Audible World Models: Spatially Aware Sound Generation for 3D Worlds

Duowen Chen
Georgia Institute of Technology
dchen322@gatech.edu

Jinjin He
Georgia Institute of Technology
jhe433@gatech.edu

Gouthaman KV
Dolby Laboratories
gkv@dolby.com

Sandeep Bangalore Venkatesh
Dolby Laboratories
sbang@dolby.com

Bo Zhu
Georgia Institute of Technology
bo.zhu@gatech.edu

Abstract

Text- and image-conditioned world generators can create visually rich 3D environments, yet these worlds often remain silent or rely on soundtracks synthesized solely from text or rendered video. Although such audio can convey what should be heard, it lacks an explicit representation of where sound sources are located and how their perceived sound should vary with listener movement. We introduce *Audible World Models*, a training-free framework that incorporates sound into the generated world state. Starting from a text prompt, our system constructs a panoramic 3D proxy, separates it into semantic layers, identifies sound-producing foreground objects and ambient background regions, and synthesizes dry audio for each sound label. It then anchors these sources to reconstructed geometry and renders listener-dependent spatial audio using geometric acoustic propagation. By explicitly linking semantics, geometry, and sound propagation, the framework maintains persistent source locations while adapting the rendered audio to changes in listener viewpoint and motion. Experiments across 80 generated scenes demonstrate substantial gains in spatial consistency over text-, video-, and panorama-conditioned baselines, while preserving competitive semantic alignment. VLM-based assessments and human evaluations further indicate that our soundtracks are preferred for their audio–visual consistency, spatial plausibility, and motion-dependent behavior.

1 Introduction

Generative 3D world models are rapidly turning lightweight inputs such as text prompts or single images into explorable environments. Recent systems [45, 23] connect panoramic generation [63, 62, 35], depth-based lifting [12], scene completion [64], and long-horizon navigation [34, 55, 54, 15] to produce visually compelling worlds with increasing geometric consistency and user control. Yet the generated experience is usually incomplete: the world can be seen and explored, but it is silent.

Adding sound to generated worlds requires more than plausible soundtrack synthesis. Modern text-to-audio [31, 38, 13, 56, 16] and video-to-audio models [42, 65, 11, 50] can synthesize high-quality environmental sounds, ambience, music, and Foley effects. Their conditioning is usually a prompt or short rendered video window, which gives only weak control over source identity, 3D placement,

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

occlusion, reflection, and left–right changes under motion. Thus, a generated street may contain plausible traffic and voices while the waveform lacks a road-side source, a stable crowd region, or a geometric reason for changing as the listener turns. This gap becomes especially visible on long trajectories, camera rotations, and revisited viewpoints.

We present a world-state representation for generated audio. An audible world stores semantic source labels, dry audio assets, 3D placements, source attributes, scene geometry, and acoustic parameters before rendering. This mirrors the structural ingredients that make visual worlds explorable: layers, geometry, and motion. Classical geometric sound simulation [47] can render distance attenuation, occlusion, diffraction, and reverberation [49], while assuming pre-authored meshes, sources, and acoustic parameters. Generated worlds already provide approximate geometry and visual semantics, and audio generators provide sound content. Our framework connects these pieces so the same world can be heard from multiple trajectories with persistent source behavior. This turns audio into a reusable property of the generated world state.

We implement this representation with an agentic, training-free pipeline. Starting from a generated panorama, VLM agents identify audible foreground objects and ambient background regions, turn them into label-conditioned sound descriptions, synthesize dry clips, anchor sources to reconstructed 3D geometry, and render binaural audio with a geometric acoustic simulator along the listener trajectory. The agents act as semantic coordinators between visual generation, audio generation, and acoustic rendering. The result is a modular system whose components can be replaced independently and whose source-level outputs can be tested against known listener–source geometry.

We summarize our contributions as:

- We formulate *audible world modeling* as spatial audio generation from an explicit world state containing geometry, semantic sound labels, dry audio assets, source placements, source attributes, and acoustic parameters.
- We present a training-free agentic pipeline that discovers audible scene elements, synthesizes label-conditioned audio, grounds sources on reconstructed geometry, and renders listener-dependent binaural sound through geometric propagation.
- We evaluate geometry-referenced direction recovery for isolated sources and final mixtures, alongside semantic alignment and perceptual preference. Our method achieves 6.91° isolated-source DoA MAE and a 67.0-percentage-point human Rank-1 preference gain over the strongest compared baseline.

2 Related Work

Text to Audio generation. Text-to-audio models based on autoregressive, diffusion, and flow-matching architectures [7, 10] have greatly improved the quality and controllability of generated sound. AudioGen [31], MusicGen [13], AudioLDM [38], TANGO/TANGO2 [19, 43], AudioBox [56], Stable Audio 1.0 [16], and TangoFlux [25] provide strong backbones for synthesizing sound effects, ambience, music, and general audio from natural-language descriptions. These models are useful for our setting because they can produce diverse dry audio assets; however, they do not by themselves determine persistent 3D source placement or listener-dependent acoustic effects.

Audio for visual worlds. A growing body of work conditions audio on images, videos, panoramic observations, or neural scene representations. Earlier methods infer binaural [17, 18] or ambisonic cues [44, 33] from visual observations, while novel-view acoustic synthesis [5] and neural scene methods [36, 2, 9, 8] render audio from new listener poses using learned 3D representations. Recent systems further target video-to-spatial-audio [30, 14, 39], panorama-conditioned audio [22], and 360-degree sound generation [53, 60]. Video-to-audio [42, 11] and neural Foley systems [65, 50] provide strong semantic and temporal synchronization, but most remain conditioned on rendered observations rather than a persistent world representation. Concurrently, SonoWorld also generates explorable 3D audio-visual scenes from a single image, grounding sound sources in 3D for listener-dependent ambisonic rendering [26].

Spatial acoustics simulation. Physically based sound propagation has long studied how geometry, materials, and listener motion shape the acoustic signal. Interactive geometric systems such as

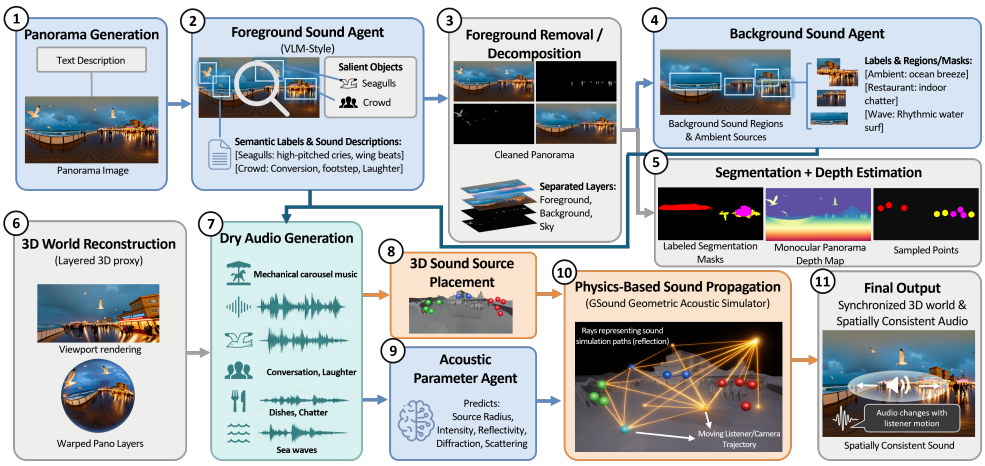

Figure 1: Overview of the proposed Audible World Models pipeline. The numbered blocks correspond to the stages described in Section 3. Stages 1–6 construct the sound-aware world proxy from a text prompt: a panorama, foreground audible objects, cleaned background regions, segmentation masks, depth, and layered 3D geometry. Stages 7–8 convert semantic sound labels into dry source assets and ground them as persistent 3D sources. Stages 9–11 predict acoustic parameters, propagate each source through the reconstructed geometry with GSound, and output synchronized spatial audio for the moving listener.

GSound [47] render specular reflection, diffuse reflection, diffraction [49, 3], and multi-source propagation in dynamic scenes [48]. Simulation frameworks [4, 6] and neural acoustic-field methods [52, 41, 37, 32] have also enabled embodied audio rendering, viewpoint interpolation [57, 40], and learned impulse-response prediction [27]. These methods provide the physical grounding needed for spatial consistency, but they typically require manually specified scenes and sources. Our contribution is to connect this physical rendering machinery to generative world and audio models.

3 From Visual Worlds to Audible Worlds

Given a text prompt c and a listener trajectory $\tau(t)$, our goal is to generate an audio signal $y(t)$ that is both semantically appropriate for the scene and spatially consistent with listener motion. The method is built around three design requirements. First, the sound representation should be *compositional*: scene-level ambience, a river, a crowd, and a vehicle are represented as distinct components. Second, it should be *persistent*: once a component is placed in the generated world, it keeps the same world-space identity as the listener moves. Third, it should be *renderable*: the final binaural signal is produced from source positions, listener motion, and scene geometry.

We represent an audible world as

$$\mathcal{W}_a = \left(\mathcal{G}, \{(\ell_i, d_i, a_i, \mathbf{x}_i, \rho_i)\}_{i=1}^{M}, \Theta\right). \tag{1}$$

The numbered pipeline in Figure 1 instantiates this representation. Stages 1–6 construct $\mathcal{G}$, the semantic labels ℓ_i, the descriptions d_i, and their visual support from the panorama. Stages 7–8 generate the dry clips a_i and convert masks, depth, and reconstructed surfaces into source placements $\mathbf{x}_i$ and source attributes ρ_i. Stages 9–11 estimate the acoustic parameters Θ and render the final listener-dependent waveform $y(t)$ along trajectory $\tau(t)$. This representation is the main distinction of our formulation: a generated world becomes a reusable sound field whose sources can be rendered from different trajectories and evaluated against known listener–source geometry.

3.1 Constructing a sound-aware world proxy

The first part of the pipeline builds the visual and semantic support for the audible world state, as shown in stages 1–6 of Figure 1. We first generate a panoramic world proxy from the input prompt using the panoramic synthesis module of HunyuanWorld [55]. The panorama gives a global

observation of the generated scene before a particular perspective trajectory is rendered. This is useful for sound because many audible objects may be outside the current camera view but still affect what the listener hears. A video-window method can miss such sources until they enter view, whereas a panorama-level parse can place them in the world from the beginning.

The proxy is decomposed into foreground, background, and sky layers. The foreground pass identifies salient localized objects that may emit sound, such as vehicles, animals, crowds, machinery, musical performers, or water features. The background pass operates on a foreground-removed panorama and identifies broader ambient regions, such as forests, streets, markets, rivers, coastlines, or indoor rooms. Separating these two cases is important because they should be spatialized differently. A barking dog or passing car should remain attached to a compact location, while wind through trees or a crowd murmur may be better modeled as a distributed field over a larger region.

For each audible label ℓ_i, the VLM produces a concise sound description d_i and a locality attribute indicating whether the source should be modeled as localized or diffuse. The same semantic labels are used to obtain segmentation masks [46], which are lifted into the layered 3D representation together with monocular depth [1] and world-sheet reconstruction [24, 55]. The output of this stage is not an audio clip, but a structured acoustic inventory: a set of named scene entities, each linked to a visual region and a proposed sound behavior. Low-level prompt templates (Appendix C.1), inpainting details and mask-processing rules (Appendix D) are left to the appendix.

3.2 Synthesizing and grounding sound sources

The next part of the pipeline turns the acoustic inventory into source assets and persistent source locations, as shown in stages 7–8 of Figure 1. The parsed labels give the system a structured inventory of what should be heard. We synthesize one dry audio clip for each foreground or background sound description, plus an optional global ambience track. In the current implementation we use Stable Audio 1.0 as the text-to-audio backend [16], but the framework only requires a model that maps textual descriptions to audio clips. Because the audio clips are generated independently before spatial rendering, they can be treated as source assets analogous to visual objects in a 3D scene.

The key step is to attach each generated sound to the world. Localized foreground sounds are sampled on their corresponding object meshes or world sheets. Background sounds are sampled on semantic submeshes associated with their masks, and diffuse ambience is instantiated as multiple distributed sources over the relevant background support. This converts label-conditioned sound generation into a persistent source field: the same dog, fountain, crowd, or river remains at the same position as the listener moves.

This grounding step also resolves a common ambiguity in text- or video-conditioned audio generation. A prompt such as "a lively street with traffic and people" may imply traffic, voices, footsteps, and distant city ambience, but it does not specify which side of the listener each sound should occupy. In our representation, this ambiguity is resolved by the visual world itself: traffic is placed on the reconstructed road region, voices are placed near crowd or pedestrian regions, and diffuse city ambience is distributed across background support. Details for surface sampling, normal offsets, jitter, and source-count normalization are left to the Appendix E.

3.3 Acoustic parameterization and rendering

The final part of the pipeline converts grounded source assets into a synchronized spatial soundtrack, as shown in stages 9–11 of Figure 1. A geometric acoustic renderer requires more than source positions. It also needs scene-level and source-level parameters controlling reflectivity, scattering, source radius, volume, and propagation behavior. We therefore use an acoustic parameter agent that reads the scene category, audible labels, approximate source layout, and source counts, then predicts conservative acoustic settings. These estimates are clamped to stable ranges before rendering (details are provided in Appendix C.3). This gives the system different behavior without training a new audio model.

