Audio Engineering Society

Convention Paper 10235

Presented at the AES 159th Convention
2025 October 23–25, Long Beach, CA, USA

*This paper was peer-reviewed as a complete manuscript for presentation at this convention. This paper is available in the AES E-Library (http://www.aes.org/e-lib), all rights reserved. Reproduction of this paper, or any portion thereof, is not permitted without direct permission from the Journal of the Audio Engineering Society.*

# Toward Agentic Immersive Audio Generation

Toni Hirvonen[1], Carlos Tejeda-Ocampo[2], and Ema Souza-Blanes[1]

[1]*Samsung Research America*
[2]*Samsung Research Tijuana*

Correspondence should be addressed to Toni Hirvonen (t.hirvonen@samsung.com)

**ABSTRACT**

A method for immersive audio generation from natural language prompts is presented. We circumvent end-to-end modeling of modern intricate audio formats by utilizing an agentic AI workflow. In addition to a conceptual specification of the agent, an example immersive generator is implemented. A language model coordination/interface unit is combined with single-source audio models and other tools, and the system is equipped with the proposed theoretical concepts. All communication between the agent, the tools and the user is done via natural language, implying a general concept of language-driven signal processing. The performance is evaluated in listening test with an immersive 7.1.4 loudspeaker layout. The results illustrate the present capabilities of audio agents and give indications of further potential as the individual tools improve.

## 1 Introduction

Content creation is a principal aspect for any medium of expression, especially for multimedia formats. The utilized production methods have implications for the expressive capability of the medium, and therefore for the content itself [1]. This paper examines audio content creation with natural language prompts. This upcoming area has potential for both intuitive usage and deep creative capacity.

With complex next-generation immersive audio formats [2, 3, 4], the ease and speed of production has hinged on the availability and the learning curve of related tools, e.g. digital audio workstation plugins. If problematic, these aspects can result in the lack of content quality and amount, hurting both the format and the creator. This is especially true for non-professional content creators interested in artistic expression over technology. On the other hand, natural language is a highly intuitive user interface. Applying prompting would avoid the need for technical expertise on conventional tools or audio processing. Also, the creator ideally has the freedom to use as much or as little detail and effort as desired; They can decide how much should be left for AI to extrapolate and improvise regarding the outcome. Unfortunately, implementing AI for low-effort creation of quality content is not trivial.

Prompting has notably been utilized in Large Language Model (LLM) - based applications, where Transformer-based token predictors [5] are trained with a large text corpus (e.g. [6, 7]). A traditional LLM can mostly be utilized in a chat application. However, giving these models *agency* enables them for tasks that go beyond chatting, including decision-making, action and interaction with external systems. Unlike traditional reinforced learning agents that act based on heuristic policy functions and learn in restricted environments for specific situations, LLM-based agents can leverage

more comprehensive internal world knowledge, allowing impressive results even without training on specific domain data [8]. LLM agents also provide a natural language interface.

Despite LLM agents being deployed across many fields [8] and prompt-based generative AI being an active area of research, they have not been applied specifically to spatial and immersive audio. LLM agents are however gaining traction in audiovisual production. Recent developments include: ByteComposer, an LLM-powered melody composition agent [9], agent-driven lyric generation [10], ComposerX for multi-agent symbolic music composition [11], MusicAgent, an agent for music understanding, classification and generation [12], FilmCompose for soundtrack generation based on film [13], JAMMIN-GPT for live improvisation [14] and WavCraft for audio editing [15]. WavCraft escpecially includes aspects of prompt-based audio generation, but lacks the immersive audio - aspect.

Also complicating the immersive audio AI application is the effort needed to train statistical models for it directly. Immersive audio formats have several different content subtypes [16] (e.g. channel-based, object-based and scene-based audio, as well as mixed presentations). Training such end-to-end models is costly, and the data requirements often prohibiting. Thus, traditional generative audio has not been focused on immersive formats or aspects, but rather on the model details and improving the quality for traditional, content-abundant formats (i.e. mono or stereo). Meta's AudioGen [17] and MusicGen [18] are recent open-source examples. While in [19] generative modeling was outlined for spatial audio context, the model produced frequency-dependent upmix parameters akin to DirAC [20] and was only implemented for stereo audio compression.

