Unified Synthesis of Compositional Speech and Sound from Free-Form Text Prompts

Yuyue Wang[1*] **Xihua Wang**[1] **Xin Cheng**[1] **Yijing Chen**[1] **Ruihua Song**[1†]
[1] Renmin University of China

Abstract

Generating unified audio that naturally integrates speech and sounds from free-form text prompts remains challenging. For example, the prompt "A fast drum roll ends with a cymbal crash, after which a man shouts: 'Come on!'." describes several events that should appear in order and form a coherent recording rather than independent tracks that lack coherent temporal and semantic interactions. Current methods broadly follow two paradigms: pipeline methods couple separate text-to-sound and text-to-speech models, while unified models rely on a single generator that still requires structured, multi-field prompt inputs. Both paradigms require upstream prompt decomposition or rewriting, limiting free-form flexibility and risking cascading errors. To enable direct generation from free-form prompts within a single model, we introduce PlanAudio, an end-to-end framework that maps unconstrained free-form text to unified audio without intermediate prompt preprocessing. To address the challenges of direct free-form generation, PlanAudio integrates latent planning into a single autoregressive framework, predicting continuous audio layout before acoustic token synthesis. We construct a large-scale training corpus and introduce Free-Form Unified Audio Generation Benchmark (FUAG-Bench) for evaluation, both spanning speech, sound, and joint speech-and-sound generation. Experiments show that a single PlanAudio outperforms existing pipeline and unified baselines on speech-and-sound generation, while remaining competitive with specialist models on speech and sound generation. Further analysis validates the effectiveness of audio layout planning over textual planning, acoustic planning, and direct generation. Sample Available: `https://planaudio.github.io/PlanAudio2026/`.

1 Introduction

In real-world scenarios, speech and environmental sounds often coexist and interact within the same audio scene. While recent Text-to-Speech (TTS) [1–3] and Text-to-Sound (T2S) [4–6] models excel in their respective domains, they are often limited to their target task and struggle with open-domain scenarios where speech and sounds are naturally described and composed together.

For instance, a free-form prompt such as "A fast drum roll ends with a cymbal crash, after which a man shouts: 'Come on!' " describes multiple audio events that must be synthesized as a coherent recording with accurate temporal relationships, rather than generated independently and simply mixed together. This calls for a unified generation framework that can directly synthesize speech, sound effects, and their combinations from free-form text prompts.

A naive pipeline (e.g., AudioLDM2$_{\text{Speech}}$ + AudioLDM2$_{\text{Sound}}$ [5]) rewrites the free-form prompt into separate T2S and TTS tracks, generates them independently, and then merges them. As shown

*Contact: wangyuyue123@ruc.edu.cn
†Corresponding author.

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

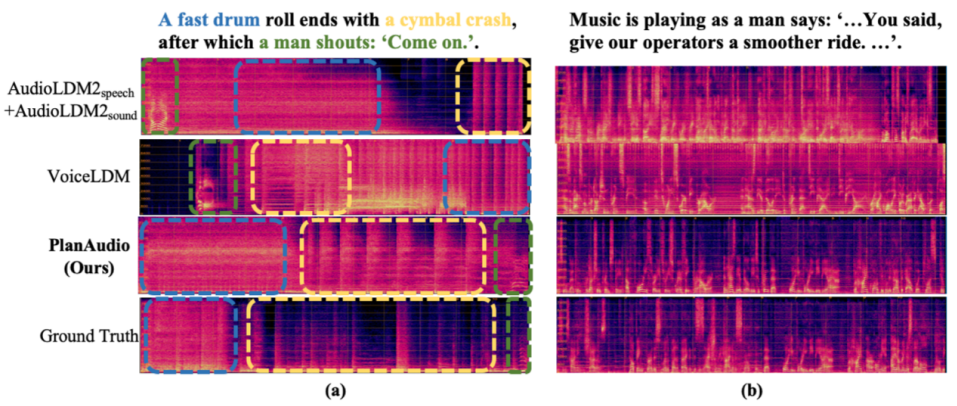


Figure 1: Qualitative comparison of audio mel-spectrograms. AudioLDM2$_{\text{Speech}}$ + AudioLDM2$_{\text{Sound}}$ denotes a pipeline approach with separate sound and speech generation, while VoiceLDM denotes a unified baseline. Textual keywords and corresponding audio regions are color-coded. (a) Pipeline and baseline models exhibit temporal misalignments (e.g., out-of-order events), whereas PlanAudio more faithfully reproduces the event sequence. (b) PlanAudio achieves more consistent relative acoustic prominence between speech and sound, closely matching the Ground Truth.

in Fig. 1, it fails to capture temporal and semantic interactions, causing misaligned event timing and unbalanced relative levels. Recent unified models [7–10] avoid explicit track merging but still require structured multi-inputs or external LLM rewriting. For example, VoiceLDM [7] (its generated mel-spectrogram also shown in Fig. 1) rewrites the prompt into "speech content" and "environment description," rather than inferring from the original prompt. Such rewriting restricts free-form prompting, risks error propagation, and increases system complexity.

We therefore introduce **Free-Form-Prompt-to-Unified-Audio Generation**, where a model synthesizes speech, sound, or joint speech-and-sound from a single unstructured prompt, with no structured templates, task tags, or external rewriting. This task raises three challenges: (1) eliminating the prompt decomposition that prior paradigms depend on; (2) avoiding the omission of events specified in the prompt when mapping free-form text directly to audio; and (3) addressing the scarcity of joint speech-and-sound annotations, as existing corpora [11–15] treat speech and sound in isolation.

To address these challenges, we propose PlanAudio, an autoregressive LLM framework that directly processes free-form prompts using the LLM's native tokenizer, without relying on external prompt rewriting or text encoders. To tackle the difficulty of direct free-form generation, PlanAudio incorporates a latent planning mechanism into the autoregressive generation process, predicting continuous audio layout to capture the global temporal and semantic relationships of audio events prior to acoustic token synthesis. We further construct a large-scale training corpus and introduce Free-Form Unified Audio Generation Benchmark (FUAG-Bench) for evaluation, both spanning speech, sound, and joint speech-and-sound generation. FUAG-Bench incorporates 4500 real-world speech-and-sound recordings with automatically generated and human-verified joint annotations. A single PlanAudio model outperforms both pipeline and unified baselines on speech-and-sound joint generation, while remaining competitive with single-task specialists on speech generation and sound generation. Further analysis confirms that our continuous audio layout plan outperforms textual planning, acoustic planning, and direct generation. Audio samples are available at `https://planaudio.github.io/PlanAudio2026/`.

2 Related works

2.1 Audio Generation: From Specific to Unified

Text-to-Speech (TTS) [1–3] and Text-to-Sound (T2S) [5, 4, 6] focus on linguistic and environmental audio, respectively. Recent efforts enhance generation control via natural language, such as structuring prompts to guide temporal order in T2S [16–18] or specifying instructions to control prosody and

style in TTS [19–21]. However, these scenario-specific methods cannot model intricate interactions between speech and sounds, failing to generate unified audio from free-form text.

To overcome this, unified models have emerged, broadly falling into two paradigms: (1) Task-level unification [22, 23], which integrates multiple audio tasks into a single model but lacks speech-and-sound joint generation; and (2) Generation-level unification [7, 10, 8, 9], which can synthesize joint speech-and-sound content. However, generation-level unification always requires explicit prompt decomposition or structured templates, shifting the burden of input structuring to users or external LLMs. In contrast, PlanAudio offers a unified framework that seamlessly generates speech, sound, and speech-and-sound content from free-form text prompts without explicit prompt rewriting.

2.2 Planning in Audio Generation

Intermediate planning has been explored in generative models to improve the controllability and quality of complex outputs [24–26]. In audio generation, existing methods mainly represent intermediate plans as textual reasoning chains or structured instructions. For example, CoT-VTM [24] generates explicit reasoning chains for vision-to-music generation, and OV-InstructTTS [25] leverages structured intermediate instructions for controllable speech synthesis. Similarly, prompt decomposition or rewriting in existing unified audio generation frameworks [8, 7, 10, 17] can also be viewed as a form of explicit textual plan. They use manual or external LLMs to decompose the prompt into environment description and speech content, then feed them separately into the model. However, constructing explicit textual plans incurs additional annotation or rewriting costs, making them expensive to obtain at scale.

Instead of relying on explicit textual plans, an alternative is to represent intermediate plans in latent spaces. In particular, latent reasoning [27–29] operates directly on continuous representations, preserving fine-grained structural information that are difficult to verbalize. This capability is especially critical for complex audio generation, where simple text cannot specify the subtle interplay between speech and sound events, such as relative loudness balance and temporal intertwining. Despite this potential, leveraging continuous semantic latents as an intermediate global plan for unified audio generation from free-form prompts remains underexplored.