The agentic parameterization is intentionally lightweight. It does not attempt to recover exact material properties from the generated image; instead, it chooses plausible propagation settings that preserve the qualitative acoustic character of the scene. For example, hard urban surfaces should generally permit stronger reflections than open vegetation. This level of acoustic control is sufficient for our

Table 1: Semantic evaluation on 80 generated scenes. Higher is better for all metrics.

Method	**CLAP ↑**	**IB-Text ↑**	**IB-Image ↑**	**Caption ↑**
See-2-Sound	0.1032	0.0233	0.0245	0.0989
OmniAudio	0.0706	0.1224	0.0875	0.1055
Stable Audio 1.0	**0.3366**	0.1813	0.2477	0.2552
MMAudio	0.3049	0.1953	**0.2632**	0.2519
Ours	0.3172	**0.2199**	0.2618	**0.2760**

Table 2: DoA on final mixtures. Matched MAE is in degrees; F_1 accounts for missed and extra directions.

Method	MAE ↓	F_1@15° ↑	F_1@45° ↑
See-2-Sound	45.6	0.202	**0.526**
OmniAudio	54.35	0.089	0.265
Ours	**17.44**	**0.297**	0.424

Table 3: Automated VLM-based evaluation using uniformly truncated 5 s clips, including all five methods. MOS values are model-assigned ratings, not human ratings. Higher is better except for mean rank.

Method	Mean MOS ↑	Mean rank ↓	Rank-1 ↑	Top-2 ↑	Borda ↑
Stable Audio 1.0	2.98	2.59	21.2%	48.5%	3.41
MMAudio	2.95	2.68	24.2%	47.0%	3.32
See-2-Sound	2.93	3.33	7.6%	24.2%	2.67
OmniAudio	2.52	4.38	1.5%	9.1%	1.62
Ours	**3.01**	**2.02**	**45.5%**	**71.2%**	**3.98**

goal: producing spatial audio whose variation with listener motion is consistent and perceptually plausible.

Finally, we render the soundtrack using the GSound C++ API [47, 49]. The reconstructed geometry is used as the propagation scene, the sampled sources provide dry audio and source metadata, and the camera path supplies the moving listener trajectory. The output is a binaural, listener-dependent audio signal $y(t)$ synchronized with the generated world. Because the sound field is tied to geometry rather than only to a rendered video window, spatial effects can remain consistent over longer trajectories and revisited viewpoints. The same representation also supports diagnostic evaluation: each source can be rendered in isolation, letting us measure whether its binaural cues follow its world-space position.

4 Evaluation and Results

We evaluate whether the generated soundtrack satisfies two requirements: it should contain sounds that match the scene semantics, and those sounds should change consistently as the listener moves. Our primary semantic benchmark contains 80 generated scenes, with additional controlled evaluations reported separately. Our method is training-free: no model weights are updated. The full offline pipeline is run on a single NVIDIA H200 GPU and requires up to 48 GB of GPU memory; GSound rendering is CPU-based. End-to-end timing is reported in Appendix H.

We compare against a strong text-to-audio baseline, Stable Audio 1.0 [16]; a representative video-to-audio baseline, MMAudio [11]; and panorama- or scene-conditioned spatial audio baselines, SEE-2-SOUND [14] and OmniAudio [39]. The human study compares Stable Audio 1.0, MMAudio, and our method over 20–40 s trajectories. The released spatial-baseline configurations used here produce shorter 5–10 s outputs, so we evaluate all five methods separately using duration-matched 5 s clips. Mixing output durations would confound the blinded long-horizon comparison.

4.1 Semantic alignment

Semantic evaluation asks whether the generated audio is appropriate for the target world. Stable Audio 1.0 receives the refined scene prompt, video-to-audio baselines receive a rendered trajectory video, and panorama- or scene-conditioned baselines receive the generated panorama. To isolate semantic content from spatialization, we compute semantic metrics on mono mixdowns of all outputs.

We report three complementary scores. CLAP [59] measures audio–text alignment between the scene prompt and generated audio using the LAION-CLAP checkpoint adopted in prior text-to-audio evaluation [38, 16, 11]. ImageBind measures cross-modal consistency with both the prompt and four

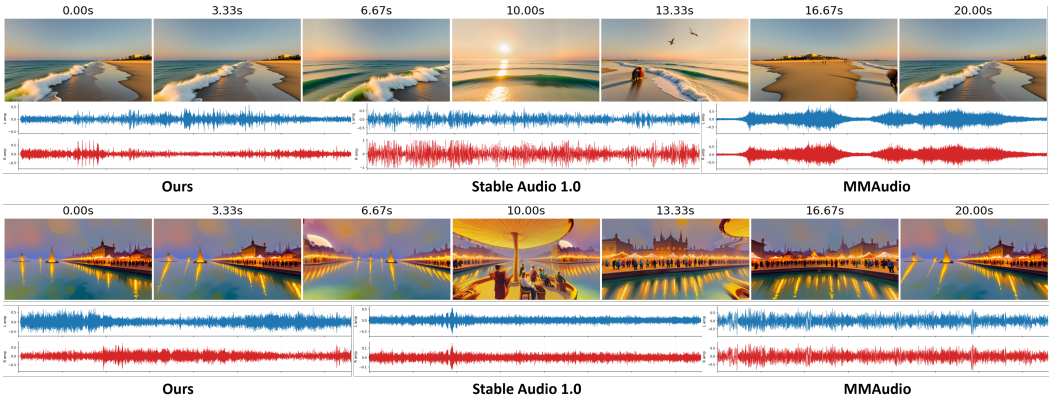

Figure 2: Synchronized 20-second audiovisual examples. Compared with scene-level text-to-audio and video-to-audio baselines, our method produces clearer motion-dependent changes because each sound remains tied to a persistent 3D source location.

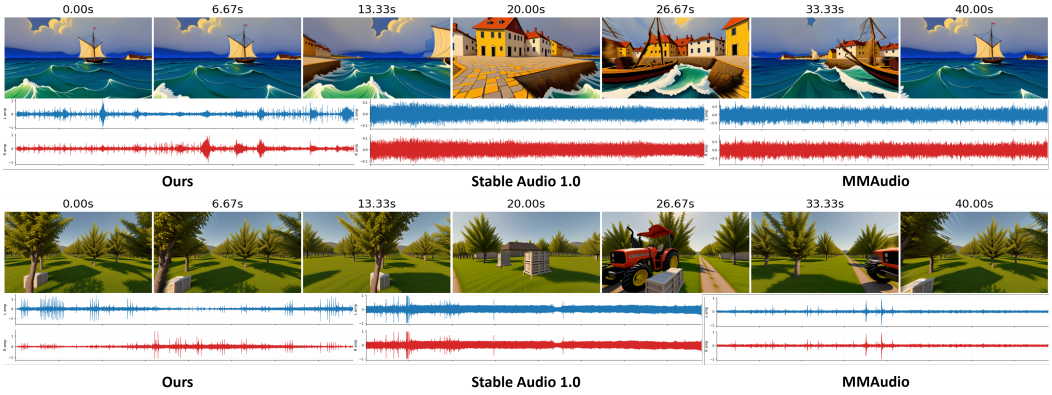

Figure 3: Synchronized 40-second audiovisual examples. The spatial behavior remains stable over longer trajectories, supporting the claim that explicit source placement and acoustic propagation help maintain long-horizon consistency.

rendered perspective views covering the scene [21]. The caption score first captions the generated audio using an audio captioning model [20] and then compares the caption with the scene prompt in a shared text-embedding space. We provide detailed metric definitions in Appendix G.1.

Table 1 shows that explicit source decomposition preserves strong semantic alignment. Stable Audio 1.0 obtains the highest CLAP score at 0.3366, while our method reaches 0.3172, or 94.2% of the best score. Our method achieves the best ImageBind text score, improving from 0.1953 to 0.2199 over the strongest baseline, and the best caption score, improving from 0.2552 to 0.2760. It also reaches 0.2618 on ImageBind image alignment, within 0.5% of the best score. These results indicate that source-level structure adds spatial grounding while keeping the audio semantically aligned with the generated world.

We additionally report scene-bootstrap 95% confidence intervals for the semantic metrics over the original 80 scenes; for example, the CLAP score is 0.317 with interval $[0.286, 0.348]$. Full intervals are provided in Appendix G.1.5.

4.2 Spatial consistency

We evaluate directional consistency against source azimuths computed from the constructed world and the known listener trajectory. This tests whether directions are recoverable from the rendered sound, rather than whether an ILD curve can be fitted to listener–source geometry.

Isolated-source direction recovery. We estimate full-circle horizontal direction of arrival (DoA) from isolated binaural tracks using a calibrated combination of multi-band ILD, interaural time difference, and a weak pinna-spectral cue for front–back disambiguation. Our method achieves 6.91° mean absolute error (95% CI: 3.51–12.04°), 2.96° median error, and 96.1% accuracy within 15°. Calibration and full intervals are reported in Appendix G.3.

Direction recovery from mixtures. We additionally estimate directions directly from the final mixed soundtracks and compare them with active reference sources using one-to-one Hungarian matching with circular angular distance. Table 2 reports matched angular error and F_1 at 15° and 45° tolerances. Our method obtains the lowest matched MAE (17.44°) and the highest $F_1@15^\circ$ (0.297), whereas See-2-Sound obtains the highest $F_1@45^\circ$ (0.526). Thus, our advantage is most evident at the stricter angular tolerance; recovering all simultaneously active sources remains challenging. We compare methods with defined spatial channel geometries. MMAudio's mono output and Stable Audio 1.0's generic stereo do not provide a defined absolute-azimuth reference and are not scored in this DoA comparison. The reference inventory, activity mask, and channel-specific estimators are specified in Appendix G.3.

Complementary spatial diagnostics. The original ILD and energy-ranking analyses remain complementary source-level diagnostics (Appendix G.2); our ILD correlation is 0.8958, compared with 0.3050 for the strongest baseline. Neither these diagnostics nor mixture DoA establishes a proportional relationship with perceptual source separation. The human study below provides complementary evidence on complete mixtures.

4.3 VLM-based and human preference studies

The numerical metrics above test targeted aspects of the output, but spatial audio is ultimately perceptual. We therefore evaluate synchronized audio–video outputs using five criteria: overall audio–visual consistency, clarity of spatial changes, consistency with listener viewing direction and motion, spatial plausibility, and audio naturalness. Each criterion is rated on a 1–5 MOS scale, and evaluators also rank the candidate methods.

VLM-based evaluation. We use a two-stage blind protocol. First, rendered frames are paired with trajectory metadata, including camera positions, viewing directions, clip duration, and path statistics. GPT-5-mini uses these inputs to produce a structured description of the scene layout, listener motion, and spatial cues. Second, GPT-audio-mini evaluates each anonymized candidate soundtrack conditioned on this shared context and metadata. Candidate names are hidden behind labels such as `candidate_a` to avoid method leakage; the full prompts are provided in the Appendix F. The original long-horizon VLM results are retained in Appendix F.1.

Duration-matched comparison. We additionally compare all five methods using uniformly truncated 5 s clips, allowing See-2-Sound and OmniAudio to be evaluated under the same duration constraint. In this automated evaluation, Table 3 shows that our method obtains the highest mean MOS (3.01), the lowest mean rank (2.02), and the highest Rank-1 rate (45.5%). The MOS differences are small, while the ranking measures more clearly favor our method. This is a separate short-horizon VLM-based comparison, not an additional human study; its protocol is discussed in Appendix F.2.

Human evaluation. We also conduct a blind human study on 20 generated scenes with 20 participants. For each scene, participants view the same rendered 3D trajectory video and evaluate three anonymized audio tracks: our method, Stable Audio 1.0, and MMAudio. They provide MOS ratings under the same five criteria and rank the three candidates from best to worst. Table 4, Figure 5, and Figure 4 show that our method is strongly preferred, with mean MOS 4.145, Rank-1 preference 79.5%, and Top-2 preference 92.0%. The strongest baseline reaches mean MOS 2.498, Rank-1 preference 12.5%, and Top-2 preference 58.9%, giving gains of 1.647 MOS points, 67.0 percentage points in Rank-1, and 33.1 percentage points in Top-2. This gap suggests that listeners value source grounding and motion-dependent acoustic rendering even when baselines generate plausible sound content.

Table 4: Human evaluation of methods using MOS and ranking-based evaluation metrics.

Method	Mean MOS ↑	Mean Rank ↓	Rank-1 ↑	Top-2 ↑	Borda ↑
Stable Audio 1.0	2.268	2.429	8.0%	49.1%	1.571
MMAudio	<u>2.498</u>	<u>2.286</u>	<u>12.5%</u>	<u>58.9%</u>	<u>1.714</u>
Ours	**4.145**	**1.286**	**79.5%**	**92.0%**	**2.714**

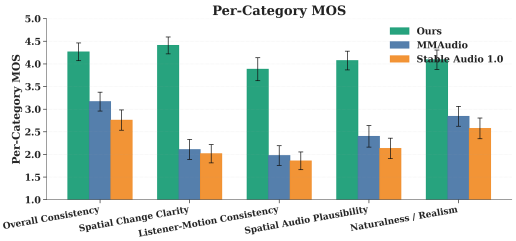

Figure 4: Per-category MOS comparison across methods.

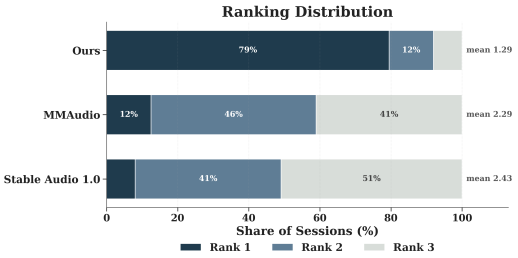

Figure 5: Rank distribution comparison across methods.