Summarizing our motivation as 1) specialized end-to-end models being difficult to train 2) prompting being an attractive alternative for content creation; The aim of this paper is to probe the capabilities of agentic immersive audio generation. In contrast with end-to-end modeling, we propose an agentic system benefiting from smaller, reusable tools. This study also implies and exemplifies a broader concept of performing signal processing using natural language. The following sections detail the system concept and an example implementation. Finally, listening tests aiming to validate the proposed approach are discussed. To our knowledge, generative audio has not previously been subjectively evaluated for immersive aspects.

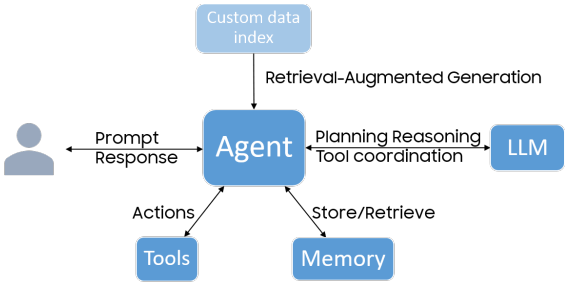



**Fig. 1:** Overview of the agentic system and its subcomponents utilized in this paper, custom data usage being optional. The agent is weighted toward processing using tools and memory. Natural language is used for communication between all parts.

## 2 System Concept

This section proposes principles and a theoretical architecture for an agentic immersive audio generator.

### 2.1 Overview

Figure 1 shows an overview of the proposed approach. Compared to typical agents in other fields, the system emphasizes tools and memory to be more *processing-based* versus data-based. Agentic AI circumvents the issues in training end-to-end models for immersive audio by re-using existing simpler models and relying on tools (computer program functions) to do smaller sub-tasks. During the process, the intermediate results can be stored to, and retrieved from the memory. Using an agent enables both language-based processing and task monitoring, as all system parts including the tool functions communicate via natural language.

As a key component to achieve this, the LLM handles prompt parsing, process planning and tool interaction. Various pre-trained LLMs can be utilized depending on the available resources and quality requirements. The LLM knowledge can be optionally extended with custom/local data for Retrieval-Augmented Generation (RAG). The system is generic and flexible toward available LLMs, computational resources and future tool developments.

### 2.2 Language Specification

Natural language as used and interpreted by humans is fundamentally non-deterministic due to ambiguity, context-dependence, pragmatics and individual variation. For effective prompt-based generation, there is nevertheless a need for common abstractions, notions and concepts. We thus aim to define key descriptions of immersive audio so that the listening impression is according to the user's intention.

The terminology, concepts and their hierarchical relations described in Figure 2 serve to include aspects of a formal language to the generation process. By internalizing the language interface specification, the agent can produce a more unambiguous result. On the other hand, the shared understanding enables extrapolation according to what is appropriate in case the prompt leaves room for interpretation. The present proposal is inspired by the hierarchical organization of similar concepts in the IAMF transmission format [4]. The dissemination of this information can be implemented in both the tool API documentation, and in the RAG custom data index.

A "sound" is an audio item that can be generated by describing the source. Intuitively, a sound corresponds to a single physical source (e.g. a snare drum), but can also have several related sources (e.g. a drum beat). Sounds are associated with temporal properties like duration, and optionally with spatial attributes like location or the capture environment description.

An "element" consists of a sound or sounds with an associated format (channel-, object-, or scene-based). A format can have a rendering method that generates the mix element from individual sounds, and include dynamic spatial metadata (in case of objects). Rendering can also involve simple channel mapping. While the spatial attributes of a single sound can be undefined or loosely described, they are explicitly realized in an element (via static mixing or as [x,y,z] Cartesian coordinates etc.).

A "presentation" has one or more mix elements intended for a joint presentation via automatic or interactive rendering [21]. Rules for combining elements of different types can be specified. Information about playback adjustments (e.g. gains or reproduction system) can be included. For transmission and storage, a presentation can be associated with a coding configuration describing the bitrates and codec instances for each element.

## 3 Tested System

An example implementation of the design discussed in Section 2 is presented. In practical terms, the most visible contribution is to extend the Meta end-to-end text-to-audio models [17, 18] with the handling of immersive aspects and more elaborate prompting. However, the goal is also to survey the future potential.

### 3.1 Agent Toolset

Given a prompt following the language specification guidelines in Section 2.2, an agent can use the tools listed in Table 1 to generate the outcome. One tool can be related to several language concepts. The tools have side effects (usually memory I/O) and return a natural language description of the tool process and outcome (error or success) according to the language specification. The LLM interpreting these outputs enables guiding the multi-stage processing and error correction.