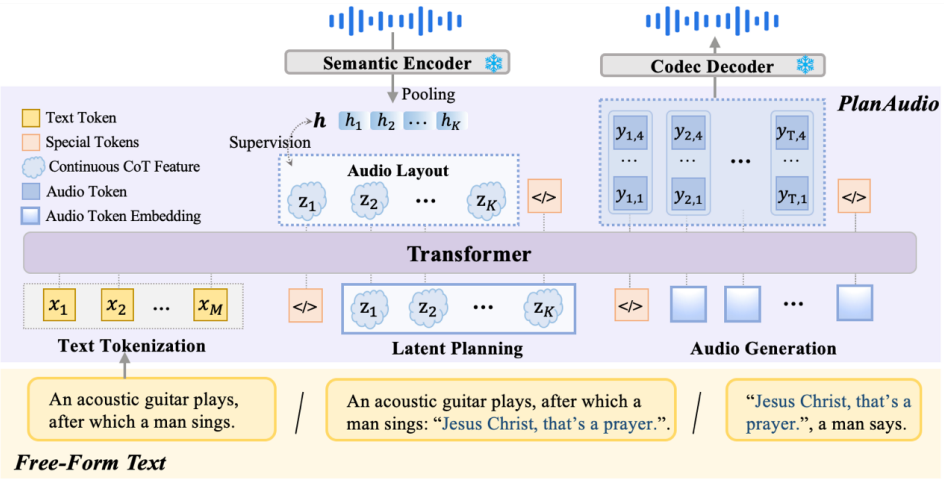


Figure 2: PlanAudio framework. Given a free-form text prompt, PlanAudio first performs latent planning to predict an audio layout, which captures the temporal and semantic organization of the target audio. The predicted audio layout then guides the autoregressive generation of hierarchical audio tokens, which are subsequently decoded into the final unified audio waveform.

3 Methodology

We formulate Free-Form-Prompt-to-Unified-Audio Generation as generating a unified audio waveform y directly from a single unconstrained text prompt x, without external prompt decomposition

or rewriting. The prompt may describe sound events, speech, or the both, and the model generates corresponding audio while preserving the specified semantic and temporal relationships.

We present **PlanAudio** (Fig. 2), an LLM-based framework that directly maps free-form text to unified audio within a single autoregressive model. Instead of explicitly decomposing the prompt into separate speech and sound instructions, we introduce a latent planning mechanism. PlanAudio first predicts a continuous audio layout $\mathbf{z}$ that captures the temporal and semantic organization of the target audio, which then guides subsequent acoustic token synthesis.

3.1 Text and Audio Representations

We first introduce the representations and notions used by PlanAudio:

- The free-form text prompt x is processed by the built-in LLM tokenizer without external text encoders. The resulting tokens $\mathbf{x} = \{x_1, \ldots, x_M\}$ serve as the conditional prefix.
- We use two representations of the target audio y. For **audio layout**, we extract a sequence of continuous audio features using the pre-trained Audio Flamingo 3 Encoder (AF3Encoder) [30] and apply segment-wise mean pooling to obtain a fixed-length sequence $\mathbf{h} = \{h_1, \ldots, h_K\}$, where each h_k captures *semantic* information from a corresponding temporal segment of the target audio. These features provide the training targets for learning the continuous audio layout $\mathbf{z}$ and are not involved in inference. For **acoustic representation**, the target audio is discretized into hierarchical tokens $\mathbf{y} = [y_1, \ldots, y_N] \in \mathbb{N}^{N \times Q}$ using the AudioCraft tokenizer[3], where y_n denotes the discrete audio tokens at time step n and $Q = 4$ is the number of codebooks. These tokens provide the training targets for acoustic generation.

3.2 PlanAudio Framework

As illustrated in Fig. 2, PlanAudio factorizes the Free-Form-Text-Prompt-to-Unified-Audio generation into a two-phase process mediated by a continuous audio layout $\mathbf{z} = \{z_1, \ldots, z_K\}$. The generation process is formulated as:

$$P(\mathbf{y}, \mathbf{z} \mid \mathbf{x}) = \underbrace{P(\mathbf{z} \mid \mathbf{x})}_{\text{Latent Planning}} \cdot \underbrace{P(\mathbf{y} \mid \mathbf{x}, \mathbf{z})}_{\text{Audio Generation}}, \tag{1}$$

where $\mathbf{x}$ represents the text tokens and $\mathbf{y}$ denotes the hierarchical discrete audio tokens.

To enable end-to-end processing, we organize all modalities into a single sequence, using special tokens to demarcate different segments. This allows latent planning and acoustic generation to be performed within a single Transformer backbone. Specifically, a data point is formatted as:

$$\mathbf{S} = [\texttt{<|sot|>}, \mathbf{x}, \texttt{<|sol|>}, \mathbf{z}, \texttt{<|soa|>}, \mathbf{y}, \texttt{<|eoa|>}], \tag{2}$$

where $\mathbf{x}$, $\mathbf{z}$, and $\mathbf{y}$ represent text tokens, continuous audio layout, and discrete audio tokens, respectively. Special tokens <|sot|> (start of text), <|sol|> (start of latent), <|soa|> (start of audio), and <|eoa|> (end of audio) serve as functional delimiters to guide the model through different phases.

Latent Planing. PlanAudio predicts a continuous audio layout $\mathbf{z}$ from the textual prefix $\mathbf{x}$.

$$P(\mathbf{z} \mid \mathbf{x}) = \prod_{k=1}^{K} P(z_k \mid z_{<k}, \mathbf{x}), \tag{3}$$

where K is a predefined constant representing the planning horizon. The predited audio layout provides a compact representation of the temporal and semantic organization of the target audio, including event orchestration and the prosodic orientation of speech. By introducing this latent planning stage, the model first establishes the global organization of the target audio before generating low-level acoustic details.

[3]https://github.com/facebookresearch/audiocraft

Acoustic Generation. In this phase, the model generates audio tokens conditioned on both the original prompt $\mathbf{x}$ and the continuous audio layout $\mathbf{z}$ predicted in the preceding phase. Following the hierarchical structure of the discrete tokenizer, the model predicts audio tokens across Q codebooks:

$$P(\mathbf{y} \mid \mathbf{x}, \mathbf{z}) = \prod_{n=1}^{N} \prod_{q=1}^{Q} P(y_{n,q} \mid y_{<n,*}, y_{n,<q}, \mathbf{x}, \mathbf{z}). \tag{4}$$

Here, $\mathbf{z}$ provides global guidance for acoustic generation by conveying the intended temporal and semantic organization of the target audio, while $\mathbf{x}$ preserves direct access to the original prompt.

3.3 Training and Inference

Training Objectives. To optimize the two-phase generation process, we introduce a dual-objective loss that supervises both latent planning and acoustic generation.

For **latent planning**, each planning step maps the LLM hidden state into the AF3 feature space by $\phi(\cdot)$, producing one frame of the latent audio layout $\mathbf{z}_{1:T}$. During training, $\psi(\cdot)$ maps the ground-truth AF3Encoder sequence $\mathbf{h}$ into the LLM input space, and the sequence is fed as the teacher-forced input with a one-frame shift. Both $\phi(\cdot)$ and $\psi(\cdot)$ are a linear layer followed by layer normalization. We train z_t against h_t with mean squared error (MSE) and cosine similarity:

$$\mathcal{L}_{\text{latent}} = \frac{1}{T} \sum_{t=1}^{T} \left[\|z_t - h_t\|_2^2 + \lambda \left(1 - \frac{\langle z_t, h_t \rangle}{\|z_t\| \, \|h_t\|} \right) \right], \tag{5}$$

where MSE ensures Euclidean proximity while cosine similarity encourages directional alignment.

For **audio generation**, the Q codebooks at each step are predicted in parallel. Cross-entropy loss maximizes the likelihood of the target audio tokens given the text $\mathbf{x}$, the teacher-forced layout $\mathbf{h}$, and the tokens from previous steps:

$$\mathcal{L}_{\text{audio}} = -\sum_{n=1}^{N} \sum_{q=1}^{Q} \log P(y_{n,q} \mid y_{<n}, \mathbf{x}, \mathbf{h}). \tag{6}$$

The final training objective is a weighted sum of the layout and acoustic losses: $\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{latent}} + \lambda_2 \mathcal{L}_{\text{audio}}$, where λ_1 and λ_2 are hyperparameters balancing semantic planning and generation quality.

Inference. Inference proceeds in two phases. Given the free-form text prompt $\mathbf{x}$, PlanAudio autoregressively predicts the continuous audio layout $\mathbf{z}$ for K steps. At each step, $\phi(\cdot)$ maps the LLM hidden state to a layout frame z_k, and $\psi(z_k)$ maps that frame into the LLM input space as the input for the next planning step, using the same input mapping as teacher forcing during training. After the planning phase, the <|soa|> token triggers acoustic generation. Conditioned on both $\mathbf{x}$ and $\mathbf{z}$, PlanAudio autoregressively synthesizes audio token sequence $\mathbf{y}$ until the <|eoa|> token is generated, after which the tokens are decoded into the final waveform.