4.4 Qualitative spatial analysis

Figure 8 illustrates why the quantitative spatial scores improve. The rendered ILD follows the trend predicted by the listener–source geometry, and the channel-wise spectrograms show clear left–right changes as the listener moves. Figures 2 and 3 provide synchronized examples over short and long trajectories. Baseline methods often produce plausible audio events, but their waveforms exhibit weaker motion-dependent variation because the sound is not anchored to persistent 3D sources. In contrast, our output changes as the listener approaches, passes, or turns relative to the sources.

4.5 Runtime and scalability

The system is an offline pipeline, and its constructed audible world can be cached for subsequent listener trajectories. Holding the reconstructed world fixed, source setup and acoustic propagation with 32 sources take 21.36 s for a 5 s trajectory and 184.14 s for a 160 s trajectory. At a fixed 20 s duration, increasing the source cap from 1 to 128 changes runtime from 22.32 s to 92.10 s. Appendix H.1 reports the complete sweeps and reported timing variability, separately from end-to-end construction costs. A controlled moving-source test with supplied source and listener trajectories also renders 32 moving sources over 20 s in 106.09 s (Appendix H.2); this tests the rendering stage, not automatic dynamic-world construction.

5 Ablation Studies

We ablate the main components of the pipeline to test which parts are responsible for semantic quality and spatial consistency. The original component-removal ablations use localized per-source renders for spatial diagnostics and final full-scene soundtracks for semantic metrics. The additional heuristic and module-replacement experiments are reported separately with their own full-model references.

Trajectory length. Table 7 compares 20-second and 40-second trajectories. The spatial scores remain stable across both settings, indicating that the method preserves source geometry over extended motion rather than only matching local, short-window cues.

Component ablations. Table 5 shows that spatial consistency is most sensitive to acoustic propagation and geometric grounding. Removing the acoustic simulation stage nearly eliminates ILD correlation and produces unstable proxy fits, confirming that text-conditioned sound alone does not create reliable listener-dependent directional cues. Removing geometry-aware placement or layered 3D reconstruction similarly degrades both ILD and energy-ranking metrics, showing that sources must be tied to the generated scene rather than placed with a listener-centric heuristic.

Variant	Spatial metrics				Semantic metrics			
	ILD ↑	ILD R^2 ↑	Pair Acc. ↑	Spearman ↑	CLAP ↑	IB-Text ↑	IB-Image ↑	Caption ↑
Full model	**0.888**	**0.807**	**0.823**	**0.758**	0.298	**0.191**	0.179	**0.301**
No acoustic simulation	0.007	$\ll 0$	0.480	0.035	**0.328**	0.176	0.177	0.275
No agentic acoustic parameters	0.881	0.753	0.690	0.581	0.287	0.172	0.176	0.204
No semantic decomposition	–	–	–	–	0.324	0.177	0.126	0.246
No geometry-aware placement	0.033	$\ll 0$	0.565	0.154	0.290	0.183	0.177	0.267
No foreground/background separation	0.872	0.342	0.729	0.706	0.287	0.164	**0.185**	0.221
No layered 3D reconstruction	-0.043	$\ll 0$	0.561	0.160	0.187	0.119	0.114	0.187
Scene-level TangoFlux [25]	–	–	–	–	0.225	0.158	0.172	0.190

Table 5: World-sound ablation study. Spatial metrics are computed from localized per-source renders using the original scene geometry and a fixed ILD calibration. Semantic metrics are computed from the final full-scene soundtrack. "–" denotes cases where localized spatial metrics are not applicable.

Table 6: Semantic comparison with heuristic world-sound construction. Higher is better. This controlled comparison is reported separately from the original 80-scene benchmark and component-removal ablations.

Method	CLAP ↑	IB-Text ↑	IB-Image ↑	Caption ↑
Weak heuristic	0.121	0.129	0.087	0.129
Strong heuristic	0.225	0.154	0.169	0.270
Full method	**0.303**	**0.196**	**0.188**	**0.301**

The ablation of agentic acoustic parameters retains strong ILD correlation but reduces energy-ranking performance, suggesting that scene-adaptive parameters mainly improve relative source strength and attenuation. Removing foreground/background separation weakens both semantic and spatial behavior, likely because a single-pass parse produces less reliable source structure. Finally, the semantic metrics show that global audio–text similarity and spatial consistency are not equivalent: several variants remain competitive under CLAP while failing on spatial metrics. This supports our core claim that audible world generation requires explicit scene structure, not only a strong audio generator. We refer readers to Appendix A for details of ablation study settings.

Comparison with heuristic construction. We compare against two heuristic baselines that retain the reconstructed scene mesh and the GSound renderer. The weak baseline assigns generic sounds from four broad scene categories; the strong baseline matches prompt keywords to a fixed sound-prompt bank using sentence-embedding similarity and synthesizes the retrieved prompts with Stable Audio. Both place sources randomly on the scene mesh and randomly sample acoustic parameters within the same permitted ranges. Our full method scores higher on all four semantic metrics (Table 6). These comparisons test the complete construction strategy rather than isolating any single agent; implementation details are provided in Appendix A.2.

Single- and multi-module replacement. We test the modular interface through controlled substitutions of the audio, segmentation, VLM, and depth backends (Appendix A.3). Replacing the source-audio backend with TangoFlux changes CLAP from 0.303 to 0.222 and ILD correlation from 0.889 to 0.796. Replacing MoGe-2 with Depth Anything V2 yields CLAP 0.316 and ILD correlation 0.854. Jointly replacing the VLM, segmentation, and depth modules while retaining Stable Audio yields CLAP 0.323 and ILD correlation 0.893. These results support interchangeability of the tested components, with metric-dependent tradeoffs, rather than invariance to backend choice. The depth substitution measures downstream behavior, not depth accuracy, because ground-truth depth is unavailable.

Inventory and agent repeatability. Scenes contain, on average, 1.17 foreground labels, 4.20 background labels, and one global ambience track. These semantic components are distinct from the sampled emitters used by the renderer. Across five repeated parses of each evaluated scene, foreground/background semantic-label agreement is 74.2%/73.3%, and localized-versus-diffuse decision agreement is 82.6%/88.1%. Only 0.869% of predicted sources require parameter clamping. Appendix C.4 reports count distributions, uncertainty, and two-run parameter variation. These are repeatability and range-compliance diagnostics, not detection-accuracy measurements.

6 Conclusion and Limitations

We presented Audible World Models, a training-free framework for adding spatially aware audio to generated 3D worlds. The central idea is to treat sound as part of the world representation: audible objects and ambient regions are parsed from the generated scene, grounded on reconstructed geometry, assigned dry audio, and rendered through physically motivated acoustic propagation. This design yields a listener-dependent audiovisual experience with persistent source identities. Our evaluation combines semantic alignment over 80 generated scenes with geometry-referenced spatial tests: isolated-source DoA MAE is 6.91°, and mixture DoA reaches 17.44° matched MAE and $F_1@15^\circ = 0.297$. Human evaluation further increases mean MOS from 2.498 to 4.145 and Rank-1 preference from 12.5% to 79.5%. These results identify explicit world-space source grounding as the paper's central contribution.

The automatic construction pipeline currently produces static geometry and fixed source placements. The supplied-trajectory experiment in Appendix H.2 demonstrates moving-source rendering, but does not infer object motion, reconstruct dynamic geometry, or establish action–sound synchronization. The system is offline rather than real-time; training-free operation does not remove the cost of world construction. Audio fidelity depends on the audio backend, and acoustic realism depends on reconstructed geometry and estimated parameters. Our geometric DoA references test consistency with the constructed world, not recovery of real-world source positions or exact material properties. Dynamic world construction and more faithful geometry and material estimation remain future work, including integration with long-horizon world models such as Lyra 2.0 [51].

Acknowledgments and Disclosure of Funding

We sincerely thank the reviewers for their valuable feedback. We gratefully acknowledge support from the Dolby ATG Research Award. We also thank NVIDIA for providing computational resources through the NVIDIA Academic Grant. The authors declare no competing interests.

References

[1] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023.

[2] Swapnil Bhosale, Haosen Yang, Diptesh Kanojia, Jiankang Deng, and Xiatian Zhu. Av-gs: Learning material and geometry aware priors for novel view acoustic synthesis. *Advances in Neural Information Processing Systems*, 37:28920–28937, 2024.

[3] Chunxiao Cao, Zhong Ren, Carl Schissler, Dinesh Manocha, and Kun Zhou. Interactive sound propagation with bidirectional path tracing. *ACM Transactions on Graphics (TOG)*, 35(6):1–11, 2016.

[4] Changan Chen, Unnat Jain, Carl Schissler, Sebastia Vicenc Amengual Gari, Ziad Al-Halah, Vamsi Krishna Ithapu, Philip Robinson, and Kristen Grauman. Soundspaces: Audio-visual navigation in 3d environments. In *European conference on computer vision*, pages 17–36. Springer, 2020.

[5] Changan Chen, Alexander Richard, Roman Shapovalov, Vamsi Krishna Ithapu, Natalia Neverova, Kristen Grauman, and Andrea Vedaldi. Novel-view acoustic synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6409–6419, 2023.

[6] Changan Chen, Carl Schissler, Sanchit Garg, Philip Kobernik, Alexander Clegg, Paul Calamia, Dhruv Batra, Philip Robinson, and Kristen Grauman. Soundspaces 2.0: A simulation platform for visual-acoustic learning. *Advances in Neural Information Processing Systems*, 35:8896–8911, 2022.

[7] Lele Chen, Sudhanshu Srivastava, Zhiyao Duan, and Chenliang Xu. Deep cross-modal audio-visual generation. In *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, pages 349–357, 2017.

[8] Mingfei Chen, Israel D Gebru, Ishwarya Ananthabhotla, Christian Richardt, Dejan Markovic, Jake Sandakly, Steven Krenn, Todd Keebler, Eli Shlizerman, and Alexander Richard. Soundvista: Novel-view ambient sound synthesis via visual-acoustic binding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8331–8341, 2025.

[9] Mingfei Chen and Eli Shlizerman. Av-cloud: Spatial audio rendering through audio-visual cloud splatting. *Advances in Neural Information Processing Systems*, 37:141021–141044, 2024.

[10] Ziyang Chen, Prem Seetharaman, Bryan Russell, Oriol Nieto, David Bourgin, Andrew Owens, and Justin Salamon. Video-guided foley sound generation with multimodal controls. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18770–18781, 2025.

[11] Ho Kei Cheng, Masato Ishii, Akio Hayakawa, Takashi Shibuya, Alexander Schwing, and Yuki Mitsufuji. Mmaudio: Taming multimodal joint training for high-quality video-to-audio synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28901–28911, 2025.

[12] Jaeyoung Chung, Suyoung Lee, Hyeongjin Nam, Jaerin Lee, and Kyoung Mu Lee. Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. *arXiv preprint arXiv:2311.13384*, 2023.

[13] Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. *Advances in neural information processing systems*, 36:47704–47720, 2023.

[14] Rishit Dagli, Shivesh Prakash, Robert Wu, and Houman Khosravani. See-2-sound: Zero-shot spatial environment-to-spatial sound. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Posters*, pages 1–2, 2025.

[15] Edoardo A Dominici, Jozef Hladkỳ, Floor Verhoeven, Lukas Radl, Thomas Deixelberger, Stefan Ainetter, Philipp Drescher, Stefan Hauswiesner, Arno Coomans, Giacomo Nazzaro, et al. Dreamanywhere: Object-centric panoramic 3d scene generation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1–11, 2026.

[16] 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)*, pages 1–5. IEEE, 2025.

[17] Ruohan Gao and Kristen Grauman. 2.5 d visual sound. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 324–333, 2019.

[18] Rishabh Garg, Ruohan Gao, and Kristen Grauman. Visually-guided audio spatialization in video with geometry-aware multi-task learning. *International Journal of Computer Vision*, 131(10):2723–2737, 2023.

[19] 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*, pages 3590–3598, 2023.

[20] Sreyan Ghosh, Zhifeng Kong, Sonal Kumar, S Sakshi, Jaehyeon Kim, Wei Ping, Rafael Valle, Dinesh Manocha, and Bryan Catanzaro. Audio flamingo 2: An audio-language model with long-audio understanding and expert reasoning abilities. *arXiv preprint arXiv:2503.03983*, 2025.

[21] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15180–15190, 2023.

[22] 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)*, pages 1–5. IEEE, 2025.

[23] Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2room: Extracting textured 3d meshes from 2d text-to-image models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7909–7920, 2023.

[24] Ronghang Hu, Nikhila Ravi, Alexander C Berg, and Deepak Pathak. Worldsheet: Wrapping the world in a 3d sheet for view synthesis from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12528–12537, 2021.

[25] Chia-Yu Hung, Navonil Majumder, Zhifeng Kong, Ambuj Mehrish, Amir Ali Bagherzadeh, 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. *arXiv preprint arXiv:2412.21037*, 2024.

[26] Derong Jin, Xiyi Chen, Ming C Lin, and Ruohan Gao. Sonoworld: From one image to a 3d audio-visual scene. *arXiv preprint arXiv:2603.28757*, 2026.

[27] Derong Jin and Ruohan Gao. Differentiable room acoustic rendering with multi-view vision priors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 37–47, 2025.

[28] Beomyoung Kim, Chanyong Shin, Joonhyun Jeong, Hyungsik Jung, Se-Yun Lee, Sewhan Chun, Dong-Hyun Hwang, and Joonsang Yu. Zim: Zero-shot image matting for anything. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 23828–23838. IEEE, 2025.