We favored the Single-Responsibility Principle (SRP) [22]; If a tool has a single, well-defined goal, there is less ambiguity. However, increasing the number of tools to handle every atomic task can lead to confusion and process failures, due to the nonzero error probability at each step especially with consumer-grade LLMs. Keeping the number of tools reasonable can require complicated internal processing chains and many input parameters. This does not necessarily violate SRP, as long as the tool purpose-, usage- and parameter descriptions are clearly documented.

The generation of single sounds is split into two tools (1. and 2. in Table 1). This coincides with the two Meta models [17, 18]. Despite having some prompting ability, these models are utilized for single sounds since they lack immersive output and handling of complex prompts. The tool separation is further motivated by SRP, and the fact that music has been recognized as a separate neurophysiological entity [23]. Parameters for both of the single-sound tools include sound type (a single word or a multi-word internal description), gain (in dB) and output duration (in s). The latter two give the agent control over the mix in more detail, e.g. *"Add a relatively quiet 5-second sound to the mix"*.

The agentic system is flexible to utilize other models for single-sound generation as they become available. For example, speech, singing, poetry, etc. are out of

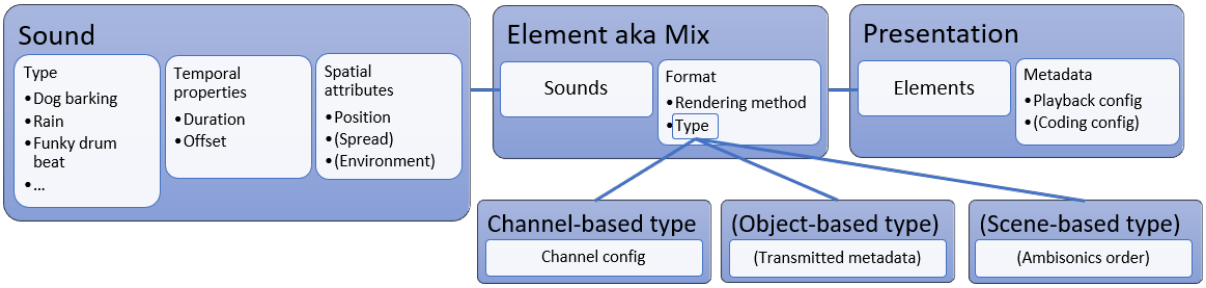



**Fig. 2:** The concepts and components of the hierarchical immersive audio language specification utilized in this paper. This knowledge is injected to the agent via a document-based RAG process and tool API instructions. The human user then utilizes the same concepts in order to have a common language for prompting. The components in parenthesis () are not used in the subjective tests of this paper, but were examined informally.

scope for the current single-sound tools similarly as in [17, 18]. Since current neural audio models are often limited in sampling frequency and quality, a simple way to complement them is by looping fixed sound files from memory. To prove this concept, our implementation used the ESC-10 database [24] as part of the environmental sound tool (1. in Table 1). The tool checks the "sound type" input parameter with a simple keyword search, and selects a random appropriate file to loop if there is a match. Otherwise, the synthesis proceeds with [17].

The mix tool (3. in Table 1) uses relatively complex internal processing. We found that having the mix done in a single tool resulted in fewer process failures compared to mixing sounds separately and then combining mixes. The tool reads lists, telling it the input sound file names, temporal offsets and spatial attributes. Despite the complexity of the API, we found the agent capable of correctly utilizing it. A format-parameter indicates the element type in Figure 2. In the tests, we limit the format to channel-based options of 1) 7.1.4 aka "immersive", 2) 5.1 aka 'surround' and 3) stereo. Each sound rendering is summed to the mix, and the final result is normalized for level.

