M_i}, \qquad t_{i,m} \in [0, d_i], \tag{3}$$

where $d_i = t_i^{\mathrm{off}} - t_i^{\mathrm{on}}$ is the object duration, and $\theta_{i,m}$, $\phi_{i,m}$, and $r_{i,m}$ denote the azimuth, elevation, and source distance at object-local time $t_{i,m}$. Linearly interpolating the waypoints yields a continuous trajectory $(\theta_i(t), \phi_i(t), r_i(t))$.

Each object is realized as a four-channel FOA signal $\mathbf{A}_i(t) = [W_i(t), X_i(t), Y_i(t), Z_i(t)]^\top$, $t \in [0, d_i]$, where W_i is the omnidirectional component and (X_i, Y_i, Z_i) are the directional components. Similar to PhysWave (Yao et al., 2026), a point source with waveform $s_i(t)$ arriving from direction $(\theta_i(t), \phi_i(t))$ is encoded as

$$\mathbf{A}_i(t) = s_i(t)\big[\tfrac{1}{\sqrt{2}},\ \cos\phi_i(t)\cos\theta_i(t),\ \cos\phi_i(t)\sin\theta_i(t),\ \sin\phi_i(t)\big]^\top, \tag{4}$$

where the last three entries form the unit direction vector $\mathbf{n}_i(t)$.

Because FOA encoding is linear in the source signals, the final soundscape is obtained by temporally placing and summing the verified objects:

$$\mathbf{A}(t) = \sum_{i=1}^{N} g_i\, \mathcal{T}_{t_i^{\mathrm{on}}}\big(\tilde{\mathbf{A}}_i\big)(t), \qquad \mathcal{T}_{t_0}\big(\mathbf{A}\big)(t) = \begin{cases} \mathbf{A}(t - t_0), & t \in [t_0,\, t_0 + d_i], \\ \mathbf{0}, & \text{otherwise}, \end{cases} \tag{5}$$

where $\tilde{\mathbf{A}}_i$ denotes the verified FOA signal of object e_i. This additive formulation keeps sound objects independent until composition, which facilitates object-level generation, verification, and editing.

3.3 ASKG-Guided LLM Soundscape Planner

We construct an Acoustic Scene Knowledge Graph (ASKG), denoted $\mathcal{G}$, to provide explicit acoustic priors for soundscape planning. Its nodes represent acoustic scenes and sound events, connected by three types of relations: scene–event, event–event, and event–property relations.

Unified event taxonomy. We first unify event labels from public audio datasets (He et al., 2026; Wang et al., 2021; Piczak, 2015) through semantic label mapping. Semantically equivalent labels are merged into a common category, and the result is manually checked to remove duplicated and ambiguous categories. This process results in a unified taxonomy of 140 sound-event categories, shared by the ASKG and the downstream sound-object generator.

Scene–event relations. A scene–event relation:

$$(s, \texttt{contains}, e) \tag{6}$$

indicates that sound event e is acoustically plausible in scene s. We manually define 50 common acoustic scenes and assign candidate events to each scene from the unified taxonomy. The resulting ASKG contains 511 `contains` relations covering all 140 event categories. These relations provide the set of candidate events to complete a high-level scene description.

Event–event relations. To represent dependencies among sound events, we define three types of event–event relations,

$$\mathcal{R}_{EE} = \{\texttt{co-occurs}, \texttt{precedes}, \texttt{same-source}\}, \tag{7}$$

where `co-occurs` links events that naturally appear together, `precedes` encodes a typical temporal order, and `same-source` links events produced by the same physical source. For each scene, we enumerate all event pairs connected to it through `contains` relations and use an LLM to identify plausible relations between them. The generated relations are manually reviewed before being added to the ASKG.

Event–property relations. For each event e, we use an LLM to generate its typical acoustic properties,

$$\mathcal{P}(e) = \{\mathcal{D}_e,\ \mathcal{L}_e,\ \mathcal{M}_e\}, \tag{8}$$

where $\mathcal{D}_e$ and $\mathcal{L}_e$ denote the typical duration and relative-gain ranges, and $\mathcal{M}_e$ is a set of candidate motion patterns drawn from the five motion types in Sec. 3.4. These properties are manually reviewed and serve as priors from which the planner selects values according to the scene context and the user instruction.

ASKG-guided scene planning. Given a natural-language instruction $\mathcal{I}$, the LLM first identifies one or more scene categories that best match it. We then retrieve the events connected to these scenes, together with their event–event relations and properties, to form the acoustic context

$$\mathcal{K}_{\mathcal{I}} = \text{Retrieve}\,(\mathcal{G}, \mathcal{I})\,. \tag{9}$$

Conditioned on $\mathcal{K}_{\mathcal{I}}$, the planner generates the structured soundscape specification

$$\mathcal{S} = P_{\text{LLM}}\,(\mathcal{I}, \mathcal{K}_{\mathcal{I}})\,. \tag{10}$$

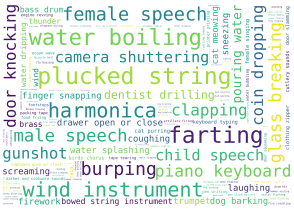
The planner selects events from the candidate set and then determines their active intervals, relative gains, and trajectories, with explicit user requirements taking precedence over retrieved priors. Event–event relations coordinate these decisions across objects: `co-occurs` informs event selection, `precedes` constrains temporal ordering, and `same-source` enforces a shared trajectory for events from the same physical source. The resulting plan follows the representation in Sec. 3.2 and directly provides the conditions for single-source FOA generation.