4 Experiments

4.1 Implementation Details

PlanAudio is initialized from Qwen2.5-1.5B[4], which provides the text tokenizer and autoregressive backbone, with no external text encoder or rewriter. Target audio is discretized by the AudioCraft tokenizer (4 codebooks, 2048 entries, 50 Hz), and the resulting tokens are predicted under the delayed token interleaving pattern [31] (App. A.5). Semantic supervision comes from AF3Encoder [30], whose 750 output frames are mean-pooled every 125 frames into $K = 6$ anchors, and the planning and acoustic losses are weighted equally. The planning horizon K is fixed, while the audio length is determined by when <|eoa|> is generated, rather than by a preset length.

Training samples are drawn from the three scenarios (sound, speech, speech-and-sound) in a prescribed ratio, and one epoch contains as many samples as the sum of the three datasets. We use a

[4] https://huggingface.co/Qwen/Qwen2.5-1.5B

maximum batch size of 16 with a token budget of 4000 per batch, and accumulate gradients over 8 steps. All parameters are trained with Adam at an initial learning rate of 10^{-4}, using a 3000-step warmup followed by inverse square-root decay with a decay coefficient of 0.1. Training runs for 70 epochs on 8 NVIDIA A800 GPUs (80 GB) and takes about 10 days. Inference uses top-k sampling with $k = 25$. We also report, in App. A.6, the codec reconstruction comparison behind the tokenizer choice and the trade-off with generative performance, and analyze the choices of planning horizon K and the loss weights in App. A.4.1 and App. A.4.3.

4.2 Data Construction and Benchmark

Free-form-prompt-to-unified-audio generation takes no task label: the model infers from the prompt alone whether speech, sound, or both are requested. For analysis only, we group prompts into three scenarios: **sound**, **speech**, and **speech-and-sound**. There is a bottleneck in joint annotation: public corpora provide either speec transcripts [11, 15] or sound descriptions [12, 13], but never both for the same recording. We therefore build our own training corpus and benchmark, using only real recordings and generating only the prompts.

Training corpus. For **speech-and-sound**, we use raw AudioSet recordings [32] to preserve real-world speech-sound interaction. Automatically writing free-form prompts for these recordings in a single pass is prone to hallucination in both scene and speech, so we generate the two parts separately and then merge them: Whisper-large [33] transcribes the speech, and Gemini-2.5 Pro describes the scene with a `<speech content>` placeholder that is then filled with the transcript. The retained *description-transcript pairs* later serve as the Oracle-Split input for baselines (Sec. 4.3). **Sound** and **speech** data preserve single-scenario ability and teach the model to tell scenarios apart without tags. Sound prompts are rewritten from AudioCaps [12] and WavCaps [13] captions; speech prompts combine LibriTTS [11] and Common Voice [15] transcripts with speaker attributes and an explicit mention of a quiet environment. We also add speech segments extracted from the speech-and-sound recordings, so that identical transcripts appear with and without their acoustic scene and only the prompt can distinguish them. Since free-form prompts can describe the same audio in many ways, each clip is paired with 5 prompt variants to improve flexibility and generalization to unseen phrasings. The corpus contains 1.43M audio clips (372k speech-and-sound, 451k sound, 605k speech). Please refer to App. A.1 for the specific data construction prompt.

Evaluation benchmark. No existing benchmark evaluates unified audio generation from free-form prompts. We therefore introduce the **Free-Form Unified Audio Generation Benchmark (FUAG-Bench)**, in which each test item pairs a free-form prompt with a real recording. Its three subsets follow the scenarios above. The **Sound** subset uses the AudioCaps [12] test set with its original captions. The **Speech** subset uses the LibriTTS [11] test set, with prompts generated in the same way as for the training data and one of the 5 variants kept at random per clip. The **Speech-and-sound** subset uses 4500 held-out AudioSet [32] clips, with prompts generated and sampled likewise; these clips and their derived speech segments are excluded from all training data. For the speech-and-sound subset, we further provide a human audit of annotation quality in App. A.3.1, and the distribution of speech-sound interaction types in App. A.3.2.

4.3 Evaluation Setup

Metrics. We evaluate a single PlanAudio checkpoint on all three subsets of FUAG-Bench (Sec. 4.2), using the same free-form input throughout. Objective metrics mainly cover two dimensions. (1) Speech-related (Speech & Speech-and-Sound): Word Error Rate (WER) for intelligibility, and UTMOS for perceived voice quality. (2) Sound-related (Sound & Speech-and-Sound): Fréchet distance (FD) and KL divergence, each computed with PANNs [34] and PaSST [35], for audio fidelity; Inception Score (IS) for diversity; and CLAP [36] similarity for text-audio alignment. To evaluate temporal structure of sound events, we follow [37] and report the Wasserstein distance (W-dis) between the onset detection function curves of the generated and reference audio, where lower values indicate closer timing. To measure whether sound events named in the prompt are missing from the audio, we also report the Semantic Coverage Factor (SCF; see details in App. A.7).

Since automatic metrics cannot fully judge whether the audio follows the prompt or whether speech and sound blend naturally, we further conduct a subjective test on 50 random speech-and-sound

Table 1: Results on the speech-and-sound subset of FUAG-Bench. Sound- and Speech-related metrics are evaluated using official checkpoints. Best results are in **bold**; second-best are underlined. PlanAudio achieves superior overall performance, dominating sound generation while remaining competitive in speech.

Model	Decomposition	Sound-related						Speech-related	
		FD_{PANNs} ↓	FD_{PaSST} ↓	KL_{PaSST} ↓	CLAP ↑	SCF ↑	W-dis ↓	WER ↓	UTMOS ↑
GroundTruth	-	0.00	0.00	0.00	0.17	0.50	0.0	0.10	2.69
Reconstruction	-	3.82	112	0.23	0.21	-	-	0.21	2.49
$AudioLDM2_{S+Sp}$ [5]	Oracle-Split	14.3	240	1.10	0.21	0.43	3.97	0.71	2.16
VoiceLDM-s [7]	Oracle-Split	25.2	379	1.39	0.15	0.25	5.38	0.70	2.35
VoiceLDM-m [7]	Oracle-Split	23.4	343	1.34	0.19	0.38	5.14	**0.13**	**2.81**
VoiceLDM-m[7]	Gemini-2.5-pro	23.7	347	1.34	0.19	0.35	5.21	0.19	2.71
	Qwen3-0.6B	23.8	362	1.36	0.15	0.30	5.26	0.27	2.73
PlanAudio	None	**8.96**	**197**	**0.98**	**0.22**	**0.45**	**3.04**	0.30	2.53

Table 2: Subjective results for speech-and-sound subset of FUAG-Bench. Ratings (0-5) across four criteria: Acoustic Quality (Quality), Temporal Correctness (Temporal), Semantic Alignment (Semantic), and Authenticity. Confidence intervals are 95%. PlanAudio performs best in all criteria.

Model	Decompisition	Quality ↑	Temporal ↑	Semantic ↑	Authenticity ↑
VoiceLDM-s [7]	Oracle-Split	2.78 ± 0.30	2.65 ± 0.24	2.67 ± 0.25	2.71 ± 0.28
VoiceLDM-m [7]	Oracle-Split	2.83 ± 0.11	2.78 ± 0.13	2.95 ± 0.13	2.93 ± 0.17
$AudioLDM2_{S+Sp}$ [5]	Oracle-Split	2.24 ± 0.31	2.20 ± 0.32	2.38 ± 0.36	2.40 ± 0.39
PlanAudio	None	$\mathbf{3.23} \pm 0.13$	$\mathbf{3.16} \pm 0.16$	$\mathbf{3.36} \pm 0.13$	$\mathbf{3.47} \pm 0.21$

samples. Ten listeners rate each sample from 0 to 5 on *Acoustic Quality*, *Temporal Correctness*, *Semantic Alignment*, and *Authenticity*.

Baselines. No prior model accepts free-form prompts, so we compare with three types of models, each using its own input format. (1) *Unified*: VoiceLDM [7], which takes a transcript and an environment description and can generate all three scenarios[5]. (2) *Specialists*: AudioLDM2 [5], Tango [38], and Make-An-Audio [39] for sound; PromptTTS++ [40] and CosyVoice2 [1] for speech. (3) *Pipelines*: for speech-and-sound, $AudioLDM2_{Speech}$ + $AudioLDM2_{Sound}$ ($AudioLDM2_{S+Sp}$), whose two outputs are mixed.

For sound subset and speech subset, specialists receive their native inputs; VoiceLDM receives the scene description or transcript in the matching condition, with the other left empty. For the speech-and-sound subset, every baseline needs the prompt split into a scene description and a transcript. Because the quality of this split directly affects the generated audio, we obtain it in three ways: with Qwen3-0.6B [41] as a lightweight local rewriter, with Gemini-2.5 Pro as a strong API-based rewriter, and with Oracle-Split, the *description-transcript pairs* kept during data construction (Sec. 4.2). The setting used by each baseline is listed in the result tables.

4.4 Main Results and Analysis

All results below come from the same PlanAudio checkpoint.