Spatial parameters for the mix tool include the start and end positions for each sound in Cartesian format [25]. While the Cartesian representation is popular in content creation, polar representation is also viable. The information is used for balance-based rendering as in [25]; for each coordinate axis, the system first finds the relevant channels enclosing the sound location with four loudspeakers at each height layer of the layout. The left-right and front-back axis are then individually panned with a pair-panning law resulting in the gains for the four speakers. If the layout has more than one height layer (like with the 7.1.4 format), layer-wise panning is calculated based on the height coordinate value. Each pair panning of two gains $g_1$ and $g_2$ can be realized with e.g. the Tangent Law [26], given the listening position angles for the sound and the loudspeaker locations ($\varphi_A$ and $\varphi_S$, respectively):

$$\frac{\tan\varphi_A}{\tan\varphi_S} = \frac{g_1 - g_2}{g_1 + g_2} \tag{1}$$

Exact control of the sound location as a function of time was not implemented. Rather, each sound was dynamically panned between the requested start and end positions by generating the renderings at the two positions separately and fading between them over the whole duration. This was hypothesized as sufficient for the present prompting, where sounds were moved with descriptions of simple linear trajectories. While limited, the method avoids having to specify and implement spatiotemporal metadata. The mix tool API documentation further instructed to use the same start and end positions for static sources.

The tool for summing mixes together (4. in Table 1) implements the "presentation" outlined in Figure 2. Since the channel-based formats share channel orders, the tool sums mixes up to the lowest common channel count, and normalizes the result. Tools 5 - 7 in Table 1 were investigated as informal proofs of concept. Both object- or scene-based output formats were deemed possible. However, e.g. precisely controlling

| |
|---|
| 1. Generate an environmental sound, given sound type, gain, duration and output file name<br>• Can be realized by lookup database files, or via a separate sound generation model like AudioGen [17]<br>• Example prompts: *"Generate a helicopter sound."*, *"Generate 10 seconds of a forest animal sound."* |
| 2. Generate a musical sound, given sound type, gain, duration and output file name<br>• Can be realized by lookup database files, or via a separate sound generation model like MusicGen [18]<br>• Example prompts: *"Rock band playing."*, *"Generate a funky drum loop at 120 BPM."* |
| 3. Make a multichannel mix element, given a list of sounds, temporal offsets, spatial parameters, multichannel format and output file name<br>• Reads sounds from the Agent memory, and renders them to the given format<br>• Example prompts: *"Generate one minute surround mix of 5 sounds in random locations."*, *"Generate a song in 7.1.4 format."* (these prompts would also use the single-sound tools) |
| 4. Sum mixes together, given list of mixes and output file name<br>• Create a presentation combining several mixes of potentially different formats. |
| (5. Make an object-based mix element, given a sound, spatial information and output file name)<br>• Formal positional metadata can be extrapolated and generated from a verbal description. |
| (6. Make scene-based Ambisonics mix element, given Ambisonics order, sounds, spatial and environment information and output file name)<br>• Simulates sounds in a scene-based capture. |
| (7. Make presentation coding metadata, given elements and output file name)<br>• Utilizes e.g. a config template, and fills it with given elements and rates etc.<br>• Bitrate/importance per element can be determined via specialized analysis. |

**Table 1:** Overview of the tools utilized by the proposed agentic system. "Example prompt" may trigger the tool but not necessarily in isolation. The tools have side effects, e.g. saving the output file. They return a natural language description of the outcome, which is used by the agent to guide the generation process. The tools in parenthesis () are not utilized in the subjective tests of Section 4, but were examined informally.

object spatial positions as a function of time may result in complicated tool input parameters that make the agent error-prone, or need a very powerful LLM. For Ambisonics processing, source audio is mixed to the spherical egocentric position defined by the polar source location-, and room dimension input parameters. For transmission, it is desirable to process the output with an audio codec, e.g. [4]. There, parameters controlling the quality vs. bandwidth tradeoff can be used, and the coding tool can analyze the resource usage per presentation element [21].

### 3.2 Practical Implementation Details

The implementation was done in Python with the LlamaIndex[1] framework. 4-bit version[2] of the *DeepSeek-R1-Distill-Qwen-32B* [7] was used as LLM. The framework handled all control and communication between the components. As mentioned, the sound tools utilized the GPU-accelerated neural models [17, 18]. The system was run on a 2-GPU machine with one GPU used for the LLM and the other for the tools.

The agent tools utilized helper functions for memory I/O, e.g. listing current audio files in memory, checking for file overwrites and instructing using different file name suffixes. These could provide feedback for the agent self-correction. For the RAG custom data index, we parsed both 1) Wikipedia articles on sound mixing, spatial sound etc. and 2) a short custom essay on the principles of immersive audio formats and utilization of the Cartesian coordinate system. The tool API documentations included descriptions of the function and the input parameters, as well as basic usage examples.

