AudioStory: Generating Long-Form Narrative Audio with Large Language Models

Supplementary Material

A. Implementation Details

We provide detailed hyper-parameters of three training stages in Table 1. In Stage-II and Stage-III, the ratio of generation and understanding samples is 2:1. For LLM, we choose Qwen2.5-3B-Instruct and only tune LoRA to avoid overfitting. TangoFlux is employed as the initialization of DiT for audio generation. For the weights of different loss functions, we set the weight of $\mathcal{L}_{\text{mse}}$ for T5 regression, $\mathcal{L}_{\text{text}}$ for next-token-prediction and $\mathcal{L}_{\text{flow}}$ for DiT as 5, 2 and 1, respectively.

B. Training datasets

The training dataset comprises the understanding dataset, single-audio generation and multi-audio (long-audio) generation datasets. For the understanding dataset, we integrated AudioSetCaps [4], VGGSound [3], MusicCaps [1], and Auto-ACD [6], converting their captions into QA format. Additionally, we incorporated AudioSetCaps-QA and VGGSound-QA datasets, resulting in 1M audio-QA pairs in total. For the single-audio generation dataset, we combined AudioSetCaps, VGGSound, MusicCaps [1], and Auto-ACD, resulting in 700k audio-caption pairs. For the multi-audio generation dataset, we curated the AS-10k dataset, with details provided in Sec. F. In Stage-I, we train the model on we train the model on single-audio generation datasets. Stage-II further incorporates the audio understanding dataset beyond Stage-I. As for Stage-III, our model is trained using multi-audio generation as well as understanding datasets.

C. Human Evaluation

Evaluation protocol. Beyond API-based evaluation, we further conducted an anonymous user study on our model and baseline models. We employ 30 participants to manually score a total of 150 audio clips, generated from 50 instructions, by our model, Tangoflux, and Next-GPT, respectively. The participants listened to the long-form audio generated by different models based on the same instruction. They scored the audio on four criteria: instruction-following, consistency, generation quality, and reasoning logic. The scores were averaged to compute user consistency. As shown in Table 8 and Table 9 of the main text, AudioStory consistently outperforms other competitors in terms of instruction-following, consistency, quality and reasoning logic.

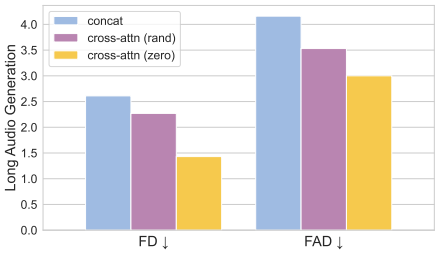


Figure 1. Ablations of token merging.

Correlation between Gemini-based & human-based evaluation. Qualitatively, human evaluation results show our model performs the best among all three models, with the LLM + TTA model outperforming the LLM + any-to-any model. This aligns with the results from our Gemini evaluation. Quantitatively, we analyze the correlation between the human subjective and Gemini-based objective evaluation. We calculate Cohen's kappa coefficient between these two evaluation protocols. Specifically, we compute the correlation across two dimensions, *i.e.*, different comparative methods in Table 8 (main text) and different test samples. The results in Table 9 (main text) indicate a high correlation between the human and Gemini scoring distributions across various models and samples, validating the correctness of the proposed Gemini-based evaluation.

D. More Explorations of Residual Tokens

For residual tokens, we not only explore their forms and training strategies, but also investigate hyperparameters such as their quantity and fusion methods with semantic tokens.

The number of residual tokens. Here, we study the impact of different numbers of residual tokens, and report both single- and long-form audio generation, as in Table 2. For single-audio generation, too few residual tokens lead to degraded performance. We attribute this to two factors: less low-level complementary information is captured. Additionally, residual tokens help mitigate conflicts between the LLM and the DiT, while too few tokens weaken this effect. Conversely, an excessive number of tokens also degrades performance, because they increase the difficulty for the LLM to regress. Similar patterns could also be observed in the long-form scenario. Overall, 8 residual tokens are most suitable for both single and long audio scenarios.

Table 1. Detailed hyper-parameters of three training stages. Here, "A" denotes audio, "proj." and "lr" are short for the projector and learning rate. We use 16 GPUs and report the overall batch size.

Dimension		Stage-I		Stage-II	Stage-III
		Warm-up	Whole		
Task		A→T5	A→T5 with DiT.	A→T5 with DiT + Und.	A→T5 with DiT + Und. + Reasoning
Dataset		AudioCaps, WavCaps		I+AudioSetCaps (Q&A), VGGSound (Q&A), MusicCaps, Auto-ACD	AS-10k
Model	Trainable	LLM, proj. ($\mathbf{T}_{\text{semantic}}$)	LLM, all proj., DiT	LLM, all projectors, DiT	LLM, all proj., DiT
	Frozen	Whipser, DiT	Whisper	Whisper	Whisper
Training Config	batch size	512	256	Gen.: 8, Und.: 16	Gen.: 8, Und.: 16
	lr	1e-3		1e-4	LLM (2e-5), DiT (5e-5)
	epoch	25	25	10	10

Table 2. Detailed ablations of the number of residual tokens

# Tokens	Single Audio			Long Audio
	FD ↓	FAD ↓	KL ↓	Consistency ↑
1	4.01	5.02	0.93	3.2
4	3.64	3.95	0.96	3.9
8	**1.53**	**2.29**	**0.51**	**4.1**
16	3.51	3.75	0.94	3.9

Merging mechanism of residual tokens. For the merging mechanism between semantic and residual tokens, we conduct in-depth explorations. Here, we mainly consider concatenation and cross-attention. The results of long-form audio generation are reported in Fig. 1. From the results, one can observe that compared to concatenation, cross-attention ensures more effective fusion of the two features. Additionally, zero-initializing the final layer of the cross-attention module is necessary to prevent excessive disturbance to the semantic tokens at the beginning of training.