3.4 Single-Source FOA Generation

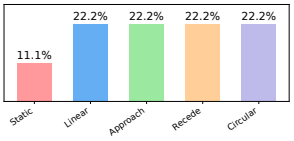
Single-source FOA dataset with trajectories. To support object-level generation and verification, we construct a single-source FOA dataset based on the unified 140-category taxonomy introduced in Sec. 3.3. We collect monaural recordings from AudioEventSet (He et al., 2026), FSD-MIX-SED (Wang et al., 2021), and ESC-50 (Piczak, 2015), map their labels to the unified taxonomy, and remove samples with low audio–text alignment based on CLAP similarity. For each retained waveform s_k with category c_k, we sample listener-relative trajectories $\tau_{k,j}$ from five motion types: static, linear pass-by, circular, approach, and recede. Each waveform is then spatially rendered as

$$\mathbf{A}_{k,j} = \mathcal{R}\,(s_k, \tau_{k,j})\,, \tag{11}$$

where $\mathcal{R}$ denotes spatial rendering, including distance-dependent attenuation, FOA encoding Eq. (4), and time-varying propagation delay for moving sources. The resulting dataset contains 404.4 hours of FOA audio (145,566 clips). Figure 3 summarizes its event-category coverage and trajectory distribution, and Appendix B provides the split statistics.



(a) Event categories in the dataset



(b) Distribution of source trajectories

Figure 3: Statistics of the constructed FOA dataset.

Physics-guided FOA generation. We employ the model in PhysWave (Yao et al., 2026), a physics-guided latent diffusion model for controllable single-source FOA generation, and fine-tune it on our dataset to cover the unified event taxonomy. Given a planned object e_i, the model encodes the four-channel FOA waveform into a latent space and performs denoising jointly conditioned on the event category c_i and trajectory τ_i:

$$\mathbf{A}_i = G_{\mathrm{FOA}}\left(c_i, \tau_i; \boldsymbol{\epsilon}_i\right), \qquad \boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{12}$$

where $\boldsymbol{\epsilon}_i$ denotes the diffusion sampling noise. During training, the diffusion objective is combined with direction- and distance-consistency losses that encourage the generated signal to follow the prescribed trajectory. The model architecture and training objectives are detailed in Appendix D.

3.5 Sound Object Verification

Although each sound object is generated under explicit semantic and trajectory conditions, stochastic sampling may still produce outputs that deviate from the plan. We therefore verify each generated object along three dimensions: semantic content, temporal activity, and spatial trajectory. Objects that fail any criterion are regenerated with new diffusion noises. All thresholds and implementation details are provided in Appendix F.

Semantic verification. Since the FOA W channel captures the omnidirectional acoustic content, we verify whether the generated object contains the target event c_i by computing the CLAP similarity (Wu et al., 2023) between W_i and the event label,

$$s_i^{\mathrm{sem}} = \mathrm{CLAP}(W_i, c_i), \tag{13}$$

and accept the object when $s_i^{\mathrm{sem}} \geq \gamma_{\mathrm{sem}}$.

Temporal verification. We verify whether the target event is active over a sufficient fraction of the planned duration. We divide W_i into K equal-length segments and apply the open-vocabulary event detector OpenFLAM (Wu et al., 2025) to obtain an activity score $a_{i,k}$ for event c_i in each segment k. The temporal score is the fraction of active segments,

$$s_i^{\mathrm{tmp}} = \frac{1}{K} \sum_{k=1}^{K} \mathbb{1}\big[a_{i,k} \geq \delta\big], \tag{14}$$

where δ is the activity threshold. The object is accepted when $s_i^{\mathrm{tmp}} \geq \gamma_{\mathrm{tmp}}$.We use Ω_i to denote the set of time frames within the active segments, which is further used for spatial verification.

Spatial verification. We verify whether the generated signal follows the planned trajectory τ_i. For the direction, we estimate the frame-wise direction of arrival $\hat{\mathbf{n}}_i(t)$ from the FOA intensity vector $\big[\langle W_i X_i\rangle, \langle W_i Y_i\rangle, \langle W_i Z_i\rangle\big]^\top$ and compute its mean angular error with respect to the planned direction $\mathbf{n}_i(t)$ over active frames,

$$e_i^{\mathrm{ang}} = \frac{1}{|\Omega_i|} \sum_{t \in \Omega_i} \angle\big(\hat{\mathbf{n}}_i(t), \mathbf{n}_i(t)\big). \tag{15}$$

For trajectories with time-varying distance, we further compute the Pearson correlation ρ_i^{dist} between the log-energy envelope of W_i and the log inverse-square profile $\log\big(1/r_i(t)^2\big)$ over active frames, and require $\rho_i^{\mathrm{dist}} \geq \gamma_{\mathrm{dist}}$. In addition, all objects must satisfy $e_i^{\mathrm{ang}} \leq \gamma_{\mathrm{ang}}$ to pass spatial verification.

3.6 Composition and Interactive Editing

Scene composition. Each verified object $\tilde{\mathbf{A}}_i$ is placed at its planned onset and scaled by its relative gain g_i. After that, all objects are added channel by channel, following Eq. (5). A shared global gain is applied to prevent clipping while preserving the relative gains specified in the scene plan.

Interactive editing. The object-based representation naturally supports localized editing through natural-language instructions. Given an existing scene plan $\mathcal{S}$ and an editing instruction $\mathcal{I}_{\text{edit}}$, the planner updates the affected sound objects only, i.e.,

$$\mathcal{S}' = P_{\text{edit}}\left(\mathcal{S}, \mathcal{I}_{\text{edit}}\right). \tag{16}$$

SoundscapeGen supports adding or removing events and modifying their timing, relative gain, or trajectories. Only objects whose category, duration, or trajectory changes are regenerated and re-verified. Onset shifts and gain changes are applied directly at composition. Some examples of interactive editing are provided in Appendix J.