[29] Duck O Kim, Brian Bishop, and Shigeyuki Kuwada. Acoustic cues for sound source distance and azimuth in rabbits, a racquetball and a rigid spherical model. *Journal of the Association for Research in Otolaryngology*, 11(4):541–557, 2010.

[30] Jaeyeon Kim, Heeseung Yun, and Gunhee Kim. Visage: Video-to-spatial audio generation. *arXiv preprint arXiv:2506.12199*, 2025.

[31] Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi. Audiogen: Textually guided audio generation. *arXiv preprint arXiv:2209.15352*, 2022.

[32] Zitong Lan, Chenhao Zheng, Zhiwei Zheng, and Mingmin Zhao. Acoustic volume rendering for neural impulse response fields. *Advances in Neural Information Processing Systems*, 37:44600–44623, 2024.

[33] Dingzeyu Li, Timothy R Langlois, and Changxi Zheng. Scene-aware audio for 360 videos. *ACM Transactions on Graphics (TOG)*, 37(4):1–12, 2018.

[34] Zizhang Li, Hong-Xing Yu, Wei Liu, Yin Yang, Charles Herrmann, Gordon Wetzstein, and Jiajun Wu. Wonderplay: Dynamic 3d scene generation from a single image and actions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9080–9090, 2025.

[35] Hanwen Liang, Junli Cao, Vidit Goel, Guocheng Qian, Sergei Korolev, Demetri Terzopoulos, Konstantinos N Plataniotis, Sergey Tulyakov, and Jian Ren. Wonderland: Navigating 3d scenes from a single image. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 798–810, 2025.

[36] Susan Liang, Chao Huang, Yapeng Tian, Anurag Kumar, and Chenliang Xu. Av-nerf: Learning neural fields for real-world audio-visual scene synthesis. *Advances in Neural Information Processing Systems*, 36:37472–37490, 2023.

[37] Susan Liang, Chao Huang, Yapeng Tian, Anurag Kumar, and Chenliang Xu. Neural acoustic context field: Rendering realistic room impulse response with neural fields. *arXiv preprint arXiv:2309.15977*, 2023.

[38] 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.

[39] Huadai Liu, Tianyi Luo, Kaicheng Luo, Qikai Jiang, Peiwen Sun, Jialei Wang, Rongjie Huang, Qian Chen, Wen Wang, Xiangtai Li, et al. Omniaudio: Generating spatial audio from 360-degree video. *arXiv preprint arXiv:2504.14906*, 2025.

[40] Xiulong Liu, Anurag Kumar, Paul Calamia, Sebastia V Amengual, Calvin Murdock, Ishwarya Ananthabhotla, Philip Robinson, Eli Shlizerman, Vamsi Krishna Ithapu, and Ruohan Gao. Hearing anywhere in any environment. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5732–5741, 2025.

[41] Andrew Luo, Yilun Du, Michael Tarr, Josh Tenenbaum, Antonio Torralba, and Chuang Gan. Learning neural acoustic fields. *Advances in Neural Information Processing Systems*, 35:3165–3177, 2022.

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

[43] 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*, pages 564–572, 2024.

[44] Pedro Morgado, Nuno Nvasconcelos, Timothy Langlois, and Oliver Wang. Self-supervised generation of spatial audio for 360 video. *Advances in neural information processing systems*, 31, 2018.

[45] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[46] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[47] Carl Schissler and Dinesh Manocha. Gsound: Interactive sound propagation for games. In *Audio Engineering Society Conference: 41st International Conference: Audio for Games. Audio Engineering Society*, volume 6, 2011.

[48] Carl Schissler and Dinesh Manocha. Interactive sound propagation and rendering for large multi-source scenes. *ACM Transactions on Graphics (TOG)*, 36(4):1, 2016.

[49] Carl Schissler, Ravish Mehra, and Dinesh Manocha. High-order diffraction and diffuse reflections for interactive sound propagation in large environments. *ACM Transactions on Graphics (TOG)*, 33(4):1–12, 2014.

[50] Sizhe Shan, Qiulin Li, Yutao Cui, Miles Yang, Yuehai Wang, Qun Yang, Jin Zhou, and Zhao Zhong. Hunyuanvideo-foley: Multimodal diffusion with representation alignment for high-fidelity foley audio generation. *arXiv preprint arXiv:2508.16930*, 2025.

[51] Tianchang Shen, Sherwin Bahmani, Kai He, Sangeetha Grama Srinivasan, Tianshi Cao, Jiawei Ren, Ruilong Li, Zian Wang, Nicholas Sharp, Zan Gojcic, Sanja Fidler, Jiahui Huang, Huan Ling, Jun Gao, and Xuanchi Ren. Lyra 2.0: Explorable generative 3d worlds. *arXiv preprint arXiv:2604.13036*, 2026.

[52] Kun Su, Mingfei Chen, and Eli Shlizerman. Inras: Implicit neural representation for audio scenes. *Advances in Neural Information Processing Systems*, 35:8144–8158, 2022.

[53] Peiwen Sun, Sitong Cheng, Xiangtai Li, Zhen Ye, Huadai Liu, Honggang Zhang, Wei Xue, and Yike Guo. Both ears wide open: Towards language-driven spatial audio generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[54] Wenqiang Sun, Haiyu Zhang, Haoyuan Wang, Junta Wu, Zehan Wang, Zhenwei Wang, Yunhong Wang, Jun Zhang, Tengfei Wang, and Chunchao Guo. Worldplay: Towards long-term geometric consistency for real-time interactive world modeling. *arXiv preprint arXiv:2512.14614*, 2025.

[55] HunyuanWorld Team, Zhenwei Wang, Yuhao Liu, Junta Wu, Zixiao Gu, Haoyuan Wang, Xuhui Zuo, Tianyu Huang, Wenhuan Li, Sheng Zhang, et al. Hunyuanworld 1.0: Generating immersive, explorable, and interactive 3d worlds from words or pixels. *arXiv preprint arXiv:2507.21809*, 2025.

[56] Apoorv Vyas, Bowen Shi, Matthew Le, Andros Tjandra, Yi-Chiao Wu, Baishan Guo, Jiemin Zhang, Xinyue Zhang, Robert Adkins, William Ngan, et al. Audiobox: Unified audio generation with natural language prompts. *arXiv preprint arXiv:2312.15821*, 2023.

[57] Mason Long Wang, Ryosuke Sawata, Samuel Clarke, Ruohan Gao, Shangzhe Wu, and Jiajun Wu. Hearing anything anywhere. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11790–11799, 2024.

[58] Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. *Advances in Neural Information Processing Systems*, 38:35928–35959, 2026.

[59] 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)*, pages 1–5. IEEE, 2023.

[60] Siyi Xie, Hanxin Zhu, Xinyi Chen, Tianyu He, Xin Li, and Zhibo Chen. Sonic4d: Spatial audio generation for immersive 4d scene exploration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 11087–11095, 2026.

[61] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in neural information processing systems*, 37:21875–21911, 2024.

[62] Hong-Xing Yu, Haoyi Duan, Charles Herrmann, William T Freeman, and Jiajun Wu. Wonderworld: Interactive 3d scene generation from a single image. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5916–5926, 2025.

[63] Hong-Xing Yu, Haoyi Duan, Junhwa Hur, Kyle Sargent, Michael Rubinstein, William T Freeman, Forrester Cole, Deqing Sun, Noah Snavely, Jiajun Wu, et al. Wonderjourney: Going from anywhere to everywhere. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6658–6667, 2024.

[64] Songchun Zhang, Yibo Zhang, Quan Zheng, Rui Ma, Wei Hua, Hujun Bao, Weiwei Xu, and Changqing Zou. 3d-scenedreamer: Text-driven 3d-consistent scene generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10170–10180, 2024.

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

Appendix Contents

A Implementation Details of Each Ablation **17**

A.1 Trajectory-length diagnostic 17

A.2 Heuristic world-sound construction 17

A.3 Controlled module replacement 18

B Video Rendering and Baseline Audio Generation **19**

B.1 3D World Rendering 19

B.2 Baselines Generation 19

C Agent Prompts and Structured Outputs **19**

C.1 Foreground sound-source parser 20

C.2 Background and ambience parser 21

C.3 Acoustic parameter agent 22

C.4 Source inventory and agent diagnostics 24

D Layer Decomposition and Mask Processing **25**

E Source Placement and Energy Normalization **25**

E.1 Surface sampling 25

E.2 Source-count normalization 25

E.3 Localized versus diffuse labels 26

F VLM Qualitative Evaluation Prompts **26**

F.1 Original long-horizon VLM results 28

F.2 Duration-matched automated comparison 28

G Metric Definition **29**

G.1 Semantic Metrics 29

G.1.1 LAION-CLAP. 29

G.1.2 ImageBind text–audio and image–audio scores. 29

G.1.3 Audio Flamingo 2 caption-based evaluation. 30

G.1.4 Reasons for the three metrics 30

G.1.5 Scene-level uncertainty 30

G.2 Spatial Metrics 30

G.2.1 Windowed ILD 30

G.2.2 ILD proxy fit. 31

G.2.3 Energy-ranking consistency. 31

G.3 Ground-truth-referenced direction of arrival 31

H Runtime Breakdown **33**

H.1 Controlled scaling with a cached world 33
H.2 Controlled moving-source rendering 34

I User Study Forms **34**

A Implementation Details of Each Ablation

For clarity and reproducibility, we summarize the exact implementation of each ablated variant used in our experiments.

No acoustic simulation. This variant keeps the same source set and 3D source positions as the full model, but bypasses the GSound propagation renderer entirely. Each localized source is evaluated using its original generated stereo waveform, resampled to the target sample rate, and scaled only by the source-level gain determined from its volume and source-power parameters. As a result, this ablation removes reflections, diffraction, transmission, and geometry-dependent listener-motion effects from the rendering stage.

No agentic acoustic parameters. This variant keeps the full geometry-aware placement and acoustic propagation pipeline, but replaces all per-source acoustic parameters with a fixed global preset. Specifically, reflectivity and scattering are clamped to constant values, and all sources share the same volume, source radius, and source power. This ablation isolates the effect of scene- and source-dependent acoustic parameter estimation.

No semantic decomposition. This variant bypasses per-label source generation and instead synthesizes a single scene-level soundtrack from the global scene prompt using Stable Audio 1.0. This setting reduces the pipeline to a text-to-audio baseline without explicit semantic source decomposition, and therefore matches the Stable Audio 1.0 comparison setting used in our experiments.

No geometry-aware placement. This variant keeps the same source waveforms and acoustic parameters as the full model, but discards the reconstructed scene-aware source locations. Instead, each source is placed on a fixed ring of radius $3\,\mathrm{m}$ around the listener, with azimuths evenly spaced from -150° to 150° while preserving the original source height. This ablation directly tests whether geometry-grounded source placement is necessary for directional correctness.

No foreground/background separation. This variant starts from a single-pass panorama parse rather than the layered foreground/background decomposition used in the full pipeline. Sources whose labels are absent from the single-pass parse are removed. The remaining sources are treated as background-style scene objects: foreground sources inherit mean background acoustic parameters and are resampled on the corresponding semantic meshes. This ablation isolates the contribution of explicit foreground/background separation to source discovery, placement, and acoustic parameter assignment.

No layered 3D reconstruction. This variant uses the same single-pass panorama parse as the previous ablation, but further removes the explicit 3D lifting stage. Instead of reconstructing source positions in the 3D scene, each source is placed directly from its panorama bounding box. The bounding-box center determines azimuth and elevation, the source radius is chosen using a simple scene-scale heuristic, and duplicate instances of the same label receive small deterministic offsets in azimuth, elevation, and radius. This ablation tests whether the explicit layered 3D representation provides benefits beyond semantic parsing alone.

Scene-level alternative audio backend. This variant follows the global-track setup of the semantic-decomposition ablation, but regenerates the same scene prompt using TangoFlux [25] instead of the default audio backend. This experiment evaluates the sensitivity of the pipeline to the choice of text-to-audio generator and illustrates that the audio synthesis module can be substituted with alternative modern backends with minimal changes.

A.1 Trajectory-length diagnostic

We show spatial metric for different trjectory length in Table 7

A.2 Heuristic world-sound construction

We evaluate two baselines that replace world-aware source construction with heuristic sound selection, source placement, and acoustic settings. Their semantic results are reported in Table 6.

Table 7: Spatial consistency under different trajectory lengths. ILD-based metrics measure agreement between the measured ILD curve and the fitted proxy ILD curve. Energy-ranking metrics compare the predicted source-energy ordering against the distance-based ordering. Higher is better for all metrics.

Method	**ILD Corr.** ↑	**ILD** R^2 ↑	**Pairwise Rank Acc.** ↑	**Spearman Score** ↑
Ours (20 sec)	0.8994	0.8110	0.8346	0.8063
Ours (40 sec)	0.8875	0.8066	0.8232	0.7577

Table 8: Controlled single- and multi-module replacement. All metrics are higher-is-better. Caption is the Audio Flamingo 2 caption-similarity metric (AF2 in the experiment record). Joint variants retain Stable Audio. These results have their own reported full-model reference and do not replace the original benchmark or component-removal results.