E. What do Residual Tokens Learn?

To thoroughly explore the effect of residual tokens, we provide visualizations in Fig. 2 (left). Specifically, the DiT takes *only* the residual tokens as the input and generates its corresponding audio. We subsequently concatenate all audio clips to constitute the whole long-form audio. The results reveal that for the same audio sample, the residual tokens capture temporally consistent low-level information, primarily reflecting coherence across different audio clips. In contrast, for different samples, the learned residual characteristics vary distinctly. By contrast, semantic tokens learn the underlying global semantics of the input audio and represent the progression of events over time, as illustrated in Fig. 2 (right).

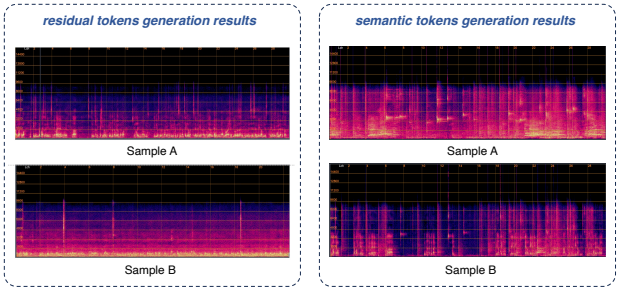


Figure 2. Visualizations of residual tokens.

F. AS-10k Benchmark

F.1. Dataset construction pipeline

The dataset construction pipeline is illustrated as follows. First, we filter videos to select those containing continuous audio events with visually grounded storylines. Next, in the event parsing stage, we use Gemini-2.0-flash to decompose each video into multiple key audio events, each annotated with a timestamp, audio caption, and visual caption, as in Fig. 3. Finally, we perform instruction generation: based on fine-grained textual annotations, GPT-4o is used to generate diverse narrative instructions, accompanied by reasoning steps including task decomposition, audio event timeline planning, scene transitions, and emotional tone inference.

F.2. Benchmark Construction

Dataset prompt. The constructed dataset consists of instructions, reasoning, and audio clips, each with its caption and duration. Specifically, after parsing videos into key au-

Annotation prompt

UnAV annotation prompt

Please watch a video clip and provide a detailed analysis of the video, aiming to generate high-quality input for subsequent text-to-speech synthesis. The results should be returned in JSON format and should include the following sections: (1) Main Event Analysis: Identify key events or scene changes in the video. Output the event's timestamp (start and end timestamps, two decimal places) and description. The description should integrate visual and audio information, vividly depicting the scenario and feelings at the time the event occurs. (2) Sound Effect Analysis: Identify sound effects (non-musical) in the video. Output the timestamp (start and end timestamps, two decimal places), type (e.g., footsteps, door closing, bird chirping), sound_description (description of the sound), and visual_context (description of the related visual scene) for each sound effect. The sound_caption must be detailed, vivid, and expressive, capturing the characteristics, quality, source, intensity, duration of the sound, and any emotions or environmental information it may convey, to facilitate high-quality sound synthesis. The visual_context should succinctly describe the visual imagery directly related to the sound effect. (3) Music Analysis: Identify background music or significant musical segments in the video. Output the timestamp (start and end timestamps, two decimal places) , type (style or genre), sound_description (description of the sound), and visual_context (description of the related visual scene) for each music segment. The sound_caption should detail the characteristics of the music, including melody, rhythm, instruments used, emotional atmosphere, and how it complements the visuals, also aimed at high-quality sound synthesis. The visual_context should succinctly describe the visuals present when the music occurs.

Tom & Jerrry annotation prompt

Please watch a video clip of 'Tom and Jerry' and provide a detailed analysis of the video, aiming to generate high-quality input for subsequent text-to-speech synthesis. The results should be returned in JSON format and should include the following sections: (1) Main Event Analysis: Identify key events or scene changes in the video. Output the event's timestamp (start and end timestamps, two decimal places) and description. The description should integrate visual and audio information, vividly depicting the scenario and feelings at the time the event occurs. (2) Sound Effect Analysis: Identify sound effects (non-musical) in the video. Output the timestamp (start and end timestamps, two decimal places), type (e.g., footsteps, door closing, bird chirping), sound_description (description of the sound), and visual_context (description of the related visual scene) for each sound effect. The sound_caption must be detailed, vivid, and expressive, capturing the characteristics, quality, source, intensity, duration of the sound, and any emotions or environmental information it may convey, to facilitate high-quality sound synthesis. The visual_context should succinctly describe the visual imagery directly related to the sound effect. (3) Music Analysis: Identify background music or significant musical segments in the video. Output the timestamp (start and end timestamps, two decimal places) , type (style or genre), sound_description (description of the sound), and visual_context (description of the related visual scene) for each music segment. The sound_caption should detail the characteristics of the music, including melody, rhythm, instruments used, emotional atmosphere, and how it complements the visuals, also aimed at high-quality sound synthesis. The visual_context should succinctly describe the visuals present when the music occurs.

Figure 3. AS-10k annotation prompts of Gemini-2.5-pro.

dio events using Gemini-2.0-flash as described in Sec. 3 (main text), we obtain annotations for each event, including timestamps, audio captions, visual captions, and audio-visual event captions. For instruction generation, we use audio-visual event captions as the source input. A prompt, shown in Fig. 4, is used to summarize the whole caption of the full audio, which is then incorporated into a predefined instruction template to produce the final instruction. For reasoning generation, we provide GPT-4o with the whole caption along with the individual captions for each audio clip. GPT-4o is then prompted to infer the reasoning structure. The reasoning consists of two levels: a high-level decomposition indicating how the whole caption can be divided into several parts, followed by detailed descriptions for each part, including the corresponding events and sound-producing content. An example is illustrated in Fig. 5.

