SAGA: Script-to-Audio-Generation-Agent

Tanish Sahu, Gautham Das, Shubham Sharma,
Nishant Singh, Mudit Batra, Shrikant Venkataramani, Vasu Sharma

Pocket FM
shubham.s@pocketfm.com

***Abstract*—Producing high-quality audio dramas requires herculean efforts and hours of intensive labor across analysis, generation, alignment, and mixing. This makes the process tedious for professional sound engineers and prohibitively difficult for novice creators. While a few recent automated approaches address high-level planning and speech generation, the low-level layering, placement, and mixing of background music (BGM) and sound effects (SFX) remain largely unaddressed within such end-to-end pipelines. To bridge this gap, we present SAGA (Script-to-Audio-Generation-Agent), a novel three-stage agentic framework that automates an expert sound engineer's workflow end to end, from script understanding to a final mastered episode. We show that SAGA matches the quality of a human-in-the-loop workflow in objective evaluation scores while reducing production time from several hours to a few minutes, making it an effective tool for both expert and novice creators.**

***Index Terms*—Audio drama, Audio mastering, Sound Design, Text-to-Speech**

I. Introduction

Producing a compelling audio drama requires a meticulous fine-tuning and orchestration of multiple audio layers: (i) dialogues and character interactions, (ii) background music (BGM) to set the overall emotional tone and enhance the key moments, (iii) ambient sounds to establish the spatial environments, and (iv) sound-effect tracks to support the happenings in the scene. With recent advances in large language models (LLMs) and generative audio models, each of these tracks can now be synthesized from text based prompts. For example, we can use LLMs for structural planning [1], [2], Text-to-Speech (TTS) models for synthesizing dialogue and speaker interactions [3], [4], and text-to-audio models for generating a suitable BGM and sound effects (SFX) tracks [5], [6]. In the traditional workflow that involves a human in the loop, a sound engineer is thereafter responsible for bridging the underlying gaps. The text-prompts often need iterative refinements and finetuning to get the desired acoustic outputs from the models. Then, these individual tracks need to be manually aligned, layered and mixed in a multi-track session, at the right volume levels and necessary cross-fades to produce the final mix. This manual alignment and mixing approach can be extremely tedious and time-consuming even for expert sound engineers. For novice creators, without the underlying domain knowledge of sound engineering, this process can be prohibitively difficult and inaccessible.

There have been a few recent attempts [7], [8] to make the creation process more accessible and scalable using LLM agents. The closest approach to our work, AuDirector [9], produces long-form audiobooks using a critic agent that listens and regenerates the wrong portions. However, most of these approaches have largely focused on the overall planning and content direction parts of the pipeline. The mixing and mastering stages themselves have been studied only in isolation [10], [11], not as part of an end-to-end audio-drama system. We approach the problem in a much more holistic way and try to recreate the individual steps of an expert sound-engineer in an automated agentic way to deliver the final mastered audio.

We approach the problem using the following 3-stage pipeline:

Lane A: Multi-Agent Script Analysis
In the first lane, we setup an agentic creative director for the audio drama. Given the script, we use a combination of LLMs to analyze the script and plan out the entire show. This planning involves multiple underlying steps like (i) finding suitable voice styles and descriptions for the characters, (ii) planning speaker turns and assigning emotional tags, (iii) understanding narrative beats and scene intensity curves, and (iv) designing the nature, styles, and placements of background music and sound effects. It's important to note that there is not audio level timing information yet at this stage.

Lane B: Generating and Placing Audio Layers
In the second lane, we use the upgraded script-specific extractions from the first lane to synthesize the underlying audio layers. We pass character and voice-design specifications to a TTS model [12] to generate emotive dialogues and the corresponding speaker turns in case of multi-character scenes. Similar prompts and other corresponding generative text-to-audio models are used to generate the BGM and SFX tracks [13]. Now that we have the audios and the aligned text available, we also establish a global clock using anchor words generated in lane 1. These anchor words and their corresponding timestamps ensure that the SFX and BGM sounds are placed at the right locations in the scenes.

Lane C: Layering, Mixing and Mastering
In the final lane, we mix these individual audio tracks. We use the scene intensities to get suitable the volume levels for the BGM and Ambience tracks thereby fitting these sounds appropriately to the script's narrative. The individual layers are also processed and loudness normalized [14], [15] before being combined to produce a professionally mastered final audio output.

