AudioCALM: Continuous Autoregressive Language Modeling for Universal Audio Generation

Huadai Liu[1,2], **Kaicheng Luo**[2], **Wen Wang**[2], **Qian Chen**[2]
Bin Ma[2], **Xiangang Li**[2], **Wei Xue**[1] *
[1]Hong Kong University of Science and Technology (HKUST)
[2]Alibaba Token Hub, Alibaba Group

Abstract

Unifying speech, sound, and music generation in one model is hindered by tradeoffs between fidelity, end-to-end training, in-context conditioning, and variable-length synthesis that no current paradigm fully resolves. To address this challenge, we present **AudioCALM**, a universal audio generation framework that extends autoregressive (AR) next-token prediction from discrete tokens to continuous audio latents: a thin flow-matching head replaces the softmax to predict rectified-flow velocities at each position, and a block-causal AR-Flow attention pattern produces arbitrary-length output. Joint training of multiple audio generation tasks faces an asymmetric text–audio mismatch: speech transcripts align to specific time spans and demand *tight, time-aligned* attention, whereas sound and music captions describe only overall semantics and rely on *diffuse, holistic* attention; mixing the two disproportionately degrades sound and music generation. We address this asymmetry at two levels: a **data reformulation** strategy that unifies all three tasks under a single description-style conditioning interface, and a novel architecture **Asymmetric Mixture-of-Modality-Experts (A-MoME)**, which adds a dedicated residual expert for speech while sound and music share the backbone, incurring no inference overhead on non-speech inputs. Experimental results demonstrate that AudioCALM matches modality-specific state-of-the-art and outperforms prior unified baselines on speech, sound, and music generation benchmarks. The project page is at `https://AudioCALM-Project.github.io`.

1 Introduction

One long-standing goal of audio generation is building a universal model that could synthesize a wide range of audio types (that is, modalities or domains), such as speech, music, and environmental sound, from natural-language conditioning in a unified manner: real-world audio rarely occurs in isolation, and a unified model can in principle leverage data and shared structure and knowledge across various audio domains, hence reducing development cost and improving generation performance. In practice, however, the field remains dominated by domain-specific systems built on three generative paradigms: discrete-token autoregressive (AR) modeling over neural audio codecs [Wang et al., 2023a, Copet et al., 2023, Kreuk et al., 2022, Borsos et al., 2023], cascaded language-model-then-acoustic-generator pipelines [Du et al., 2024, Wang et al., 2024, Ju et al., 2024], and non-autoregressive (NAR) latent diffusion or flow matching over continuous representations [Liu et al., 2024a, Evans et al., 2025, Le et al., 2023, Mehta et al., 2024]. Each of these paradigms imposes structural constraints on its generative interface, limiting how uniformly a single instantiation can support text-to-speech (TTS), text-to-sound, and text-to-music (T2M) tasks. Note that "Audio" is our umbrella term for speech, sound, and music, and T2A denotes the text-to-non-speech regime (sound and music together). On

*Corresponding author.

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

the other hand, truly unified attempts spanning all three modalities are far fewer, concentrated on autoregressive language modeling over discrete audio tokens [Yang et al., 2023, 2024] and non-autoregressive flow matching with task-specific input formatting [Vyas et al., 2023, Xu et al., 2025b], with the cascaded paradigm comparatively less explored under the same unified setting.

However, all three paradigms suffer from severe limitations. Specifically, discrete-token autoregression [Wang et al., 2023a, Kreuk et al., 2022, Borsos et al., 2023] imposes its constraint by compressing audio into a finite codebook, an information bottleneck especially restrictive for high-bandwidth content such as polyphonic music and environmental sound, where token-based methods trail continuous-latent approaches [Liu et al., 2024a, Evans et al., 2025] in performance. Cascaded systems sidestep discretization by handing off to a separate diffusion or flow-matching module [Du et al., 2024, Wang et al., 2024], but the hand-off introduces a rigid intermediate representation and severs end-to-end optimization of the language model and the acoustic generator. Non-autoregressive flow matching models avoid both issues, but forgo the autoregressive in-context conditioning over previously generated content, and depend on external duration prediction or forced alignment for variable-length generation [Ju et al., 2024, Vyas et al., 2023]. These limitations motivate four desiderata for universal audio generation—*fidelity*, *end-to-end optimization*, *in-context understanding*, and *natural variable-length generation*—which no existing paradigm fully satisfies.

We address this gap with **AudioCALM**, a universal audio generation framework instantiating *Continuous Autoregressive Language Modeling (CALM)*: a paradigm that extends autoregressive next-token prediction from discrete tokens to continuous audio latents while retaining the autoregressive language-model backbone. A single LM processes a unified text–audio sequence, and at each audio position, a thin *flow-matching head* on the LM hidden state predicts a rectified-flow velocity over the continuous latent space—the continuous counterpart of the softmax in standard next-token prediction. The two components share one latent space and are jointly trained under a single rectified-flow objective, removing any intermediate discretization. Consequently, different from all three paradigms, **AudioCALM retains the LM interface of discrete-token autoregressive models, the high-fidelity continuous representation of non-autoregressive flow matching models, and the end-to-end optimization that cascaded LM-then-generator systems sacrifice**. To support streaming arbitrary-length generation, we further introduce **AR-Flow**, a novel block-causal attention pattern in which each latent block is generated by flow matching while attending autoregressively to all preceding blocks and the full text condition, thus variable length becomes a property of the attention mask rather than a predefined hyperparameter.

Within this unified architecture, text still plays substantially different roles across tasks: TTS exhibits largely *local* text–audio correspondence—transcript segments align to specific time spans—while T2A is largely *global*, with captions fixing only overall semantics. Therefore, speech favors sharp, locally aligned text–audio attention, while sound and music favor diffuse, globally distributed text–audio attention. Our joint-training experiments reveal an asymmetric mismatch (Table 3): adding speech to the training mixture degrades sound and music generation more than sound and music degrade speech generation, which is consistent with speech's stricter alignment crowding out the dispersed attention that non-speech audio depends on. Symmetric designs that allocate capacity uniformly across modalities, therefore, fail to match the structure of this mismatch.

We address this asymmetry at two levels. At the *data* level, we adopt a single description-style conditioning interface across all three modalities. An audio-conditioned multimodal LLM (MLLM) consumes each training clip together with its bare textual annotation—the transcript for speech, the short caption for sound and music—and emits a long-form description that surfaces modality-relevant attributes from the waveform as natural-language phrases: for speech, speaker timbre, fine-grained prosody, and acoustic context (e.g., reverberation, background noise, recording quality); for sound, acoustic events and their sources, scene, and event sequencing; for music, genre, instrumentation, tempo, mood, and recurring motifs. Under this interface, TTS, text-to-sound, and text-to-music all reduce to conditional next-block prediction over audio latents under one model, and zero-shot voice cloning is supported natively by prepending a speaker embedding as a global prefix. At the *architecture* level, the residual parameter-level mismatch is handled by **Asymmetric Mixture-of-Modality-Experts (A-MoME)**, an asymmetric variant of modality-expert designs [Bao et al., 2022, Wang et al., 2023b, Liang et al., 2024]: self-attention and the main feed-forward network remain shared across speech, sound, and music, while a speech-specific residual expert is added in parallel to capture the tighter text–audio correspondence required by speech alone. The asymmetry, rather than the modality count, defines the design: sound and music share the backbone unchanged, and

only the modality whose objective diverges from the others receives dedicated capacity. Notably, the integration of A-MoME layers into AudioCALM does not modify the flow-matching head or the AR-Flow attention pattern, and incurs no inference overhead on non-speech inputs.

Our contributions are summarized as follows:

- We propose **AudioCALM**, a universal audio generation framework that extends next-token prediction to continuous audio latents via a flow-matching head, paired with a block-causal AR-Flow attention pattern that yields arbitrary-length speech, sound, and music generation in a single model.
- We identify an asymmetric local–global text–audio mismatch in joint training and address it at two levels: a data-level recasting of speech as a special case of audio that unifies the three modalities under one description-style conditioning interface, and an architecture-level innovation of **A-MoME**, which mirrors this asymmetry by adding a dedicated residual expert for speech alone while keeping sound and music to share the backbone, allocating capacity by demand rather than by modality.
- AudioCALM achieves strong performance across speech, sound, and music generation benchmarks, outperforming the compared unified models and matching or exceeding modality-specific systems on most reported metrics under our evaluation protocol.

2 Related Work

Unified Audio Generation We use *unified audio generation* to refer to a single model whose *output modality* jointly covers speech, music, and sound—replacing the historically separate TTS, T2A, and T2M systems—with text as the primary conditioning *input*, optionally augmented with reference-audio conditioning for voice cloning or style imitation. Here the same model performs these tasks separately; within-sample speech–sound–music composition is outside the evaluated scope. Existing output-unified frameworks fall along three architectural lines. The first category is discrete-token autoregressive modeling, exemplified by VALL-E [Wang et al., 2023a] for TTS, MusicGen [Copet et al., 2023] for T2M, AudioGen [Kreuk et al., 2022] for T2A, and the unified UniAudio [Yang et al., 2023, 2024] in the audio language model (AudioLM) family [Borsos et al., 2023], which tokenizes audio with a neural codec [Défossez et al., 2022, Kumar et al., 2023] and casts every task as next-token prediction; this yields a clean LM interface, but the codec bottleneck caps fidelity on polyphonic music and complex sound generation [Liu et al., 2024a, Evans et al., 2025, Liu et al., 2024c,b]. The second category is the cascaded LM-then-generator family, such as CosyVoice2 [Du et al., 2024] and MaskGCT [Wang et al., 2024], in which an LM emits semantic tokens for a downstream diffusion or flow-matching synthesizer, removing the fidelity ceiling but freezing the intermediate semantic code as a fixed contract between two separately optimized stages, which blocks end-to-end optimization. The third category is non-autoregressive flow matching [Liu et al., 2025a,b,c], e.g., Audiobox [Vyas et al., 2023], Stable Audio Open [Evans et al., 2025], and the recent UniFlow-Audio [Xu et al., 2025b], which conditions flow-matching networks on task-specific input formatting and attains strong fidelity, but generates in parallel and thereby requires an externally specified target length and drops the token-level in-context conditioning of AR models [Ju et al., 2024]. Orthogonal to the three output-unified lines above, a related strand of speech-output universal models—early speech-only LMs [Zhang et al., 2023, Rubenstein et al., 2023, Défossez et al., 2024] and recent omni-modal audio LMs [Ding et al., 2025, Xu et al., 2025a, Zhang et al., 2025, Team et al., 2025]—ingests speech, music, and sound as *input* for understanding but produces only speech (and text) as *output*, leaving music and sound *generation* unaddressed; concurrent work [Liu et al., 2025d] tackles the resulting cross-modal interference via mixture-of-experts in a unified speech–music generator, yet still excludes general sound.