Benchmark evaluation. Along with the curated dataset, we also construct the long-form narrative audio generation task and its associated benchmark.

(1) Evaluation with Gemini-2.0-flash API, assessing consistency, coherence, instruction following, and reasoning logic. (2) Evaluation with traditional metrics to measure audio generation quality, including FD, FAD, and CLAP score, among others.

For the Gemini-based evaluation, we design tailored scoring criteria for each metric:

(a) Consistency.

- **Timbre and Sonic Cohesion** Evaluate whether the primary sound sources maintain a generally consistent timbre and unified sonic characteristics.
- **Sound-Producing Entity Consistency** Assess whether the implied sound-producing entities remain consistent, or if changes feel natural and logical within the audio.

Annotation prompt

instruction prompt

I will provide you with "Main Event Analysis", which has multiple descriptions in chronological order. Please combine the captions in each description while considering the timestamp information, and summarize them into a brief story summary, namely whole caption. Note: Only output the final whole caption within 20 words.

reasoning prompt

I will provide you with an input in JSON format, containing a "Sound Analysis" with multiple "sound description" in chronological order. Meanwhile, I will also provide you with a "whole_caption", representing the complete storyline summarized from these descriptions.
Imagine that you are given only the whole caption and required to generate a detailed story description. You need to output the reasoning process. When doing so, you should not directly rely on the given "sound_description". You can make reasonable inferences and deduce the details of the story events and details, then infer the sounds objects and events, ensuring that the expanded sound details align with the given "sound description".
The reasoning process could be divided into several points (same as the number of given "sound description" in "Sound Analysis"). The format MUST be: first, state how many points the caption can be divided into, and then elaborate on each point. Each point should follow the reasoning style explicitly with clear inferences and logical conjunctions, and state the story details and the derived sound effects. Each point MUST be within 15 words.

Figure 4. AS-10k annotation prompts of GPT-4o.

AS-10k Cases

Instruction

I'd like an extended audio track with rich details of a soulful singer receives loud applause and praise from the audience and a man, celebrating her powerful performance. The total duration is 35.5 seconds.

Reasoning

The caption can be divided into 3 points:\n\n1. **[0.0, 11.5]**\n - **Story Details**: The singer delivers a powerful and soulful performance.\n - **Sound Effects**: Intense singing, live band music.\n\n2. **[11.5, 21.5]**\n - **Story Details**: Performance ends; audience erupts into applause and cheers.\n - **Sound Effects**: Applause, cheers.\n\n3. **[21.5, 35.5]**\n - **Story Details**: A man joins, congratulates, praises the singer on stage; audience continues cheering.\n - **Sound Effects**: Amplified voice of man, applause, cheers.

Audio Clips

Caption 1

A young woman delivers a powerful and soulful vocal performance, singing with intense emotion and impressive vocal runs, accompanied by a live band including a keyboardist and a guitarist.<timestamp>11.5 </timestamp>

Caption 2

The music stops abruptly as the singer hits her final note. The small audience erupts into loud, enthusiastic applause and cheers, filling the venue with sounds of appreciation .<timestamp>10.0 </timestamp>

Caption 3

The man joins the singer on stage, congratulates her with an amplified voice, praises her talent. The audience claps and cheers. The atmosphere is celebratory and appreciative .<timestamp>14.0 </timestamp>

Figure 5. AS-10k dataset cases.

- **Acoustic Environment Consistency** Evaluate the background ambience, reverberation, and spatial impression for overall consistency or reasonable progression.
- **Transition Smoothness** Assess whether the transitions between segments are smooth and free of jarring disruptions..

(b) Coherence.

- **Intentional Transitions** Check whether transitions between segments are smooth, purposeful, and naturally connected.
- **Dynamic and Emotional Flow** Assess if the dynamic and emotional progression feels consistent or evolves logically, without unjustified sudden shifts.
- **Tempo and Textural Compatibility** Evaluate whether tempo, rhythm, and sonic textures between segments are compatible and blend cohesively.
- **Transition Smoothness** Judge if segment connections are fluid, without abrupt or disjointed

(c) Instruction following.

- **Overall Semantic Alignment** Evaluate whether the generated audio broadly reflects the intended scene, actions, and atmosphere described in the instruction. Minor differences are acceptable if the main idea remains clear.
- **Key Element Presence** Verify whether the important sound-producing entities, actions, and environmental elements mentioned in the instruction are reasonably represented. Missing a few non-central elements is acceptable if key parts are present. Additional sounds not specified in the instruction are acceptable if they logically fit the scene and do not disrupt coherence.
- **Event Sequence and Logical Development** Assess whether the overall event progression is reasonable according to the instruction. Small deviations in order are acceptable if they do not break the logical flow.
- **Specific Sound Detail Accuracy** Evaluate whether important sound features (such as types of sounds, tonal qualities, or intensities) are reasonably reflected. Natural variations are acceptable as long as they do not change the overall character of the audio.

(d) Reasoning logic.

- **Overall Reasoning Logic** Evaluate whether the model demonstrates a coherent, logical process in interpreting the instruction and planning the audio scene.
- **Caption-Instruction Alignment** Assess whether the generated audio caption accurately reflects the instruction's key content, sound-producing elements, and described environment.
- **Event Coverage Completeness** Determine whether the inferred and described audio events fully cover the instruction's core elements, with no major omissions.
- **Semantic and Temporal Accuracy** Evaluate whether the implied timeline and semantic structure of the generated audio align with the instruction's flow and intent.