Audio demos are available at: Pocket FM SAGA

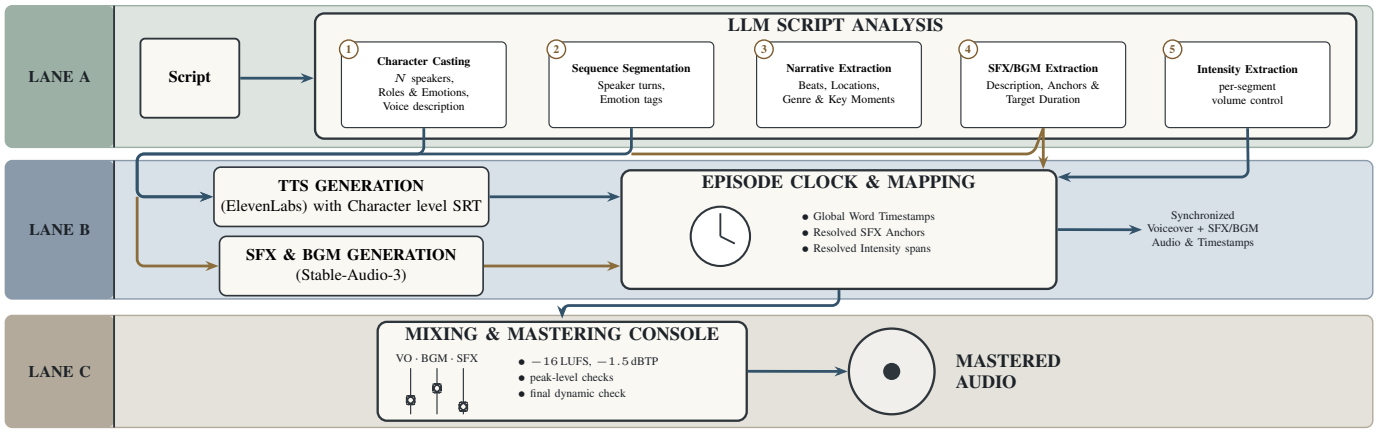

Fig. 1. Lane A generates script annotations via LLM; Lane B renders voices and sound beds into the master clock; Lane C mixes/masters the output.

The key contributions of this work are as follows:

- **A Holistic, Automated End-to-End Production Pipeline:** We present a novel, agentic three-lane structure to automate the entire production pipeline for producing high-quality audio dramas. Our system replicates and handles the comprehensive workflow of a professional sound engineer to generate the complete mastered file only from the script.
- **Empirical Validation Against Professional Human Mixing:** Through our evaluations, we demonstrate that our approach delivers compelling audio dramas more consistently than the traditional human-in-the-loop approach while delivering the same quality across multiple aesthetic evaluation metrics.
- **Democratization and Turnaround Optimization:** Our evaluations also show that our framework drastically reduces the end-to-end production time from hours to mere minutes of automated execution. The text-prompt based pipeline removes the need for deep domain expertise and makes this automated approach a viable and accessible alternative for both experts and novice creators.

II. Algorithm Details

The system is organized into three sequential lanes that transform a script into a mastered audio episode as shown in Fig. 1. The first lane operates as a creative-direction pipeline composed of multiple specialized LLM passes. A character casting pass identifies all speaking characters and assigns each a voice specification, and a segmentation pass divides the script into speaker-level turns while embedding performance-oriented emotion tags to guide the speech model's emotional rendering. The remaining passes carry the heavier analytical load, producing the design of the show's non-speech soundscape: they extract the narrative structure, including story beats, locations, and key moments; construct a word-anchored intensity curve on a 0–10 scale that later governs both music design and the final mix; define the overarching creative direction, including directorial intent, genre balance, hook placement, and climax positioning; and specify the planned background-music and sound-effect events together with their intended style, function, and target span.

The second lane converts this script-level plan into a synchronized episode timeline. Dialogue is rendered segment by segment using the assigned voice profiles for each character. To prevent word loss or unwanted silence when joining independently generated TTS clips, temporary spoken boundary markers are added around each segment so that any clipping or edge silence affects only the markers. These markers are then removed using silence-aware trimming. Because TTS timing metadata can drift, each segment is placed using a cursor that advances according to the actual rendered duration of the clip. This produces a stable timeline and converts the word anchors from the first lane into real timestamps. Once the true scene durations are known, the system produces a global timestamp map that fixes the onset of each voice segment, sound effect event, music cue, and ambience layer on this shared clock; each asset is then processed according to its own function — a sound effect punctuated at its anchored instant, a music cue swelling across its span, an ambience bed sustained beneath the scene, and time aligned against the others, so that every layer fulfills its individual role, yet the layers remain mutually coherent as a single mix. This lane therefore provides the synchronized timing specification required for the final mixing and mastering stage.

