

Zen AI Model Family

Zen-Musician

Packaging YuE for Long-Form Lyrics-to-Song Generation in the Zen Stack

Technical Report v2025.01

Hanzo AI Research Team*

Zoo Labs Foundation†

January 2025

Abstract

Zen-Musician is a packaging and integration of **YuE**, the open long-form music generation foundation model developed by **M-A-P (Multimodal Art Projection)** and HKUST and released under the permissive **Apache-2.0** license [1, 2]. We make no claim of a from-scratch model: Zen-Musician redistributes YuE’s published weights unmodified and exposes them through the Zen tooling, runtime, and API surface. This report documents the *real* architecture so that integrators understand what they are deploying. YuE is a lyrics-to-song system: given lyrics together with a short genre/style description, it generates full songs containing both sung vocals and accompaniment. It is built on a decoder-only LLaMA-2-style autoregressive transformer that operates over discrete audio tokens. The flagship Stage-1 checkpoint (e.g. `m-a-p/YuE-s1-7B-anneal-en-cot`) has approximately 7B parameters and performs track-decoupled next-token prediction to produce vocal and accompaniment token streams; a separate Stage-2 model (approximately 1B parameters) refines and upsamples those tokens toward the final waveform [3]. According to its authors, YuE can generate up to roughly five minutes of music while maintaining lyrical alignment, coherent musical structure, and vocal melodies with appropriate accompaniment, and supports style transfer and reference conditioning through an in-context-learning (ICL) variant [1]. This document supersedes an earlier draft that incorrectly described a from-scratch “Zen MoDE” architecture and reported fabricated benchmarks; those claims were false and have been removed.

Contents

1	Introduction	3
1.1	What Zen-Musician Is (and Is Not)	3
1.2	Correcting the README	3
1.3	The Problem YuE Addresses	3
1.4	Model Overview	4
2	Architecture (as published by YuE’s authors)	4
2.1	Audio Tokenization	4
2.2	Stage-1: LLaMA-2-style Autoregressive Generator	4
2.3	Stage-2: Refinement / Upsampling	4
2.4	Conditioning Interface	4
2.5	Checkpoint Family	5
3	Capabilities Reported Upstream	5

*research@hanzo.ai

†foundation@zoo.ngo

4	Integration in the Zen Stack	5
4.1	Resource Notes	5
5	Usage	6
6	Provenance and Attribution	6
7	Conclusion	6

1 Introduction

1.1 What Zen-Musician Is (and Is Not)

Zen-Musician is the Zen stack’s integration of **YuE** ([multimodal-art-projection/YuE](#)), an open foundation model for full-song generation released by M-A-P and HKUST under the **Apache-2.0** license [2, 1]. The Apache-2.0 license permits redistribution, modification, and commercial use subject to its attribution and notice requirements; YuE’s authors explicitly encourage creators to incorporate model outputs, with attribution to “YuE by HKUST/M-A-P” recommended [2].

To be explicit about provenance:

- Zen-Musician **does not** introduce a new architecture. There is no “Zen MoDE,” no “Section-Aware Transformer,” and no unified MIDI/audio model. Those concepts appeared in a prior draft of this report and were not real.
- Zen-Musician **does not** report its own benchmark numbers. The FAD, MOS, genre-consistency, structure-F1, and accompaniment-latency tables in the earlier draft were fabricated and have been deleted. Where this report mentions evaluation, it attributes claims to YuE’s authors and does not invent metrics.
- Zen-Musician **is** a redistribution of YuE’s published weights, wrapped with Zen tooling (a consistent loader, runtime integration, and the Zen API surface) so that YuE can be used alongside other Zen models.

1.2 Correcting the README

A previous Zen-Musician description listed the model as “1B” parameters and as a MIDI/symbolic system. Both were wrong. The packaged flagship is the **7B Stage-1** checkpoint; the “1B” figure corresponds only to the *Stage-2* refinement/upsampling model, not the main generator. YuE generates *audio* (sung vocals plus accompaniment) from *lyrics and genre text*. It is not a MIDI model and does not generate symbolic scores.