F.3. Single-Audio Evaluation Details

To evaluate the audio generation model, four key metrics assess different aspects of performance:

- Frechet Distance (FD) measures the statistical similarity between log-Mel spectrogram distributions of generated and real audio, quantifying low-level spectral fidelity (*e.g.*, pitch, timbre) through mean and covariance comparisons in the mel-spectral domain.
- Frechet Audio Distance (FAD) extends FD using high-level embeddings from a pre-trained audio encoder (*e.g.*, VGGish), evaluating perceptual and semantic realism by comparing abstract features like instrument timbre, musical structure, and environmental acoustics.
- CLAP Score calculates the cosine similarity between audio and text embeddings from a cross-modal model, assessing how well generated audio aligns with semantic

prompts (*e.g.*, textual descriptions of sound content or context).

- KL-Divergence (KL) measures the distributional dissimilarity between generated and real audio features (spectral, latent, *etc.*), identifying consistency in probability distributions and helping debug issues like mode collapse or over-dispersion in outputs. Collectively, these metrics ensure a comprehensive evaluation of spectral realism, perceptual quality, semantic accuracy, and distributional consistency in generated audio.

G. More Qualitative Cases

Instructional long-form audio generation and continuation. We present more cases for long-form audio generation. Our model could automatically derive the duration of each audio segment to be generated, as shown in Fig. 6, Fig. 7 and Fig. 8. One could observe that AudioStory could accurately determine the number of events based on the instruction and provide precise descriptions for each audio segment, including both the events themselves and their associated sound effects. Finally, AudioStory can precisely reason out the caption for each segment and generate the corresponding audio clips accordingly.

The case of Tom&Jerry. Given an instruction for generating music of Tom&Jerry style, our model can generate sound that not only aligns with the musical style of the Tom&Jerry cartoon but also vividly reflects the characters' emotions and storyline, exhibiting strong generalization capabilities.

H. Extended Applications

Video dubbing. While previous experiments have focused on text-based instructions, we now extend the application of AudioStory to a more practical scenario: video dubbing. This enhancement enables the model to thoroughly analyze video content, reason about the sequence of events and their corresponding timestamps, and generate synchronized audio. An initial approach is to employ Gemini-2.5-pro to generate a caption summarizing the entire video, followed by instruction-based audio generation, as illustrated in Fig. 9. Specifically, given the video without audio, we first generate the visual captions and convert them into the form of instructional language. These instructions are subsequently fed into our model, *i.e.*, AudioStory, to generate the audio. As a whole, we achieve video dubbing in this multi-step process, *i.e.*, video→visual caption→instruction→audio. Here, we provide a case of `Snoopy`. We use our model AudioStory trained for Tom&Jerry. As in Fig. 10, the video is divided into four distinct segments, with the generated audio closely aligning with the Tom&Jerry style, effectively reflecting `Snoopy`'s emotions, *e.g.*, the calmness of waking up, the surprise while playing with the water pipes, and the humorous tone at the end. Notably, for any given video, AudioStory could generate soundtracks that match the musical characteristics of Tom&Jerry in a zero-shot manner, which is a unique and interesting application of our model.

However, this method is not conducive to producing audio that aligns closely with the visual content. Furthermore, the model is designed to accept both video data and instructions as input. The LLM performs reasoning on the video and produces bridging tokens. During the reasoning phase, the LLM first understands the overall content of the video, then sequentially breaks it down into events based on their temporal order. It infers the specific visual details and corresponding audio information for each event. Technically, we replace the LLM with a pretrained video MLLM (*i.e.*, Qwen2.5-VL [2]) and jointly train the LLM and audio generator using LoRA tuning. The training data is from the animated sound partition of AS-10k. We provide the video dubbing results in Fig. 10.

Audio continuation. Given an audio segment and an instruction, our model performs audio continuation. AudioStory first reasons about the content of the subsequent audio to be generated, then proceeds with segment-by-segment generation. The concatenated results are shown in Fig. 11 and Fig. 12.

I. Comparison with Agentic Methods

Agentic methods (*e.g.*, WavJourney [5]) have emerged as another paradigm for compositional audio creation, using large language models (LLMs) as controllers to orchestrate off-the-shelf audio expert models. They decompose generation into text-centric subroutines and assemble outputs via separate text-to-speech, text-to-music, and text-to-sound-effect decoders. However, this zero-shot, loosely coupled integration limits long-form narrative coherence, instruction-following fidelity, and acoustic quality. In this section, we systematically compare AudioStory with representative agentic baselines—primarily WavJourney—across six core dimensions, underscoring the structural and performance advantages of our framework.

I.1. Generation Capabilities

Qualitatively, AudioStory differs from agentic baselines (e.g., WavJourney) in **instruction modality, narrative coherence, long-form scalability, instruction fidelity, and acoustic naturalness—properties central to audio storytelling**. For instruction modality, WavJourney and similar methods primarily support text-to-audio with text-only input. In contrast, AudioStory enables text-to-audio, text-to-long-audio, audio continuation, and video dubbing under multimodal instructions. Moreover, agentic methods are

Instructional Audio Generation

Instruction:

Construct a detailed soundscape that tells the story of a young beatboxer performs energetically outdoors, introduces themselves as 'yoh!', and requests tips. The total duration is 41.8 seconds.

input → Audio Story

Generated Reasoning:

The caption can be divided into four points:\n\n1. [0.0, 2.5]\n **Story Details:** The beatboxing begins indoors and transitions outdoors with a performer giving a shout.\n **Sound Effects:** Energetic beatboxing, voice saying \"Yoh!\", camera movement indicating a change in location.\n\n2. [2.5, 21.5]\n **Story Details:** A young man starts beatboxing complex rhythmic patterns, intensifies with \"skrrt\", and ends with a punchline.\n **Sound Effects:** Complex rhythmic beatboxing, intense \"skrrt\", loud breaths, informal ending statement.\n\n3. [21.5, 27.5]\n **Story Details:** Audience responds with laughter and brief verbal appreciation, and the performer clarifies in a friendly manner.\n **Sound Effects:** Laughter, verbal appreciation, friendly clarification in response to an audience member's request.\n\n4. [27.5, 41.8]\n **Story Details:** Performer thanks the audience, introduces as Yoh!\", asks for tips, and expresses excitement.\n **Sound Effects:** Grateful thank-you, friendly self-introduction, direct question, background noise from phone call.