Audio Language Models Among the three paradigms above, the AR LM family uniquely affords the standard LLM toolbox—streaming inference, in-context conditioning, and scalable training—yet discrete-token AR is bounded by lossy codec reconstruction and intricate multi-codebook residual-quantization decoding patterns, while continuous-latent diffusion and flow matching [Le et al., 2023, Liu et al., 2022, 2023] avoid the codec ceiling only by sacrificing the streaming and token-level in-context behavior of AR models. A more recent line preserves the LM interface while removing the discrete bottleneck, as in GIVT [Tschannen et al., 2024] and MAR [Li et al., 2024] in vision. In speech, DiTAR [Jia et al., 2025] processes aggregated patch representations with a causal LM and conditions a separate local diffusion Transformer on the LM output to generate each patch. Block

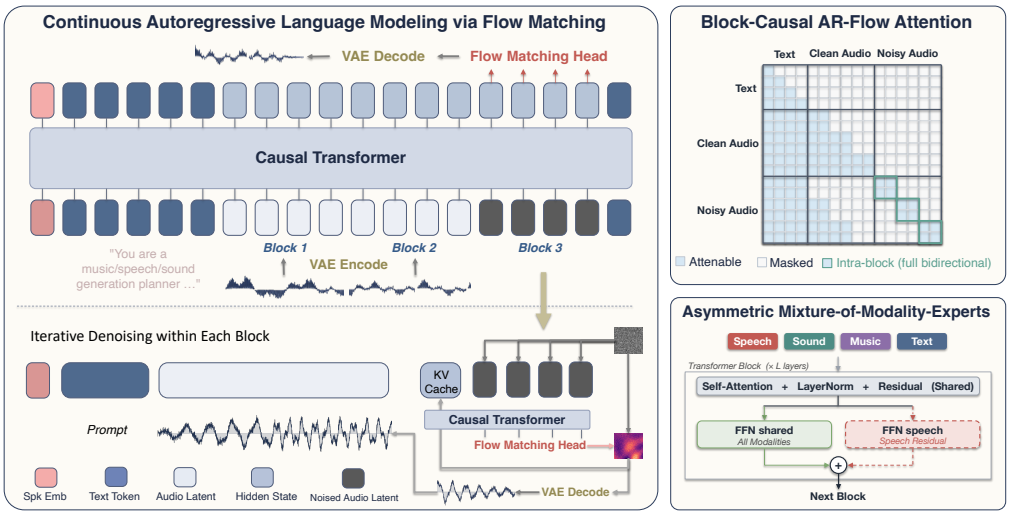


Figure 1: **Overview of AudioCALM.** *Left:* a causal Transformer autoregresses over fixed-size blocks of continuous audio latents; a flow-matching head and a stop head are attached to its hidden states (top), and each block is produced by iterative denoising with KV-cache reuse (bottom). *Top right:* the block-causal AR-Flow attention mask, causal across blocks and bidirectional within the active noisy block (red). *Bottom right:* the Asymmetric Mixture-of-Modality-Experts layer, in which a deterministically routed speech-only FFN is added to the shared backbone.

Diffusion [Arriola et al., 2025] also combines autoregression across blocks with joint denoising within each block, but models discrete text tokens. AudioCALM shares this blockwise factorization while learning rectified-flow velocities over continuous audio latents through the LLM itself and a single linear output projection. AudioCALM extends this continuous-latent AR line from the speech-only DiTAR to universal audio generation. Beyond this domain extension, AudioCALM further differs from DiTAR in three respects: (i) we directly use the LLM itself as the denoiser across speech, sound, and music; (ii) we introduce **A-MoME** to resolve the cross-modal local–global text–audio mismatch that does not arise in DiTAR's single-modality setting; and (iii) we recast speech as a special case of audio under a unified description-style conditioning interface, so that TTS, T2A, and T2M reduce to one conditional next-block prediction problem within a single model.

3 AudioCALM

3.1 Overview

Figure 1 illustrates the AudioCALM architecture. Section 3.2 introduces the flow-matching head that extends next-token prediction from a discrete softmax to a continuous rectified-flow velocity over VAE latents. Section 3.3 develops the block-causal AR-Flow attention pattern that couples autoregressive commitment across blocks with bidirectional flow matching within a block, supporting streaming variable-length generation. Section 3.4 addresses the speech–audio asymmetry through a description-style conditioning pipeline and the Asymmetric Mixture-of-Modality-Experts (A-MoME). Section 3.5 specifies the training objective, exposure-bias regularizers, and autoregressive inference procedure.

3.2 Continuous Autoregressive Language Modeling via Flow Matching

Sequence formulation. A training example pairs a textual condition c with continuous audio latents $x = (x_1, \ldots, x_L) \in \mathbb{R}^{L \times C}$ from a frozen VAE [Kingma and Welling, 2013, Evans et al., 2025], with L latent positions and C channels per position. For contiguous blocks $\mathcal{B}_1, \ldots, \mathcal{B}_K$, generation factorizes as $p(x \mid c) = \prod_{k=1}^{K} p(x_{\mathcal{B}_k} \mid x_{\mathcal{B}_{<k}}, c)$, where $x_{\mathcal{B}_{<k}}$ contains all preceding blocks and the flow jointly models the positions within the current block. For speech, c is preceded by a single speaker embedding $s \in \mathbb{R}^{D_s}$ from a frozen speaker encoder (Spk Emb in Fig. 1). Text is embedded

via the LM's native token embedding; for each audio position i, a noisy latent $x_i^{(t)}$ at flow-matching timestep $t \in [0, 1]$ enters the LM through a linear projection $\phi_{\text{in}} : \mathbb{R}^C \to \mathbb{R}^H$, summed with a learned position embedding p_i and a sinusoidal timestep embedding $\tau(t)$,

$$e_i^{(t)} = \phi_{\text{in}}\big(x_i^{(t)}\big) + p_i + \tau(t), \tag{1}$$

where $t = 0$ marks a clean latent and $t \in (0, 1]$ a noisy one. Text and audio embeddings are concatenated into a text-prefix/audio sequence and processed by a shared transformer backbone f_θ initialized from a pretrained text LLM, yielding hidden states $h_i \in \mathbb{R}^H$.

Flow-matching head and training objective. At every audio position, a linear projection

$$v_\theta(h_i) := \phi_{\text{out}}(h_i) \in \mathbb{R}^C \tag{2}$$

maps the hidden state to a velocity in the same latent space as x_i, providing a per-position component of the current block's joint velocity field v_θ. We zero-initialize ϕ_{out} so the velocity field starts at zero, preventing the untrained head from disturbing the backbone in early training. Following rectified flow [Liu et al., 2022] with a logit-normal timestep schedule [Esser et al., 2024], we draw $t = \sigma(u)$ with $u \sim \mathcal{N}(0, 1)$ and σ the logistic sigmoid, concentrating probability mass near $t \approx 0.5$ where denoising is hardest. With per-position noise $\epsilon_i \sim \mathcal{N}(0, I_C)$, the linear interpolation and target velocity are

$$x_i^{(t)} = (1 - t)\, x_i + t\, \epsilon_i, \qquad v_i^\star := \epsilon_i - x_i, \tag{3}$$

and the training loss, averaged over the set $\mathcal{N}$ of noisy audio positions in the example, is

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{x,\, c,\, u,\, \epsilon}\left[\frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} \big\| v_\theta(h_i) - v_i^\star \big\|_2^2\right]. \tag{4}$$

The same projection ϕ_{in} embeds clean prefix latents and noisy targets. After denoising, the generated latent is re-embedded as clean context for subsequent blocks; accumulated context error is addressed by the regularizers in Section 3.5.

3.3 Block-Causal AR-Flow Attention

Combining autoregression with flow matching raises a structural question: causal attention serves AR commitment, while flow matching requires bidirectional access among the noisy latents being denoised together. AR-Flow resolves this by attending causally across blocks and fully within each block, yielding a single mask under which a flow-matching trajectory denoises one block while the LM commits the previous one.

Block-causal mask. Let $\tilde{x}_i^{(t)}$ denote a noisy latent at timestep t, and call a clean latent *committed* once it has been denoised and written to the KV cache. At each generation step, the LM processes a packed sequence corresponding to one active noisy block (illustrated in Fig. 1 top-right):

$$\big[\, c_1, \ldots, c_T,\ x_1, \ldots, x_i,\ \tilde{x}_{i+1}^{(t)}, \ldots, \tilde{x}_{i+B}^{(t)} \,\big], \tag{5}$$

formed by a T-token textual condition, i committed clean latents, and a contiguous block of B noisy latents sharing a block-level timestep t. The *AR-Flow* mask enforces three rules:

- every text token c_j attends only to text tokens $c_{\le j}$;
- every clean audio latent x_k ($k \le i$) attends to all text tokens and to clean latents $x_{\le k}$;
- every noisy latent $\tilde{x}_m^{(t)}$ ($i < m \le i + B$) attends to the entire textual condition, all committed clean latents $x_{1:i}$, and all noisy latents within its block, but not to any other noisy block.

The mask is therefore causal across blocks and full within each block: the bidirectional intra-block context is what flow matching needs to denoise the block jointly, while the inter-block causality lets the cached prefix grow incrementally as blocks commit, so the model accommodates arbitrary length under the same mask at training and inference.

Single-pass teacher-forced training. A naïve realization would require one forward pass per noisy block, scaling linearly with the number of blocks per utterance and forfeiting the parallelism that makes LM training efficient. We instead collapse all blocks into a single pass over the extended packed sequence

$$\big[\, c,\ x_1, \ldots, x_L \;\big|\; \tilde{x}_1^{(t_1)}, \ldots, \tilde{x}_L^{(t_L)} \,\big], \tag{6}$$

whose right half holds per-position noisy copies of the audio. Let $\beta(m)$ denote the index of the noisy block containing position m and $b_s(m)$ its first position. Each $\tilde{x}_m^{(t_m)}$ attends to c, to the clean prefix $x_{1:\, b_s(m)-1}$ *strictly preceding* its own block, and to all noisy positions in the same block. No noisy position attends to its clean counterpart inside its own block, matching the target block's conditioning pattern in Eq. 5. Noisy copies use the same positional indices as their clean counterparts under RoPE [Su et al., 2024]. Timesteps are sampled independently once per block. The loss in Eq. 4 is computed on the noisy half, while the clean half supplies teacher-forced prefix states. This trains all target blocks in one forward pass over the extended sequence; generated-prefix drift remains a train–inference difference and is addressed in Section 3.5.

3.4 Modality-Asymmetric Specialization

The local–global text–audio mismatch identified in Section 1 manifests on two levels: a *conditioning* gap, with speech tightly aligned to a transcript while sound and music are described by global captions, and a *computation* gap, with speech demanding alignment-bound processing absent from the others. We close the former at the data level and the latter at the architecture level.