Superiority in Speech-and-Sound Scenario. Tab. 1 and Tab. 2 report objective and subjective results on the speech-and-sound subset. This subset tests the core difficulty of the task: the model must realize every prompted event and place speech and sound in the right order and balance.

On sound-related metrics, PlanAudio achieves the best FD and KL among all baselines, even when they receive Oracle-Split. It also achieves the lowest W-dis (3.04 vs. 3.97 for the best baseline), indicating event timing closest to the reference, and the highest SCF (0.45, close to 0.50 for the ground truth), indicating that it omits the fewest prompted events. The subjective test agrees: PlanAudio

[5]Audiobox [10], VoiceDit [8], and ControlAudio [9] also accept structured inputs but lack open-source checkpoints, restricting our comparisons to available models.

Table 3: Results on the sound subset of FUAG-Bench, evaluated on Sound-related dimensions only. Results are derived from official checkpoints, with the highest scores in **bold** and the second highest underlined. Rows in gray denote specialist baselines designed solely for text-to-sound. PlanAudio surpasses unified baselines on all metrics and maintains competitive against specialist models.

Model	FD_{PANNs} ↓	FD_{PaSST} ↓	KL_{PANNs} ↓	KL_{PaSST} ↓	IS ↑	CLAP ↑	SCF ↑	W-dis ↓
GroundTruth	1.05	0.29	0.02	0.00	13.2	0.29	0.35	0.0
Reconstruction	11.4	130	0.49	0.48	9.11	0.23	-	-
AudioLDM2 [5]	32.5	395	1.56	1.51	**8.54**	**0.21**	0.29	4.30
Make-An-Audio [39]	27.9	**182**	1.60	1.62	7.44	**0.21**	0.27	4.13
Tango [38]	26.1	276	**1.37**	**1.29**	8.23	0.19	**0.33**	**3.78**
VoiceLDM-s [7]	58.4	430	3.27	3.01	4.41	0.10	0.14	6.23
VoiceLDM-m [7]	55.8	433	3.37	3.05	4.18	0.07	0.15	4.88
PlanAudio	**24.7**	233	1.93	1.89	8.02	0.19	0.32	3.89

scores highest on all four criteria, with clear margins in temporal correctness and semantic alignment (Tab. 2). Fig. 1 also illustrates both aspects. In case (a), PlanAudio follows the prompted event order, whereas the baselines place events out of order. In case (b), PlanAudio matches the relative loudness of speech and sound in the ground truth, whereas the baselines do not.

On speech-related metrics, PlanAudio falls behind VoiceLDM-m, mainly for two reasons. First, the codec limits its speech quality: reconstructed audio already has a WER of 0.21. Second, VoiceLDM's training data include clean speech overlaid onto separately recorded sound, which keeps speech easy to recognize but makes the scene less natural, whereas PlanAudio is trained on real recordings, where speech is mixed with the scene. This trade-off appears in the subjective *Authenticity* score, where PlanAudio leads VoiceLDM-m by 0.54. Moreover, VoiceLDM-m's intelligibility depends on the quality of the prompt split: when the Oracle-Split is replaced by Gemini-2.5 Pro and Qwen3-0.6B, its WER rises from 0.13 to 0.19 and 0.27, whereas PlanAudio reads raw free-form prompts.

Table 4: Results on the speech subset of FUAG-Bench, evaluated on Speech-related metrics only. All results are derived from official checkpoints, with the highest scores in **bold** and the second highest underlined. Rows in gray denote specialist baselines designed solely for text-to-speech. PlanAudio outperforms unified baselines on all metrics and maintains competitive performance against specialist baselines.

Model	WER ↓	UTMOS ↑
GroundTruth	0.03	3.69
Reconstruction	0.04	3.13
Prompt TTS++ [40]	0.12	**3.51**
VoiceLDM-s [7]	0.62	2.75
VoiceLDM-m [7]	0.13	2.99
PlanAudio	0.11	3.11

Generalization to Domain-Specific Audio Synthesis. Tab. 3 and Tab. 4 report performance on the sound subset and the speech subset. They evaluate whether unifying all three scenarios in a single model degrades quality when the prompt requests only sound or only speech. Across both subsets, PlanAudio consistently outperforms the unified baseline VoiceLDM and remains competitive with task-specific specialists, so unification does not degrade quality. On the sound subset, PlanAudio achieves CLAP, SCF, and W-dis scores close to those of the best specialist, showing that it captures both the events in the prompt and their timing. On the speech subset, PlanAudio surpasses VoiceLDM on both WER and UTMOS, unlike on the speech-and-sound subset, highlighting its high-fidelity synthesis when the prompt requests only speech. Since PlanAudio takes free-form prompts without task tags, results also show that it infers from the prompt alone whether sound or speech is requested.

4.5 Does the Latent Plan Really Matter?

To test whether the latent planning mechanism is doing real work, we apply two inference interventions to the fully trained model. **GT-Plan** replaces the predicted audio layout $\mathbf{z}$ with the AF3 feature $\mathbf{h}$ of the ground-truth audio, removing planning error; **Reverse-Plan** supplies the AF3 feature $\mathbf{h}$ in reversed order, preserving content while destroying temporal structure. Results are shown in Tab. 5.

Compared with PlanAudio, GT-Plan substantially reduces FD, from 197 to 140 for Speech-and-Sound and from 233 to 176 for Sound, closing approximately 67% and 55% of the respective gaps to the Reconstruction reference. This indicates that more accurate latent plans substantially improve distributional fidelity. However, the improvement in intelligibility is relatively limited (Speech-and-Sound WER: $0.30 \rightarrow 0.28$, compared with 0.21 for Reconstruction), suggesting that planning accuracy alone is insufficient to fully recover intelligibility. This residual gap may also reflect limitations in the phonetic information retained by the latent representation and errors in the generation process. Conversely, Reverse-Plan substantially degrades performance across all three scenarios. This highlights the importance of temporal ordering in the latent plan: although PlanAudio has no explicit timing module, the ordering of latent features implicitly organizes audio events over time. Disrupting this order while preserving the feature content severely compromises generation quality. Together, these interventions demonstrate that the latent plan encodes not only event-level content but also its temporal organization.

Table 5: Analysis of the latent planning mechanism. *GT-Plan* replaces the predicted audio layout $\mathbf{z}$ with AF3 feature $\mathbf{h}$ extracted from the ground-truth audio; *Reverse-Plan* supplies the same $\mathbf{h}$ in reversed temporal order. Both are inference interventions on the fully trained model. FD and KL are computed via the PaSST classifier.

Model	Speech-and-Sound					Sound			Speech	
	FD ↓	KL ↓	CLAP ↑	WER ↓	UTMOS ↑	FD ↓	KL ↓	CLAP ↑	WER ↓	UTMOS ↑
Reconstruction	112	0.37	0.21	0.21	2.49	130	0.48	0.23	0.04	3.13
PlanAudio	197	0.98	0.22	0.30	2.53	233	1.89	0.19	0.11	3.11
GT-Plan	140	0.61	0.22	0.28	2.51	176	1.24	0.21	0.09	3.05
Reverse-Plan	235	1.03	0.20	0.67	2.45	288	2.03	0.16	0.77	3.13

4.6 Analysis of Latent Planning Mechanism Design

We compare five different Planning mechanisms: (1) *PlanAudio*: latent planning in continuous semantic space; (2) *w/o Planning*: direct audio generation from text without an intermediate planning; (3) *Textual Planning*: a natural language reasoning chain (generated by Gemini-2.5 Pro) serving as a textual bridge for planning supervision; (4) *Acoustic Planning*: latent planning in continuous acoustic space, using codebook-based acoustic vectors as supervision; (5) *Discrete Planning*: latent planning in discrete semantic space, supervised by a semantic token sequence generated from k-means quantization of AF3 features. All variants follow the main experimental configuration, but are trained for 50 epochs on a one-third data subset. Tab. 6 yields several insights:

Table 6: Planning mechanism comparison on three subsets of FUAG-Bench. FD and KL metrics are computed via the PasST classifier. Results indicate that PlanAudio's semantic latent planning achieves superior performance.