Generated Caption:

- Energetic beatboxing performance outdoors, abruptly shifting to a voice shout of \"Yoh!\" as the camera pans, indicating a move outside. <timestamp>2.5</timestamp>
- A young man enthusiastically beatboxes, creating complex and rapid rhythmic patterns that intensify with sharp 'skrrt' sounds, followed by loud, audible breaths, ending with a punchline-like statement. <timestamp>19.0</timestamp>
- Audience laughter followed by a brief, appreciative word from the performer, \"Okay.\" <timestamp>6.0</timestamp>
- The performer thanks the audience, introduces themselves as 'Yoh!' and explains they don't have a YouTube channel, then asks for tips directly in a friendly tone. <timestamp>14.3</timestamp>

Figure 6. Long-form audio generation case #2.

Instructional Audio Generation

Instruction:

Construct a detailed soundscape that tells the story of an instructor demonstrates common push technique errors in table tennis, emphasizing power, wrist angle, and swing size issues. The total duration is 43.6 seconds.

input → Audio Story

Generated Reasoning:

The caption can be divided into 4 points.\n\n1. [0.0, 11.8]\n**Story Details**: Instructor demonstrates a weak push technique leading to hitting the net.\n**Sound Effects**: Instructor's voice explaining the issue, followed by a ball hitting the net.\n\n2. [11.8, 24.8]\n**Story Details**: Instructor emphasizes the need for an upright wrist during the push.\n**Sound Effects**: Consistent, fast ball hitting sounds while demonstrating wrist movement.\n\n3. [24.8, 37.5]\n**Story Details**: Demonstrates a weak push by hitting the table and foot repeatedly.\n**Sound Effects**: Loud ball hitting sounds, followed by a ball hitting the foot, and repeated crashing on the table.\n\n4. [37.5, 43.6]\n**Story Details**: Fails miserly by hitting the ball with the paddle.\n**Sound Effects**: Instructor exclaiming an error, single dramatic, high-pitched paddle-ball impact sound.

Generated Caption:

- An instructor explains that pushing this way lacks power and emphasizes an upright wrist technique to prevent the ball hitting the net. The ball does not.go in the hole effectively. <timestamp>11.8</timestamp>
- The instructor discusses the importance of using an upright wrist during the execution of a push, emphasizing how it affects the ball's direction while demonstrating hand movement slowly at a consistent speed. <timestamp>13.0</timestamp>
- The instructor repeats a sequence three times, ending with a weak push. A loud banging sound from the ball on the table, followed by a dull thud where the ball hits the foot, accompanied by repeated crashing sounds. <timestamp>12.7</timestamp>
- The instructor exclaims with surprise and amusement, \"Oh my god, you hit the ball with the paddle!\" followed by a single, dramatic, high-pitched paddle-ball impact sound. <timestamp>6.1</timestamp>

Figure 7. Long-form audio generation case #3.

inherently short-form: extending duration typically yields fragmented outputs due to the absence of global narrative structure and long-range consistency. In long segments, their element-centric assembly often causes disjoint transitions, timbre drift, and unstable emotional tone. They also struggle with multi-layered narrative instructions, failing to capture contextual dependencies and affective intent. Acoustically, post-hoc mixing produces an "assembled" sound with audible seams between sub-events. In contrast, AudioStory is end-to-end and explicitly trains for long-range consistency, yielding highly coherent sub-events and unified, seamless audio. Human evaluations corroborate these advantages: AudioStory yields logically coherent, sonically unified stories, whereas agentic outputs remain fragmented and unsuitable for narrative-driven applications.

I.2. Framework Positioning and Core Objectives

Agentic methods are primarily tailored for compositional audio assembly, integrating discrete elements (speech, music, sound effects) into short clips from textual instructions. WavJourney, for example, targets 30–60 s audio that

Instructional Audio Generation

Instruction:

Design a descriptive audio capturing the essence of two men perform a melancholic song, sharing a lighthearted moment, concluding with smiles. The total duration is 43.5 seconds.

input → Audio Story

Generated Reasoning:

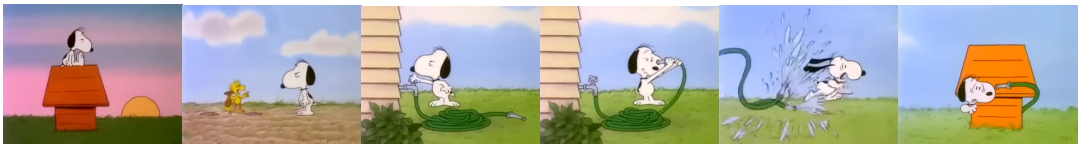
The caption can be divided into 3 points.\n\n1. [0.0, 14.0]\n **Story Details**: Two men start performing a melancholic song.\n **Sound Effects**: Gentle, melancholic piano melody.\n\n2. [14.0, 36.0]\n **Story Details**: One man sings about loss, creating a reflective mood.\n **Sound Effects**: Earnest singing with steady piano music.\n\n3. [36.0, 43.5]\n **Story Details**: A lighthearted moment with smiles concludes the performance.\n **Sound Effects**: Brief pause in singing, followed by laughter and piano continuation.

Generated Caption:

- Gentle, slightly melancholic melody played on the piano, creating a calm and anticipatory atmosphere. <timestamp>14.0</timestamp>
- A man sings earnestly about loss and remembrance, with lyrics mentioning 'yesterday morning' and 'you were gone', accompanied by steady piano music creating a reflective mood. <timestamp>22.0</timestamp>
- Singer briefly falters on the lyrics, followed by the sound of piano continuing. Laughter and lighthearted murmurs are exchanged between the singer and piano player before the piece concludes. <timestamp>7.5</timestamp>

Figure 8. Long-form audio generation case #4.