Description-style conditioning. Each clip is paired offline with two textual conditions. The *short* variant is the corpus annotation—transcript for speech, original short caption for sound and music. The *long-form* variant is produced by an audio-conditioned MLLM that consumes the clip together with the short variant and surfaces modality-relevant attributes the short variant cannot recover (speaker prosody and acoustic context for speech; acoustic events and scene structure for sound; instrumentation, tempo, and recurring motifs for music) as natural-language phrases grounded in the waveform; for speech, the verbatim transcript is wrapped in explicit content delimiters so local alignment is preserved. Both variants are sampled with equal probability per step, exposing the LM to terse and verbose prompts under one description-style interface usable with either at inference. The MLLM choice, prompt templates, decoding settings, and speech delimiter convention are in Appendix A.3.

Asymmetric Mixture-of-Modality-Experts. Inspired by modality-expert designs that share self-attention while specializing the feed-forward sub-layer per modality [Bao et al., 2022, Wang et al., 2023b, Liang et al., 2024], A-MoME keeps self-attention, layer normalization, residual connections, *and* the main feed-forward network shared, introducing a single residual expert active only on speech positions. Let h denote the hidden state at any token position and $m(h) \in \{\texttt{speech}, \texttt{sound}, \texttt{music}, \texttt{text}\}$ the modality tag; the modified feed-forward computation is

$$\mathrm{FFN}_{\text{A-MoME}}(h) = \begin{cases} \mathrm{FFN}_{\text{shared}}(h) + \mathrm{FFN}_{\text{speech}}(h), & m(h) = \texttt{speech}, \\ \mathrm{FFN}_{\text{shared}}(h), & \text{otherwise}, \end{cases} \tag{7}$$

where $\mathrm{FFN}_{\text{speech}}$ matches $\mathrm{FFN}_{\text{shared}}$ in form but has its own parameters, zero-initialized as with ϕ_{out} so A-MoME begins as a no-op and only departs from the shared backbone as training progresses. Modality tags are read from each position's source corpus, so the branch selection in Eq. 7 is deterministic with no gating network. Unlike symmetric modality-expert variants that allocate a separate FFN per modality, A-MoME adds only one extra FFN sub-layer per block, and non-speech inference is unaffected since sound, music, and text activate only $\mathrm{FFN}_{\text{shared}}$.

3.5 Training and Inference

Training objective. The total objective combines the rectified-flow loss in Eq. 4 on the noisy half of the extended sequence (Eq. 6) with a binary cross-entropy on the stop head (described below).

Closing the train–inference gap. In inference, each committed block is the denoised latent $\hat{x}$ from the previous step's flow-matching trajectory, re-embedded as a clean prefix for the next step. Drift from the true clean latent accumulates across blocks, opening a gap that pure teacher forcing does not capture. We close it with two complementary regularizers:

- **Per-block clean-prefix noise** (σ_{clean}). We perturb only the input embeddings $\phi_{\text{in}}(x_i)$ of clean prefix tokens with Gaussian noise; the noise scale ramps linearly from 0 at the first block to σ_{clean} at the last block of the utterance and is held constant within each block. The interpolation $x_i^{(t)}$ and target $v_i^\star$ remain computed from the unperturbed x_i, so the supervision is unchanged and the perturbation only simulates per-block drift in the cached prefix.
- **Exposure-bias perturbation** (γ_{exp}). For each sample we draw $\alpha \sim \mathcal{U}[0, \gamma_{\text{exp}}]$ and form a perturbed clean target $\tilde{x}_i = x_i + \alpha\,\xi_i$ with $\xi_i \sim \mathcal{N}(0, I_C)$. Both the prefix embedding and the target $v_i^\star = \epsilon_i - \tilde{x}_i$ are computed from $\tilde{x}_i$, training the model to recover from a slightly noisy commitment.

Autoregressive inference. AR-Flow accommodates arbitrary lengths by construction, so AudioCALM needs no external duration predictor. A lightweight binary stop head $g_{\text{stop}} : \mathbb{R}^H \to (0, 1)$ on the clean hidden states is trained with binary cross-entropy whose targets ramp linearly from 0 to 1 over the last K_{stop} tokens of each clip, so the model learns to anticipate clip end. At inference, each block is denoised by Euler integration of the learned velocity field [Liu et al., 2022] and committed to the KV cache; g_{stop} is then queried at every position of the new block, and the first position exceeding a fixed threshold τ_{stop} cuts the output, yielding token- rather than block-level termination granularity.

4 Experiments

4.1 Experimental Setup

Training Data. We jointly train AudioCALM on three corpora covering the universal audio setting. For speech, we combine LIBRITTS [Zen et al., 2019] with the English subset of EMILIA [He et al., 2024]; for general sound, we aggregate the audio of VGGSOUND [Chen et al., 2020], AUDIOCAPS [Kim et al., 2019], and WAVCAPS [Mei et al., 2024]; for music, we combine FMA [Defferrard et al., 2016] with MTG-JAMENDO [Bogdanov et al., 2019]. Only the official train splits are used; LibriTTS *test-clean*, the AudioCaps and Song-Describer evaluation sets, and SeedTTS-eval are held out to avoid test-set contamination. All audio is resampled to 44.1 kHz, and every clip is re-captioned with GEMINI 3 PRO[2] into long-form natural-language descriptions through the description-style pipeline of Section 3.4, with the original transcript or tag retained as a parallel short-form conditioning channel. Corpus details are in Appendix A.2, and the verbatim GEMINI 3 PRO prompts in Appendix A.3.

Evaluation Benchmarks. We follow standard evaluation protocols for each domain. Text-to-speech is evaluated on the full LIBRITTS *test-clean* split and the *en* split of SEEDTTS-EVAL [Anastassiou et al., 2024]; text-to-sound on AUDIOCAPS and text-to-music on the held-out SONG-DESCRIBER [Manco et al., 2023] captions.

Evaluation Metrics. We adopt both objective and subjective metrics. For speech, **WER** (Whisper-large-v3 [Radford et al., 2023]) measures intelligibility, and **SIM** (cosine similarity between WAVLM-LARGE [Chen et al., 2022] speaker-verification embeddings of the generation and the reference) measures speaker fidelity. For sound and music, **FAD** [Kilgour et al., 2018]—the Fréchet distance in VGGish [Hershey et al., 2017] embedding space—measures distributional fidelity, and LAION-**CLAP** [Wu et al., 2023] text–audio cosine similarity measures prompt adherence. Subjective evaluation uses **MOS** (overall quality) for speech and **MOS-Q/MOS-T** (quality, text relevance) for sound and music, on $N{=}50$ samples per system; full definitions and protocol are in Appendix B.

Baselines. We compare against modality-specific and unified baselines spanning the dominant generative paradigms in each domain. *Modality-specific:* F5-TTS [Chen et al., 2025] and

[2] `https://deepmind.google/technologies/gemini/`

Table 1: Text-to-speech evaluation on LIBRITTS *test-clean* and the *en* split of SEEDTTS-EVAL. MOS is reported as mean±std. Best per column in **bold**, second-best underlined.

	LibriTTS			SeedTTS-eval (en)		
Model	WER↓	SIM↑	MOS↑	WER↓	SIM↑	MOS↑
Modality-specific baselines						
F5-TTS	0.033	0.616	$3.85_{\pm 0.08}$	0.018	0.648	$3.78_{\pm 0.09}$
CosyVoice 3.0	<u>0.022</u>	**0.697**	<u>$3.96_{\pm 0.07}$</u>	0.015	**0.695**	<u>$3.88_{\pm 0.08}$</u>
Unified baselines						
UniAudio	0.120	0.265	$3.30_{\pm 0.11}$	0.113	0.363	$3.22_{\pm 0.12}$
UniMoE-Audio	0.078	0.361	$3.52_{\pm 0.09}$	0.019	0.573	$3.72_{\pm 0.08}$
UniFlow-Audio	0.032	0.570	$3.50_{\pm 0.10}$	0.058	0.573	$3.45_{\pm 0.10}$
Ming-omni-TTS	0.025	0.553	$3.82_{\pm 0.08}$	<u>0.013</u>	0.633	$3.80_{\pm 0.07}$
AudioCALM (ours)	**0.020**	<u>0.668</u>	$\mathbf{4.02_{\pm 0.06}}$	**0.011**	<u>0.672</u>	$\mathbf{3.95_{\pm 0.07}}$

COSYVOICE 3.0 [Du et al., 2025] for speech; AUDIOLDM 2-LARGE [Liu et al., 2024a], TANGOFLUX [Hung et al., 2024], and STABLE AUDIO OPEN [Evans et al., 2025] for sound; MUSICGEN-LARGE [Copet et al., 2023] and STABLE AUDIO OPEN for music. *Unified:* UNIAUDIO [Yang et al., 2023], UNIMOE-AUDIO [Liu et al., 2025d], UNIFLOW-AUDIO-LARGE [Xu et al., 2025b], and MING-OMNI-TTS-16.8B-A3B [InclusionAI, n.d.]. Tables 1–2 abbreviate the last two as UniFlow-Audio and Ming-omni-TTS. All baselines are evaluated from their official checkpoints under the recommended decoding configurations. These are complete-system benchmark comparisons; controlled attribution uses Table 3.

Implementation Details. The backbone is initialized from QWEN3-1.7B [Yang et al., 2025]; audio is encoded by our own continuous VAE on 44.1 kHz waveforms. We train for 300 k steps on A800×8 GPUs under FSDP HYBRID-SHARD with AdamW (β_1=0.9, β_2=0.95, weight decay 0) at a constant learning rate of 1×10^{-4} following 2,000 warmup steps, with a global batch of $\sim$64 k tokens (8 k tokens per GPU) and gradient clipping at 1.0, optimizing the rectified-flow objective. We apply 10% caption dropout for classifier-free guidance with scale w=3. Inference uses K_{flow}=24 rectified-flow steps per block and an AR-Flow block size of B=1.0 s, which corresponds to 11 VAE latent tokens at the bottleneck's $\approx$10.75 Hz rate (Appendix A). VAE architecture and training are in Appendix A, and the full hyperparameter ledger is in Appendix A.4.

4.2 Universal Audio Generation Results

Tables 1 and 2 compare AudioCALM against the two baseline families on the three target domains. Under the reported protocol, AudioCALM—a single set of weights—matches or exceeds the compared modality-specific systems on most metrics and outperforms the compared unified baselines. AR-Flow supports incremental latent generation without an external duration predictor; the current waveform-output implementation and its measured costs are detailed in Appendix C.8.

Zero-Shot TTS. Both benchmarks follow the zero-shot TTS protocol described in Appendix B, reporting WER (intelligibility), SIM (speaker fidelity), and MOS (overall quality). Among the systems in Table 1, AudioCALM achieves the lowest WER on LIBRITTS *test-clean* (0.020) and SEEDTTS-EVAL *en* (0.011), and the highest MOS means (4.02 and 3.95). Its SIM is second to CosyVoice 3.0 on both benchmarks (0.668 vs. 0.697 and 0.672 vs. 0.695). These results show competitive TTS within the same model used for sound and music generation.