[1] https://github.com/run-llama/llama_index
[2] https://huggingface.co/deepseek-ai

### 3.3 Prompt engineering

This section illustrates the seven prompts that were heuristically selected to showcase the implementation. Generally, the prompts are intended to cover different levels of detail. We utilized straightforward and verbose requests versus more refined techniques, such as zero-shot chain-of-thoughts and plan-and-solve [27]. We found these sometimes useful to prevent premature halting, but not mandatory.

Prompt 0 generated a random mix in 7.1.4 format. It functioned as a baseline anchor and was not evaluated directly. The variation in such random request can be limited based on the knowledge of the agent. Our outputs for prompt 0 typically included only few (2-4) sounds, but did include varying sound types.

Prompt 0

*"Create a random immersive mix."*

Prompt 1 represents an open-ended request with ample room for interpretation from both the agent and the listener. In this sense, this is a relatively easy generation task.

Prompt 1

*"Create an immersive mix of a busy city street."*

Prompt 2 is also open-ended but with a specific context, presumably increasing the distinction from a random scene.

Prompt 2

*"You are a sound mixer for a blockbuster movie. Create an immersive mix for an intense action scene."*

Prompt 3 consists of four distinct sounds with less room for interpretation.

Prompt 3

*"Create an immersive mix with four different sounds. First sound should be something that an animal makes. Second sound should be human sneezing. Third and fourth sounds should be different musical instrument sounds."*

Prompts 4 and 5 include spatial positioning aspects in addition to requesting specific sounds. While prompt 4 has three static sounds, prompt 5 adds the aspect of movement and uses only one sound source for critical spatial evaluation.

Prompt 4

*"Create an immersive mix with three different and equally loud sounds. Place the sounds spatially in the mix. First sound should be located left front. Second sound should be located right front. Third sound should be moving from left to right."*

Prompt 5

*"Create an immersive mix with a helicopter flying from center front to center back overhead."*

Prompts 6 and 7 aimed to push the complexity limitations of the agentic system. The former includes summing together two different formats (stereo and immersive), while the latter aimed for high presentation complexity. We omit specific dB-valued gain adjustments from the prompts for the sake of subjective evaluation simplicity.

Prompt 6

*"Create a stereo mix of a rock band with different instruments in different directions. Create a second immersive mix with a helicopter flying from center front to center back overhead. Sum the first and second mix together."*

Prompt 7

*"Create an immersive mix for a horror movie scene soundtrack. Have different instruments of the soundtrack be at different spatial directions. Add some appropriate environmental sounds for the scene. Each environmental sound should be 4 seconds long and placed overhead. Use a total of 8 environmental and musical sounds in the mix."*

## 4 Subjective Test

This section details the double-blind listening tests conducted with an immersive 7.1.4 loudspeaker layout [28] in an ITU-compliant listening room [29].

### 4.1 Design and Preparation

Options for testing included 1) using human-made references from the few existing databases and mimicking those with generative prompts, 2) having professional content creators craft references based on prompts, or 3) omitting human-made reference audio. The issue with crafting absolute references is that human interpretation creates variance. The sampling frequency of the present tools is also limited (16 kHz for [17] and 32 kHz for [18]), giving an unfair bias toward human-made full-band material that would be resolved by future tool developments. Thus, option 3) was chosen. Instead of using a reference of an absolute sound quality, the test goal was relaxed to show how well specific prompts performed against a random one.

The participants were asked to score how well each listened item corresponded to a verbal description of the audio scene displayed by the test user interface. That description was the actual prompt used for the generation of the specific sample. In total, there were seven trials corresponding to prompts 1-7 in Section 3.3. Since the generative process is inherently stochastic, eight different outputs of the same prompt were scored. A traditional test of merely using an single generation snapshot per prompt would in this case have presented additional issues of e.g. how to choose the snapshot that is the most representative. Generative prompt-based models have much higher variation than typical listening test samples, as they may sound completely different at waveform level, while still being valid for the given prompt at higher semantic level. Using several generation examples attempts to cover the generative scope variations.

These eight generated outputs split into two test sessions each having four different generated items per prompt. In addition, an anchor item of a random immersive scene (prompt 0) was added. Thus the participants evaluated five items at each sample trial in double-blind, MUSHRA-like manner, i.e. a multiple stimulus presentation with randomly-ordered stimuli rated on continuous scales ranging from 0-100 with, score 100 indicated the best imaginable match to the verbal description, and 0 the worst. Verbal anchors for the score scale were used as in MUSHRA [30] ("Excellent" etc.).