Variant	CLAP	IB-T	IB-I	Caption	ILD Corr.	ILD R^2
Full model	0.303	0.196	0.188	0.301	0.889	0.812
Audio → TangoFlux	0.222	0.167	0.189	0.205	0.796	0.740
Segmentation → ZIM	0.280	0.180	0.204	0.255	0.885	0.791
VLM → Sonnet 4	0.300	0.171	0.222	0.224	0.875	0.776
Depth → DA-V2	0.316	0.185	0.215	0.336	0.854	0.746
VLM + segmentation	0.321	0.180	0.209	0.279	0.876	0.788
VLM + segmentation + depth	0.323	0.184	0.205	0.274	0.893	0.806

Weak heuristic. Each scene-generation prompt is assigned to one of four broad scene categories, and generic default sounds associated with that category are used as its semantic sound inventory.

Strong heuristic. We extract key terms from the scene-generation prompt and match them to a fixed bank of sound-generation prompts. An LLM constructs the bank to cover commonly occurring sounds. Sentence-transformer embeddings provide cosine similarities between extracted keywords and bank labels; the best-matched prompts are passed to Stable Audio to generate the semantic soundtracks. The fixed resource is a prompt bank, not a set of measured spatial source positions.

Shared geometry and rendering. For both baselines, sources are placed at random locations on the mesh reconstructed by the full pipeline. Acoustic parameters are randomly sampled within the same clamping ranges used by the full method, and GSound remains the renderer. Spatial scores are not reported because the heuristics do not infer semantically corresponding world-space source positions; omitted scores should not be interpreted as zero error. Because source selection, placement, and acoustic parameterization change together, the comparison measures the complete construction strategy.

A.3 Controlled module replacement

We replace one module at a time while retaining unrelated intermediate states and downstream settings. The four substitutions replace the Stable Audio source generator with TangoFlux [25], SAM2 [46] with ZIM [28] ViT-L, the GPT-based VLM with Claude Sonnet 4, and the MoGe-2 [58] depth backend with Depth Anything V2 [61]. We additionally replace the VLM and segmentation modules together, and then also replace depth. The joint variants use Claude Opus 5 and retain Stable Audio, separating these perception/geometry changes from a simultaneous audio-backend change.

Table 8 shows that the tested substitutions remain compatible with the pipeline but affect metrics differently. For example, the audio substitution reduces CLAP and Caption, while IB-Image changes from 0.188 to 0.189. The depth substitution increases CLAP and Caption but decreases ILD correlation and R^2. Thus, these results demonstrate modular execution with quality tradeoffs, not statistical equivalence between backends. Without ground-truth depth, the depth experiment cannot establish which reconstruction is geometrically more accurate. The source-level audio substitution here is

distinct from the original *scene-level* TangoFlux ablation, which synthesizes a single global track without semantic source decomposition.

B Video Rendering and Baseline Audio Generation

B.1 3D World Rendering

For each scene session, we first render a canonical silent video clip. This clip serves as the visual input to video-conditioned baselines and as the carrier stream for all final audio–video composites. Each session directory contains a reconstructed scene representation and is paired with a predefined camera-trajectory text file. We parse the trajectory, uniformly sample frame timestamps over the target clip duration, and linearly interpolate camera positions along the path. In our rendering script, the short benchmarks use a $20\,\mathrm{s}$ clip and the long benchmarks use a $40\,\mathrm{s}$ clip. Frames are rendered at $15\,\mathrm{fps}$ with resolution 960×540 and a 75° field of view. The renderer then outputs one RGB frame for each sampled camera pose.

B.2 Baselines Generation

MMAudio. MMAudio is conditioned directly on the rendered silent video. For each session, we provide the silent clip as input, together with the selected model variant, the target duration, the classifier-free guidance strength, the number of diffusion steps, and a fixed random seed. In our experiments, we use the `large_44k_v2` variant. When available, we also parse the original scene-generation prompt and pass it to MMAudio as an optional text condition. MMAudio produces both a synthesized audio waveform and an audio–video composite obtained by attaching the generated waveform to the input clip, which we use for qualitative inspection.

Waveform-only baselines and audio–video compositing. For methods that generate audio but do not directly output a composite video, we reuse the same silent clip and remux the generated audio onto it. This category includes our acoustic renderings, Stable Audio 1.0, and other waveform-only baselines such as TangoFlux. We combine video and audio with a single `ffmpeg` call that maps the video stream from rendered world video and the audio stream from the generated waveform, copies the original video bitstream without re-encoding, encodes the audio as AAC, truncates the output to the desired clip length, and stops at the shorter stream if needed. If a generated waveform contains non-finite values, we first sanitize it by replacing NaNs/Infs before muxing. This ensures that all methods are compared using exactly the same visual content.

Stable Audio 1.0 We first rewrite the scene-generation prompt into an audio-oriented prompt using an LLM (gpt-5-mini), and then generate a single audio clip from the refined prompt. We use the released `StableAudioPipeline(stabilityai/stable-audio-open-1.0)` for generating all the results for comparison with default number of inference steps and guidance scale. For video with sound tracks, we mux the generated sound track back onto the rendered world view video as mentioned above.

SEE-2-SOUND. SEE-2-SOUND is conditioned on the scene panorama rather than the rendered camera clip. For each session, we provide the generated `panorama.png` to the released SEE-2-SOUND inference wrapper and use the resulting sound track as the baseline output.

OmniAudio. OmniAudio is video-conditioned but expects a shorter input clip than our full rendered sequence. We therefore create a temporary $10\,\mathrm{s}$ version of `render_silent.mp4` and pass it to OmniAudio's released inference pipeline using the FOA model configuration.

C Agent Prompts and Structured Outputs

All agent calls use the generated panorama and the original scene prompt as context. The original prompt is treated as a weak prior: when the prompt and generated image disagree, the visible/generated scene is prioritized. We request structured JSON-like outputs so that downstream modules can consume labels, masks, sound descriptions, and acoustic parameters without free-form parsing.

C.1 Foreground sound-source parser

The foreground parser is applied in two passes. The first pass identifies salient unobscured objects. The second pass operates after conceptual removal of the first layer and identifies remaining localized foreground entities. We restrict the second pass to compact object-like entities to avoid assigning global regions such as sky, fog, atmosphere, or general lighting to foreground layers.

Task Prompt

```
Given the panoramic image above, please complete the following tasks:
1) Identify up to ONE unobscured and most salient foreground object categories
for FOREGROUND LAYER 1 (fg1).
   - Return 1 labels only. Do not return more than one labels.
2) Now imagine fg1 objects are removed/inpainted. Identify up to One remaining
salient FOREGROUND object categories for FOREGROUND LAYER 2 (fg2).
   - Return 1 labels only. Do not return more than one labels.
   - fg2 must be discrete foreground entities (countable/localized), not global
scene regions.
   - Do NOT include sky/atmosphere/non-object descriptors (sky, clouds, sun,
moon, stars, haze, fog, horizon, background).
   - trees/buildings/water/roads ARE allowed only if clearly localized
foreground entities (e.g., 'nearby tree trunk', 'foreground building facade', '
boat').
3) Provide an inpainting prompt for removing fg1 objects, and another for
removing fg2 objects.
   IMPORTANT (CRITICAL FOR QUALITY):
   - Each inpainting prompt must ONLY describe what should appear in the removed
/masked region.
   - The prompt MUST NOT mention or name the removed objects (do not include
their nouns at all).
   - Describe plausible continuation of the surrounding content (materials/
structures/lighting/perspective).
   - Keep it SHORT: ONE sentence, <= 20 words, <= 140 characters.
   - Do NOT use long comma-separated lists.
4) Determine whether the scene is indoor or outdoor.
5) For any fg labels likely to produce sound, provide per-label:
   - sound description, give descriptions that descriptive and informational but
 always keep in mind of the context prompt., and
   - whether the sound should be DIRECTIONAL (true/false).
   - if there's animal/instrument/engine/etc that can be reasonably assumed to
be making sound, label it and give a sound description even if it's not salient
visually.
   Directional=true for localized point-like sources (e.g., waterfall, car
engine, bird near camera).
   Directional=false for diffuse/ambient sources (e.g., wind, crowd murmur,
broad rustling).
```

Schema Hint

```
Return ONLY a JSON object with this schema:
{
  "classes": "indoor" | "outdoor",
  "fg1": {
    "labels": ["label1"],
    "inpainting_prompt": "ONE sentence, <= 20 words, <= 140 chars; describes
ONLY what should appear in the masked region; MUST NOT mention removed objects"
  },
  "fg2": {
    "labels": ["label1"],
```

```
    "inpainting_prompt": "ONE sentence, <= 20 words, <= 140 chars; describes
ONLY what should appear in the masked region; MUST NOT mention removed objects"
  },
  "sound_descriptions": {
    "label": {
      "description": "sound description, give descriptions that descriptive and
informational but always keep in mind of the context prompt.",
      "directional": true
    }
  }
}
Rules:
- labels must be concrete nouns (or short noun phrases) suitable for
segmentation.
- Avoid duplicates and synonyms; pick canonical labels.
- fg1.labels must contain 1 to 2 labels.
- fg2.labels must contain 0 to 2 labels.
- fg2.labels MUST NOT contain sky/atmosphere/non-object descriptors.
- If fg2 has no objects, set fg2.labels=[] but still provide fg2.
inpainting_prompt.
```

Full Prompt Wrapper Sent to the LLM

```
{OPENAI_TASK_PROMPT}

Panorama generation prompt, you should perform task by keeping in mind this
prompt, especially for generating sound description:
'''{pano_prompt}'''

{OUTPUT_SCHEMA_HINT}
```

C.2 Background and ambience parser

After foreground removal, a second parser identifies broader sound-producing regions and global ambience. This pass is designed to recover off-screen or distributed sources that a short rendered video might miss, such as trees, rivers, markets, ocean surf, city traffic, or room tone.

Task Prompt

```
Given the panoramic image above, please complete the following tasks:
1. Identify the semantic labels in the image that will generate sound. For each
label:
   - sound description, give descriptions that descriptive and informational but
 always keep in mind of the context prompt.
   - specify whether the sound is DIRECTIONAL (true/false)
     (Directional=true for localized point-like sources; false for diffuse/
ambient sounds).
2. For each label, provide likely bounding box locations in a json file.
3. If there should be a background/ambience sound for the image, provide:
   - sound description, give descriptions that descriptive and informational but
 always keep in mind of the context prompt.
   - if there's animal/instrument/engine/etc that can be reasonably assumed to
be making sound, label it and give a sound description even if it's not salient
visually.
   - For animals, the description should just be their normal sound (e.g., '
birdsong', 'cow mooing')
   - directional (usually false for ambience)
```

Alignment Prompt

```
Important context: the panorama was generated from the following prompt.
- Use the prompt as semantic prior for what sounds/objects are likely and how to
 describe them.
- If the prompt conflicts with the image, ALWAYS follow the image.
- Do NOT invent objects not supported by the image.

Label requirements:
- Labels must be segmentable concrete nouns (e.g., 'trees', 'waves', 'traffic',
'waterfall', 'birds'),
  NOT abstract phrases.
- Avoid synonyms for the same thing (pick one canonical label).

Bounding boxes:
- Prefer PIXEL coordinates [x1,y1,x2,y2] for the provided image size.
- If you output normalized coordinates, set coord_type='normalized'.
- Scores should be in [0,1].
```

Schema Hint

```
Return ONLY a JSON object with this schema:
{
  "coord_type": "pixel" | "normalized",
  "sound_objects": [
    {
      "label": "string",
      "sound_description": "string",
      "directional": true/false,
      "boxes": [
        {"x1": number, "y1": number, "x2": number, "y2": number, "score": number
}
      ]
    }
  ],
  "background_sound": {"present": true/false, "description": "string"}
}
Notes: If coord_type='pixel', x in [0,{img_w}] and y in [0,{img_h}]. If
coord_type='normalized', coords are in [0,1] and will be scaled.
```

Full Prompt Wrapper Sent to the LLM

```
{OPENAI_TASK_PROMPT}
{OPENAI_ALIGNMENT_PROMPT}
Image size: width={img_w}, height={img_h}
Panorama generation prompt:
'''{pano_prompt}'''
you should perform task by keeping in mind this prompt, especially for
generating sound description

{schema_hint}
```

C.3 Acoustic parameter agent

The acoustic parameter agent proposes conservative scene-level and source-level settings. These values are clamped before simulation and are intended to capture qualitative acoustic character, not exact material recovery.

```
You are tuning acoustic simulation parameters for a ray-based propagation
renderer.
Return ONLY JSON using the given schema.

Scene type: {classes}
Panorama prompt (prior): {pano_prompt}
Scene generation prompt (hint): {scene_prompt_hint}

Sound labels and descriptions:
{json.dumps(per_label, indent=2, ensure_ascii=False)}

Directional flags (true=localized point-like, false=diffuse):
{json.dumps(per_label_dir, indent=2, ensure_ascii=False)}

Approx bbox counts (proxy for how many sources are placed per label):
{json.dumps(counts, indent=2, ensure_ascii=False)}

Per-label source location summary (distance to origin [0,0,0]):
{json.dumps(source_loc_summary, indent=2, ensure_ascii=False)}

Preferred ranges (must follow):
- volume: 40..90
- radius_ratio: 0.0001..0.05
- power_db: 65..80

Default source presets (baseline; modify based on scene and distances):
{json.dumps(default_source_presets, indent=2, ensure_ascii=False)}

Schema:
{
  "global": {
    "reflectivity": number (0..1),
    "scattering": number (0..1),
    "render_volume": number (0.1..2),
    "ir": number (0.5..6),
    "toggles": {
      "direct": bool, "specular": bool, "diffuse": bool,
      "diffraction": bool, "transmission": bool, "air": bool
    }
  },
  "source_presets": {
    "background_ambience": {"volume": number(40..90), "radius_ratio": number
(0.0001..0.05), "power_db": number(65..80)},
    "bg_label_dir":        {...},
    "bg_label_diff":       {...},
    "fg_dir":              {...},
    "fg_diff":             {...}
  },
  "per_label_overrides": {
    "label": {"volume": number(40..90), "radius_ratio": number(0.0001..0.05), "
power_db": number(65..80)}
  }
}

Guidelines:
- Outdoor scenes: lower reflectivity, higher scattering than indoor.
- Directional sources: smaller radius_ratio, higher power_db than diffuse.
- Background ambience: low power_db and larger radius_ratio.
- [IMPORTANT!] Use distance-to-origin as a volume and power_db setting hint:
  labels farther from origin should be boosted (higher volume, power_db, and
slightly larger radius_ratio)
  via source_presets and/or per_label_overrides so they remain audible.
- Use real-world scene priors from the scene generation prompt:
```

Table 9: Distribution of semantic audio components per scene. These counts describe labels/tracks, not sampled acoustic emitters.