Model	Speech-and-Sound						Sound			Speech	
	FD ↓	KL ↓	CLAP ↑	WER ↓	UTMOS ↑	SCF ↑	FD ↓	KL ↓	CLAP ↑	WER ↓	UTMOS ↑
PlanAudio	**177**	**1.07**	**0.20**	**0.86**	**2.24**	**0.34**	**242**	**1.94**	**0.17**	**0.35**	**3.05**
– w/o Planning	<u>217</u>	1.43	0.15	<u>0.92</u>	2.16	0.12	299	3.26	0.03	<u>0.50</u>	<u>2.90</u>
– w/ Explicit Planning	230	1.38	<u>0.16</u>	1.12	2.13	0.20	<u>279</u>	<u>3.10</u>	<u>0.05</u>	1.29	2.34
– w/ Acoustic Planning	319	<u>1.37</u>	0.14	1.21	**2.24**	0.09	374	3.21	0.04	2.68	2.70
– w/ Discrete Planning	213	1.32	0.17	1.19	2.18	0.12	254	3.10	0.05	1.88	2.33

(1) **Effectiveness of semantic latent planning.** *PlanAudio* is consistently the best across all subset of FUAG-Bench, confirming that a semantic latent space bridges the text-audio modality gap more effectively than textual or acoustic representations. (2) **Expressive limits of textual and discrete planning.** Although *textual planning* benefits sound generation, it suffers from an expressive bottleneck when modeling dense, overlapping speech-and-sound interactions, which our latent method successfully captures. Additionally, PlanAudio outperforms *discrete planning* by preserving fine-grained acoustic nuances and semantic density, whereas quantization-induced information loss disrupts acoustic continuity. (3) **The Reconstruction-Semantic Mismatch.** *Acoustic planning* performs worst overall, as codec-based features optimized for reconstruction lack the semantic depth required for planning. Unlike semantic latents, they fail to bridge the gap between textual prompts

and acoustic realization. (4) **Robustness to Semantic Omission.** *PlanAudio* achieves the highest SCF, proving that the semantic plan effectively acts as a robust semantic anchor to ensure accurate event synthesis.

5 Conclusion

We present PlanAudio, a framework for unified audio generation from free-form text. The model performs latent planning in a continuous semantic space, bridging high-level semantics and low-level acoustic synthesis. We also introduce a large-scale corpus and FUAG-Bench for free-form-prompt-to-unified-audio traning and evaluation. PlanAudio surpasses baselines in speech-and-sound joint generation while remaining competitive in speech and sound. Future work will target stronger audio codecs and better semantic-acoustic trade-offs. We also acknowledge the risk of misuse for deceptive audio generation.

Acknowledgments and Disclosure of Funding

This work is supported by the National Natural Science Foundation of China (No.62276268) and Huawei Inc. The authors declare no competing interests.

References

[1] Z. Du, Y. Wang, Q. Chen, X. Shi, X. Lv, T. Zhao, Z. Gao, Y. Yang, C. Gao, H. Wang, F. Yu, H. Liu, Z. Sheng, Y. Gu, C. Deng, W. Wang, S. Zhang, Z. Yan, and J. Zhou, "Cosyvoice 2: Scalable streaming speech synthesis with large language models," *CoRR*, vol. abs/2412.10117, 2024.

[2] Z. Du, C. Gao, Y. Wang, F. Yu, T. Zhao, H. Wang, X. Lv, H. Wang, C. Ni, X. Shi, K. An, G. Yang, Y. Li, Y. Chen, Z. Gao, Q. Chen, Y. Gu, M. Chen, Y. Chen, S. Zhang, W. Wang, and J. Ye, "Cosyvoice 3: Towards in-the-wild speech generation via scaling-up and post-training," *CoRR*, vol. abs/2505.17589, 2025.

[3] Q. Team, "Qwen3-tts technical report," *CoRR*, vol. abs/2601.15621, 2026.

[4] F. Kreuk, G. Synnaeve, A. Polyak, U. Singer, A. Défossez, J. Copet, D. Parikh, Y. Taigman, and Y. Adi, "Audiogen: Textually guided audio generation," in *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[5] H. Liu, Y. Yuan, X. Liu, X. Mei, Q. Kong, Q. Tian, Y. Wang, W. Wang, Y. Wang, and M. D. Plumbley, "Audioldm 2: Learning holistic audio generation with self-supervised pretraining," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 32, pp. 2871–2883, 2024.

[6] Z. Evans, J. D. Parker, C. Carr, Z. Zukowski, J. Taylor, and J. Pons, "Stable audio open," in *2025 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2025, Hyderabad, India, April 6-11, 2025*. IEEE, 2025, pp. 1–5.

[7] Y. Lee, I. Yeon, J. Nam, and J. S. Chung, "Voiceldm: Text-to-speech with environmental context," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2024, Seoul, Republic of Korea, April 14-19, 2024*. IEEE, 2024, pp. 12 566–12 571.

[8] J. Jung, J. Ahn, C. Jung, T. D. Nguyen, Y. Jang, and J. S. Chung, "Voicedit: Dual-condition diffusion transformer for environment-aware speech synthesis," in *2025 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2025, Hyderabad, India, April 6-11, 2025*. IEEE, 2025, pp. 1–5.

[9] Y. Jiang, Z. Chen, Z. Ju, Y. Dai, W. Dou, and J. Zhu, "Controlaudio: Tackling text-guided, timing-indicated and intelligible audio generation via progressive diffusion modeling," *CoRR*, vol. abs/2510.08878, 2025.

[10] A. Vyas, B. Shi, M. Le, A. Tjandra, Y. Wu, B. Guo, J. Zhang, X. Zhang, R. Adkins, W. Ngan, J. Wang, I. Cruz, B. Akula, A. Akinyemi, B. Ellis, R. Moritz, Y. Yungster, A. Rakotoarison, L. Tan, C. Summers, C. Wood, J. Lane, M. Williamson, and W. Hsu, "Audiobox: Unified audio generation with natural language prompts," *CoRR*, vol. abs/2312.15821, 2023.

[11] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "Libritts: A corpus derived from librispeech for text-to-speech," in *20th Annual Conference of the International Speech Communication Association, Interspeech 2019, Graz, Austria, September 15-19, 2019*, G. Kubin and Z. Kacic, Eds. ISCA, 2019, pp. 1526–1530.

[12] C. D. Kim, B. Kim, H. Lee, and G. Kim, "Audiocaps: Generating captions for audios in the wild," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 119–132.

[13] X. Mei, C. Meng, H. Liu, Q. Kong, T. Ko, C. Zhao, M. D. Plumbley, Y. Zou, and W. Wang, "Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 32, pp. 3339–3354, 2024.

[14] Z. Jin, J. Jia, Q. Wang, K. Li, S. Zhou, S. Zhou, X. Qin, and Z. Wu, "Speechcraft: A fine-grained expressive speech dataset with natural language description," in *Proceedings of the 32nd ACM International Conference on Multimedia, MM 2024, Melbourne, VIC, Australia, 28 October 2024 - 1 November 2024*, J. Cai, M. S. Kankanhalli, B. Prabhakaran, S. Boll, R. Subramanian, L. Zheng, V. K. Singh, P. César, L. Xie, and D. Xu, Eds. ACM, 2024, pp. 1255–1264.

[15] R. Ardila, M. Branson, K. Davis, M. Kohler, J. Meyer, M. Henretty, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," in *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, and S. Piperidis, Eds. European Language Resources Association, 2020, pp. 4218–4222.

[16] Y. Wang, H. Chen, D. Yang, Z. Wu, and X. Wu, "Audiocomposer: Towards fine-grained audio generation with natural language descriptions," in *2025 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2025, Hyderabad, India, April 6-11, 2025*. IEEE, 2025, pp. 1–5.

[17] Y. Jiang, Z. Chen, Z. Ju, C. Li, W. Dou, and J. Zhu, "Freeaudio: Training-free timing planning for controllable long-form text-to-audio generation," in *Proceedings of the 33rd ACM International Conference on Multimedia, MM 2025, Dublin, Ireland, October 27-31, 2025*, C. Gurrin, K. Schoeffmann, M. Zhang, L. Rossetto, S. Rudinac, D. Dang-Nguyen, W. Cheng, P. Chen, and J. Benois-Pineau, Eds. ACM, 2025, pp. 9871–9880.

[18] Z. Xie, X. Xu, Z. Wu, and M. Wu, "Picoaudio: Enabling precise temporal controllability in text-to-audio generation," in *2025 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2025, Hyderabad, India, April 6-11, 2025*. IEEE, 2025, pp. 1–5.

[19] Y. Zhou, X. Qin, Z. Jin, S. Zhou, S. Lei, S. Zhou, Z. Wu, and J. Jia, "Voxinstruct: Expressive human instruction-to-speech generation with unified multilingual codec language modelling," in *Proceedings of the 32nd ACM International Conference on Multimedia, MM 2024, Melbourne, VIC, Australia, 28 October 2024 - 1 November 2024*, J. Cai, M. S. Kankanhalli, B. Prabhakaran, S. Boll, R. Subramanian, L. Zheng, V. K. Singh, P. César, L. Xie, and D. Xu, Eds. ACM, 2024, pp. 554–563.

[20] D. Chen, X. Zhang, Y. Wang, K. Dai, L. Ma, and Z. Wu, "Flexivoice: Enabling flexible style control in zero-shot TTS with natural language instructions," *CoRR*, vol. abs/2601.04656, 2026.

[21] H. Xue, X. Song, Y. Tang, J. Chen, Y. Chen, Y. Li, and Y. Zhou, "Moe-tts: Enhancing out-of-domain text understanding for description-based TTS via mixture-of-experts," *CoRR*, vol. abs/2508.11326, 2025.

[22] D. Yang, J. Tian, X. Tan, R. Huang, S. Liu, X. Chang, J. Shi, S. Zhao, J. Bian, X. Wu, Z. Zhao, S. Watanabe, and H. Meng, "Uniaudio: An audio foundation model toward universal audio generation," *CoRR*, vol. abs/2310.00704, 2023.