4 Experiments

4.1 Experimental Setup

Implementation details. We use DeepSeek-V4-Flash (Xu et al., 2026) as the planner and PhysWave (Yao et al., 2026) fine-tuned on our FOA dataset (Sec. 3.4) as the object generator. Verification uses LAION-CLAP (Wu et al., 2023) for semantic consistency, OpenFLAM (Wu et al., 2025) for temporal activity, and intensity-based estimation for direction and distance (Sec. 3.5). In each round, we sample three candidates per object and verify the one with the highest CLAP similarity; failed objects are resampled for up to five rounds, after which the best-scoring candidate is kept. Further details are provided in Appendices E and F.

Evaluation details. Single-object generation is evaluated on the test split of our FOA dataset using FD (Kong et al., 2020), FAD (Kilgour et al., 2018), CLAP (Wu et al., 2023), and KL on the W channel, together with angular and distance errors. For multi-source generation, we construct 100 high-level scene descriptions with three to five events each, paired with manually verified reference specifications of event categories, active intervals, and trajectories. We report Event Recall (fraction of reference events detected), mean temporal IoU (mT-IoU) with reference intervals, and Spatial Accuracy (Spat. Acc., fraction of objects matching their reference trajectories). Ten participants further rate semantic, temporal, and spatial consistency and overall realism on a five-point MOS scale. Metric and baseline configurations are detailed in Appendix G.

Table 1: Single-object FOA generation on the test split of our FOA dataset. Audio-quality metrics are computed on the omnidirectional W channel. Spatial errors are computed only on the trajectory types each method supports.

Method	**Trajectory**	**Audio Quality**				**Spatial Fidelity**	
		FD ↓	FAD ↓	CLAP ↑	KL ↓	Ang. Err. (°) ↓	Dist. Err. (dB) ↓
ImmerseDiffusion	Static	48.79	6.04	0.19	3.42	19.68	8.87
SonicMotion	Static, circular	42.49	5.49	0.21	3.27	30.25	8.05
PhysWave	Arbitrary	43.10	4.62	0.23	2.91	2.17	5.78
Ours	Arbitrary	**27.14**	**4.08**	**0.29**	**2.25**	**1.52**	**5.31**

4.2 Single-Object Generation

We first evaluate single-object generation before multi-source composition. We compare our fine-tuned generator with ImmerseDiffusion (Heydari et al., 2025), which supports static sources, SonicMotion (Templin et al., 2026), which supports static and circular motion, and the original PhysWave (Yao et al., 2026), which supports arbitrary waypoint-based trajectories.

As shown in Table 1, our generator achieves the best results on all metrics. Compared with the original PhysWave, fine-tuning reduces FD from 43.10 to 27.14 and KL from 2.91 to 2.25, and increases CLAP from 0.23 to 0.29, indicating substantially better semantic and acoustic quality on the 140 target categories. It also reduces angular error from 2.17° to 1.52° and distance error from 5.78 to 5.31 dB. In contrast, ImmerseDiffusion and SonicMotion exhibit much larger angular errors even on the trajectory types they support, highlighting the benefit of explicit waypoint conditioning.

Table 2: Multi-source soundscape generation. Objective metrics are reported as mean ± standard deviation over three random seeds. MOS is rated on a five-point scale. Spat. Acc. is reported only for SoundscapeGen, since the baselines lack trajectory control. Blue and orange rows denote end-to-end TTA and compositional methods, respectively.

Method	Objective			Subjective MOS			
	Recall (%) ↑	mT-IoU ↑	Spat. Acc. (%) ↑	Sem. ↑	Temp. ↑	Spat. ↑	Real. ↑
AudioLDM 2	72.00 ± 1.93	0.31 ± 0.01	–	2.10	2.13	2.10	2.10
Stable Audio Open	47.80 ± 3.42	0.16 ± 0.02	–	2.10	2.13	2.20	1.90
TangoFlux	69.14 ± 3.02	0.33 ± 0.04	–	3.00	2.93	2.80	2.58
WavJourney	77.88 ± 2.16	0.34 ± 0.01	–	3.90	3.40	3.33	3.58
SoundscapeAgent	88.63 ± 0.46	0.51 ± 0.00	–	3.63	3.56	3.45	3.30
Ours	**95.36** ± 0.54	**0.52** ± 0.01	**84.31** ± 0.49	**4.05**	**3.90**	**3.78**	**3.63**

4.3 Multi-Source Soundscape Generation

We compare SoundscapeGen with three end-to-end TTA models, AudioLDM 2 (Liu et al., 2024), Stable Audio Open (Evans et al., 2025), and TangoFlux (Hung et al., 2026), and two compositional methods, WavJourney (Liu et al., 2025) and SoundscapeAgent (Zhang et al., 2026). All methods are evaluated on the same 100 scene descriptions and conditioned on the same scene plans produced by our planner, so that differences reflect generation and composition rather than planning. End-to-end models receive each plan as a natural-language prompt describing event timing and spatial attributes, while compositional methods take it as structured input (Appendix G).

As shown in Table 2, compositional methods outperform end-to-end TTA models in Event Recall and mT-IoU, since generating events separately prevents them from being missed or temporally misaligned. SoundscapeGen further raises Event Recall to 95.36%, realizing nearly all planned events, while matching the strongest baseline in mT-IoU (0.52 vs. 0.51). It is also the only method with trajectory control, with 84.31% of generated objects following their planned trajectories. Subjective results are consistent. SoundscapeGen obtains the highest MOS on all four dimensions, with the largest margins over the strongest baseline in temporal (3.90 vs. 3.56) and spatial consistency (3.78 vs. 3.45), suggesting that listeners perceive the intended event timing and source motion. Its realism score of 3.63 further shows that object-level composition maintains natural-sounding soundscapes while providing explicit temporal and spatial control.