All the eight samples per prompt were generated offline a priori to a sample pool, from where they were randomly assigned to different listeners. While the generation process occasionally terminated without producing audio, the prompt could be repeated until an output was detected. This is easy for the user to determine by monitoring the agent process and self-evaluation. Approximately 25% of the time the process did not finish, or produced only a verbal solution proposal etc. Some generated samples were manually truncated or looped so that all items in a single trial were of similar duration. No further vetting or selection of the test samples was done.

### 4.2 Results

A total of 10 participants with normal hearing completed the grading task. They were experienced with critical immersive audio listening tests of professional material. The authors were not among the participants. No outlier listeners were found and all results were used for the analysis.

The scores given to the scenes of the appropriate prompt were combined to be compared against the random prompt 0 - generated anchor. A two-way ANOVA revealed that the condition of using the generative prompt as the scene description ("Random" versus "Prompt") had a significant effect on the results ($p<0.001$). Similarly, the interaction between the prompt as the scene description - condition and specific prompt (prompt 1-7) was significant ($p=0.003$). The specific prompt alone was not found to have a significant effect on the scores. Figure 3 presents a boxplot summary of the scores. The appropriate prompts were generally rated higher than the random scene. This validates the proposed system's general capability to generate content based on user requests. However, the variances are notable, indicating differences in subjective interpretation.

To investigate the interactions, paired comparisons using estimated marginal means and a Bonferroni adjustment were used to compare the random and specific prompts, for subsets of data isolating each prompt at a time. These comparisons revealed that the random prompt 0 samples are rated significantly lower for cases

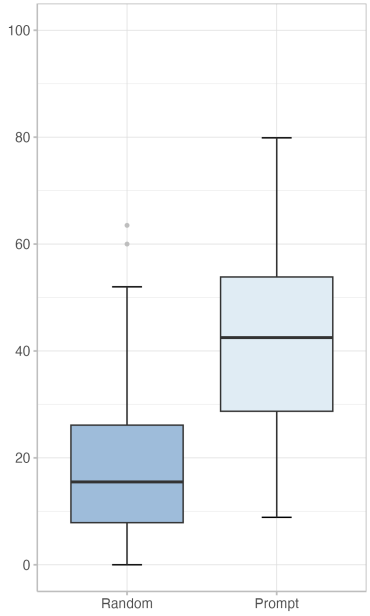



**Fig. 3:** Randomly generated immersive mix (prompt 0) compared against the mix generated with the described prompt, presented to human listeners using a 7.1.4 loudspeaker layout. At each trial, the listener was asked to rate how well the items correspond to a written description (same as the prompt) shown in the graphical user interface.

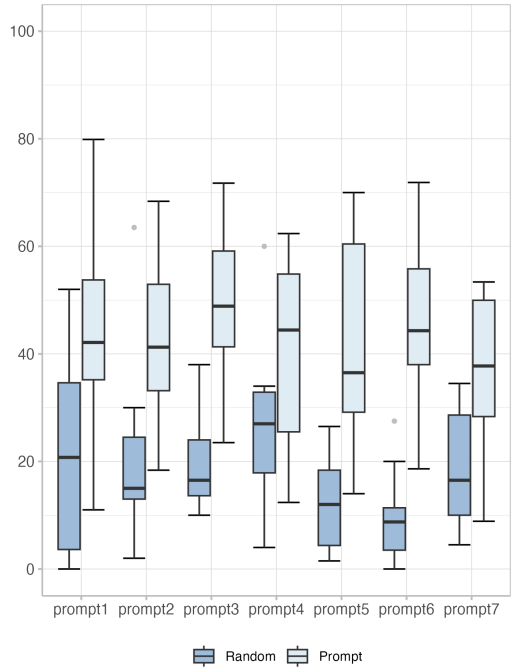



**Fig. 4:** Randomly generated immersive mix (prompt 0) compared against the mix generated with the described prompt, presented to human listeners using a 7.1.4 loudspeaker layout, for each prompt separately.

2, 3, 5 and 6 compared to the appropriate prompt. This indicates that the agentic system is capable of following instructions for detailed, as well as for open-ended generation tasks. Prompts 3 and 6 especially represent complex tasks that the agent was nevertheless able to negotiate.