Sound and Music Generation. On AUDIOCAPS, AudioCALM dominates every column, lowering FAD from the strongest sound generation specialist (TangoFlux, 2.70) to 1.95—a 28% relative reduction—and edging LAION-CLAP from 0.36 to 0.37. On SONG-DESCRIBER, AudioCALM likewise dominates: lowest FAD (2.02 vs. Stable Audio Open's 2.23) and highest text–audio CLAP (0.36 vs. 0.32). Both subjective axes (MOS-Q and MOS-T) also place AudioCALM first on every non-speech track.

Table 2: Text-to-sound (AudioCaps) and text-to-music (Song-Describer) evaluation. "–" marks domains a modality-specific model is not trained for; UniMoE-Audio additionally does not support text-to-sound.

	Text-to-sound (AudioCaps)				Text-to-music (Song-Describer)			
Model	FAD↓	CLAP↑	MOS-Q↑	MOS-T↑	FAD↓	CLAP↑	MOS-Q↑	MOS-T↑
Modality-specific baselines								
AudioLDM 2-Large	5.36	0.22	$3.25_{\pm 0.10}$	$3.10_{\pm 0.11}$	–	–	–	–
TangoFlux	2.70	0.36	$3.82_{\pm 0.07}$	$3.85_{\pm 0.08}$	–	–	–	–
Stable Audio Open	4.13	0.25	$3.65_{\pm 0.08}$	$3.45_{\pm 0.09}$	2.23	0.32	$3.95_{\pm 0.07}$	$3.85_{\pm 0.08}$
MusicGen-Large	–	–	–	–	5.28	0.19	$3.65_{\pm 0.08}$	$3.45_{\pm 0.10}$
Unified baselines								
UniAudio	6.64	0.13	$3.20_{\pm 0.11}$	$2.95_{\pm 0.13}$	11.25	0.06	$2.80_{\pm 0.14}$	$2.65_{\pm 0.15}$
UniMoE-Audio	–	–	–	–	3.71	0.22	$3.80_{\pm 0.08}$	$3.60_{\pm 0.09}$
UniFlow-Audio	4.22	0.35	$3.62_{\pm 0.08}$	$3.80_{\pm 0.08}$	6.39	0.15	$3.45_{\pm 0.10}$	$3.25_{\pm 0.11}$
Ming-omni-TTS	2.46	0.27	$3.85_{\pm 0.07}$	$3.60_{\pm 0.09}$	7.98	0.07	$3.25_{\pm 0.11}$	$2.92_{\pm 0.13}$
AudioCALM (ours)	**1.95**	**0.37**	$\mathbf{3.98_{\pm 0.06}}$	$\mathbf{3.95_{\pm 0.07}}$	**2.02**	**0.36**	$\mathbf{3.99_{\pm 0.06}}$	$\mathbf{3.92_{\pm 0.07}}$

Table 3: Ablation on text-to-speech (LIBRITTS *test-clean*), text-to-sound (AUDIOCAPS), and text-to-music (SONG-DESCRIBER). Top: specialists using variant (b)'s architecture without joint speech/non-speech training. Bottom: joint-training variants sharing the Qwen3-1.7B backbone and VAE encoder/decoder; (a) uses a quantized bottleneck with discrete prediction, whereas (b) uses continuous latents with a flow-matching head.

	Speech (LibriTTS)		Sound (AudioCaps)		Music (Song-Describer)	
Variant	WER↓	SIM↑	FAD↓	CLAP↑	FAD↓	CLAP↑
Single-modality specialists (variant (b) architecture, no joint training)						
Speech-only training	0.022	0.628	–	–	–	–
Non-speech-only training	–	–	2.45	0.34	2.55	0.30
Joint speech/sound/music training, cumulative ablation						
(a) Discrete-token AR baseline (no flow head)	0.040	0.560	4.50	0.22	4.80	0.20
(b) Continuous AR + flow head, raw transcripts	0.024	0.620	3.30	0.30	3.45	0.26
(c) (b) + description-style reframing	0.023	0.625	2.70	0.33	2.85	0.29
(d) (c) + symmetric MoME (3 modality experts)	0.022	0.660	2.30	0.35	2.40	0.33
(e) (c) + **A-MoME (speech residual only)**	**0.020**	**0.668**	**1.95**	**0.37**	**2.02**	**0.36**

Performance across Domains. AudioCALM ranks first or second on every reported metric in Tables 1 and 2. The compared unified systems have larger gaps on at least one domain; for example, music FAD is 6.39 for UniFlow-Audio and 7.98 for Ming-omni-TTS, compared with 2.02 for AudioCALM. These comparisons establish complete-system performance; the matched training-mixture and architecture comparisons below examine cross-domain interference.

4.3 Ablation Studies

We ablate AudioCALM's three core designs on text-to-speech (LIBRITTS *test-clean*; Sp. WER, SIM), text-to-sound (AUDIOCAPS; Sn. FAD, CLAP), and text-to-music (SONG-DESCRIBER; Mu. FAD, CLAP). The cumulative variants (a)–(e) in Table 3 share the joint training of Section 4.1, so each column reflects per-modality fidelity *and* any cross-modal crowd-out; the two single-modality rows at the top use variant (b)'s architecture on one modality, isolating crowd-out from architectural effects. Backbone, CFG, block-size, and inference-step sweeps are deferred to Appendices C.1–C.4.

Asymmetric Crowd-Out. The specialist rows and joint variant (b) use the same architecture with different training mixtures. Joint training changes speech WER/SIM from $0.022/0.628$ to $0.024/0.620$, while sound FAD increases from 2.45 to 3.30 and music FAD from 2.55 to 3.45 (about 35% each), with lower CLAP in both domains. This pattern supports the asymmetric mismatch motivating our data and architecture designs; WER and FAD measure different properties, so their changes are not treated as a common numerical interference scale.

Continuous Flow-Matching Head. Moving from (a)'s quantized bottleneck and discrete prediction to (b)'s continuous-latent flow improves all three domains (Sp. WER $0.040 \rightarrow 0.024$, SIM $0.560 \rightarrow 0.620$; Sn. FAD $4.50 \rightarrow 3.30$, CLAP $0.22 \rightarrow 0.30$; Mu. FAD $4.80 \rightarrow 3.45$, CLAP $0.20 \rightarrow 0.26$). With the VAE encoder and decoder fixed, this supports the continuous-latent formulation; the comparison changes both quantization and the prediction objective.

Description-Style Reframing. Description-style conditioning (c vs. b) improves sound FAD/CLAP from $3.30/0.30$ to $2.70/0.33$ and music from $3.45/0.26$ to $2.85/0.29$, with speech WER/SIM changing from $0.024/0.620$ to $0.023/0.625$. The controls in Appendix C.5 separate improvements from a shared textual format and additional audio-grounded information, and show that A-MoME also improves generation with raw conditions alone.

A-MoME vs. Symmetric MoME. Symmetric three-expert MoME (d) reaches Sp. $0.022/0.660$, Sn. $2.30/0.35$, and Mu. $2.40/0.33$. A-MoME (e) improves all six metrics to $0.020/0.668$, $1.95/0.37$, and $2.02/0.36$, with fewer total parameters and no extra non-speech inference cost. The parameter and route counts are reported in Appendix C.7; speech uses the shared FFN plus its residual expert, whereas sound and music retain the shared route. Supplementary regularizer and listening ablations are in Appendix C.6.

5 Conclusion

We presented **AudioCALM**, a universal audio generation framework instantiating *Continuous Autoregressive Language Modeling (CALM)*: a single autoregressive LM parameterizes per-position rectified-flow velocities via a thin flow-matching head, paired with block-causal AR-Flow attention for incremental, variable-length latent generation across speech, sound, and music. We identified an asymmetric mismatch between locally aligned text-to-speech and globally conditioned text-to-audio, and addressed it with a unified description-style conditioning interface and A-MoME, whose capacity allocation follows this mismatch rather than the modality count. Under the reported protocol, one set of weights delivers competitive performance against dedicated systems across all three domains and outperforms the compared unified baselines. Corpus coverage, backbone scale, and long-form generation remain open directions; deployment also requires provenance and consent safeguards against voice-cloning and synthetic-media risks (Appendices E, F, and G).

References

Inclusion AI, Biao Gong, Cheng Zou, Chuanyang Zheng, Chunluan Zhou, Canxiang Yan, Chunxiang Jin, Chunjie Shen, Dandan Zheng, Fudong Wang, et al. Ming-omni: A unified multimodal model for perception and generation. *arXiv preprint arXiv:2506.09344*, 2025.

Philip Anastassiou, Jiawei Chen, Jitong Chen, Yuanzhe Chen, Zhuo Chen, Ziyi Chen, Jian Cong, Lelai Deng, Chuang Ding, Lu Gao, et al. Seed-tts: A family of high-quality versatile speech generation models. *arXiv preprint arXiv:2406.02430*, 2024.

Marianne Arriola, Aaron Gokaslan, Justin T. Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block Diffusion: Interpolating between autoregressive and diffusion language models. In *International Conference on Learning Representations*, 2025. URL `https://arxiv.org/abs/2503.09573`.

Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, Songhao Piao, and Furu Wei. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. *Advances in neural information processing systems*, 35:32897–32912, 2022.

Dmitry Bogdanov, Minz Won, Philip Tovstogan, Alastair Porter, and Xavier Serra. The mtg-jamendo dataset for automatic music tagging. In *Machine learning for music discovery workshop, international conference on machine learning (ICML 2019)*, pages 1–3. Long Beach, CA, United States, 2019.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, et al. Audiolm: a language modeling approach to audio generation. *IEEE/ACM transactions on audio, speech, and language processing*, 31:2523–2533, 2023.

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.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, 2022.

Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, JianZhao JianZhao, Kai Yu, and Xie Chen. F5-tts: A fairytaler that fakes fluent and faithful speech with flow matching. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6255–6271, 2025.

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

Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. Fma: A dataset for music analysis. *arXiv preprint arXiv:1612.01840*, 2016.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*, 2022.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*, 2024.

Ding Ding, Zeqian Ju, Yichong Leng, Songxiang Liu, Tong Liu, Zeyu Shang, Kai Shen, Wei Song, Xu Tan, Heyi Tang, et al. Kimi-audio technical report. *arXiv preprint arXiv:2504.18425*, 2025.

Zhihao Du, Yuxuan Wang, Qian Chen, Xian Shi, Xiang Lv, Tianyu Zhao, Zhifu Gao, Yexin Yang, Changfeng Gao, Hui Wang, et al. Cosyvoice 2: Scalable streaming speech synthesis with large language models. *arXiv preprint arXiv:2412.10117*, 2024.