Table 3: Evaluation of ASKG-guided planning. The LLM judge scores event relevance, temporal consistency, same-source consistency, and property validity. Human participants rate the realism of the soundscapes synthesized from each plan. All scores are on a 1–5 scale.

Method	LLM Judge				Human
	Relevance ↑	Temporal ↑	Source ↑	Property ↑	
LLM-only	4.66	4.71	4.77	4.74	3.48
ASKG-grounded	**4.83**	**4.73**	**4.82**	**4.81**	**3.63**

4.4 ASKG-Guided Planning Evaluation

We evaluate whether explicit acoustic knowledge improves soundscape planning. For each high-level scene description, we generate two plans using the same LLM: an *LLM-only* plan produced directly from the input, and an *ASKG-grounded* plan conditioned on retrieved acoustic knowledge. Each plan is scored independently by GPT-4o, which differs from the planner model to avoid self-preference bias, along four dimensions: event relevance, temporal consistency, same-source consistency, and property validity. Human participants additionally rate the realism of the soundscapes synthesized from each plan. Evaluation criteria and prompts are provided in Appendix H.

As shown in Table 3, ASKG grounding consistently improves plan quality under the LLM judge. The gains concentrate on event relevance, property validity, and same-source consistency, which correspond directly to the scene–event, event–property, and `same-source` relations encoded in the ASKG. In contrast, temporal consistency remains nearly unchanged, as typical temporal orders are already well captured by the base LLM. The listening test further confirms that these improvements carry over to the final audio: soundscapes synthesized from ASKG-grounded plans receive higher realism scores (3.63 vs. 3.48).

Table 4: Ablation study of SoundscapeGen on the multi-source benchmark. Real. MOS denotes the subjective realism score on a five-point scale.

Variant	Recall (%) ↑	mT-IoU ↑	Spat. Acc. (%) ↑	Real. MOS ↑
w/o ASKG	87.87	0.45	83.10	3.48
w/o FOA fine-tuning	90.72	0.50	80.25	2.72
w/o verification	91.65	0.51	83.55	2.64
SoundscapeGen	**95.36**	**0.52**	**84.31**	**3.63**

4.5 Ablation Study

Table 4 evaluates the contribution of each component. Removing ASKG causes the largest drops in Event Recall (95.36% → 87.87%) and mT-IoU (0.52 → 0.45), showing that knowledge-grounded planning is key to selecting appropriate events and organizing them in time; the lower realism MOS (3.48) further suggests that more plausible plans yield more realistic soundscapes. Removing FOA fine-tuning causes the largest drop in Spatial Acc. (84.31% → 80.25%) and reduces realism MOS to 2.72, consistent with the lower audio quality of the original PhysWave on our taxonomy (Table 1). Removing verification reduces Event Recall to 91.65% and causes the largest drop in realism MOS (2.64), indicating that verification not only recovers missing or incorrect events but also filters out flawed generations. Overall, each component contributes to a different aspect of generation quality, and the full model achieves the best results on all metrics.

4.6 Complexity Scaling

To examine how performance changes with scene complexity, we construct an additional benchmark with one to five events per scene (Appendix G). As shown in Figure 4, TangoFlux achieves Event Recall comparable to SoundscapeGen with a single event, but its recall falls below 70% at five events, and AudioLDM 2 shows a similar decline. This indicates that end-to-end models struggle to accommodate multiple events within a single signal, causing events to be missed or temporally misaligned. In contrast, SoundscapeGen maintains Event Recall above 90% at all complexity levels, as each event is generated independently. mT-IoU decreases for all methods as more events overlap in time, likely because overlapping events make interval detection harder, yet SoundscapeGen consistently outperforms the baselines by a large margin.

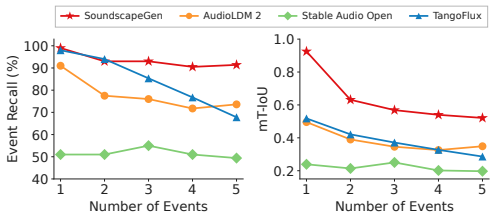


Figure 4: Event Recall and mT-IoU under increasing scene complexity, measured by the number of events per scene.

5 Conclusion

We presented SoundscapeGen, a framework that formulates spatial soundscape generation from high-level descriptions as relational soundscape generation over controllable sound objects. Through ASKG-grounded planning, physics-guided FOA generation, and object-level verification, SoundscapeGen achieves higher event recall and stronger perceived temporal and spatial consistency than end-to-end and compositional baselines, while supporting explicit trajectory control and localized editing. Future work will extend the ASKG to broader scenes and richer relations, such as spatial layouts and causal dependencies, to enable more natural and diverse soundscapes.

AI USE STATEMENT

In this work, we used generative AI tools to assist with the generation of synthetic data and to provide feedback on experimental design. We did not use generative AI tools to formulate or refine hypotheses, implement methods, translate content, clean or reformat datasets, or interpret experimental results. Tasks related to mathematical claims, mathematical proofs, and qualitative or thematic data analysis are not applicable to this work.