The final lane performs audio processing, layering, mixing, and mastering. Each layer is processed independently before the final mix. The voice chain applies click repair, noise and synthetic-reverb removal, de-essing, per-character adaptive EQ and loudness normalization. Generated music beds, ambience, and sound effects are normalized within category-specific loudness ranges, peak-limited, and given a subtle high-frequency lift to reduce the dullness often present in generative audio. Music loudness is shaped according to the story by converting the script-based intensity spans into a continuous BGM target curve Fig. 2, with controlled adjustment relative to voice energy near scripted peaks so that major narrative moments remain impactful even when the narration is delivered evenly. The final mix is then tonally matched to a professionally

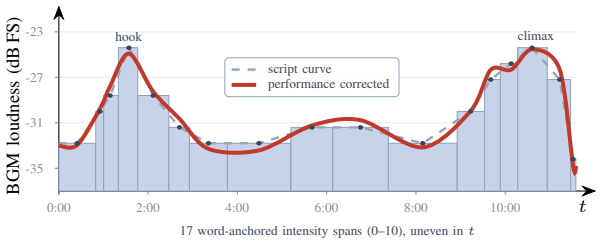

Fig. 2. An example BGM loudness envelope over 17 intensity segments. Each segment gets an intensity score in Lane A that is interpolated and used to transition the BGM track across the whole audio. Depending on the loudness of the dialogue and performance, the loudness is corrected to maintain suitable BGM loudness levels.

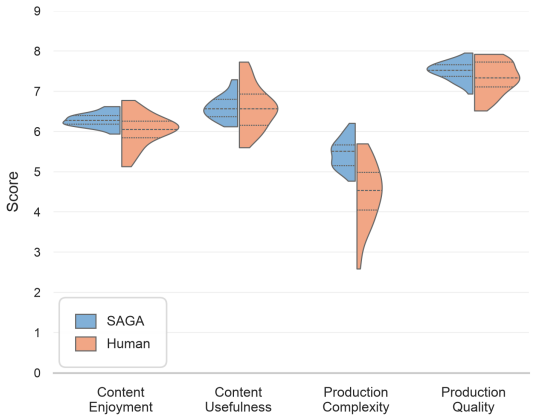

Fig. 3. Objective evaluation using Audiobox-Aesthetics scores (1–10) for SAGA (left) vs. human (right) mastering. The evaluation is split over four axes: Content Enjoyment (CE), Content Usefulness (CU), Production Complexity (PC), and Production Quality (PQ). Each side aggregates 35 episodes. The evaluated audios range in duration from 3 mins to 17 mins with an average duration of 8 mins. The dashed lines in the violins show the median, 25-percentile and 75-percentile points of the scores respectively. *Observations:* (i) SAGA matches or exceeds human on every axis (higher on three, level on CU); (ii) the largest gap is Production Complexity ($+0.99$)—the model rates SAGA masters as noticeably more "complex/layered"; (iii) SAGA is far more consistent, its standard deviation $\sim 2\times$ smaller than human on every axis.

mastered reference, normalized to -16 LUFS under a -1.5 dBTP true-peak limiter following EBU R128, quality-checked, and delivered as a 48 kHz, 24-bit stereo master.

III. Evaluation, Results and Discussion

We compared SAGA's fully automatic master with that of a professional sound engineer, scoring both with Meta Audiobox-Aesthetics [16]. Fig. 3 shows that SAGA is competitive with human mastering on every axis and significantly leads on Production Complexity. SAGA is also more consistent in quality across episodes.

We also compared the two workflows on overall and lane-wise production time (Table I). A human team spends roughly 11 hours per episode, selecting sounds from a library, recording voice talent, and mixing by hand. SAGA instead synthesizes every layer on demand and completes the mixing process in about 18 minutes on average, leading to an end-to-end speed-up of nearly $40\times$ over the human-in-the-loop approach. This shifts the bottleneck from audio-engineering expertise to creative intent, thereby enabling a solo writer to take a script all the way to a finished, broadcast-ready episode.

Finally, we ran a subjective study in which professional sound engineers rated both SAGA and human mastered files on seven production-oriented dimensions (Table II). Here expert listeners rate the human mastered higher on every dimension compared to SAGA and the gap is widest on SFX quality. Despite this gap, SAGA still attains a consistent, broadcast-ready and usable level across the same dimensions at a fraction of the time and human effort. Closing this residual gap is a clear direction for future work.