Prompt 5 intentionally included only one sound that was typically created using pre-recordings (see Section 3.1) instead of neural synthesis. This enables quality assessment focusing on spatial positioning. Interestingly, the absolute scores are still roughly similar with other prompts. Based on verbal comments, the source was not heard overhead very well despite the agent usually requesting the tool do so. For this reason, it is hypothesized that the simple spatial movement algorithm and the internal spatial representation can be improved. Also, prompt 5 is slightly ambiguous on should the sound be overhead all the time or only at the end.

No statistically significant difference is found for prompts 1, 4 and 7 compared to random prompt 0. Prompt 1 is very generic and open to interpretation from both the agent and listener, so this result is not surprising, especially if the random prompt occasionally included urban sounds. Prompt 4 and the random prompt 0 often produced a similar number of sounds, complicating the comparison. Furthermore, asking for "equal loudness" may hurt the prompt 4 scores, since the mix tool did not include loudness measurements. Prompt 7 represents an intentionally difficult case due to its complexity and not surprisingly received low scores. The visual comparison of the results for different prompts is seen in Figure 4. This figure shows that all score medians are higher with a specific prompt compared to a random one. However, the statistical significance is not always evident due to high variance.

The absolute scores indicate that there is room for improvement in sound quality. The implementation does

not yet compare to a professional content as the participants imagine it. On the other hand, the difficulty of judging this aspect is evidenced by the large variance and the listener comments. A caveat of the present listening test is that separating the effect of the single-sound output quality from the agentic process is challenging. For maximal fidelity, the sound generators could have been implemented purely with lookup and looping of pre-recorded sounds.

While it is challenging to detail the higher-level generation quality in objective metrics, the agent was capable of using the tools correctly. As mentioned, the cases when the process halted prematurely or did not produce an output are trivial to detect. Otherwise, the agent generated the number of sounds, types, amounts, mix formats etc. according to what a human would interpret from the prompts approximately 95% of the time. The spatial parameters, which involve more interpretation and are less quality-critical, were correctly specified approximately 80-90% of the time as well. The agent also performed self-reflection and corrections during the process. The main bottleneck of the output quality seemed thus to be the individual tools, rather than the language specification, LLM or any other aspects. Nevertheless, the results represent the current realistic capabilities of the agentic system as a whole.

## 5 Summary

This paper examines the immersive audio generation capabilities of AI agents. For efficient communication within an agentic workflow, we propose a natural language specification for immersive audio creation that is interpretable by both humans and LLMs. To survey the present potential, an example agent implementation using commonly available open-source packages and models is tested. This method also implies a broader framework for language-driven signal processing and hopefully inspires further studies on the topic.

Subjective tests show that while the agentic system is capable of generating immersive audio that corresponds to various prompts, the overall audio quality still leaves room for improvement. Based on result analysis, listener comments and observing the agent, most of the improvement should happen within the specific tools. These include e.g. the models that generate the individual sounds and methods for spatial rendering. The agentic system itself seems to be able to fulfill the tasks at the higher conceptual level. To characterize the potential as the individual tools improve, we include the expression "toward" in the title.

## References

[1] McLuhan, M., *Understanding Media : The Extensions of Man*, New American Library, 1988.

[2] ISO/IEC 23008-3:2022, "Information Technology - High Efficiency Coding and Media Delivery in Heterogeneous Environments - Part 3: 3D audio," Standard, International Organization for Standardization, Geneva, CH, 2022.

[3] ETSI TS 103 190, "Digital Audio Compression (AC-4) Standard," Standard, European Telecommunications Standards Institute, Geneva, CH, 2018.

[4] Alliance for Open Media, "What is IAMF?" `https://aomedia.org/iamf/`, 2024.

[5] Vaswani, A. et al., "Attention Is All You Need," *Advances in neural information processing systems*, 30, 2017.

[6] Touvron, H. et al., "LLaMA: Open and Efficient Foundation Language Models," *arXiv preprint arXiv:2302.13971*, 2023.

[7] Bi, X. et al., "Deepseek LLM: Scaling Open-source Language Models with Longtermism," *arXiv preprint arXiv:2401.02954*, 2024.

[8] Wang, L. et al., "A Survey on Large Language Model Based Autonomous Agents," *Frontiers of Computer Science*, 18(6), p. 186345, 2024.