[23] R. Valle, R. Badlani, Z. Kong, S. Lee, A. Goel, S. Kim, J. F. Santos, S. Dai, S. Gururani, A. Aljafari, A. H. Liu, K. J. Shih, R. Prenger, W. Ping, C. H. Yang, and B. Catanzaro, "Fugatto 1: Foundational generative audio transformer opus 1," in *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

[24] X. Guan, Z. Gu, J. Huo, T. Ding, and Y. Gao, "Cot-vtm: Visual-to-music generation with chain-of-thought reasoning," in *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, ser. Findings of ACL, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds. Association for Computational Linguistics, 2025, pp. 12 493–12 510.

[25] Y. Ren, J. Yi, J. Tao, H. Sun, Z. Wen, H. Gu, L. Xu, and Y. Bai, "Ov-instructtts: Towards open-vocabulary instruct text-to-speech," *CoRR*, vol. abs/2601.01459, 2026.

[26] H. Xue, Y. Tang, H. Liu, J. Zhang, X. Geng, and L. Xie, "Enhancing non-core language instruction-following in speech llms via semi-implicit cross-lingual cot reasoning," in *Proceedings of the 33rd ACM International Conference on Multimedia, MM 2025, Dublin, Ireland, October 27-31, 2025*, C. Gurrin, K. Schoeffmann, M. Zhang, L. Rossetto, S. Rudinac, D. Dang-Nguyen, W. Cheng, P. Chen, and J. Benois-Pineau, Eds. ACM, 2025, pp. 10 984–10 993.

[27] J. Geiping, S. McLeish, N. Jain, J. Kirchenbauer, S. Singh, B. R. Bartoldson, B. Kailkhura, A. Bhatele, and T. Goldstein, "Scaling up test-time compute with latent reasoning: A recurrent depth approach," *CoRR*, vol. abs/2502.05171, 2025.

[28] S. Hao, S. Sukhbaatar, D. Su, X. Li, Z. Hu, J. Weston, and Y. Tian, "Training large language models to reason in a continuous latent space," *CoRR*, vol. abs/2412.06769, 2024.

[29] X. Chen, A. Zhao, H. Xia, X. Lu, H. Wang, Y. Chen, W. Zhang, J. Wang, W. Li, and X. Shen, "Reasoning beyond language: A comprehensive survey on latent chain-of-thought reasoning," *CoRR*, vol. abs/2505.16782, 2025.

[30] A. Goel, S. Ghosh, J. Kim, S. Kumar, Z. Kong, S. Lee, C. H. Yang, R. Duraiswami, D. Manocha, R. Valle, and B. Catanzaro, "Audio flamingo 3: Advancing audio intelligence with fully open large audio language models," *CoRR*, vol. abs/2507.08128, 2025.

[31] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023.

[32] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*. IEEE, 2017, pp. 776–780.

[33] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds. PMLR, 2023, pp. 28 492–28 518.

[34] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 2880–2894, 2020.

[35] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," in *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*. ISCA, 2022, pp. 2753–2757.

[36] Y. Wu*, K. Chen*, T. Zhang*, Y. Hui*, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2023.

[37] X. Wang, Y. Wang, Y. Wu, R. Song, X. Tan, Z. Chen, H. Xu, and G. Sui, "Tiva: Time-aligned video-to-audio generation," in *Proceedings of the 32nd ACM International Conference on Multimedia, MM 2024, Melbourne, VIC, Australia, 28 October 2024 - 1 November 2024*, J. Cai, M. S. Kankanhalli, B. Prabhakaran, S. Boll, R. Subramanian, L. Zheng, V. K. Singh, P. César, L. Xie, and D. Xu, Eds. ACM, 2024, pp. 573–582.

[38] D. Ghosal, N. Majumder, A. Mehrish, and S. Poria, "Text-to-audio generation using instruction-tuned LLM and latent diffusion model," *CoRR*, vol. abs/2304.13731, 2023.

[39] R. Huang, J. Huang, D. Yang, Y. Ren, L. Liu, M. Li, Z. Ye, J. Liu, X. Yin, and Z. Zhao, "Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models," in *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, ser.

Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds. PMLR, 2023, pp. 13 916–13 932.

[40] R. Shimizu, R. Yamamoto, M. Kawamura, Y. Shirahata, H. Doi, T. Komatsu, and K. Tachibana, "Prompttts++: Controlling speaker identity in prompt-based text-to-speech using natural language descriptions," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2024, Seoul, Republic of Korea, April 14-19, 2024*. IEEE, 2024, pp. 12 672–12 676.

[41] Q. Team, "Qwen3 technical report," *CoRR*, vol. abs/2505.09388, 2025.

A Technical appendices and supplementary material

A.1 Data Construction Prompts

We present detailed prompts tailored for our data construction pipeline, comprising three key components: (1) prompts for speech-and-sound annotation; (2) prompts for sound caption enhancement; (3) prompts for speech caption enhancement.

Prompts for Speech-and-Sound Annotation
The user will input an audio track. Briefly describe the audio, focusing exclusively on what can be perceived aurally, ignore those things that cannot be verified from the audio. Generate five diverse descriptions for the same audio, separate them by inserting "`\t`" in the end.

- Describe audio events in order, but avoid specific timestamps.
- The audio always contains speech (including singing), identify the primary speaker(s) based on the audio, and always use the placeholders "`<speech content>`" to represent their transcriptions.
- Do not describe the topic of the speech, do not transcribe the speech. Do not use phrases like "continues to speak" or "continues singing", represent the entire speech content of a primary speaker with a single "`<speech content>`", unless there are interleaved conversations among multiple primary speakers.
- Background voices (indistinct chatter, crowd noise) should be described as environmental noise (e.g., "the crowd hummed") and do not require placeholders.
- Avoid using indicative phrases like "The audio contains".

Prompts for Sound Caption Enhancement The user will input an audio clip and its original caption. Please generate 5 alternative diverse audio captions. Separate different captions with "`\t`".

- If there are multiple audio events, please describe them in order, but avoid specific timestamps.
- If the audio contains speech, just consider it as a sound event and there is no need to describe the content of the speech.

Prompts for Speech Caption Enhancement The user will provide a set of tags describing a speech recording. There is no actual audio file. Write five distinct, one-sentence descriptions for this speech solely on the tags. Separate different sentences with "`\t`".

- Use neutral narrative verbs like "says", "speaks", avoid performance-based words like "recites" or "delivers". Do not use technical terms like "energy", "pitch" as nouns.
- Vary the sentence structures. Do not feel obligated to use every tag in every sentence.
- Always include the placeholder "`<speech content>`" as the spoken text, formatted like dialogue or a statement.
- The setting is a quiet environment, weave this naturally into some of the sentences (e.g., "In the silence of the room", "There's no background sound." or "In a quiet space").

A.2 Impact of Data Curriculum

We compare three data curriculum strategies (Tab. 7) to investigate their impact on multi-scenario learning: (1) *Constant*: A fixed, equal ratio of all scenarios; (2) *Gradual*: Progressively increasing the proportion of speech-and-sound data; and (3) *Decoupled*: Sequential training from single-scenario to composite audio. All variants follow the main experimental configuration, but are trained for 50 epochs on a one-third data subset. To facilitate cross-category comparison, we report a Normalized Score, calculated by averaging min-max normalized metrics within each scenario. A higher Normalized Score indicates superior overall performance in a scenario. Evaluations are

Table 7: Data sampling proportions across the three training stages (early, middle, and final). Ep. denotes the number of training epochs per stage. Values indicate Sound/Speech/Speech-and-Sound sampling weights.

Strategy	Early (Ep. 0-10)	Middle (Ep. 10-25)	Final (Ep. 25-50)
Constant	0.33/0.33/0.33	0.33/0.33/0.33	0.33/0.33/0.33
Gradual	0.40/0.40/0.20	0.40/0.20/0.40	0.25/0.25/0.50
Decoupled	0.50/0.50/0.00	0.50/0.50/0.00	0.00/0.00/1.00

implemented at epochs 10, 25, and 50. Fig. 3 illustrates several key findings: (1) **Robustness of Constant Balancing.** *Constant* generally outperforms other strategies, suggesting that a uniform data curriculum is vital for robust unified generation without favoring specific categories at the expense of others. This also demonstrates that our framework does not require meticulous ratio tuning for convergence. (2) **Catastrophic Forgetting in Decoupled Sequential Training.** In *Gradual* and *Decoupled*, sound and speech performance degrades as speech-and-sound data increases. This is most acute in *Decoupled*, where the abrupt shift to exclusive speech-and-sound training triggers significant forgetting, emphasizing the necessity of continuous multi-scenario exposure. (3) **Synergistic Transfer from Speech-and-Sound.** *Constant* and *Gradual* outperform *Decoupled* in sound/speech even when *Decoupled* focuses solely on those scenarios. This reveals a positive transfer effect: mastering speech-and-sound scenarios, which requires decoupling complex acoustic components in prompts, forces the model to learn more discriminative features, thereby accelerating proficiency in single-scenario generation.