TABLE I
Production time per episode: Human vs. SAGA

Pipeline lane	Human	SAGA	Speedup
1 Script analysis	2 h	5 min	25×
2 Audio layer creation	8 h	12 min	40×
3 Layering, mix & master	1 h	1 min	60×
Total time / episode	**∼11 h**	**∼18 min**	**∼40×**

TABLE II
Subjective ratings by professional sound engineers

Dimension	Human Mastered (mean ± 95% CI)	SAGA (mean ± 95% CI)
Overall Audio Quality	7.48 ± 0.21	5.56 ± 0.24
Music Quality	6.90 ± 0.27	5.38 ± 0.29
Music Mood Mapping	6.66 ± 0.36	5.22 ± 0.32
SFX Quality	6.86 ± 0.32	4.58 ± 0.23
SFX Sync / Timing	7.28 ± 0.24	5.78 ± 0.36
Mix Balance	6.92 ± 0.25	5.40 ± 0.25
Audio Levels	7.24 ± 0.26	6.12 ± 0.26

IV. Conclusion

In this paper, we present a novel three-lane approach to entirely automate the end-to-end production of audio dramas directly from its text script by tracking and automating the steps of a human-in-the-loop approach. Lane A first uses a multi-agent framework to analyze the script, extract character specifications, understand narrative beats, and generate scene intensity curves. Lane B uses this creative plan to generate dialogue, BGM, SFX and other layered audio tracks and also aligns them with anchor words to get precise time-stamps relative to the generated speech. Finally, Lane C mixes these tracks based on scene intensity to deliver the final mastered audio file. While subjective evaluations show that human mixing still holds the edge, our approach establishes a good baseline while significantly slashing end-to-end production time from hours to minutes.

REFERENCES

[1] Google DeepMind, "Gemini 3.5 Flash Model Card," May 2026, accessed: 2026-07-09. [Online]. Available: https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-5-Flash-Model-Card.pdf

[2] Anthropic, "Introducing Claude Opus 4.8," 2026, accessed: 2026-07-09. [Online]. Available: https://www.anthropic.com/news/claude-opus-4-8

[3] Y. Chen, Z. Niu, Z. Ma, K. Deng, C. Wang, J. JianZhao, K. Yu, and X. Chen, "F5-tts: A fairytaler that fakes fluent and faithful speech with flow matching," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025, pp. 6255–6271.

[4] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li *et al.*, "Neural codec language models are zero-shot text to speech synthesizers," *arXiv preprint arXiv:2301.02111*, 2023.

[5] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation (musicgen)," *Advances in Neural Information Processing Systems (NeurIPS 2023)*, 2023.

[6] F. Kreuk, G. Synnaeve, A. Polyak, U. Singer, A. Défossez, J. Copet, D. Parikh, Y. Taigman, and Y. Adi, "Audiogen: Textually guided audio generation," *arXiv preprint arXiv:2209.15352*, 2022.

[7] X. Liu, Z. Zhu, H. Liu, Y. Yuan, Q. Huang, M. Cui, J. Liang, Y. Cao, Q. Kong, M. D. Plumbley *et al.*, "Wavjourney: Compositional audio creation with large language models," *IEEE Transactions on Audio, Speech and Language Processing*, 2025.

[8] Y. Xiao, L. He, H. Guo, F.-L. Xie, and T. Lee, "Podagent: A comprehensive framework for podcast generation," in *Findings of the Association for Computational Linguistics: ACL 2025*, 2025, pp. 23 923–23 937.

[9] Y. Ren, X. Xu, Z. Zhang, W. Wu, B. Li, and C. Zhang, "Audirector: A self-reflective closed-loop framework for immersive audio storytelling," *arXiv preprint arXiv:2605.11866*, 2026.

[10] C. J. Steinmetz, J. Pons, S. Pascual, and J. Serrà, "Automatic multitrack mixing with a differentiable mixing console of neural audio effects," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 71–75.

[11] M. A. Martínez-Ramírez, W.-H. Liao, G. Fabbro, S. Uhlich, C. Nagashima, and Y. Mitsufuji, "Automatic music mixing with deep learning and out-of-domain data," *arXiv preprint arXiv:2208.11428*, 2022.

[12] ElevenLabs, "Eleven v3: Expressive text to speech," https://elevenlabs.io/docs, 2025, [Online].

[13] Z. Evans, J. D. Parker, M. Rice, C. Carr, Z. Zukowski, J. Taylor, and J. Pons, "Stable audio 3," 2026. [Online]. Available: https://arxiv.org/abs/2605.17991

[14] R. EBU-Recommendation, "Loudness normalisation and permitted maximum level of audio signals," *Eur. Broadcast. Union*, 2011.

[15] C. J. Steinmetz and J. D. Reiss, "pyloudnorm: A simple yet flexible loudness meter in python," in *150th AES Convention*, 2021.

[16] A. Tjandra, Y.-C. Wu, B. Guo, J. Hoffman, B. Ellis, A. Vyas, B. Shi, S. Chen, M. Le, N. Zacharov *et al.*, "Meta audiobox aesthetics: Unified automatic quality assessment for speech, music, and sound," *arXiv preprint arXiv:2502.05139*, 2025.