Zhihao Du, Changfeng Gao, Yuxuan Wang, Fan Yu, Tianyu Zhao, Hao Wang, Xiang Lv, Hui Wang, Chongjia Ni, Xian Shi, et al. Cosyvoice 3: Towards in-the-wild speech generation via scaling-up and post-training. *arXiv preprint arXiv:2505.17589*, 2025.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Zach Evans, Julian D Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Stable audio open. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

Haorui He, Zengqiang Shang, Chaoren Wang, Xuyuan Li, Yicheng Gu, Hua Hua, Liwei Liu, Chen Yang, Jiaqi Li, Peiyang Shi, et al. Emilia: An extensive, multilingual, and diverse speech dataset for large-scale speech generation. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, pages 885–890. IEEE, 2024.

Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 131–135. IEEE, 2017.

Chia-Yu Hung, Navonil Majumder, Zhifeng Kong, Ambuj Mehrish, Amir Ali Bagherzadeh, Chuan Li, Rafael Valle, Bryan Catanzaro, and Soujanya Poria. Tangoflux: Super fast and faithful text to audio generation with flow matching and clap-ranked preference optimization. *arXiv preprint arXiv:2412.21037*, 2024.

InclusionAI. Ming-omni-tts. `https://github.com/inclusionAI/Ming-omni-tts`, n.d. Official implementation and model releases.

Dongya Jia, Zhuo Chen, Jiawei Chen, Chenpeng Du, Jian Wu, Jian Cong, Xiaobin Zhuang, Chumin Li, Zhen Wei, Yuping Wang, et al. Ditar: Diffusion transformer autoregressive modeling for speech generation. *arXiv preprint arXiv:2502.03930*, 2025.

Zeqian Ju, Yuancheng Wang, Kai Shen, Xu Tan, Detai Xin, Dongchao Yang, Yanqing Liu, Yichong Leng, Kaitao Song, Siliang Tang, et al. Naturalspeech 3: Zero-shot speech synthesis with factorized codec and diffusion models. *arXiv preprint arXiv:2403.03100*, 2024.

Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fréchet audio distance: A metric for evaluating music enhancement algorithms. *arXiv preprint arXiv:1812.08466*, 2018.

Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee 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, Volume 1 (Long and Short Papers)*, pages 119–132, 2019.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

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

Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved rvqgan. *Advances in Neural Information Processing Systems*, 36:27980–27993, 2023.

Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, et al. Voicebox: Text-guided multilingual universal speech generation at scale. *Advances in neural information processing systems*, 36:14005–14034, 2023.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37:56424–56445, 2024.

Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, et al. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. *arXiv preprint arXiv:2411.04996*, 2024.

Haohe Liu, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Qiao Tian, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D Plumbley. Audioldm 2: Learning holistic audio generation with self-supervised pretraining. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:2871–2883, 2024a.

Huadai Liu, Rongjie Huang, Xuan Lin, Wenqiang Xu, Maozong Zheng, Hong Chen, Jinzheng He, and Zhou Zhao. Vit-tts: Visual text-to-speech with scalable diffusion transformer. *arXiv preprint arXiv:2305.12708*, 2023.

Huadai Liu, Rongjie Huang, Yang Liu, Hengyuan Cao, Jialei Wang, Xize Cheng, Siqi Zheng, and Zhou Zhao. Audiolcm: Text-to-audio generation with latent consistency models. *arXiv preprint arXiv:2406.00356*, 2024b.

Huadai Liu, Jialei Wang, Rongjie Huang, Yang Liu, Heng Lu, Zhou Zhao, and Wei Xue. Flashaudio: Rectified flows for fast and high-fidelity text-to-audio generation. *arXiv preprint arXiv:2410.12266*, 2024c.

Huadai Liu, Kaicheng Luo, Jialei Wang, Wen Wang, Qian Chen, Zhou Zhao, and Wei Xue. Thinksound: Chain-of-thought reasoning in multimodal large language models for audio generation and editing. *arXiv preprint arXiv:2506.21448*, 2025a.

Huadai Liu, Kaicheng Luo, Wen Wang, Qian Chen, Peiwen Sun, Rongjie Huang, Xiangang Li, Jieping Ye, and Wei Xue. Prismaudio: Decomposed chain-of-thoughts and multi-dimensional rewards for video-to-audio generation. *arXiv preprint arXiv:2511.18833*, 2025b.

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

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Zhenyu Liu, Yunxin Li, Xuanyu Zhang, Qixun Teng, Shenyuan Jiang, Xinyu Chen, Haoyuan Shi, Jinchao Li, Qi Wang, Haolan Chen, et al. Unimoe-audio: Unified speech and music generation with dynamic-capacity moe. *arXiv preprint arXiv:2510.13344*, 2025d.

Ilaria Manco, Benno Weck, Seungheon Doh, Minz Won, Yixiao Zhang, Dmitry Bogdanov, Yusong Wu, Ke Chen, Philip Tovstogan, Emmanouil Benetos, et al. The song describer dataset: a corpus of audio captions for music-and-language evaluation. *arXiv preprint arXiv:2311.10057*, 2023.

Shivam Mehta, Ruibo Tu, Jonas Beskow, Éva Székely, and Gustav Eje Henter. Matcha-tts: A fast tts architecture with conditional flow matching. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11341–11345. IEEE, 2024.

Xinhao Mei, Chutong Meng, Haohe Liu, Qiuqiang Kong, Tom Ko, Chengqi Zhao, Mark D Plumbley, Yuexian Zou, and Wenwu Wang. Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32: 3339–3354, 2024.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR, 2023.

Paul K Rubenstein, Chulayuth Asawaroengchai, Duc Dung Nguyen, Ankur Bapna, Zalán Borsos, Félix de Chaumont Quitry, Peter Chen, Dalia El Badawy, Wei Han, Eugene Kharitonov, et al. Audiopalm: A large language model that can speak and listen. *arXiv preprint arXiv:2306.12925*, 2023.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Tongyi Fun Team, Qian Chen, Luyao Cheng, Chong Deng, Xiangang Li, Jiaqing Liu, Chao-Hong Tan, Wen Wang, Junhao Xu, Jieping Ye, et al. Fun-audio-chat technical report. *arXiv preprint arXiv:2512.20156*, 2025.

Michael Tschannen, Cian Eastwood, and Fabian Mentzer. Givt: Generative infinite-vocabulary transformers. In *European Conference on Computer Vision*, pages 292–309. Springer, 2024.

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

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*, 2023a.

Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for vision and vision-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19175–19186, 2023b.

Yuancheng Wang, Haoyue Zhan, Liwei Liu, Ruihong Zeng, Haotian Guo, Jiachen Zheng, Qiang Zhang, Xueyao Zhang, Shunsi Zhang, and Zhizheng Wu. Maskgct: Zero-shot text-to-speech with masked generative codec transformer. *arXiv preprint arXiv:2409.00750*, 2024.

Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-omni technical report. *arXiv preprint arXiv:2503.20215*, 2025a.

Xuenan Xu, Jiahao Mei, Zihao Zheng, Ye Tao, Zeyu Xie, Yaoyun Zhang, Haohe Liu, Yuning Wu, Ming Yan, Wen Wu, et al. Uniflow-audio: Unified flow matching for audio generation from omni-modalities. *arXiv preprint arXiv:2509.24391*, 2025b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Dongchao Yang, Jinchuan Tian, Xu Tan, Rongjie Huang, Songxiang Liu, Xuankai Chang, Jiatong Shi, Sheng Zhao, Jiang Bian, Xixin Wu, et al. Uniaudio: An audio foundation model toward universal audio generation. *arXiv preprint arXiv:2310.00704*, 2023.

Dongchao Yang, Haohan Guo, Yuanyuan Wang, Rongjie Huang, Xiang Li, Xu Tan, Xixin Wu, and Helen Meng. Uniaudio 1.5: Large language model-driven audio codec is a few-shot audio task learner. *Advances in Neural Information Processing Systems*, 37:56802–56827, 2024.

Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. Libritts: A corpus derived from librispeech for text-to-speech. *arXiv preprint arXiv:1904.02882*, 2019.

Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15757–15773, 2023.

Dong Zhang, Gang Wang, Jinlong Xue, Kai Fang, Liang Zhao, Rui Ma, Shuhuai Ren, Shuo Liu, Tao Guo, Weiji Zhuang, et al. Mimo-audio: Audio language models are few-shot learners. *arXiv preprint arXiv:2512.23808*, 2025.

A Additional Implementation Details

A.1 Continuous Audio VAE

Architecture. Our autoencoder is a CNN–GAN of the STABLE AUDIO OPEN [Evans et al., 2025]/DAC [Kumar et al., 2023] family with three deviations: (i) an iSTFT synthesis head replaces the time-domain transposed-convolution stack at the decoder output, (ii) self-attention layers are inserted at the lowest-resolution stages of both the encoder and the decoder, and (iii) a learned patch-with-`[CLS]` aggregator at the bottleneck further reduces the latent rate [AI et al., 2025].

Encoder and bottleneck. The encoder takes 44.1 kHz stereo waveforms through a stack of strided 1D residual blocks (Snake activations, weight normalisation) whose cumulative stride of $2048\times$ yields a 21.5 Hz feature sequence; self-attention blocks at the lowest-resolution stages let the encoder reach beyond its convolutional receptive field. This sequence is right-padded to a multiple of $P{=}2$, split into patches of P frames each augmented with a learnable `[CLS]` token, and processed by a small self-attention aggregator that mixes within and across patches. The `[CLS]` positions are collected and projected to the parameters of a diagonal Gaussian posterior $q(\mathbf{z}\,|\,\mathbf{x}) = \mathcal{N}(\mu, \mathrm{diag}(\sigma^2))$ at $21.5/P \approx 10.75$ Hz, so an AR-Flow block of $B{=}1.0$ s corresponds to 11 latent tokens.

Decoder. The decoder linearly interpolates the latent back to 21.5 Hz, refines it through 1D ResBlocks with symmetric self-attention insertions, and synthesises the waveform with an iSTFT head whose analysis grid matches the feature rate (hop $= 2048$ samples, $n_{\text{fft}} = 8192$). The head emits $2 \cdot 2 \cdot (n_{\text{fft}}/2{+}1)$ channels for two stereo-channel complex spectrograms, which a per-channel inverse short-time Fourier transform (Hann window) maps to the 44.1 kHz stereo output. Routing the final $2048\times$ of upsampling through a closed-form filterbank rather than through additional transposed convolutions removes the dominant source of phase artefacts in DAC-style decoders.

Training. The VAE is trained on the corpora of Section A.2 at 44.1 kHz stereo with a multi-resolution STFT reconstruction loss, a KL term on the Gaussian posterior with free-bits regularisation, and an adversarial loss combining a multi-period time-domain discriminator with a multi-resolution complex-spectrogram discriminator (topology following the public STABLE AUDIO OPEN release). We then freeze the VAE for the remainder of AudioCALM training.