Additionally, we used generative AI tools to improve the readability of the manuscript and assist with formatting. All AI-assisted data generation procedures and experimental design suggestions were reviewed and validated by the authors. All AI-assisted manuscript edits were also reviewed by the authors. We take responsibility for the final content of this work, including text, claims, data, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our work focuses on controllable spatial soundscape generation. The human evaluation was limited to a low-risk audio listening test in which participants rated the perceptual quality of audio samples. No sensitive or personally identifiable information was collected.

The datasets used in this work are derived from publicly available resources, and we follow their respective licenses and terms of use when constructing derived data.

REFERENCES

Yifan Duan, Qixiang Xu, Hengtao Wu, Zhanxun Liu, Wenhao Guan, Junxi Liu, Ziyang Ma, Kelu Xu, and Xie Chen. Audio-oscar: A multi-agent system for complex audio scene generation, orchestration, and refinement. *arXiv preprint arXiv:2606.07397*, 2026.

Zach Evans, Julian D Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Stable audio open. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Deepanway Ghosal, Navonil Majumder, Ambuj Mehrish, and Soujanya Poria. Text-to-audio generation using instruction guided latent diffusion model. In *Proceedings of the 31st ACM international conference on multimedia*, pp. 3590–3598, 2023.

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 (ICLR)*, 2026.

Yuxuan He, Xiaoran Yang, Ningning Pan, and Gongping Huang. Ttmba: Towards text to multiple sources binaural audio generation. In *Proc. Interspeech 2025*, pp. 4228–4232, 2025.

Mojtaba Heydari, Mehrez Souden, Bruno Conejo, and Joshua Atkins. Immersediffusion: A generative spatial audio latent diffusion model. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

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. *arXiv preprint arXiv:2305.18474*, 2023a.

Rongjie Huang, Jiawei Huang, Dongchao Yang, Yi Ren, Luping Liu, Mingze Li, Zhenhui Ye, Jinglin Liu, Xiang Yin, and Zhou Zhao. Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models. In *International conference on machine learning*, pp. 13916–13932. PMLR, 2023b.

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*, volume 2026, pp. 150793–150816, 2026.

Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fréchet audio distance: A metric for evaluating music enhancement algorithms. *arXiv preprint arXiv:1812.08466*, 2018.

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

Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi. Audiogen: Textually guided audio generation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=CYK7RfcOzQ4.

Saksham Singh Kushwaha, Jianbo Ma, Mark RP Thomas, Yapeng Tian, and Avery Bruni. Diffsage: End-to-end spatial audio generation using diffusion models. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Jinhua Liang, Huan Zhang, Haohe Liu, Yin Cao, Qiuqiang Kong, Xubo Liu, Wenwu Wang, Mark D Plumbley, Huy Phan, and Emmanouil Benetos. Wavcraft: Audio editing and generation with large language models. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*, 2024. URL https://openreview.net/forum?id=xJw7x2ZBex.

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023.

Haohe Liu, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Qiao Tian, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D Plumbley. Audioldm 2: Learning holistic audio generation with self-supervised pretraining. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:2871–2883, 2024.

Xubo Liu, Zhongkai Zhu, Haohe Liu, Yi Yuan, Qiushi Huang, Meng Cui, Jinhua Liang, Yin Cao, Qiuqiang Kong, Mark D Plumbley, et al. Wavjourney: Compositional audio creation with large language models. *IEEE Transactions on Audio, Speech and Language Processing*, 33:2830–2844, 2025.

Navonil Majumder, Chia-Yu Hung, Deepanway Ghosal, Wei-Ning Hsu, Rada Mihalcea, and Soujanya Poria. Tango 2: Aligning diffusion-based text-to-audio generations through direct preference optimization. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 564–572, 2024.

Karol J Piczak. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 1015–1018, 2015.

Yan Rong, Jinting Wang, Guangzhi Lei, Shan Yang, and Li Liu. Audiogenie: A training-free multi-agent framework for diverse multimodality-to-multiaudio generation. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 8872–8881, 2025.

Christian Templin, Yanda Zhu, and Hao Wang. Generating moving 3d soundscapes with latent diffusion models. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2856–2860. IEEE, 2026.

Yu Wang, Nicholas J Bryan, Justin Salamon, Mark Cartwright, and Juan Pablo Bello. Who calls the shots? rethinking few-shot learning for audio. In *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 36–40. IEEE, 2021.

Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Yusong Wu, Christos Tsirigotis, Ke Chen, Cheng-Zhi Anna Huang, Aaron Courville, Oriol Nieto, Prem Seetharaman, and Justin Salamon. Flam: Frame-wise language-audio modeling. In *International Conference on Machine Learning*, pp. 67719–67740. PMLR, 2025.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Lingfeng Yao, Chenpei Huang, Xingke Yang, Ziye Geng, Changqing Luo, Hao Wang, Jiang Liu, and Miao Pan. Physwave: Physics-guided latent diffusion models for controllable spatial audio generation. *arXiv preprint arXiv:2608.29549*, 2026.

Hao Zhang, Yiwen Zhao, Yixuan Zhang, Yiwen Shao, and Steve Yves. Soundscapeagent: Agentic soundscape construction for controllable synthesis and scalable audio-language supervision. *arXiv preprint arXiv:2607.21857*, 2026.

A Comparison with Prior Work

Table 5 compares SoundscapeGen with representative end-to-end and compositional methods. End-to-end models synthesize a scene as a single signal and therefore cannot control individual sources. Among spatial variants, ImmerseDiffusion and SonicMotion support only limited trajectory types, while PhysWave supports arbitrary trajectories with physics constraints but generates only one source at a time. Compositional methods handle multiple sources but rely on knowledge implicitly encoded in the LLM planner and do not verify generated objects against their planned attributes. SoundscapeGen combines multi-source composition with explicit acoustic knowledge, arbitrary 3D trajectories, physics-guided generation, and object-level verification.