Component	Mean	Median	SD	Min	25th pct.	75th pct.	Max
Foreground labels	1.17	1	0.55	0	1	1.25	2
Background labels	4.20	4	1.53	1	3	5	8
Ambience tracks	1.00	1	0.00	1	1	1	1

Table 10: Parser repeatability over five runs per evaluated scene. Entries are mean $\pm$ scene SD; bracketed values are 95% bootstrap confidence intervals. All values are percentages.

Parser	Semantic-label agreement	Directional-flag agreement
Foreground	74.2 ± 16.5	82.6 ± 24.5
	$[67.5, 81.5]$	$[71.4, 92.5]$
Background	73.3 ± 12.1	88.1 ± 10.0
	$[68.1, 78.3]$	$[83.7, 92.2]$

```
  set relative volume/power_db/radius_ratio BETWEEN labels to match how the
scene usually sounds.
  The ratios between labels matter more than absolute values.
- Keep values conservative to avoid clipping.
- Try to keep every label audible in the mix.
```

C.4 Source inventory and agent diagnostics

Semantic inventory. Table 9 summarizes foreground labels, background labels, and global ambience tracks. The two-layer foreground decomposition limits the number of separately segmented, removed, inpainted, and reconstructed foreground layers; it does not limit the complete audible inventory to two sounds. On average, a scene contains 5.37 scene-grounded labels (1.17 foreground and 4.20 background), plus one global ambience track. Background labels have their own descriptions, dry assets, visual support, and spatial grounding, whereas global ambience is a separate non-localized component. A semantic label may be represented by multiple sampled acoustic emitters, so label counts differ from rendering-source counts.

Repeated semantic parsing. We run the GPT-5-mini parser five times for each evaluated scene with the same prompt. Labels are embedded with the `all-MiniLM-L6-v2` sentence-transformer model; labels whose cosine similarity exceeds 0.7 are treated as semantic matches. We also measure agreement of the directional flag, which denotes localized-versus-diffuse source behavior rather than a continuous source azimuth. Table 10 reports mean agreement, scene standard deviation, and 95% bootstrap confidence intervals. These results quantify repeatability, not accuracy against human-annotated semantic labels.

Acoustic parameter repeatability and clamping. The acoustic-agent prompt specifies permitted parameter ranges before prediction. In the reported runs, 0.869% of predicted sound sources contain an out-of-range parameter and therefore require clamping. Table 11 summarizes mean absolute parameter changes between two runs of the same session, both in the parameter's native scale and relative to its permitted span.

Observed inventory emptiness. No reported scene has both an empty foreground inventory and an empty background inventory. This observation and the clamping rate diagnose empty outputs and range compliance; they do not measure missed-object recall, correctness of predicted parameters, or an overall pipeline success rate against annotated ground truth.

Table 11: Acoustic parameter variation between two runs of the same session, averaged over the reported sessions. The final column normalizes the absolute change by the parameter's allowed span.

Parameter	Mean absolute change	Mean % of allowed span
Reflectivity	0.0217	2.17%
Scattering	0.0688	6.88%
Render volume	0.0210	1.11%
Source volume	3.71	7.41%
Source radius ratio	0.00197	3.95%
Source power	1.13 dB	7.53%

D Layer Decomposition and Mask Processing

Foreground layers. The foreground parser returns labels for segmentation and fill-only inpainting prompts for object removal. Fill-only prompts describe the desired replacement content inside the mask rather than restating the removed object. This choice reduces semantic regrowth, where the inpainting model reconstructs the removed category because the prompt still mentions it. For example, when removing a fountain from a plaza, the fill prompt should describe "continuous plaza paving and background architecture" rather than "remove the fountain from the plaza."

Mask generation and cleanup. For each label, we obtain a candidate mask using the segmentation backend. We discard masks whose area is too small to be reliable or too large for a compact foreground object. For overlapping foreground masks, we keep the higher-confidence or more specific label and subtract it from lower-priority masks. For background masks, overlap is allowed when a region can plausibly support multiple sounds, but duplicated near-identical masks are merged.

Background parsing after foreground removal. The background parser operates on a foreground-removed panorama. This avoids a common failure mode in which salient objects dominate the parse and suppress broader ambience. For example, a single visible car should not prevent the model from identifying city traffic, crowd murmur, or street ambience in the rest of the panorama.

Sky handling. Sky and atmosphere are not treated as localized foreground sources. They are only used for visual scene composition and, when appropriate, for global ambience descriptions such as wind, rain, thunder, or distant outdoor ambience.

E Source Placement and Energy Normalization

E.1 Surface sampling

Each localized source label is associated with a semantic mesh or world-sheet region. Given a triangular mesh with faces f and areas A_f, we sample a face with probability

$$p(f) = \frac{A_f}{\sum_{f'} A_{f'}}. \tag{2}$$

For a selected triangle with vertices $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2$, we sample barycentric coordinates uniformly and obtain a surface point $\mathbf{p}$. If $\mathbf{n}$ is the local unit normal, the source is offset slightly from the surface as

$$\mathbf{x} = \mathbf{p} - \mathbf{n}(\delta + \epsilon), \tag{3}$$

where δ is a mesh-scale-dependent offset and ϵ is small random jitter. This offset prevents degenerate source–surface coincidence in the acoustic renderer. For diffuse labels, we sample multiple points over the semantic support region. For compact labels, we use fewer samples and optionally collapse nearby samples to avoid over-counting a single object.

E.2 Source-count normalization

When a semantic label is represented by N sampled source points, naive playback can make that label louder simply because more points were sampled. We therefore normalize source energy across all

points belonging to the same label. If P_0 is the nominal source power and v_0 is the nominal volume, each sampled point receives

$$P = P_0 - 10\log_{10}(N), \qquad v = \frac{v_0}{\sqrt{N}}. \tag{4}$$

This keeps the aggregate emitted energy approximately stable as the number of samples changes. The normalization is especially important for distributed ambience, where a forest, crowd, or ocean region may be represented by many source points.

E.3 Localized versus diffuse labels

A localized label is intended to have a compact, inspectable world-space origin, such as a dog, vehicle, performer, fountain, or machine. A diffuse label is distributed across a larger visual support, such as wind in trees, crowd murmur, ocean surf, or market ambience. The locality attribute affects both the number of source samples and the acoustic radius assigned to each sample. Localized labels typically use smaller radii and fewer samples; diffuse labels use larger radii or multiple distributed samples.

F VLM Qualitative Evaluation Prompts

Scene Summarization System Prompt

```
You summarize the shared visual context for a soundtrack evaluation study. The
images come from one 3D scene and one fixed camera trajectory. Your job is to
describe the visible scene content and the listener motion cues that a later
audio judge should use. When explicit trajectory metadata is provided, treat it
as the ground-truth listener motion and viewing direction. Use the images to
connect that motion to visible objects and scene regions. Do not evaluate
rendering quality. Do not invent sounds unless they are visually grounded.
```

Scene Summarization User Prompt Template

```
Summarize the shared scene context from the attached images.

Explicit listener trajectory metadata:
{trajectory_text}

Return JSON only with this schema:
{
  "scene_summary": "brief visual description of the scene and main actions",
  "camera_trajectory_summary": "brief description of the camera path over time",
  "listener_motion_summary": "brief description of the implied listener
orientation and motion",
  "salient_visible_events": ["short event or object cue", "short event or object
 cue"],
  "spatial_audio_cues": ["short cue relevant to spatial sound alignment", "short
 cue relevant to spatial sound alignment"],
  "uncertainties": ["brief uncertainty if any"]
}

Rules:
- Keep the summary concise and concrete.
- Focus on cues that matter for judging soundtrack alignment.
- If explicit trajectory metadata is provided, use it as the authoritative
listener motion/view-direction signal.
- Use the images to identify which visible objects or scene regions align with
that motion.
- If there is no clear uncertainty, return an empty list for "uncertainties".
```

Independent Audio Evaluation System Prompt

```
You are running a human-study-style soundtrack evaluation. The video frames have
 already been summarized for you in text. Judge only the soundtrack against that
 shared scene context and apparent listener motion. Do not judge rendering
quality or video quality. Be conservative and internally consistent.
```

Independent Audio Evaluation User Prompt Template

```
Evaluate one candidate soundtrack independently.

Candidate soundtrack id: "{soundtrack_alias}"

Shared scene context:
{json_prompt_blob(scene_context)}

Explicit listener trajectory metadata:
{trajectory_text}

Rate the soundtrack on a 1-5 integer scale for:
1. overall_consistency: Overall consistency between the video and the sound
2. spatial_change_clarity: Clarity of spatial changes in the sound
3. listener_motion_consistency: How well the sound reflects the listener's
viewing direction and motion
4. spatial_audio_plausibility: Plausibility of the spatial audio
5. naturalness_realism: Naturalness / realism of the audio

Important instructions:
- Judge this candidate independently. Do not compare it against any other
candidate.
- Focus on the soundtrack, not the rendering quality.
- Use the explicit trajectory metadata as the ground-truth listener motion when
it is provided.
- If no explicit trajectory metadata is provided, fall back to motion inferred
from the frames.
- When rating spatial_change_clarity, consider whether spatial variation is
noticeable and understandable.
- When rating listener_motion_consistency, consider whether audio changes match
the listener orientation and movement implied by the trajectory metadata and
camera trajectory summary.
- Keep justifications brief but specific.
- If the spatial effect is weak or unclear, score it low rather than assuming
spatial motion.

Return JSON only with this schema:
{
  "soundtrack_id": "{soundtrack_alias}",
  "scores": {
    "overall_consistency": {"score": 1, "justification": "brief"},
    "spatial_change_clarity": {"score": 1, "justification": "brief"},
    "listener_motion_consistency": {"score": 1, "justification": "brief"},
    "spatial_audio_plausibility": {"score": 1, "justification": "brief"},
    "naturalness_realism": {"score": 1, "justification": "brief"}
  },
  "overall_comment": "brief overall comment"
}
```

Final Ranking System Prompt

```
You are producing the final human-study-style ranking for soundtrack candidates.
 Each candidate was already judged independently. Use the shared scene context
and the independent evaluations to rank the candidates best to worst overall.
Focus on sound quality, clarity of spatial motion, and consistency with the
scene and listener motion. Do not judge rendering quality.
```

Final Ranking User Prompt Template

```
Rank {len(candidate_ids)} soundtrack candidates for the same scene from best to
worst overall.

Shared scene context:
{json_prompt_blob(scene_context)}

Explicit listener trajectory metadata:
{trajectory_text}

Independent candidate evaluations:
{json_prompt_blob(independent_evaluations)}

Important instructions:
- Judge the final ranking using the shared scene context and the independent
evaluations above.
- The scene content and camera trajectory are the same for all candidates.
- Treat the explicit trajectory metadata as the ground-truth listener motion
when it is provided.
- Focus on sound quality, clarity of spatial motion, and consistency with the
scene/listener motion.
- Produce a strict total ordering with no ties.
- Keep justifications brief but specific.

Return JSON only with this schema:
{
  "ranking": [
    {"rank": 1, "soundtrack_id": "candidate_x", "justification": "brief"},
    ...
  ],
  "overall_comment": "brief summary of why the top-ranked candidate won"
}
```

F.1 Original long-horizon VLM results

The following table and plots retain the original three-method long-horizon VLM study.

F.2 Duration-matched automated comparison

The long-horizon studies evaluate audio–visual consistency and listener-motion-dependent behavior over 20–40 s trajectories. The released See-2-Sound and OmniAudio configurations used in our comparisons provide approximately 5–10 s outputs. We therefore evaluate these baselines in a separate automated comparison in which every candidate is uniformly truncated to 5 s (Table 3).

The comparison reports model-assigned mean MOS, mean rank, Rank-1 rate, Top-2 rate, and Borda score for all five methods. It does not add human participants or extend the original three-method human study. The original long-horizon VLM results are retained in Table 12 and Figures 6 and 7; those plots do not represent the new five-method comparison. Ranking and Borda values from the three- and five-candidate studies should not be directly compared because both the candidate set and evaluated duration differ.

Table 12: Original long-horizon VLM-based comparison over 20–40 s trajectories. Higher is better except mean rank. These are three-candidate results, distinct from the five-second comparison.