A.2 Training Corpora and Preprocessing

The training mixture covers three modalities: speech (LIBRITTS [Zen et al., 2019] and the English subset of EMILIA [He et al., 2024]), general sound (the audio of VGGSOUND [Chen et al., 2020], AUDIOCAPS [Kim et al., 2019], and WAVCAPS [Mei et al., 2024]), and music (FMA [Defferrard et al., 2016] and MTG-JAMENDO [Bogdanov et al., 2019]). Only the official training splits are used; LIBRITTS *test-clean*, the AUDIOCAPS and SONG-DESCRIBER [Manco et al., 2023] evaluation sets, and SEEDTTS-EVAL [Anastassiou et al., 2024] are held out to rule out test-set contamination. All audio is resampled to 44.1 kHz and converted to two-channel stereo, with mono sources upmixed by replicating the single channel.

A.3 Description-Style Conditioning Pipeline

This subsection details the offline pipeline that produces the dual textual conditions consumed by the body model (Section 3.4). For every training clip we store a *short* variant—the verbatim transcript for speech, the original short caption for sound and music—and a *long-form* variant generated by an audio-conditioned MLLM (GEMINI 3 PRO). The two variants are sampled with equal probability per training step.

Prompt design. We use a two-tier prompt: a shared *system message* fixes the output format and the audio-grounding contract; a modality-specific *user message* lists the attributes to surface, with explicit priority ordering, and (for speech) pins the transcript-delimiter rule. The two-tier split is what makes the long-form descriptions homogeneous in style across speech, sound, and music while still soliciting modality-relevant content. The templates we issue to GEMINI 3 PRO are reproduced verbatim below.

System message (identical across modalities).

```
You are an audio analyst. Listen to the supplied audio clip and
write a single fluent paragraph that will be used as a text prompt
for an audio-generation model--so describe what the audio sounds
like, not what is happening behind the recording.
Hard rules:
(1) Every adjective and attribute must be supported by the audio
itself. If you are not confident from the audio alone, omit the
attribute; do not guess and do not pad with generic adjectives.
(2) Do not begin with ‘‘This is...’’ or ‘‘The audio shows...’’. Start
directly with the most distinctive sonic content.
(3) Plain prose only: no bullet points, no markdown headings, no
bracketed metadata, no numerical confidence scores.
(4) Single paragraph, ≤ 200 words.
```

User message — speech.

```
Modality: speech. Surface, in roughly this order of
priority: (i) speaker timbre, age range, and apparent gender;
(ii) prosody--speaking rate, pitch range, emphasis pattern,
emotional tone; (iii) recording acoustics--reverberation level,
background-noise level, overall fidelity (e.g., studio-clean vs.
mobile-recorded).
At the end of the paragraph, splice the verbatim transcript wrapped
in delimiters <spoken> ... </spoken>. The text inside the delimiters
must reproduce the transcript character-for-character; do not
rephrase, expand, shorten, or translate it. The delimiter pair
must appear exactly once and must not be nested.
Transcript: {short_variant}
```

User message — general sound.

```
Modality: general sound. Surface, in roughly this order of
priority: (i) the dominant acoustic events and their likely
physical sources; (ii) the scene or environment that ties the
events together; (iii) the foreground/background structure of the
soundscape; (iv) the temporal arc--which events come first, which
recur, and how the energy evolves.
Do not invent specific named entities (particular cities, brands,
individuals) that the audio cannot disambiguate; describe categories
instead (‘‘a busy city street’’ rather than ‘‘downtown Tokyo’’).
Original short caption: {short_variant}
```

User message — music.

```
Modality: music. Surface, in roughly this order of priority:
(i) genre or stylistic register and the apparent instrumentation;
(ii) tempo, rhythmic feel, and overall mood; (iii) structural
cues--recurring melodic or rhythmic motifs, dynamic arc, vocal
vs. instrumental balance; (iv) production character--production
cleanliness, stereo image, lo-fi/hi-fi flavor.
Do not specify exact BPM values, named musical keys, or named
artists/albums even if you think you recognize them.
Original short caption: {short_variant}
```

The system message anchors a uniform prose style and the audio-grounding contract; the user messages enumerate attributes by priority, ban modality-specific failure modes (named entities for sound, BPM/keys/artists for music), and pin the speech delimiter convention. The audio-grounding clause is what makes an audio-conditioned MLLM strictly required: a text-only LLM cannot recover speaker timbre, recording acoustics, or musical mood from the bare transcript or short caption alone.

Speech transcript delimiter. For speech, the prompt additionally instructs the MLLM to splice the verbatim transcript inside the long-form description, wrapped in explicit content delimiters `<spoken>` ... `</spoken>`. The body LM can therefore attend to a globally diffuse description and

still see the exact spoken content for local content alignment, without the description-style and transcript-style conditioning collapsing into two separate prompts. The same delimiter convention is reused at inference for TTS and voice-cloning prompts.

To prevent the byte-pair tokenizer from fragmenting these markers into multiple sub-pieces (which would scatter the boundary signal across $\sim$3–4 tokens per delimiter and force the LM to re-learn the same opening/closing semantics from many distinct token combinations), both `<spoken>` and `</spoken>` are added as *special tokens* to the body LM's tokenizer, extending the Qwen3 vocabulary by two entries. Each delimiter is then a single token at training and inference, and the LM has a single dedicated embedding row per boundary that it can attend to as a unit. The two new embedding rows are initialized as the mean of Qwen3's existing chat-template special tokens, placing them within the boundary-marker subspace the backbone already uses, and are trained jointly with the rest of the model.

Quality control. For every captioning batch, 5% of the produced long-form descriptions are sampled uniformly at random and reviewed by trained annotators. A description is rejected if it (i) asserts audio attributes the clip does not actually exhibit, (ii) omits attributes the prompt template asks for, or (iii) violates the speech delimiter convention. Rejected clips are returned to the MLLM for re-captioning under the same prompt; a clip that fails review more than three times is dropped from the training set entirely. Clips that pass review (and the unsampled remainder of their batch) enter the training manifest.

Storage and loader behavior. Both variants are stored as parallel fields in the training manifest. At every step the data loader draws a Bernoulli(0.5) coin per clip to select which variant is used as the conditioning text; the audio target is identical in both cases. This dual exposure is what teaches the LM to handle both terse prompts (a bare label) and richly described prompts at inference within a single conditioning interface.

A.4 Hyperparameters and Training Setup

Table 4 summarizes the architectural, training, regularization, and inference hyperparameters of the AudioCALM run that produces all results in Tables 1, 2, and 3. The appendix sweeps (Tables 5–7) differ only along the swept axis. All values were fixed before any evaluation on held-out splits: optimizer settings follow the Qwen3 pretraining recipe, the rectified-flow schedule and CFG dropout follow [Esser et al., 2024, Liu et al., 2022], and the AR-Flow block size, exposure-bias scale, and clean-prefix noise scale were selected on a 5% in-domain validation slice held out from the training mixture, with no held-out evaluation prompts used for selection.

B Evaluation Metrics

B.1 Objective Metrics

Generated waveforms are converted to mono for objective scoring and resampled *per metric* to the rate expected by the scoring model—16 kHz for Whisper, WavLM-Large, and VGGish, and 48 kHz for LAION-CLAP—using a polyphase resampler with a Kaiser window. These scoring conversions are distinct from each system's native waveform output; AudioCALM's VAE synthesizes 44.1 kHz stereo audio.

WER (speech). Intelligibility is measured as the word error rate of the generated speech when transcribed by the open-source WHISPER-LARGE-V3 [Radford et al., 2023] ASR model under greedy decoding (`temperature=0`, `condition_on_previous_text=False`, no language hint). Both the hypothesis and the reference target text pass through the standard Whisper text-normalization pipeline—Unicode NFKC, case folding, removal of punctuation and bracketed annotations, white-space collapse, and number-to-word expansion via the Whisper `EnglishTextNormalizer`—before computing a Levenshtein word distance with `jiwer`. WER is reported as a fraction in $[0, 1]$; lower is better. We restrict speech evaluation to the *en* split of SeedTTS-eval, so CER is not reported.

SIM (speech). For zero-shot voice cloning we report SIM, the cosine similarity between the speaker embedding of the generated utterance and the speaker embedding of the prompt audio. Embeddings are extracted with WAVLM-LARGE [Chen et al., 2022] fine-tuned for speaker verification on VoxCeleb, using the model's mean-pooled embedding over the full utterance. On LIBRITTS *test-*

Table 4: Model, training, and inference hyperparameters for AudioCALM.

Backbone and head	**Value**
Backbone init	Qwen3-1.7B
Hidden size / layers / heads	2048 / 28 / 16 (Qwen3 default)
A-MoME speech FFN width	matches shared FFN; zero-init
Flow-matching head ϕ_{out}	linear, zero-init
Stop head g_{stop}	linear $\rightarrow$ sigmoid; ramp window $K_{\text{stop}} = 10$
Audio embedding ϕ_{in}	linear; C=64 VAE channels $\rightarrow$ H=2048
Position embedding	shared RoPE with text
Timestep embedding $\tau(t)$	sinusoidal, summed into $e_i^{(t)}$
Training	
Optimizer	AdamW, β_1=0.9, β_2=0.95, weight decay 0
Learning rate	1×10^{-4} constant after 2,000-step linear warmup
Gradient clipping	global norm 1.0
Total steps	300,000
Global batch	~ 64 k tokens (8 k tokens / GPU, 8 GPUs; token count includes training-time noise tokens)
Precision	bfloat16 mixed precision; FSDP HYBRID-SHARD
Sequence packing length	4096 tokens (text + audio)
Modality mixing ratio (sp. / sn. / mu.)	0.4 / 0.3 / 0.3 at the example level
Caption mode (short / long)	0.5 / 0.5 per example
Caption (CFG) dropout	10% unconditional during training
Rectified-flow timestep t	$\sigma(u),\ u \sim \mathcal{N}(0,1)$ (logit-normal)
Per-block clean-prefix noise σ_{clean}	0.1
Exposure-bias perturbation γ_{exp}	0.1
Stop-head ramp length K_{stop}	10 tokens
Inference	
AR-Flow block size B	1.0 s (11 latent tokens)
Rectified-flow steps per block K_{flow}	24, Euler integration
CFG scale w	3.0
Stop probability threshold τ_{stop}	0.95
Sampling temperature on velocity	1.0 (deterministic Euler)

clean the prompt is a 3 s clip drawn from a *different* utterance of the same speaker as the target sentence, following the protocol of [Chen et al., 2025]; on SEEDTTS-EVAL (*en*) we use the prompt audio and target text shipped with the benchmark verbatim. SIM lies in $[-1, 1]$; higher is better.

FAD (sound, music). Fréchet Audio Distance [Kilgour et al., 2018] measures the distributional gap between generated and reference audio in the VGGISH [Hershey et al., 2017] embedding space. We use Google's reference `frechet_audio_distance` implementation: both candidate and reference clips are downmixed to 16 kHz mono and split into 0.96 s VGGish frames, per-clip embeddings are pooled into a single multivariate Gaussian per set, and the Fréchet distance between the two Gaussians is reported. The reference distribution is built from the ground-truth waveforms of the AUDIOCAPS test split for sound and the SONG-DESCRIBER evaluation pool for music; the candidate distribution contains exactly one generated clip per evaluation prompt at the same nominal duration as the corresponding reference. FAD is non-negative; lower is better.