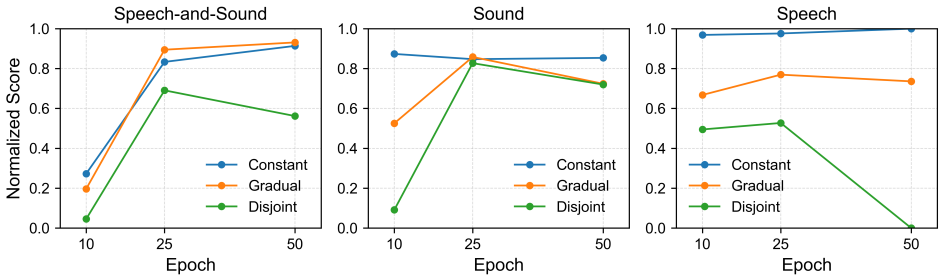


Figure 3: Normalized Score of data curriculum strategies across training epochs. Each sub-figure presents results for a scenario. This score averages strategy-wise normalized metrics; higher values indicate superior performance. Constant leads, proving uniform curricula ensure generalization.

A.3 Impact of Data Augmentation

To investigate the impact of annotation diversity, we compare two training settings: 1) *Full Annotation*: All 5 annotations per clip. 2) *Single Annotation*: One annotation per clip. Both variants follow the main experimental setup, but are trained for 25 epochs on a one-third data subset. As shown in Tab 8, *Full Annotation* consistently achieves superior semantic alignment and acoustic quality. This suggests that exposing the model to varied phrasing and multiple annotations enhances generalization to Free-Form prompts and mitigates overfitting to specific scenarios.

Table 8: Impact of annotation diversity. Comparison between using five text annotations (Full) versus one (Single) per clip throughout training. Full annotation consistently yields superior performance.

Setting	Composite					Sound			Speech	
	FD ↓	KL ↓	CLAP ↑	WER ↓	UTMOS ↑	FD ↓	KL ↓	CLAP ↑	WER ↓	UTMOS ↑
Full Annotation	234.89	1.07	0.17	0.81	2.21	251.87	1.94	0.16	0.36	3.03
Single Annotation	245.13	1.07	0.16	0.85	2.21	273.66	2.06	0.12	0.44	2.98

A.3.1 Annotation Quality of Speech-and-Sound Subset in FUAG-Bench

Since speech-and-sound data is annotated automatically, we verify the reliability through human inspection. Three annotators independently rated a random subset of 91 items, yielding 156 judgments in total. Each judgment is binary and assesses whether the prompt faithfully describes the paired audio along three dimensions: the accuracy of the speech transcription, the accuracy of the non-linguistic description, and the correctness of the stated temporal relations. Temporal relations are judged as categorical event ordering rather than millisecond-level boundary alignment, matching the granularity at which the prompts are written. Scores are averaged over all judgments, and an item counts as overall correct only when all three dimensions are satisfied.

Table 9: Human verification of Speech-and-Sound annotations in FUAG-Bench(156 judgments on 91 randomly selected items).

Dimension	Accuracy ↑
Speech Transcription	98.1%
Sound Description	99.4%
Temporal Relation	100.0%
Overall	96.8%

Accuracy is high across all three dimensions, indicating that the decoupled annotation pipeline produces prompts that can serve as reliable references for joint audio evaluation.

A.3.2 Interaction Type Coverage of Speech-and-Sound Subset in FUAG-Bench

To characterize what kinds of speech-and-sound interaction the benchmark covers, we label all 4,500 speech-and-sound samples with a five-way taxonomy using Gemini-2.5 Pro. Each sample receives a single label under a fixed priority rubric, so samples exhibiting more than one interaction are assigned to the highest-priority category rather than counted multiple times.

Table 10: Distribution of interaction types across the 4,500 speech-and-sound subset of the FUAG-Bench. Each sample is assigned a single label under a fixed priority rubric.

Interaction Type	Count	Ratio (%)
Sequential ordering	605	13.4
Overlapping speech–sound	698	15.5
Background ducking / loudness adaptation	739	16.4
Causal trigger	1,198	26.6
Rhythmic alignment	1,260	28.0
Total	4,500	100.0

The distribution is broad, with no category falling below 13%, so the benchmark does not reduce to a single dominant interaction pattern. Within the sequential ordering category, the number of ordered events ranges from two to six, with two-event cases accounting for 49.9%; the remainder therefore requires the model to maintain longer temporal chains.

A.4 Ablation Study

Full-scale training is costly, so all ablation experiments below are conducted under a common reduced-scale setting: 15 epochs on a 100k subset with an average data sampling proportion unless stated otherwise. Since this budget is considerably smaller than that of the main experiments, the absolute values are therefore not comparable to the main results; only the relative ordering is meaningful.

A.4.1 Planning Horizon

The planning horizon K determines how many latent anchors the model emits before entering acoustic generation. To justify our choice, we compare $K \in \{3, 6, 12\}$ under an identical setup. Results are shown in Tab. 11.

Table 11: Ablation on the planning horizon K. All variants are trained for 15 epochs on a 100k subset. KL is computed via the PaSST classifier.

	Speech-and-Sound		Sound	Speech
Variant	KL ↓	WER ↓	KL ↓	WER ↓
$K = 3$	1.15	1.21	2.49	1.11
$K = 6$ (PlanAudio)	1.12	1.14	2.17	1.10
$K = 12$	1.10	1.14	2.22	1.08

Increasing K from 3 to 6 improves speech-and-sound intelligibility (WER $1.21 \rightarrow 1.14$) and sound fidelity (KL $2.49 \rightarrow 2.17$), indicating that three anchors are too coarse to represent the event structure of a speech-and-sound scene. Enlarging the horizon further to $K = 12$ yields no consistent gain while doubling the length of the planning phase, so we adopt $K = 6$ as the setting that attains the same quality at lower inference cost.

A.4.2 Fixed Planning Horizon and Variable-Length Audio

A fixed K constrains only the number of semantic anchors, not the duration of the generated audio. The two phases operate at different granularities: AF3Encoder features are extracted chronologically and pooled into K anchors that form a coarse temporal backbone, while the acoustic phase renders frame-level prosodic contours and event boundaries conditioned on that backbone. Audio length is decided in this second phase, which terminates on the <|eoa|> token, so prompts describing longer or denser scenes are decoded to correspondingly longer sequences without truncation.

We also examined whether the horizon should itself be predicted. Prior work on continuous latent reasoning in text LLMs reports that fixed- and variable-length chains perform comparably [28], and our findings agree in the audio setting: a variant that terminates the planning phase through a binary classifier underperformed the fixed-length configuration, as length prediction proved unstable to optimize and tended to terminate prematurely under dense acoustic mixtures. We therefore retain a fixed horizon throughout.

A.4.3 Sensitivity to the Loss Weight

We vary the weight λ_1 of the semantic latent regression loss while keeping the acoustic objective fixed at $\lambda_2 = 1.0$. Results are shown in Tab. 12.

Table 12: Sensitivity to the semantic latent loss weight λ_1, with the acoustic objective fixed at $\lambda_2 = 1.0$. KL is computed via the PaSST classifier.

	Composite		Sound	Speech
Variant	KL ↓	WER ↓	KL ↓	WER ↓
$\lambda_1 = 0.2$	1.14	1.19	2.62	1.11
$\lambda_1 = 1.0$ (PlanAudio)	1.12	1.14	2.17	1.10
$\lambda_1 = 5.0$	1.16	1.13	2.08	1.08

Weakening the semantic objective to $\lambda_1 = 0.2$ degrades sound generation appreciably (KL 2.62 vs. 2.17), confirming that the planning phase requires sufficient supervision to be useful. Beyond this point the model is largely insensitive to the setting: $\lambda_1 = 1.0$ and $\lambda_1 = 5.0$ differ only marginally and trade places across metrics, so PlanAudio does not depend on careful tuning of this hyperparameter. We adopt $\lambda_1 = 1.0$, which balances the two objectives without scaling up gradient magnitudes in the planning phase.

A.5 Multi-stream Token Modeling with Delay Pattern

In the context of multi-stream audio token modeling, the delay pattern [31] is a sequence interleaving strategy designed to handle parallel audio tokens from a multi-codebook audio codec model. Unlike the flattening pattern, which serializes all tokens into a long 1D sequence, the delay pattern maintains the original temporal resolution by introducing a fixed offset between codebook streams.