Method	Mean MOS ↑	Mean Rank ↓	Rank-1 ↑	Top-2 ↑	Borda ↑
Stable Audio 1.0	2.93	2.33	16.7%	50.0%	1.67
MMAudio	<u>2.99</u>	<u>1.99</u>	<u>29.5%</u>	<u>71.8%</u>	<u>2.01</u>
Ours	**3.02**	**1.68**	**53.8%**	**78.2%**	**2.32**

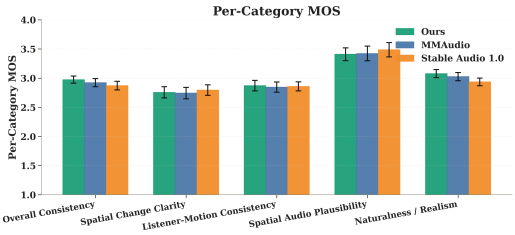

Figure 6: Original long-horizon VLM per-category MOS for three methods.

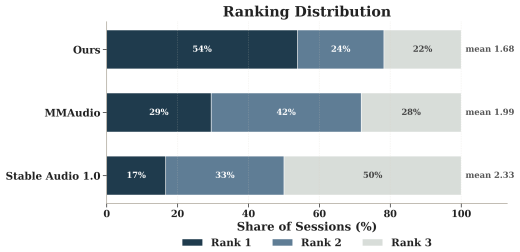

Figure 7: Original long-horizon VLM rank distribution for three methods.

G Metric Definition

G.1 Semantic Metrics

We use two complementary semantic evaluation protocols. The first is an embedding-based protocol, which scores text–audio and image–audio agreement using LAION-CLAP and ImageBind. The second is a caption-based protocol, which uses Audio Flamingo 2 to describe the generated audio and then measures semantic agreement between the generated caption and the scene prompt.

Session inputs and prompt resolution. For each session, for the embedding-based protocol, the scene text is used directly as the text query unless optional LLM rewrite is explicitly enabled. For the caption-based protocol, the default reference text is an audio-relevant filtered version of the scene prompt, in which a fixed list of mostly visual words (e.g., colors, lighting, view/composition terms, and rendering descriptors) is removed before similarity scoring.

G.1.1 LAION-CLAP.

For the CLAP-based runs reported in this work, we use the official LAION-CLAP implementation with the checkpoint `music_speech_audioset_epoch_15_esc_89.98.pt`. Given a text query x and an audio waveform a, the evaluator extracts a normalized text embedding $\mathbf{t}$ and a normalized audio embedding $\mathbf{u}$ and reports their cosine similarity

$$s_{\mathrm{CLAP}}(x, a) = \cos(\mathbf{t}, \mathbf{u}).$$

At the session level, audio is clipped to the first 20 s before scoring in order to keep evaluation length consistent across methods. The resulting scalar is stored as the session's `text_audio` score, and dataset-level results are obtained by averaging over all successfully processed sessions.

G.1.2 ImageBind text–audio and image–audio scores.

We complement CLAP with ImageBind using the released model. The text–audio score is computed analogously to CLAP: the scene text and the session audio are embedded, ℓ_2-normalized, and compared with cosine similarity,

$$s_{\mathrm{IB,text}}(x, a) = \cos(\mathbf{z}_x, \mathbf{z}_a).$$

For image–audio evaluation, we convert the session panorama from equirectangular projection into multiple perspective views before embedding. In the default session-level configuration used in our large runs, we use the four horizontal views *front*, *right*, *back*, and *left*; an optional flag also enables

Table 13: Semantic scores with bootstrap 95% confidence intervals over the original 80 scenes.

Metric	Mean	95% CI
CLAP	0.317	$[0.286, 0.348]$
IB-Text	0.220	$[0.130, 0.260]$
IB-Image	0.262	$[0.142, 0.302]$
Caption	0.276	$[0.230, 0.322]$

up and *down* views. Each perspective image is rendered at 512×512, then resized and center-cropped to the CLIP/ImageBind input resolution of 224×224. If $\{\mathbf{z}_{v_i}\}_{i=1}^{N}$ are the normalized image-view embeddings and $\mathbf{z}_a$ is the normalized audio embedding, we report

$$s_{\mathrm{IB,img}}^{\mathrm{mean}} = \frac{1}{N} \sum_{i=1}^{N} \cos(\mathbf{z}_{v_i}, \mathbf{z}_a),$$

G.1.3 Audio Flamingo 2 caption-based evaluation.

Our caption-based semantic metric uses the released Audio Flamingo 2 checkpoint `nvidia/audio-flamingo-2-0.5B`. Audio is converted to mono at 16 kHz, normalized to $[-1, 1]$, divided into CLAP-style windows according to the AF2 inference configuration, and zero-padded when needed. Caption generation is conditioned on the text prompt `"Caption the input audio."`. The generated text is then cleaned by removing separator, padding, and end-of-chunk tokens.

Prompt–caption similarity. After AF2 produces a caption $\hat{c}$ for a session, we compare it against the audio-relevant reference prompt $\tilde{x}$ using a SentenceTransformer encoder. In our runs, this encoder is `all-MiniLM-L6-v2`. Let $\phi(\cdot)$ denote the normalized sentence embedding. The final prompt–caption score is

$$s_{\mathrm{AF2}}(\tilde{x}, \hat{c}) = \cos\big(\phi(\tilde{x}), \phi(\hat{c})\big).$$

At the dataset level, we report the mean of these per-session similarities over all successfully processed sessions.

G.1.4 Reasons for the three metrics

The three metrics emphasize different aspects of semantic fidelity. LAION-CLAP and ImageBind text–audio are direct cross-modal retrieval scores against the original scene description; ImageBind image–audio further tests whether the generated sound is consistent with the scene appearance under multiple panorama views; and Audio Flamingo 2 measures whether a large audio-captioning model can recover a textual description that remains semantically close to the intended prompt. We therefore treat these metrics as complementary rather than interchangeable.

G.1.5 Scene-level uncertainty

We report bootstrap 95% confidence intervals for our semantic scores over the original 80 generated scenes (Table 13). The means are shown at the precision reported in the uncertainty analysis. These intervals characterize the scene-level evaluation; they do not constitute a paired significance test against the baselines.

G.2 Spatial Metrics

G.2.1 Windowed ILD

Since binaural cues are usually treated as locally stationary, we use the standard windowed ILD metrics for evaluation. For a binaural signal with left and right channels y_L and y_R, we compute ILD in short windows indexed by m:

$$\mathrm{ILD}(m) = 10 \log_{10} \frac{\sum_{t \in \Omega_m} y_L(t)^2 + \varepsilon}{\sum_{t \in \Omega_m} y_R(t)^2 + \varepsilon}, \tag{5}$$

Table 14: Original ILD diagnostics. Correlation and R^2 measure agreement with the fitted geometry-based proxy. Mean absolute ILD (dB) measures lateralization strength, not directional accuracy by itself.

Method	ILD Corr.	ILD R^2	Mean $\mathrm{ILD}_{\mathrm{abs}}$
See-2-Sound	0.1763	0.0408	1.6938
OmniAudio	0.3050	0.1270	2.9048
Stable Audio 1.0	0.1393	0.0291	1.2081
MMAudio	–	–	–
Ours	0.8958	0.8097	9.4735

where Ω_m is the analysis window and ε prevents numerical instability. Here we use 0.2s for window size and ensuring 50% overlapping between windows.

G.2.2 ILD proxy fit.

Prior acoustic studies show that ILD varies systematically with source azimuth and can often be approximated by a sinusoidal trend in the horizontal plane; its magnitude also increases as the source becomes closer, especially for lateral directions [29]. Motivated by this, we define a simple geometry-based proxy. We fit the geometry-based diagnostic proxy,

$$\mathrm{ILD}_{\mathrm{proxy}}(t) = k\frac{\sin(\theta(t))}{d(t)^{\alpha}} + b, \tag{6}$$

where $\theta(t)$ is the signed horizontal listener-source angle and $d(t)$ is listener-source distance. In the implementation, α is selected by grid search over

$$\alpha \in \{0, 0.05, 0.10, \ldots, 2.50\},$$

and for each fixed α, the parameters k and b are obtained by least squares. We report three metrics: ILD Corr., the Pearson correlation between the measured ILD curve and the fitted proxy; ILD R^2, the fraction of ILD variance explained by the proxy; and mean $\mathrm{ILD}_{\mathrm{abs}}$, the average absolute ILD magnitude in dB, where larger values indicate stronger left-right lateralization. Scores are averaged across sources and scenes, with results summarized in Table 14. For mono-channel baselines such as MMAudio, spatial channel cues are absent, so these metrics are not applicable.

G.2.3 Energy-ranking consistency.

For each scene, let E_i be the rendered energy of source i, and let $\bar{d}_i$ be its average distance to the listener along the trajectory. Pairwise rank accuracy is the fraction of source pairs whose predicted acoustic-energy ordering agrees with the inverse-distance ordering:

$$\mathrm{Acc}_{\mathrm{pair}} = \frac{1}{|\mathcal{P}|}\sum_{(i,j)\in\mathcal{P}} \mathbf{1}\left[(E_i - E_j)(\bar{d}_j - \bar{d}_i) > 0\right]. \tag{7}$$

We also report Spearman rank correlation between E_i and $-\bar{d}_i$. These metrics are computed only when per-source renders are available and used for our pipeline in the ablation study in Section 5.

G.3 Ground-truth-referenced direction of arrival

Reference azimuths. Ground truth is defined relative to the constructed audible world. Each evaluated localized semantic label is one logical source whose position $\mathbf{s}_i$ is the centroid of its associated emitters. Background emitters are excluded from the reference source set in this evaluation. Listener position and orientation are interpolated from the known trajectory. Let $\mathbf{l}(t)$, $\mathbf{f}(t)$, and $\mathbf{u}(t)$ be listener position, forward direction, and up direction. With

$$\mathbf{r}(t) = \frac{\mathbf{f}(t)\times\mathbf{u}(t)}{\|\mathbf{f}(t)\times\mathbf{u}(t)\|}, \tag{8}$$

the reference horizontal azimuth is

$$\theta_i(t) = \mathrm{wrap}_{[-180,180)}\left[\frac{180}{\pi}\,\mathrm{atan2}((\mathbf{s}_i - \mathbf{l}(t))\cdot\mathbf{r}(t), (\mathbf{s}_i - \mathbf{l}(t))\cdot\mathbf{f}(t))\right]. \tag{9}$$

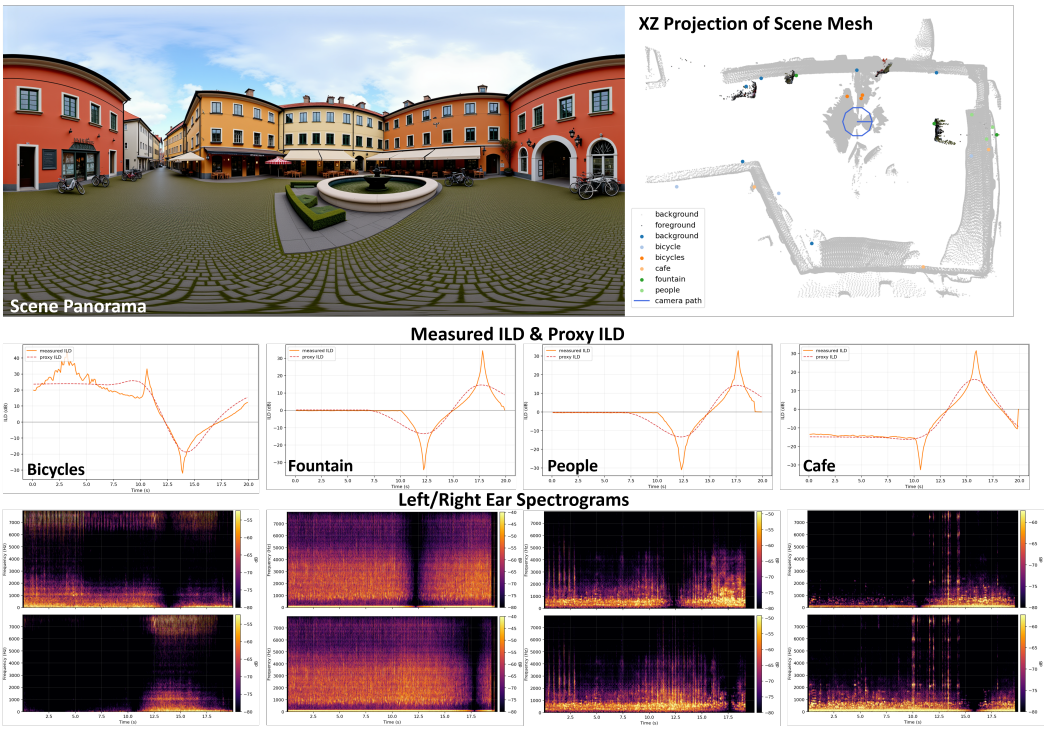

Figure 8: Qualitative spatial analysis for one generated scene. We visualize the panorama, the top-down source layout, the proxy ILD curve induced by listener–source geometry, the measured ILD from the rendered binaural audio, and channel-wise spectrograms. The measured ILD follows the geometric trend, indicating that the soundtrack changes consistently with listener motion.

Table 15: Full-circle isolated-source DoA recovery for our method. Angular errors are in degrees; accuracy is a percentage.

Metric	Estimate	95% CI
MAE ↓	6.91°	$[3.51^\circ, 12.04^\circ]$
Median error ↓	2.96°	$[2.64^\circ, 3.31^\circ]$
Accuracy@15° ↑	96.1%	[92.8%, 98.9%]

Positive azimuth is listener-right. This reference tests directional consistency with assigned source positions; it is not an independently measured real-world source-position or depth ground truth.