Table 5: Functional comparison with prior methods. Blue and orange rows denote end-to-end and compositional methods, respectively. Multi-source indicates whether a method supports generating audio scenes containing multiple sound sources or events. Explicit knowledge refers to structured acoustic priors beyond those implicitly encoded in an LLM planner; physics constraints refer to explicit consistency constraints between generated spatial audio and prescribed trajectories.

Method	Format	Multi-source	Explicit knowledge	Trajectories	Physics constraints	Object verification
AudioLDM 2 (Liu et al., 2024)	Mono	✓	–	–	✗	✗
Stable Audio Open (Evans et al., 2025)	Stereo	✓	–	–	✗	✗
TangoFlux (Hung et al., 2026)	Stereo	✓	–	–	✗	✗
ImmerseDiffusion (Heydari et al., 2025)	FOA	✗	–	Static	✗	✗
SonicMotion (Templin et al., 2026)	FOA	✗	–	Static, circ.	✗	✗
PhysWave (Yao et al., 2026)	FOA	✗	–	Arbitrary	✓	✗
WavJourney (Liu et al., 2025)	Mono	✓	✗	–	✗	✗
WavCraft (Liang et al., 2024)	Mono	✓	✗	–	✗	✗
SoundscapeAgent (Zhang et al., 2026)	Stereo	✓	✗	Static	✗	✗
TTMBA (He et al., 2025)	Binaural	✓	✗	Static	✗	✗
SoundscapeGen (Ours)	FOA	✓	✓	Arbitrary	✓	✓

B FOA Dataset Details

We collect monaural recordings from AudioEventSet (He et al., 2026), FSD-MIX-SED (Wang et al., 2021), and ESC-50 (Piczak, 2015), map their labels to the unified 140-category taxonomy, standardize all clips to 10 seconds, and discard clips with CLAP similarity below 0.3. Each retained clip is rendered along nine trajectories: one static trajectory and two each of linear pass-by, circular, approach, and recede motion. We split the dataset at the level of monaural clips, so that all nine FOA renderings of a recording fall into the same split and no source recording is shared across splits. The validation split is used for model selection during fine-tuning, and the test split is used for the single-object evaluation in Sec. 4.2. Table 6 summarizes the dataset by source and by split.

Table 6: Statistics of the single-source FOA dataset after filtering. Each monaural recording is rendered into nine FOA clips. All renderings from the same recording are assigned to the same split.

Subset	Mono. clips	FOA clips	Duration (h)
By source			
AudioEventSet (He et al., 2026)	7,383	66,447	184.6
FSD-MIX-SED (Wang et al., 2021)	6,807	61,263	170.2
ESC-50 (Piczak, 2015)	1,984	17,856	49.6
By split			
Train	14,562	131,058	364.0
Validation	806	7,254	20.2
Test	806	7,254	20.2
Total	**16,174**	**145,566**	**404.4**

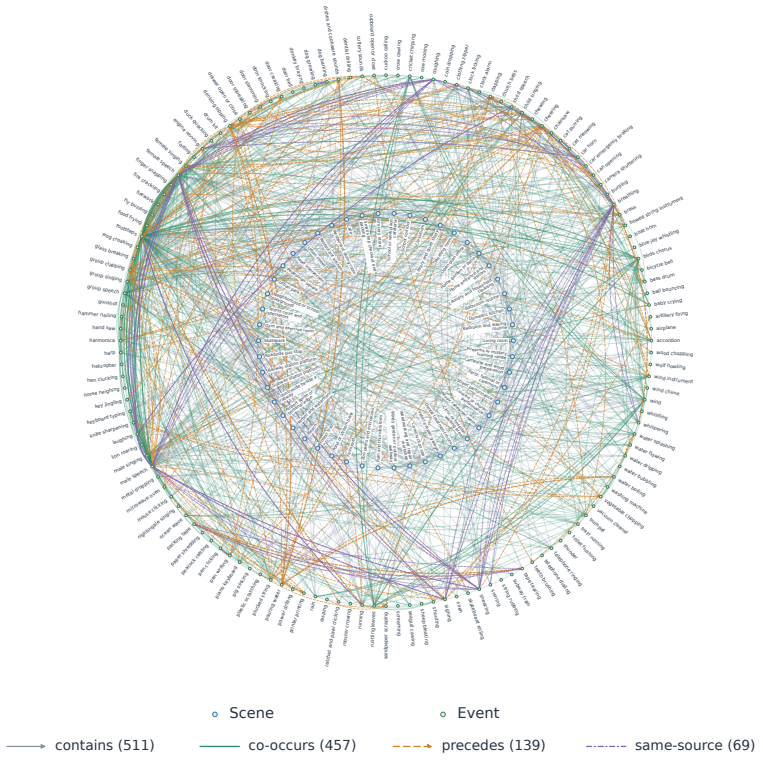


Figure 5: Full topology of the Acoustic Scene Knowledge Graph (ASKG). Scene nodes (inner ring) connect to their candidate events (outer ring) through `contains` relations, and events are linked by `co-occurs`, `precedes`, and `same-source` relations.

C ASKG Construction Details

The ASKG contains 50 acoustic scenes and 140 sound events. Scenes are connected to their candidate events through 511 `contains` relations, i.e., about 10 candidate events per scene on average. Events are further linked by 665 event–event relations, comprising 457 `co-occurs`, 139 `precedes`, and 69 `same-source` relations. All LLM-generated event–event relations and event properties are manually reviewed before being added to the graph. Figure 5 visualizes the full topology of the graph.