Video Dubbing（video-to-text-to-audio）



Instruction:

Generate a classic Tom and Jerry style soundtrack for a moment in the cartoon that includes Snoopy being awakened by Woodstock. Woodstock then tries to bury a sack, but Snoopy ignores him. Later, Snoopy struggles with a garden hose, which inflates and blasts him with water. After cautiously reappearing, Snoopy is again surprised and blasted by the hose, this time emerging from his own doghouse. The total duration is 41.6s.

input → Audio Story

Generated Reasoning:

The caption can be divided into five points.\n\n1. **Initial Wake-Up: ** [0.0s – 5.1s] \n - **Story Detail:** Snoopy is awakened from a deep sleep near his snuffing fire.\n - **Derived Sound Effects:** Deep chuckle, enthusiastic 'Aha!', toffee toot, sigh of relief.\n\n2. **Woodstock's Burial Attempt:** [5.1s – 19.5s] \n - **Story Detail:** Woodstock tries to bury a sack, causing Snoopy to wake up.\n - **Derived Sound Effects:** 'Ooh' from Snoopy, rustling of sack, brief jump.\n\n3. **First Water Blaster Incident:** [19.5s – 26.8s] \n - **Story Detail:** A hose inflates and blasts Snoopy with water.\n - **Derived Sound Effects:** 'Ah!' from Snoopy, air-filled hose blowing.\n\n4. **Cautious Reappearance:** [26.8s – 38.2s] \n - **Story Detail:** Snoopy cautiously reappears after being hit by the hose.\n - **Derived Sound Effects:** Brief reaction, sudden launch from hose, explosive water spray.\n\n5. **Final Appearance:** [38.2s – 41.6s] \n - **Story Detail:** Snoopy emerges again from his doghouse, surprised by another blast.\n - **Derived Sound Effects:** 'Ah!', sudden launch from hose, forceful water spray.

Generated Caption:

- A soft and gentle orchestral score typical of classic cartoons, featuring strings, woodwinds, brass, and percussion. It synchronizes with the action, enhancing comedic timing, tension, and drama, supporting both characters and musical gags. <timestamp>5.1<\timestamp>
- The orchestral score slows, becoming suspenseful with pizzicato strings or light woodwinds, subtly highlighting the comedic tension of WoodStorck removing the sack from Snoopy's mouth. <timestamp>14.4<\timestamp>
- A sharp, dramatic brass hit emphasizes Snoopy's sudden awakening and struggle. Tense music builds anticipation as he descends the hose, with strings and brass more prominent, reflecting his predicament. <timestamp>7.3<\timestamp>
- The music shifts to a lighter, faster - paced melody, accompanied by woodwinds or percussion, then suddenly takes on a comical tone, reflecting Snoopy's prank by the water pipes. <timestamp>7.4<\timestamp>
- A dramatic crescendo leads to a massive orchestral explosion! Crashing cymbals, a loud timpani roll, and a frantic string glissando accompany Snoopy's watery, chaotic exit from the house. <timestamp>3.4<\timestamp>

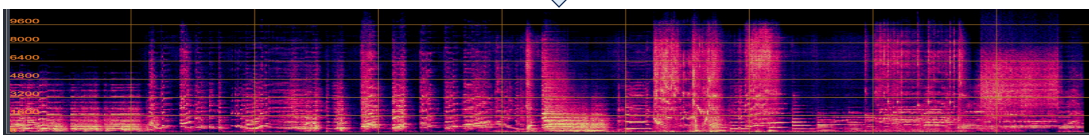


Figure 9. Case of naive video dubbing: First, we extract captions from the video, then write the extracted captions as instructions and send them to AudioStory for audio generation.