Isolated-source estimator. We estimate full-circle horizontal DoA from binaural windows of at least 0.5 s using multi-band ILD, GCC-PHAT interaural time difference, a low-weight pinna-spectral cue for front–back disambiguation, and circular trajectory decoding. Calibration uses 36 azimuths at 10° intervals over the full circle. Table 15 reports the resulting angular errors and accuracy within 15°.

Mixture activity mask. Directions are estimated from the final mixed soundtracks. A reference source is active only when its dry signal is above -60 dBFS and within 40 dB of its session-level maximum. The same reference activity mask is used for every method, avoiding method-dependent selection of favorable windows. Exclusion from the reference inventory should not be interpreted as source separation of the mixed waveform.

Channel-specific mixture estimators. Our binaural outputs use held-out GSound calibration and circular ILD peak detection. See-2-Sound [14] uses its $\pm x$ and $\pm z$ microphones to estimate horizontal direction. For OmniAudio [39], we use first-order Ambisonic active intensity and convert its coordinates to the same positive-listener-right convention. These channel-specific estimators

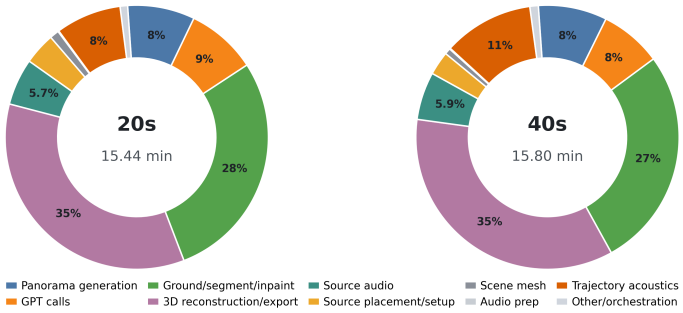

Figure 9: Runtime distribution of the full generation pipeline for 20s and 40s trajectories.

respect the different output geometries; the comparison therefore measures recoverable direction under each method's estimator. MMAudio is evaluated in its released mono configuration without post-hoc spatialization. Stable Audio 1.0 produces generic stereo without a defined binaural HRTF, array geometry, or listener-centric coordinate reference. Neither is assigned an absolute-azimuth DoA score.

Matching and metrics. We perform one-to-one Hungarian assignment between predicted and active reference directions using circular angular distance,

$$d_{\text{circ}}(\hat{\theta}, \theta) = \left| \text{wrap}_{[-180,180)}(\hat{\theta} - \theta) \right|. \tag{10}$$

Matched MAE measures error over assigned direction pairs. At angular tolerance $\delta \in \{15°, 45°\}$, a pair within tolerance counts as a true positive, while remaining predictions and references contribute false positives and false negatives. For the corresponding counts,

$$F_1@\delta = \frac{2\,\text{TP}_\delta}{2\,\text{TP}_\delta + \text{FP}_\delta + \text{FN}_\delta}. \tag{11}$$

The results in Table 2 show lower matched error and higher $F_1@15°$ for our method, but higher $F_1@45°$ for See-2-Sound. Matched MAE must be interpreted alongside F_1, since a low error over assigned pairs alone does not establish recovery of all sources. Mixture DoA is complementary to, rather than a substitute for, perceptual source-distinctness evaluation.

H Runtime Breakdown

We measure the runtime of the full pipeline on a single GPU with a network-mounted hard drive. The reported timing includes sound generation, acoustic simulation, GPT network calls, and HunyuanWorld generation. The main bottleneck is 3D world generation and file saving, while our agentic sound generation and acoustic simulation account for a relatively small fraction of the total runtime. The results are provided in Figure 9 and Table 16.

H.1 Controlled scaling with a cached world

We hold the reconstructed world fixed and vary trajectory duration or the rendering-source cap. Panorama generation, segmentation, inpainting, and reconstruction are therefore unchanged; source setup and acoustic propagation constitute the varying costs. These measurements are distinct from the end-to-end timings in Table 16.

Table 17 shows increasing runtime with longer trajectories and larger source caps over the tested range, while the reconstructed world can be reused. These measurements do not establish an asymptotic complexity bound or real-time performance. GSound is CPU-based, so absolute runtime and variability also depend on available threads and concurrent CPU load.

Table 16: Full runtime breakdown of our generation pipeline. The dominant costs are perception and 3D world reconstruction/export, while source-audio generation and acoustic rendering account for a smaller fraction of the total runtime.

Operation	**20s**		**40s**	
	Sec.	**Share**	**Sec.**	**Share**
Panorama generation	76.50	8.3%	80.00	8.4%
GPT calls	80.04	8.6%	71.74	7.6%
Ground/segment/inpaint	263.00	28.4%	256.50	27.1%
3D reconstruction/export	323.41	34.9%	334.06	35.2%
Source audio	53.00	5.7%	55.50	5.9%
Source placement/setup	36.00	3.9%	27.50	2.9%
Scene mesh	10.50	1.1%	7.00	0.7%
Audio prep	1.00	0.1%	1.00	0.1%
Trajectory acoustics	74.70	8.1%	105.07	11.1%
Other/orchestration	8.52	0.9%	9.83	1.0%
Total	**926.67**	**100.0%**	**948.20**	**100.0%**

Table 17: Controlled cached-world scaling. Left: 32 sources with varying trajectory duration. Right: a fixed 20 s trajectory with varying source cap. Times are in seconds; the $\pm$ terms reproduce the timing variability reported with the experiments. Source caps concern rendering emitters, not semantic labels.

Duration (s)	Reported time (s)
5	21.36 ± 9.86
10	34.42 ± 10.61
20	46.34 ± 13.45
40	57.67 ± 19.39
80	106.20 ± 31.77
160	184.14 ± 18.08

Source cap	Reported time (s)
1	22.32 ± 11.06
2	31.09 ± 10.42
4	32.09 ± 9.43
8	34.49 ± 10.92
16	40.00 ± 10.06
32	46.34 ± 13.45
64	63.17 ± 17.75
128	92.10 ± 17.85

H.2 Controlled moving-source rendering

To test the rendering interface beyond fixed source positions, we provide heuristic trajectories for the sources and the listener directly to GSound. Source audio is generated by our pipeline, and each source starts at the position used in the original scene. Both listener and source trajectories last 20 s. Geometry remains static: the experiment does not construct moving visual objects, infer source motion, or reconstruct dynamic geometry.

The test demonstrates that generated source assets can be rendered along provided trajectories, reaching 32 moving sources with a warm-render time of 106.09 s and reported memory of 978.6 MiB. Automatic dynamic-scene construction remains future work. The timings do not by themselves validate motion estimation, visual synchronization, or Doppler behavior.

I User Study Forms

Here we show the user study google form setup we used.

Table 18: Moving-source rendering with supplied 20 s source and listener trajectories in a static reconstructed world. Warm-render times and paired slowdowns are reported as measured; memory is in MiB. This is a rendering-stage proof of capability, not end-to-end dynamic-world construction.

Moving sources	Warm render (s)	Paired slowdown	Memory (MiB)
1	24.51	1.00×	544.5
2	51.81	2.12×	558.2
4	65.98	2.66×	586.4
8	77.86	3.22×	641.4
16	89.95	3.67×	753.6
32	106.09	4.33×	978.6

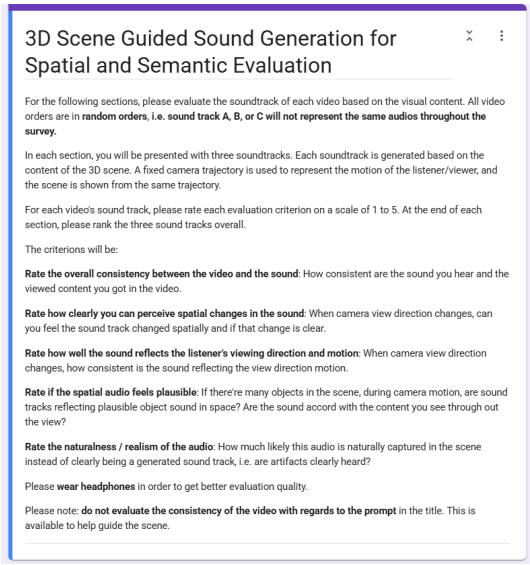

Figure 10: User Instruction Page

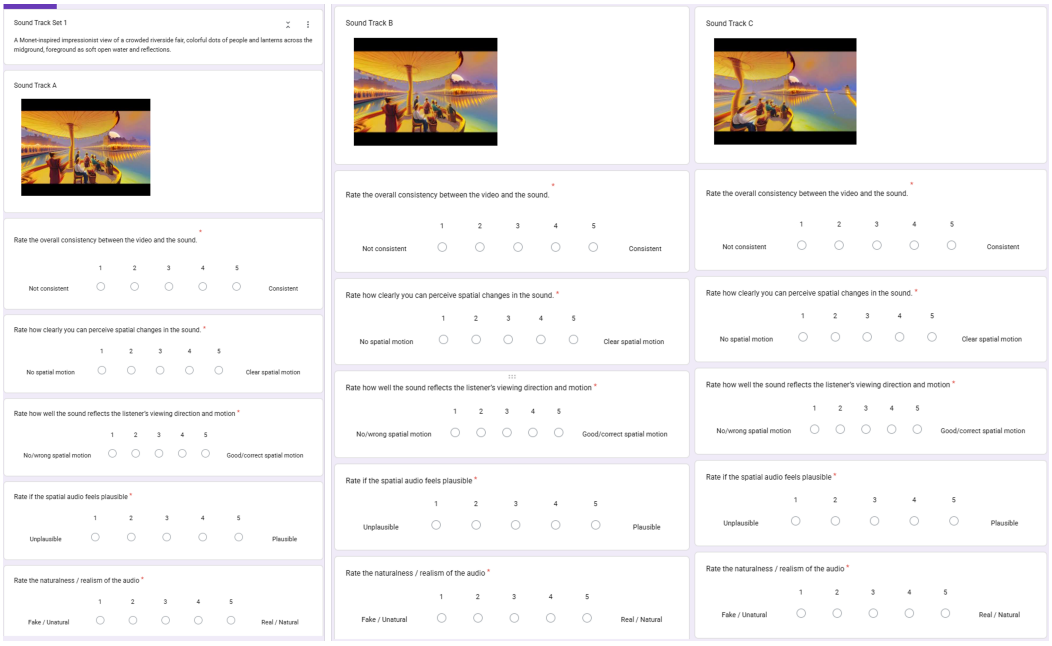

Figure 11: User Study MOS Questions

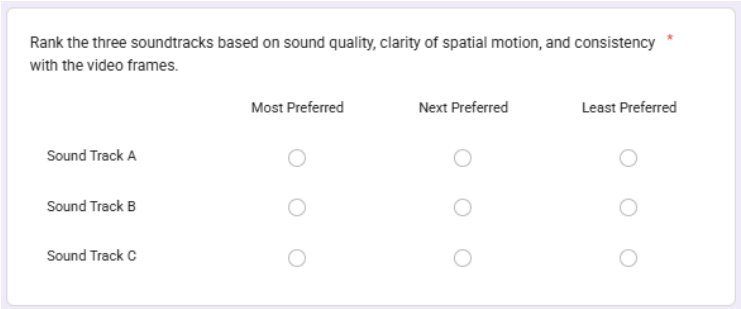

Figure 12: User Study Ranking Question

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the proposed training-free, agentic framework for audible world models in 3D world-generation pipelines where generating spatial audio from an explicit world representation improves spatial awareness, and the experiments in Sections 4 and 5 support the claims on semantic quality, spatial consistency, human/VLM evaluation, and ablations.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 discusses limitations, including the focus on only static scenes with fixed geometry, the lack of moving sound sources and dynamic scene changes, limitations in modeling complex indoor acoustics, and dependence on current text/audio generation backends.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not present theoretical theorems or formal proof results. It defines proxy metrics and evaluation equations for spatial audio analysis, but these are empirical evaluation tools rather than theoretical results.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Details of implementation is for ablation is in Section A, details for baselines methods in Section B, agents prompts in Section C, layer decomposition in Section D, source placement in Section E and definition of metrics in Section G.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Details for pipeline agent prompts and settings are provided in Section C. We plan to relase code and scripts upon acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The method is training-free, so optimizer and training hyperparameters are not applicable. The paper describes the evaluation over generated scenes, semantic and spatial metrics, baselines, mono mixdown protocol, VLM and human evaluation settings, ablation variants, and prompt templates in Section 4, 5, A, B, F, G.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Results such as averaged metrics, MOS values, ranking metrics, and ablation tables are reported as statistical average. Experiments with confidence interval are shown in Figure 2 and 6.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Provided at the beginning of Section 4. Timing break down provided in Section H.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We've read and adhered to the Code of Ethics. The work uses only publicly available code and used self-generated data. No ethical concerns were identified.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: While our work is on 3D audible world model generation which motivates positive uses for richer audible world models and immersive 3D experience, we focus mainly on system design and evaluation and doesn't introduce qualitatively new societal risks beyond the current existing sound / world generation methods. Hence, we didn't include a sperate broader-impact discussion.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release a new high-risk pretrained model with elevated misuse risk. The method is a training-free framework built from existing world-generation, audio-generation, VLM, and acoustic simulation components.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All external datasets and codebases are cited in the paper with their original licenses respected.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not release a new dataset or model checkpoint.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [Yes]

Justification: Details provided in Section I.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [No]

Justification: The current study with human subjects doesn't expose risks beyond existing sound / world generation models and, hence, didn't include a section for IRB approval.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: This paper introduced a multiagent training free world sound generation pipeline which involves usage of LLMs. Details of its usage together with prompts are provided in Section 3, 4, B, C, F.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.