CLAP (sound, music). Text–audio alignment is reported as the cosine similarity between the LAION-CLAP [Wu et al., 2023] text and audio embeddings of an evaluation caption and its corresponding generation. To allow direct comparison across sound and music we use a single checkpoint, `music_audioset_epoch_15_esc_90.14`; the audio branch consumes 48 kHz mono input zero-padded or centre-cropped to the model's 10 s receptive field, while the text branch consumes the unmodified evaluation caption (no truncation is required at the caption lengths used). We report the per-prompt cosine similarity averaged over all prompts in the evaluation set; values lie in $[-1, 1]$, higher is better.

B.2 Subjective MOS Protocol

For every system we additionally collect Mean Opinion Scores on a randomly sampled subset of $N{=}50$ prompts per benchmark. Speech is rated on a single *overall-quality* scale, denoted **MOS** in Table 1: because the target text is fixed and prompt adherence is already captured by WER, a separate text-relevance axis would be redundant. Sound and music are rated on two complementary axes: **MOS-Q** (overall perceived quality) and **MOS-T** (text relevance to the caption), the latter being essential in the absence of a ground-truth reference waveform. All axes use the standard 5-point ITU-T P.808 absolute-category-rating scale (1 bad, 2 poor, 3 fair, 4 good, 5 excellent), with anchor descriptions adapted per modality.

Listening setup. Each rater completes the study in a single session of at most 30 min on a desktop or laptop browser; mobile devices are blocked. All clips are loudness-normalized to -23 LUFS with a true-peak ceiling of -1 dBTP following ITU-R BS.1770-4 before being served, and playback level is fixed via a 1 kHz sine calibration trial at the start of the session. A short training stage shows three labelled reference clips per modality (anchored at the 1, 3, and 5 levels of each axis) before the rating session begins, and three attention checks per session—an unmodified ground-truth clip, a 50%-time-stretched clip, and a clip with 20 dB additive white noise—are inserted at random positions. Sessions that fail more than one attention check are discarded and the affected ratings re-collected.

B.3 Human-Subjects Study Operationalization

Recruitment and demographics. Listeners were recruited through a vetted commercial crowdsourcing platform restricted to workers self-reporting fluent English and normal hearing. Each listener completed a one-minute pre-screening that included three suprathreshold tone-detection trials and a short comprehension question covering the rating instructions; only listeners who passed all four were admitted. The final pool comprised $\sim$120 unique listeners (anonymized to numeric IDs at intake), self-reporting a balanced gender distribution and an age range of 19–58. No personally identifying information was collected, stored, or shared with the authors at any point.

Consent and instructions. Before any audio was played, each listener was shown a plain-language consent screen describing the purpose of the study ("rating the perceived quality of computer-generated audio"), the data collected (numeric ratings and click metadata, no audio recordings of the listener), the right to withdraw at any time without forfeiting accumulated payment for completed clips, and a contact mailbox for questions or complaints. The full text of the consent screen, the rating instructions for each axis (MOS, MOS-Q, MOS-T) and modality (speech, sound, music), the calibration trial, and screenshots of the rating UI are included verbatim in the supplemental zip under `supp/mos_protocol/`.

Compensation. Listeners were paid a fixed task rate calibrated to a target hourly wage of \$15/hour, well above the federal and state minimum wage in both the data collector's and the platform's jurisdictions, and above the platform's recommended floor. Payment was issued for every completed session, including sessions whose ratings were later discarded for failing attention checks; only the affected ratings were re-collected from a different listener. Bonus payments were awarded for sessions with all three attention checks correct. Median completion time was 22 min, yielding a realized median rate of $\sim$\$16.4/hour after bonuses.

Voice-clone consent in evaluation. Speaker-similarity (SIM) evaluation requires a reference voice prompt. We use only voices from the public LIBRITTS *test-clean* and SEEDTTS-EVAL (*en*) splits, both of which were released by their authors with explicit re-distribution permissions for research; we do not record, solicit, or generate additional voice prompts from any human participant in this study, so no consent step beyond the upstream dataset releases is required for the SIM measurements.

C Additional Experimental Results

C.1 Backbone Scale Ablation

We initialize AudioCALM from three Qwen3 checkpoints—0.6B, 1.7B (default), and 4B—training each for the same number of tokens as the main run. Table 5 reports per-modality WER (speech) and FAD/CLAP (sound, music).

Table 5: Backbone-scale ablation. All variants share the data, optimizer, and step budget of the main run. Bold marks the best result per column; the 1.7B row is the default used in the main results.

Backbone	Sp. WER↓	Sn. FAD↓	Sn. CLAP↑	Mu. FAD↓	Mu. CLAP↑
Qwen3-0.6B	0.025	2.38	0.31	3.75	0.31
Qwen3-1.7B (default)	0.020	1.95	0.37	2.02	0.36
Qwen3-4B	**0.018**	**1.89**	**0.39**	**1.97**	**0.38**

C.2 Classifier-Free Guidance Scale Sweep

We sweep the inference CFG scale $w \in \{1.0, 2.0, 3.0, 4.0, 5.0\}$ with all other hyperparameters fixed.

Table 6: CFG scale sweep on the main evaluation benchmarks. Bold marks the best result per column; w=3.0 is the default used in the main results.

w	Sp. WER↓	Sn. FAD↓	Sn. CLAP↑	Mu. FAD↓	Mu. CLAP↑
1.0	0.028	3.22	0.31	2.49	0.32
2.0	0.026	2.68	0.35	2.13	0.33
3.0 (default)	**0.020**	1.95	**0.37**	**2.02**	**0.36**
4.0	0.021	**1.88**	0.36	2.15	0.31
5.0	0.020	2.01	0.32	2.30	0.30

C.3 AR-Flow Block-Size Sweep

We sweep the AR-Flow block size $B \in \{0.25, 0.5, 1.0, 2.0, 4.0\}$ s, holding the VAE frame rate and total decoded duration fixed, and additionally report a non-autoregressive baseline without block-wise decoding. Table 7 reports per-modality quality. Speech WER is best at 0.5 s and worsens for $B \geq 2$ s, while sound and music have their best joint FAD/CLAP at 1 s. We use 1 s as the shared setting. Output duration is variable, but the block size is fixed within each generation.

Table 7: AR-Flow block-size sweep (in seconds). Bold marks the best result per column; B=1.0 s is the default used in the main results.

B (s)	Sp. WER↓	Sn. FAD↓	Sn. CLAP↑	Mu. FAD↓	Mu. CLAP↑
0.25	0.019	2.45	0.35	2.48	0.33
0.5	**0.018**	2.17	0.36	2.22	0.34
1.0 (default)	0.020	**1.95**	**0.37**	**2.02**	**0.36**
2.0	0.025	2.11	0.34	2.18	0.34
4.0	0.031	2.08	0.33	2.14	0.33
Non-Autoregressive	0.036	2.05	0.34	2.11	0.35

C.4 Inference Step Sweep

We sweep the number of inference steps with all other hyperparameters fixed. Figure 2 shows that generation quality improves substantially as the number of inference steps increases, while the gains become marginal beyond roughly 20–30 steps. Based on this trade-off, we adopt 24 inference steps as the default configuration, which provides a favorable balance between generation quality and computational efficiency.

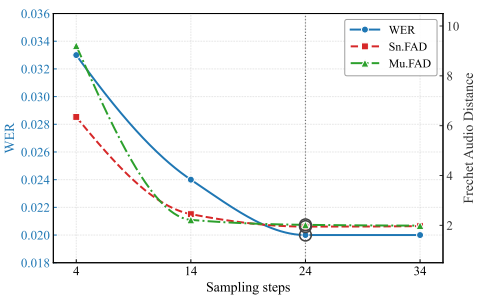


Figure 2: Inference-step sweep.

C.5 Separating Conditioning and Architecture

Table 8 complements the cumulative comparisons in Table 3. The format-only condition uses the long-description prose template and speech delimiters but rearranges only the original text, without attributes inferred from audio. Format-only and audio-grounded conditions each mix original/control text with probability $0.5/0.5$; evaluation always uses the original benchmark text. The shared-backbone rows separate textual style from additional audio information. Comparing Shared with A-MoME under raw conditions tests the architecture without Gemini-generated long descriptions.

Table 8: Supplementary conditioning and architecture controls. Sp.: WER/SIM; Sn. and Mu.: FAD/CLAP. Backbone initialization, VAE, training budget, and decoding settings are fixed.

Conditioning	Architecture	Sp. WER/SIM	Sn. FAD/CLAP	Mu. FAD/CLAP
Raw	Shared	.024/.620	3.30/.30	3.45/.26
Raw/Format-only	Shared	.024/.622	3.05/.31	3.18/.27
Raw/Audio-grounded	Shared	.023/.625	2.70/.33	2.85/.29
Raw	A-MoME	.021/.659	2.54/.34	2.62/.32
Raw/Audio-grounded	A-MoME	.020/.668	1.95/.37	2.02/.36

Both a shared format and audio-grounded attributes improve sound/music metrics. A-MoME also improves the raw-condition model, supporting an architecture contribution independent of long-description construction.

C.6 Regularization and Listening Ablations

We separately vary clean-prefix noise σ_{clean} and exposure perturbation γ_{exp} on A-MoME, keeping all other settings fixed. Table 9 shows improvements from either regularizer, with the combined setting giving the best numerical results across domains. The main Table 3 comparisons fix both scales at 0.1.

Table 9: Separate regularizer ablation on the final A-MoME architecture.

σ_{clean}	γ_{exp}	Sp. WER/SIM	Sn. FAD/CLAP	Mu. FAD/CLAP
0	0	.023/.651	2.14/.35	2.22/.34
.1	0	.022/.658	2.07/.36	2.13/.35
0	.1	.021/.662	2.04/.36	2.10/.35
.1	.1	.020/.668	1.95/.37	2.02/.36

Table 10 reports listening ablations using the same 50-prompt protocol as the main tables. Description reformulation improves all four sound/music means from (b) to (c). The final model's SIM increase over speech-only training is accompanied by a higher speech-quality MOS mean. These comparisons report numerical means and std values, without claiming statistical significance.

Table 10: Supplementary listening ablations, mean$\pm$std. Speech uses LibriTTS test-clean; sound and music use the main evaluation sets. Dashes indicate subjective ablations not reported here.