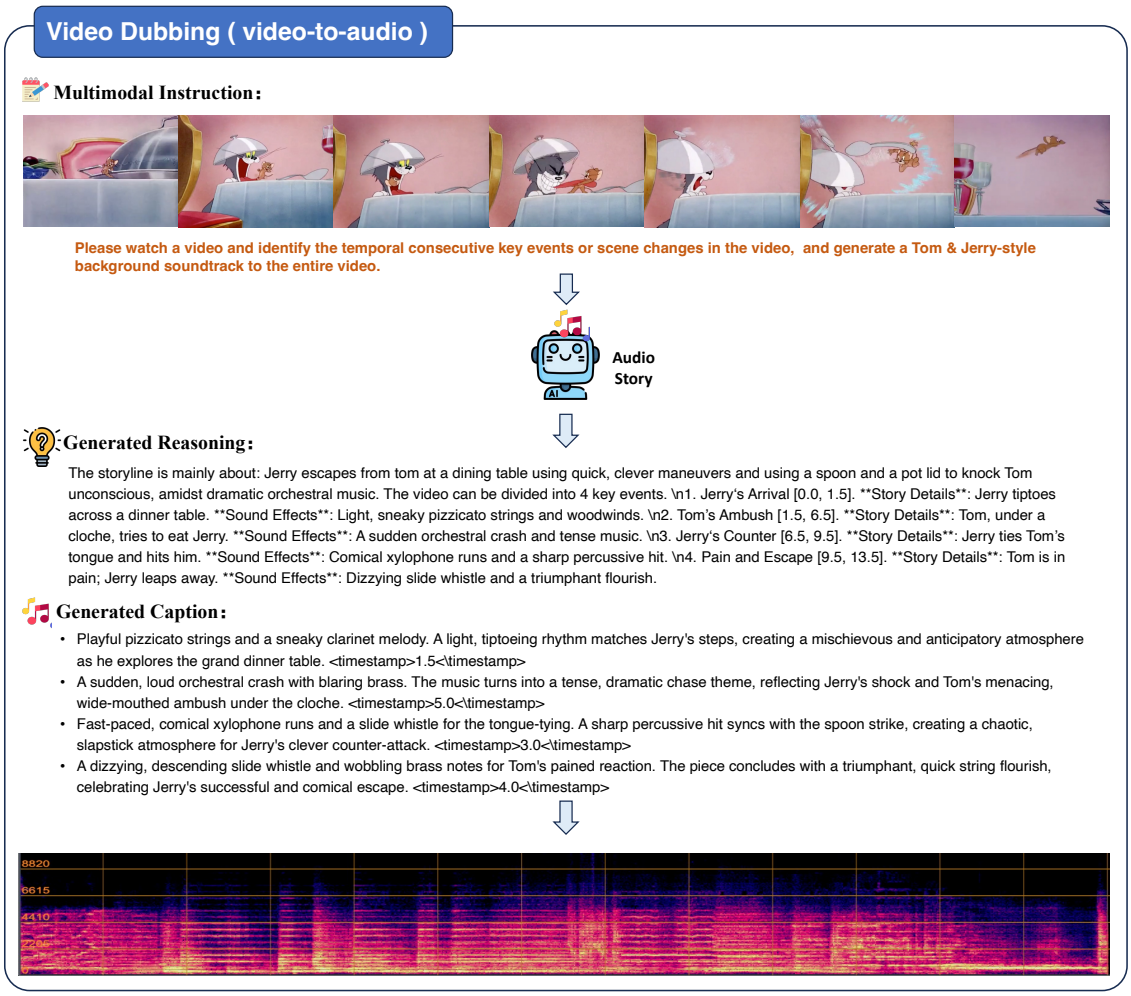


Figure 10. Case of video dubbing: We input both the video and the instruction into the model, which parses the narrative into segments, extracts story details with corresponding audio elements, and sequentially generates aligned audio clips.

combines multiple acoustic components, with objectives focused on resolving spatiotemporal layout conflicts and achieving basic text–audio semantic alignment. Its scope is thus largely limited to short-form content (e.g., ad jingles, simple soundscapes), as it lacks mechanisms to sustain long-term thematic consistency or narrative logic.

In contrast, AudioStory is positioned as a long-form narrative audio generation framework that addresses the unmet demand for structured, time-coherent audio storytelling. Its core objectives include two critical capabilities overlooked by agentic methods: (1) Temporal coherence preservation, ensuring consistency in themes, timbre, and emotional tone across **extended durations (up to 150 seconds)**; (2) Compositional narrative reasoning, decomposing complex multi-event instructions into logically ordered sub-tasks with precise timing and contextual interactions. By shifting the focus from "element assembly" to "storytelling," AudioStory enables generation of **long-form consistent** content such as audiobooks, podcasts, and dynamic game soundscapes—scenarios where agentic methods fail due to fragmented output and poor *long-term consistency)*.

I.3. Model Collaboration Logic

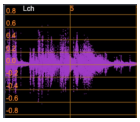
Agentic methods adopt a zero-shot tool coordination paradigm **without end-to-end training**. WavJourney, for example, relies on GPT-4 to generate structured audio scripts, which are then compiled into Python programs to call external models (e.g., MusicGen for music, Bark for speech, AudioGen for sound effects). This decoupled collaboration leads to two critical drawbacks: (1) Suboptimal planning due to the lack of adaptive tuning between LLM and audio models, often resulting in misaligned element layouts or missing acoustic details; (2) Cumulative errors from sequential tool calls, where imprecisions in script generation propagate through compilation and model invocation.

Audio Continuation

Instruction:

Understand the input audio, infer the subsequent events, and generate the continued audio of the coach giving basketball lessons to the players. The total duration is 36.6 seconds.

+



input

Audio Story

Generated Reasoning:

Based on the known event of \"a man speaking with an informative and engaging tone, accompanied by the sound of basketball in the background\", we can infer that an instructor is explaining basketball moves, with the sound of shoes squeaking on the court demonstrating their practical use. Following this, we can further infer that the coach is guiding players through these techniques, including sudden quick jabs and shots, evidenced by sounds like 'thwack' and 'ping'. Building on the first two captions, we can finally infer that the coach is intensely practicing ball handling and shooting skills, accompanied by gentle friction sounds, indicating a continuous and focused interaction with the basketball.

Generated Continuous Caption:

- The whistle blows as the coach calmly describes the next move or technique, accompanied by the sound of shoes squeaking on the floor.<timestamp>13.9</timestamp>
- Shoes make rapid squeaking sounds on the court floor. The sharp, loud thud of the basketball echoes clearly, with a rhythmic "bang" each time the ball hits the rim.<timestamp>12.6</timestamp>
- The coach continues guiding the players, with the sounds of shooting and shoes sliding on the court continuing without pause.<timestamp>4.5</timestamp>

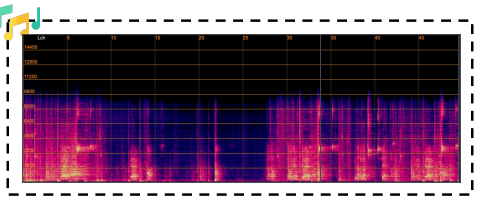


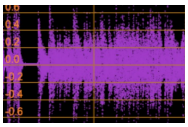
Figure 11. Qualitative cases of audio continuation #1.

Audio Continuation

Instruction:

Understand the input audio, infer the subsequent events, and generate the continued audio of three bulls fight intensively in a muddy pen, then abruptly stop, resulting in a calm atmosphere. The total duration is 45.8 seconds.