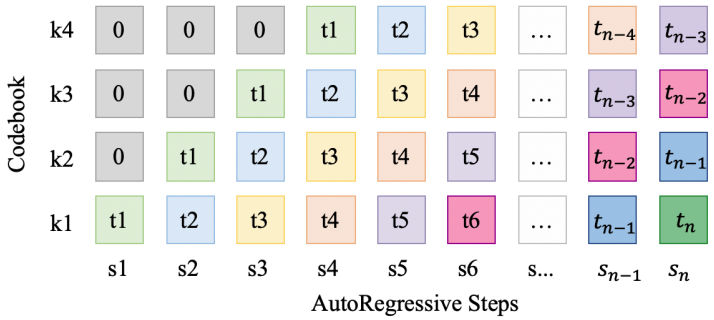


Figure 4: Delay sequence interleaving pattern. Each time step t_1, t_2, ..., t_n is composed of 4 quantized values (corresponding to k_1, ..., k_4). When doing autoregressive modelling, we interleave them and result in a new sequence with 4 parallel streams and steps s_1, s_2, ..., s_n. 0 indicates empty positions in the pattern.

As shown in Figure 4, for a codec with K codebooks, the k-th stream is shifted to the right by $k-1$ time steps. This data arrangement allows a single autoregressive Transformer to predict K tokens at each logical step through parallel linear output heads. In this configuration, the prediction of the k-th codebook at time t is conditioned on the previous $t-1$ steps of all streams, as well as the concurrent tokens of the $k-1$ higher-level codebooks due to the temporal shift. This mechanism effectively transforms the parallel codebook structure into a format compatible with autoregressive modeling without increasing the total number of transformer inference steps.

A.6 Balancing Reconstruction Quality and Generative Performance across Audio Domains

In this section, we present an experimental analysis of PlanAudio using different audio codecs. Since our task encompasses speech, sound, and composite scenarios, it is essential to select a universal codec model. Given that the reconstruction quality of a codec determines the performance ceiling of the generative model, we first evaluated the reconstruction quality of candidate codecs across various audio categories. The reconstruction results are summarized in Table 13.

Regarding reconstruction performance, among multi-layer codecs, AudioCraft and DAC exhibit lower frame rates alongside higher reconstruction quality, differing primarily in codebook size and quantity. Among single-layer codecs, UniCodec achieves an optimal balance between sound and speech reconstruction. Consequently, we conducted PlanAudio training experiments across these three codecs while keeping all other parameters identical. It is worth noting that due to the parallel delay pattern employed for multi-stream audio token prediction, there was no significant difference in training time per epoch among these configurations. The results of PlanAudio on various codecs are presented in Table 14.

Table 13: Comparison of reconstruction quality across various audio codecs and scenarios. The results demonstrate the performance ceiling of candidate codecs across sound, speech, and composite scenarios.

Codec	Nq	Codebook Size	Frame Rate	Composite					Sound			Speech	
				FD ↓	KL ↓	CLAP ↑	WER ↓	UTMOS ↑	FD ↓	KL ↓	CLAP ↑	WER ↓	UTMOS ↑
GT	-	-	-	0.00	0.00	0.17	0.10	2.69	0.29	0	0.29	-	-
Single-Layer Codec													
Xcodec	1	1024	50	**217.31**	**0.59**	0.19	0.46	2.56	**322.20**	1.48	0.10	0.11	3.26
Unicodec	1	8192	75	370.24	0.65	**0.23**	**0.27**	**2.60**	408.13	**1.31**	0.16	**0.04**	3.57
Wav_Tokenizer	1	4096	75	533.90	1.01	0.20	0.35	2.53	608.10	2.24	0.06	0.05	**3.67**
Multi-Layer Codec													
Audiocraft	4	2048	50	112	0.37	0.21	0.21	2.43	130	0.48	0.23	0.04	3.13
Encodec	8	1024	75	198	0.40	0.22	0.19	2.45	252	0.47	0.32	0.03	3.45
Xcodec	8	1024	50	155	0.34	**0.34**			285	0.95	0.18		
DAC	12	1024	50	**83.8**	**0.23**	0.22	**0.12**	**2.65**	**96.8**	**0.29**	0.25	0.03	**3.58**

Table 14: Generative performance of PlanAudio across different audio codecs and planning configurations. The results are evaluated across composite, sound, and speech scenarios.

model	Composite					Sound			Speech	
	FD ↓	KL ↓	CLAP ↑	WER ↓	UTMOS ↑	FD ↓	KL ↓	CLAP ↑	WER ↓	UTMOS ↑
DAC-Planning	429	1.64	0.19	0.81	2.72	583	3.78	0.00	0.16	3.40
DAC-w/o Planning	484	2.37	0.13	0.84	2.11	622	4.78	-0.05	0.40	3.23
UniCodec-Planning	415	1.11	0.23	0.35	2.81	477	1.87	0.20	0.06	3.61
UniCodec-w/o Planning	472	1.37	0.22	0.43	2.59	494	2.83	0.08	0.08	3.59
AudioCraft-Planning	180	1.05	0.21	0.47	2.49	233	1.89	0.19	0.11	3.11
AudioCraft-w/o Planning	252	1.38	0.18	0.57	2.35	281	3.10	0.04	0.21	3.04

From the perspective of Plan, the Planning mechanism consistently yields significant improvements over the version w/o Plan for every codec, demonstrating the robustness of our approach. From the codec perspective, although DAC achieves the best reconstruction quality, it performs the worst in terms of actual generation. We attribute this phenomenon primarily to its excessive number of codebooks, which makes it difficult for the model to learn the underlying patterns. For UniCodec and AudioCraft, by comparing the generation performance in Table 14 with the reconstruction performance in Table 13, we observe a strong correlation between the two. UniCodec, which performs better in speech reconstruction, also excels in speech generation. Similarly, AudioCraft, which is adept at sound reconstruction, shows consistent performance during generation.

Given the current results, we have yet to find a single codec that perfectly balances reconstruction quality and generation performance across all audio domains; thus, a trade-off is inevitable. Ultimately, we selected AudioCraft as the discrete audio tokenizer for PlanAudio.

A.7 The Semantic Coverage Factor

The Semantic Coverage Factor (SCF) quantifies semantic omission, i.e. the extent to which events named in the prompt are actually realized in the generated audio. We first filter out low-probability background noise from the generated audio, then pair each detected event with the events named in the prompt. Every pair contributes the cosine similarity of the two, weighted by the model's confidence in the detected event, and only pairs whose similarity exceeds a threshold t are counted. The accumulated score is normalized by the number of ground-truth events, so a value of 1.0 indicates that every prompted event is realized with high confidence.

A.8 Details about Human Study

In our human study, each of the 50 samples was rated by 10 raters across four models and four criteria, yielding 8,000 observations in total. We provide the instructions for the subjective evaluation:

This evaluation contains 50 samples in total. Each subfolder includes audio generated by four different models from the same text prompt, along with a textual description provided in description.txt. The correspondence between model names and audio files has been randomly shuffled.
Please listen to the audio samples and read the text description, then rate each sample according to the following four dimensions.
Scoring Criteria:

- Audio Quality: Score range: 0–5, with 0.5-point increments. Higher scores indicate better audio quality.
- Temporal Correctness: Evaluate whether the ordering of audio events matches the temporal sequence described in the text. Higher consistency receives higher scores. Pay particular attention to temporal cues such as “after”, “followed by”, etc.
- Semantic Alignment: Score range: 0–5, with 0.5-point increments. Evaluate how completely the events mentioned in the text are reflected in the audio. More comprehensive semantic coverage receives higher scores. For example, if the text mentions bird chirping and human speech, both should be present in the generated audio.
- Audio Authenticity: Score range: 0–5, with 0.5-point increments. Evaluate whether the environmental sounds and speech are naturally integrated, rather than sounding like a simple overlap of two separate audio tracks. Higher scores indicate more realistic and coherent audio.

Note: When rating a specific dimension, please focus only on the corresponding criterion without considering other evaluation dimensions.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The claims regarding the proposed PlanAudio framework, the semantic latent planning mechanism, and the FUAG-Bench are supported by the methodology in Section 3 and experimental results in Section 4.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss limitations in the Conclusion, specifically addressing the dependency on neural codec reconstruction quality.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: This paper is empirical in nature and does not present new theoretical theorems or formal proofs.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide detailed descriptions of the network architecture (Section 3), implementation details and data construction pipeline (Section 4, Appendix A.1).

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We have released our benchmark repository on GitHub, and the code will be made publicly available upon publication.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Core training settings (learning rate, batch size, etc.) are detailed in Section 4

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: For subjective evaluations (MOS), we report 95% confidence intervals following standard practice.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We specify the hardware and the approximate total training time in Section 4.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Our research follows the ethics guidelines. We ensure the generated content does not violate safety policies.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss both potential positive societal impacts and negative societal impacts in Section 5.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [Yes]

    Justification: We will add a usage agreement after the model release.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All the resources used in this paper are publicly available, and we have provided proper citations for them.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We provide our FUAG-Bench.

    Guidelines:

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

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

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

    Answer: [Yes]

    Justification: We provide detailed information of subjective evaluation in Section A.8

    Guidelines:

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

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

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

    Answer: [No]

    Justification: The study involved anonymous voluntary evaluations and did not require formal IRB approval under our institutional guidelines.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: Large language models are used as a core component of the proposed framework for semantic reasoning and latent chain-of-thought modeling.

    Guidelines:

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