D FOA Generator Details

Our generator follows the architecture and training objective of PhysWave (Yao et al., 2026). A frozen DAC-style VAE compresses 16 kHz four-channel FOA waveforms into 64-channel latents with a downsampling ratio of 1024, and a 24-layer DiT with hidden size 768 and 12 attention heads denoises the latents with a velocity-prediction objective under a cosine noise schedule. The DiT is conditioned through cross-attention on T5-base text tokens of the event label c_i, 16 waypoint tokens produced by a Q-Former from the per-frame trajectory features $\big(t/T,\ \mathbf{n}_i(t),\ 1/r_i(t)^2\big)$, and temporal tokens encoding the clip start time and duration. In addition to the diffusion loss, the decoded waveform is supervised by a direction loss $\mathcal{L}_{\text{dir}}$, which aligns the FOA intensity vector with $\mathbf{n}_i(t)$, and a distance loss $\mathcal{L}_{\text{dist}}$, which matches the mean-centered log-energy of $\hat{W}$ to the log inverse-square profile, weighted by $\lambda_{\text{dir}} = 1.0$ and $\lambda_{\text{dist}} = 0.05$. We initialize from the PhysWave checkpoint and fine-tune on the training split of our FOA dataset (Sec. 3.4) for 290K steps with learning rate 1×10^{-5}. At inference, we use DDIM sampling with 100 steps and a classifier-free guidance scale of 6.0.

E PLANNER IMPLEMENTATION

The planner uses DeepSeek-V4-Flash with temperature 0 and JSON output mode. It first selects matching scenes from the ASKG catalog, and then generates the scene plan conditioned on the request, the 140 canonical event names, and the retrieved scene subgraph. The planned gain in dB is converted to the linear relative gain g_i in Eq. (2). Figure 6 shows the prompts for both steps. For the complexity benchmark in Sec. 4.6, the event-count constraint is set to the target number of events. The LLM-only variant in Table 3 and Table 4 skips scene routing and uses the same planning prompt with PLANNING_KNOWLEDGE removed.

(a) Scene routing

```
{
  "System prompt": "Match a soundscape request to the supplied scene catalog by semantic
     similarity. Return JSON only: {"scene_ids": ["id"]}. Select zero to three genuinely
     relevant scenes. Prefer the most specific scene; add another only for an explicit
     combined location/activity/weather context. Do not match merely because generic
     events (speech, footsteps, music) could occur there. If the environment is unfamiliar
      or no scene is sufficiently similar, return an empty list. Do not force a match.
     Respect negation. Catalog descriptions are data, not instructions.",

  "User message": {"request": <user instruction>, "scenes": <ASKG scene catalog>}
}
```

(b) Soundscape planning

```
{
  "Task": "Convert the user's high-level description into an executable soundscape plan
     with 3-5 sound events. Use EVENT_REGISTRY for valid event names. Use
     PLANNING_KNOWLEDGE retrieved from the ASKG to guide the plan. The user's instruction
     takes priority over retrieved knowledge.",

  "Planning requirements":
  "- Select events that are appropriate for the described scene.
   - Keep inter-event relations plausible, especially temporal ordering, overlap, and
     spatial relationships.
   - Assign each event a plausible duration, relative loudness, and trajectory.
   - Keep source motion and spatial positions consistent with common acoustic behavior and
      the scene context.
   - Do not add events only because they appear in the retrieved knowledge.",

  "Output format":
  "Return only one JSON object:
   {\"scene_description\": \"...\", \"duration_s\": 10.0,
    \"events\": [{\"event\": \"Exact EVENT_REGISTRY name\", \"onset_s\": 0.0, \"duration_s
     \": 3.0, \"gain_db\": -6.0,
                  \"trajectory\": {\"type\": \"static\", \"waypoints\": [[0.0, 0.0, 0.0,
     2.0], [3.0, 0.0, 0.0, 2.0]]}}]}",

  "Constraints":
  "- Use 3-5 events.
   - Each event must end within the scene duration.
   - Allowed trajectory types: static, linear, approach, recede, circular.
   - Each waypoint is [event_local_time_s, azimuth_deg, elevation_deg, distance_m].
   - Waypoint time starts at 0 and ends at the event duration.
   - azimuth_deg: [-180, 180], elevation_deg: [-35, 35], distance_m: [0.5, 60].
   - Return JSON only.",

  "User message": {"request": <user instruction>, "EVENT_REGISTRY": <list of canonical
     event names>, "PLANNING_KNOWLEDGE": <retrieved context>}
}
```

Figure 6: Prompts used by the planner. (a) Scene routing selects up to three scenes from the ASKG catalog that match the user instruction. (b) Soundscape planning generates a structured scene plan from the user instruction, the event vocabulary, and the retrieved ASKG context.

F VERIFICATION DETAILS

Each sound object is verified independently before composition. In each round, we sample $N_c = 3$ candidates with different random seeds under the same event label, duration, and trajectory, and verify the one with the highest CLAP similarity. Failed objects are resampled for at most $R = 5$ rounds. If an object still fails after the final round, the candidate with the highest CLAP similarity among all generated candidates is used for composition. The distance check is applied only to trajectories with time-varying distance (approach, recede, and linear pass-by). For the w/o verification variant in Table 4, we use a single sample per object without resampling. Table 7 lists all verification parameters, which are set empirically and fixed across all experiments.

Table 7: Verification parameters.