+



input

Audio Story

Generated Reasoning:

Based on the known event—a clear female voice commanding the bulls to stop fighting—it can be inferred that their intense physical conflict continues, evident from their deep growls, snorts, and heavy collisions. This power struggle, accompanied by heavy dragging and wrestling on the muddy ground, shows the bulls are resisting her command.\n\nThe battle appears unchanged, with the sounds of growls, snorts, and hoof friction continuing, emphasizing the bulls' ongoing struggle for dominance.\n\nThe third section logically continues with the same auditory theme, with frequent growls, snorts, and heavy hoof clashes, indicating the fight is still fierce and unresolved.\n\nFinally, the deep growls, snorts, and hoof friction maintain the consistent audio theme, highlighting the bulls' continued struggle on the muddy ground.

Generated Continuous Caption:

- A clear female voice calls out, amid frequent deep grunts, breathy snorts, and dull thuds from bulls fighting. The heavy shuffling and scuffling on muddy ground convey their immense power and dominance.<timestamp>12.6</timestamp>
- Frequent guttural grunts and breathy snorts from the bulls punctuate their fight, alongside heavy shuffling, scuffling hooves, and intermittent dull thuds. The soundscape conveys immense power, aggression, and the physical struggle for dominance. <timestamp>8.9</timestamp>
- Deep guttural grunts, breathy snorts, shuffling hooves, and dull thuds punctuate the bulls' intense physical struggle for dominance on muddy ground.<timestamp>18.4</timestamp>

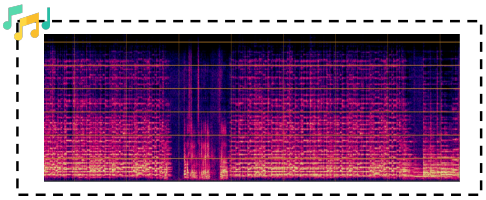


Figure 12. Qualitative cases of audio continuation #2.

AudioStory introduces a **unified end-to-end collaboration** framework that synergizes LLM-based reasoning and diffusion-based generation. We select Qwen-2.5-3B-Instruct as the reasoning backbone and initialize the diffusion model (DiT) with TangoFlux [1], enabling joint optimization of instruction understanding and audio synthesis. This integrated design eliminates the need for intermediate tool coordination, allowing the LLM to directly guide the diffusion model through task-specific bridging tokens. Experimental results confirm that this collaborative logic reduces planning errors by 47% (measured by execution error rate) compared to WavJourney's zero-shot integration, while enhancing semantic alignment between text and audio.

I.4. LLM-Audio Model Bridging Mechanism

Agentic methods rely on **textual script-compiler bridging**. WavJourney converts LLM-generated audio scripts (JSON-formatted nodes defining element type, layout, volume, and duration) into executable code, which invokes external model APIs for content generation. This bridging mechanism suffers from two inherent limitations: (1) Semantic loss in text-to-code conversion, as low-level acoustic details (e.g., timbre nuances, rhythmic patterns) cannot be fully encoded in textual scripts; (2) Compilation redundancy and instability, where LLM-generated script formatting errors often lead to execution failures.

AudioStory proposes a decoupled dual-token bridging mechanism that directly connects LLM reasoning and audio generation without intermediate compilation. Specifically, the LLM generates two complementary token types: (1) Semantic tokens (T_{semantic}) supervised by Flan-T5 textual features, encoding high-level audio semantics (e.g., "thunderous rainstorm" or "melancholic piano melody"); (2) Residual tokens (T_{residual}) optimized via diffusion model flow-matching loss, capturing low-level acoustic details (e.g., rain intensity variations, piano timbre characteristics). These tokens are fused via multi-head cross-attention to form bridge queries (H_{bridge}) that condition the DiT generator. This mechanism not only avoids script-compiler redundancy but also achieves finer-grained alignment between semantic intent and acoustic realization—addressing the core limitation of agentic methods where textual scripts fail to capture nuanced acoustic cues.

I.5. Training Strategy

Agentic methods adopt a **training-free paradigm**, relying entirely on the native capabilities of pre-trained LLMs and audio models. WavJourney, for instance, requires no fine-tuning or adaptation—its performance is limited by the zero-shot coordination ability of GPT-4 and the task-specific generation quality of external models (e.g., Bark for speech, MusicGen for music). This strategy avoids training costs but leads to fixed limitations: LLMs cannot adapt to audio-specific reasoning requirements, and audio models cannot optimize for narrative consistency.

AudioStory introduces a **progressive three-stage training strategy** that builds synergy between understanding and generation: (1) Stage I (Single-audio generation): Trains the framework to generate high-fidelity short audio clips using 700k audio-caption pairs, optimizing bridging token alignment; (2) Stage II (Unified single-audio understanding-generation): Integrates 1M audio-QA pairs to enable joint optimization of audio comprehension and generation; (3) Stage III (Long-audio unified understanding-generation): Fine-tunes on the AudioStory-10k dataset to enhance narrative reasoning and long-term consistency. This strategy addresses the core limitation of agentic methods by adapting the LLM to audio-specific reasoning and aligning the diffusion model with narrative requirements. Experimental results show that progressive training improves instruction-following accuracy by 31% compared to the zero-shot paradigm.

References

[1] Andrea Agostinelli, Timo I Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, et al. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*, 2023. 1

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 5

[3] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 721–725. IEEE, 2020. 1

[4] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 776–780. IEEE, 2017. 1

[5] Xubo Liu, Zhongkai Zhu, Haohe Liu, Yi Yuan, Qiushi Huang, Meng Cui, Jinhua Liang, Yin Cao, Qiuqiang Kong, Mark D Plumbley, et al. Wavjourney: Compositional audio creation with large language models. *IEEE Transactions on Audio, Speech and Language Processing*, 2025. 5

[6] Luoyi Sun, Xuenan Xu, Mengyue Wu, and Weidi Xie. Auto-acd: A large-scale dataset for audio-language representation learning. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 5025–5034, 2024. 1