Variant	Speech MOS	Sn. MOS-Q	Sn. MOS-T	Mu. MOS-Q	Mu. MOS-T
Speech-only	$3.94 \pm .07$	–	–	–	–
(a) Discrete	$3.65 \pm .09$	–	–	–	–
(b) Raw joint	$3.88 \pm .08$	$3.68 \pm .08$	$3.56 \pm .09$	$3.63 \pm .09$	$3.48 \pm .10$
(c) Description	$3.90 \pm .08$	$3.83 \pm .07$	$3.76 \pm .08$	$3.84 \pm .07$	$3.73 \pm .08$
(d) Symmetric	$3.98 \pm .07$	–	–	–	–
(e) A-MoME	$4.02 \pm .06$	$3.98 \pm .06$	$3.95 \pm .07$	$3.99 \pm .06$	$3.92 \pm .07$

C.7 Capacity and Corpus-Scale Reporting

Excluding the frozen VAE, the shared generator has 1.721 B parameters. One same-width speech residual FFN stack adds $28 \times 3 \times 2048 \times 6144 = 1{,}056{,}964{,}608$ parameters. Table 11 separates total capacity from the parameter sets used by each generation route. A-MoME concentrates extra capacity on speech while leaving the sound/music FFN computation unchanged. Route parameter counts and measured runtime costs are reported separately.

Table 11: Generator parameters, excluding the frozen VAE, in billions.

Architecture	Total	Speech route	Sound/music route
Shared backbone	1.721	1.721	1.721
Symmetric MoME (d)	3.835	1.721	1.721
A-MoME (e)	2.778	2.778	1.721

For context, the approximate public source-corpus sizes before training-split selection and filtering are 47.4 K hours for speech (LibriTTS and Emilia-English), 8.3 K for sound (VGGSound, AudioCaps, and WavCaps), and 12.1 K for music (8.3 K FMA plus 3.8 K MTG-Jamendo). These are source-corpus totals, not effective filtered training hours. The final joint model and the speech-only model use the same speech corpus; the joint mixture samples speech in 40% of examples. The comparisons in Table 3 do not show a SIM improvement from adding non-speech data alone; the improvement appears with modality-specific FFNs.

C.8 Inference Cost and Termination Behavior

We measure the shared generation path on one A800-80GB GPU with BF16, batch size 1, 24 Euler steps per block, KV caching, and 11-token blocks (approximately 1.022 s). The timed checkpoint does not load the speech residual expert. Timing includes flow generation and full VAE waveform decoding, but excludes model loading and file I/O. Table 12 reports this measured path.

Table 12: Measured shared-path inference cost; no inference optimization. Memory is peak allocated GPU memory.

Audio length (s)	Time (s), mean±std	RTF, mean±std	Peak (GiB)
5.11	$5.374 \pm .129$	$1.052 \pm .025$	4.56
10.22	$10.659 \pm .091$	$1.043 \pm .009$	4.86
20.43	$21.427 \pm .479$	$1.049 \pm .023$	5.45
40.87	$43.083 \pm .280$	$1.054 \pm .007$	6.63

Runtime is approximately proportional to duration over this range, with RTF near 1.05; this finite-range observation is not an asymptotic complexity claim. VAE decoding takes approximately 7–67 ms, so flow generation dominates. Generating and decoding a single 1.022-s block takes 1.149 ± 0.139 s. This is time to a first decodable latent block. The current script decodes after generating the full latent sequence and does not emit waveform blocks incrementally: a 10.22-s output becomes playable after approximately 10.66 s. Incremental waveform delivery additionally requires a compatible block-output and VAE decoding implementation.

For termination, we evaluate all $4{,}837$ LibriTTS test-clean examples at $\tau_{\text{stop}} = 0.95$ and a 40-s generation cap, without supplying reference duration to the generator. Table 13 compares generated and reference durations. Agreement with reference duration is a diagnostic rather than a direct accuracy measure: valid speech-rate and pause variation can change duration, and termination before the cap does not by itself prove content completeness.

Table 13: Duration agreement and termination on LibriTTS test-clean.

Metric	Result
Reference duration, mean / median	6.378 / 4.850 s
Generated duration, mean / median	6.270 / 4.783 s
Mean generated-minus-reference bias	-0.108 s
MAE / RMSE	1.302 / 3.437 s
Within ±1 / ±2 / ±5 s	68.47 / 85.80 / 96.38%
Shorter / longer by more than 1 s	24.54 / 6.99%
Reached cap without normal stopping	19/4,837 (0.39%)

The small aggregate bias coexists with a long tail. An early case generates 23.127 s against a 32.420-s reference; a short *You know.* case generates 39.985 s against a 0.280-s reference and reaches the cap. Both duration statistics and content inspection are needed to characterize these failures.

C.9 Additional Sound and Music Benchmarks

We evaluate Clotho and MusicCaps after excluding source-ID and acoustic-fingerprint overlaps with the training data. Table 14 provides a compact cross-benchmark check. The same model remains competitive on both additional benchmarks. These results concern sound or music prompts, not mixed speech–sound–music composition.

Table 14: Supplementary benchmark transfer; MOS-Q is mean$\pm$std.

Benchmark	System	FAD↓	CLAP↑	MOS-Q↑
Clotho	TangoFlux	3.08	.31	$3.76 \pm .05$
Clotho	AudioCALM	2.61	.34	$3.89 \pm .04$
MusicCaps	Stable Audio Open	2.41	.32	$3.91 \pm .08$
MusicCaps	AudioCALM	2.18	.35	$3.97 \pm .06$

D Asset Licenses and Use

The training and evaluation datasets, benchmarks, and pretrained models are publicly released and cited at first use in Section A.2 (data) and Section B (evaluators); we use them under their applicable research or open-source licenses. The auxiliary captioning service is closed-source, as discussed in Section H. The planned AudioCALM release includes code, weights, re-captioning prompts, and cached annotations, subject to upstream licenses; it does not re-host upstream audio datasets or pretrained checkpoints.

E Limitations

Our training mixture is restricted to English speech and publicly available sound and music corpora; non-English speech, singing, and underrepresented acoustic events require broader coverage. The scale study reaches 4 B backbone parameters, leaving larger models unexplored. Universal generation here means one model supporting TTS, sound, and music tasks separately. Mixed-source composition would require suitable training data and routing that can vary with the generated content, for example a soft gate on the speech residual conditioned on text and audio history. AR-Flow supports incremental latent generation, but the current script emits waveform audio only after full-sequence VAE decoding (Section C.8). The termination statistics expose long-tail duration deviations, including premature endings and occasional cap failures; duration agreement is not equivalent to content completeness. Longer speech may repeat or omit content, complex scenes can challenge event binding, and long-range musical structure needs further evaluation.

F Broader Impacts

A unified speech, sound, and music generator has positive applications in accessibility (text-to-speech for assistive devices), creative work (sound-design and music generation for independent creators), and research tooling (data augmentation for ASR and audio classification). The same capabilities raise misuse risks shared with the broader class of strong audio generators, most concretely zero-shot voice cloning and impersonation from a few seconds of reference audio, and the fabrication of plausible synthetic audio events that can be embedded in disinformation videos.

G Safeguards for Responsible Release

We will release AudioCALM under a research-use license that explicitly prohibits non-consensual voice cloning, impersonation of identified individuals, and use in surveillance, harassment, or fraud workflows. The voice prompts used in our speaker-similarity evaluation come exclusively from public datasets whose original release permits such use (LIBRITTS *test-clean* and SEEDTTS-EVAL), and the MOS study (Section B.3) collects no participant voice recordings, so this paper introduces no new voice-cloning targets. We do not claim that license clauses alone are sufficient against a determined adversary, and we regard parallel community work on synthetic-audio detection and content provenance as a necessary complement to this release.

H Declaration of LLM Usage

We use GEMINI 3 PRO only to construct offline training descriptions (Section A.3). Its contribution is examined separately from model architecture in Section C.5. Training samples original and long conditions with equal probability. Evaluation uses the original benchmark transcripts/captions, and inference never queries the auxiliary captioning MLLM; AudioCALM's own LLM backbone remains part of the generator. We plan to release the prompts, cached descriptions, and quality-control procedure so downstream training and evaluation can use fixed annotations without Gemini access. This enables reuse of the annotation cache, but does not guarantee exact regeneration of those annotations from a closed service. The authors also used a writing assistant for copy editing.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction accurately summarize the paper's contributions, including the continuous autoregressive framework, AR-Flow attention, and A-MoME design for unified speech, sound, and music generation. These claims are stated in Section 1 and supported by the method in Section 3 and the empirical 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: The paper discusses limitations in Section 5 and Appendix E.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: The paper does not present formal theoretical results requiring mathematical proofs; the contributions are methodological and empirical.

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: The paper describes the model, training objective, and inference procedure in the main text, and Appendix A further details the VAE, training corpora, conditioning pipeline, and full hyperparameters. Appendix B defines the evaluation metrics and MOS protocol.

Guidelines:

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

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

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

   Answer: [No]

   Justification: The current submission does not yet include an open-source code release or a full reproducibility package. We plan to release code and supporting documentation 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: Section 4.1 summarizes the training and evaluation setup, and Appendix A and Appendix B provide full implementation details, hyperparameters, metrics, and human-evaluation procedures.

   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: Tables 1 and 2 report subjective scores as mean$\pm$std for $N = 50$ prompts per system, with the protocol in Appendix B. We report numerical differences without inferring statistical significance from the std values alone.

   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: Section 4.1 and Appendix A report the training hardware, optimization setup, precision, and distributed-training configuration used in the experiments, which provide the main compute context needed for reproducibility.

   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: To the best of our knowledge, the research conforms to the NeurIPS Code of Ethics. The paper also discusses potential misuse risks and responsible deployment considerations in Section 5.

   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: The paper discusses both positive and negative societal impacts in Section 5 and Appendix F, including accessibility and creative uses as well as risks related to voice cloning, impersonation, and disinformation.

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: Appendix G describes our planned research-use license that prohibits non-consensual voice cloning, impersonation, and surveillance/harassment use; the SIM evaluation uses only public datasets with re-distribution permissions and introduces no new voice-cloning targets; we further point to community work on synthetic-audio detection and content provenance as complementary mitigations.

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: The paper cites the original datasets, pretrained models, and prior systems at first use, including the training corpora in Appendix A.2 and the evaluation models in Appendix B. Appendix D states that we use only publicly released assets and do not re-host upstream datasets or pretrained checkpoints.

Guidelines:

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

13. **New assets**

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

Answer: [N/A]

Justification: The current submission does not release new public assets alongside the paper. If such assets are released after publication, their documentation will be included in the released repository.

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: The paper describes the MOS study protocol in Appendix B and operational details in Appendix B.3, including participant instructions, consent, compensation, and the availability of interface screenshots and full study materials in the supplemental material.

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: [Yes]

    Justification: Appendix B.3 describes the anonymous and minimal-risk nature of the listening study, the information disclosed to participants through the consent process, and the compensation procedure.

    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: The paper describes LLM usage in Section 3. In particular, the proposed model is built by initializing from a pretrained LLM backbone (Qwen3) and adapting it to continuous audio generation, while an audio-conditioned multimodal LLM is additionally used offline to produce long-form conditioning descriptions for training.

    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.