[9] Liang, X., Du, X., Lin, J., Zou, P., Wan, Y., and Zhu, B., "ByteComposer: a Human-like Melody Composition Method Based on Language Model Agent," *arXiv preprint arXiv:2402.17785*, 2024.

[10] Liu, H.-H. and Liu, Y.-W., "Agent-Driven Large Language Models for Mandarin Lyric Generation," in *O-COCOSDA 27th Conference*, pp. 1–6, 2024.

[11] Deng, Q., Yang, Q., Yuan, R., Huang, Y., Wang, Y., Liu, X., Tian, Z., Pan, J., Zhang, G., Lin, H., et al., "Composerx: Multi-Agent Symbolic Music Composition with LLMs," *arXiv preprint arXiv:2404.18081*, 2024.

[12] Yu, D. et al., "MusicAgent: An AI Agent for Music Understanding and Generation with Large Language Models," *arXiv preprint arXiv:2310.11954*, 2023.

[13] Xie, Z., He, Q., Zhu, Y., He, Q., and Li, M., "FilmComposer: LLM-Driven Music Production for Silent Film Clips," *arXiv preprint arXiv:2503.08147*, 2025.

[14] Hollowell, S., Namgyal, T., and Marshall, P., "JAMMIN-GPT: Text-based Improvisation Using LLMs in Ableton Live," *arXiv preprint arXiv:2312.03479*, 2023.

[15] Liang, J. et al., "WavCraft: Audio Editing and Generation with Large Language Models," *arXiv preprint arXiv:2403.09527*, 2024.

[16] European Broadcast Union, "Next Generation Audio," `https://tech.ebu.ch/publications/next-generation-audio`, 2019.

[17] Kreuk, F. et al., "Audiogen: Textually Guided Audio Generation," *arXiv preprint arXiv:2209.15352*, 2022.

[18] Copet, J. et al., "Simple and Controllable Music Generation," *arXiv preprint arXiv:2306.05284*, 2023.

[19] Kleijn, W. B., Chinen, M., Lim, F. S. C., and Skoglund, J., "Multi-Channel Audio Signal Generation," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023.

[20] Pulkki, V., "Spatial Sound Reproduction with Directional Audio Coding," *Journal of the Audio Engineering Society*, 55, pp. 503–516, 2007.

[21] Tejeda-Ocampo, C., Hirvonen, T., Souza-Blanes, E., Namazi, M., and Skoglund, J., "Perceptual Evaluation of Mix Presentation for Immersive Audio with IAMF," in *AES 158th Convention*, Audio Engineering Society, 2025.

[22] Martin, R. C., *Agile Software Development: Principles, Patterns, and Practices*, Pearson Education, 2003.

[23] Norman-Haignere, S., Kanwisher, N. G., and McDermott, J. H., "Distinct Cortical Pathways for Music and Speech Revealed by Hypothesis-Free Voxel Decomposition," *Neuron*, 88, pp. 1281 – 1296, 2015.

[24] Piczak, K. J., "ESC: Dataset for Environmental Sound Classification," `https://github.\com/karolpiczak/ESC-10`, 2015.

[25] Riedmiller, J. C. and Tsingos, N., "Recent Advancements in Audio – How a Paradigm Shift in Audio Spatial Representation Delivery Will Change the Future of Consumer Audio Experiences," in *Spring Proceedings*, The Technical Forum, 2015.

[26] Leakey, D. M., "Some Measurements on the Effect of Interchannel Intensity and Time Difference in Two Channel sound Systems," *J. Acoust. Soc. Am.*, 31, p. 977–986, 1959.

[27] Wang, L., Xu, W., Lan, Y., Hu, Z., Lan, Y., Lee, R. K.-W., and Lim, E.-P., "Plan-and-solve Prompting: Improving Zero-shot Chain-of-thought Reasoning by Large Language Models," *arXiv preprint arXiv:2305.04091*, 2023.

[28] ITU-R, "Advanced Sound System for Programme Production," Recommendation BS.2051-3, International Telecommunication Union, Geneva, 2022.

[29] ITU-R BS.1116-3, "Methods for the Subjective Assessment of Small Impairments in Audio Systems," Standard, International Telecommunication Union, Geneva, CH, 2015.

[30] ITU-R BS.1534-3, "Method for the Subjective Assessment of Intermediate Quality Level of Audio Systems," Standard, International Telecommunication Union, Geneva, CH, 2015.