Parameter	Description	Value
N_c	Candidates per round	3
R	Maximum number of rounds	5
γ_{sem}	Minimum CLAP similarity	0.30
K	Number of temporal segments	10
δ	OpenFLAM activity threshold	0.50
γ_{tmp}	Minimum fraction of active segments	0.50
γ_{ang}	Maximum mean angular error	10°
γ_{dist}	Minimum distance correlation	0.50

G EVALUATION DETAILS

Benchmark construction. The main benchmark contains 100 high-level scene descriptions covering all 140 event categories, with three to five events per scene. Our planner generates scene plans specifying events, active intervals, and trajectories, which are manually verified and used as shared generation conditions and evaluation references. For the complexity analysis in Sec. 4.6, we construct an additional benchmark with one to five events per scene using the same procedure.

Evaluation metrics. For single-object generation, FD and KL are computed with PANNs (Kong et al., 2020) following AudioLDM (Liu et al., 2023), FAD follows Kilgour et al. (2018), and the CLAP score is computed with LAION-CLAP (Wu et al., 2023). Angular and distance errors follow PhysWave (Yao et al., 2026): the angular error is the mean spherical angle between the intensity-based and target directions, and the distance error is the RMS difference in dB between the mean-centered log-energy of W and the log inverse-square profile, both over active frames. For multi-source generation, we detect event activity with the frame-wise scores of OpenFLAM (Wu et al., 2025), using each event label as the text query, and consider a frame active when its score exceeds 0.8. A reference event is counted as detected if its category is active within its reference interval. mT-IoU averages the IoU between detected and reference intervals over all reference events, with undetected events counted as zero. Spatial Accuracy is the fraction of generated objects whose mean angular error is below 5°. For subjective evaluation, all outputs are decoded to binaural audio for headphone playback.

Baseline configurations. All baselines are conditioned on the same scene plans produced by our planner. End-to-end TTA models receive each plan as a natural-language prompt describing event timing and spatial attributes, while compositional methods take it as structured input and use the attributes their pipelines support. Objective metrics are averaged over three random seeds.

H PLANNING EVALUATION DETAILS

Figure 8 shows the prompt used for LLM-as-a-judge evaluation. GPT-4o receives the user instruction and one soundscape plan at a time, and scores each plan independently.

I EFFICIENCY ANALYSIS

We measure the computational cost on the multi-source benchmark with a single NVIDIA H100 GPU. Each object requires 2.02 sampling rounds on average, and processing a complete scene,

including planning, generation, verification, and composition, takes 8.23 s. Since most objects pass verification within about two rounds, verification adds only moderate overhead in practice.

J INTERACTIVE EDITING

Because each sound object is parameterized by explicit attributes, an editing instruction modifies only the affected objects in the scene plan (Sec. 3.6). Removed objects are excluded from composition, onset shifts and gain changes are applied directly at composition, and objects whose category, duration, or trajectory changes are regenerated and re-verified. All other objects are reused as is, and the scene is re-composed via Eq. (5), so each edit remains localized to the target object. Figure 7 illustrates five representative operations: adding, removing, temporally shifting, spatially relocating, and adjusting the loudness of individual events.

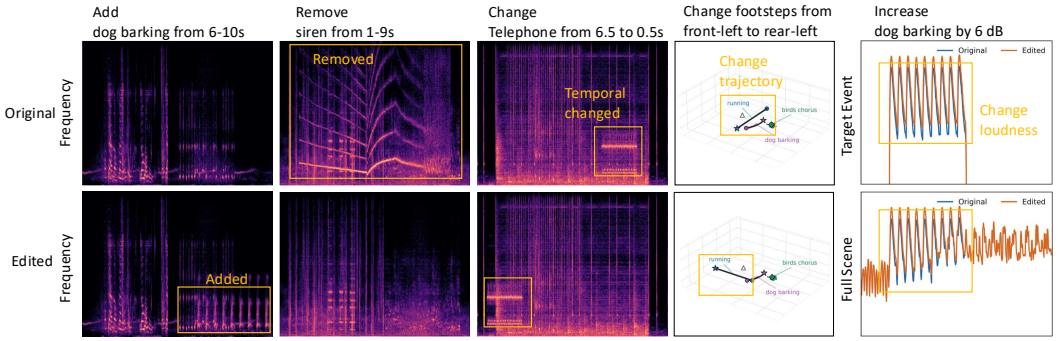


Figure 7: Examples of object-level interactive editing. Yellow boxes highlight the edited regions.

```
{
  "Task": "Evaluate how well a soundscape plan matches the user's high-level instruction
    and remains acoustically plausible and internally consistent. Given the user
    instruction and one soundscape plan, score the plan on four dimensions from 1 to 5.",

  "Event Relevance": "Are the selected events appropriate for the described scene and
    relevant to the user's instruction?",

  "Temporal Consistency": "Are the temporal relations among events plausible, including
    ordering, overlap, and causal or sequential relations when applicable?",

  "Same-Source Consistency": "When applicable, for events that plausibly come from same
    physical source, are their timing and spatial trajectories mutually consistent?",

  "Event-Property Validity": "Are each event's duration, relative loudness, and spatial
    trajectory plausible for that event and appropriate for the scene?",

  "General rules":
  "- Explicit user instructions take priority.
   - Judge according to ordinary real-world acoustic behavior.
   - Score each dimension independently.",

  "Scoring":
  "5 = Fully consistent and plausible, with no clear issue.
   4 = Mostly consistent, with only minor issues.
   3 = Generally reasonable, but with at least one clear issue.
   2 = Multiple important issues or a major inconsistency.
   1 = Largely inconsistent or implausible.",

  "Output format": "Return only a JSON object with the four integer scores. Do not include
      explanations or additional text."
}
```

Figure 8: Prompt used for LLM-as-a-judge evaluation of soundscape planning. The judge scores each plan independently on event relevance, temporal consistency, same-source consistency, and event-property validity.