1.3 The Problem YuE Addresses

Open music models have historically struggled with *long-form* generation: producing a coherent multi-minute song with intelligible, aligned lyrics, a sensible verse/chorus structure, and a singing voice that stays on top of a suitable accompaniment. YuE targets this lyrics-to-song setting directly, scaling an autoregressive transformer over audio tokens to song-length outputs [1].

1.4 Model Overview

Property	Value
Upstream model	YuE (M-A-P / HKUST)
License	Apache-2.0 [2]
Task	Lyrics + genre $\rightarrow$ full song (vocals + accompaniment)
Backbone	Decoder-only, LLaMA-2-style autoregressive transformer [1, 4]
Stage-1 generator	$\approx$ 7B parameters (e.g. <code>YuE-s1-7B-anneal-en-cot</code>)
Stage-2 refiner	$\approx$ 1B parameters (token refinement / upsampling)
Representation	Discrete audio tokens (xcodec-style tokenizer)
Languages	English, Mandarin, Cantonese, Japanese, Korean (checkpoint-specific)
Conditioning	Lyrics with section tags; genre/style tags; optional audio reference (ICL)
Reported length	Up to $\approx$ 5 minutes of music [1]

Table 1: Zen-Musician = packaged YuE: upstream specifications

2 Architecture (as published by YuE’s authors)

The description below summarizes YuE’s design from its paper and model documentation [1, 2, 3]. Zen-Musician does not alter this architecture.

2.1 Audio Tokenization

YuE operates over *discrete audio tokens* rather than raw waveforms. Audio is encoded into token sequences by an xcodec-style neural audio codec; the language model predicts tokens, and the decoder side reconstructs audio from them [2]. This lets a standard autoregressive transformer model music the same way a text LLM models language.

2.2 Stage-1: LLaMA-2-style Autoregressive Generator

The Stage-1 model is a **decoder-only, LLaMA-2-style transformer** [1, 4] with approximately **7B parameters**. Its central design choice, as described by the authors, is **track-decoupled (“dual-track”) next-token prediction**: rather than modeling a single dense mixture signal, it predicts *vocal* and *accompaniment* token streams in a decoupled fashion, which the authors report helps the model keep the singing voice coherent over long contexts [1]. The authors further describe *structural progressive conditioning* to maintain lyric alignment across long-form generation [1].

2.3 Stage-2: Refinement / Upsampling

A separate **Stage-2** model (approximately **1B parameters**) takes the Stage-1 token output and performs residual refinement / upsampling toward the final audio, improving fidelity before waveform reconstruction [3, 2]. This is the source of the “1B” figure that an earlier Zen description mistook for the size of the whole system.

2.4 Conditioning Interface

YuE is prompted with three kinds of input [2]:

- **Genre / style tags**: a short free-text description (the project suggests including genre, instrumentation, mood, vocal gender, and timbre).

- **Lyrics:** the song’s lyrics, annotated with structural labels such as `[verse]` and `[chorus]`, organized into sessions/segments.
- **Optional audio reference** (ICL variants): a reference clip used for style transfer or voice-style conditioning via in-context learning [1].

2.5 Checkpoint Family

YuE ships multiple Stage-1 checkpoints, differentiated by language coverage and by mode — a chain-of-thought (`cot`) variant and an in-context-learning (`icl`) variant for reference-conditioned generation [2]. Example flagship checkpoint: `m-a-p/YuE-s1-7B-anneal-en-cot`.

3 Capabilities Reported Upstream

The following are claims made by YuE’s authors [1]; Zen-Musician inherits them by redistributing the same weights and does not independently re-benchmark them in this report.

- **Long-form songs:** generation of up to roughly five minutes of music with maintained lyrical alignment and coherent musical structure.
- **Vocals with accompaniment:** engaging vocal melodies sung over an appropriate instrumental backing, produced by the dual-track scheme.
- **Style transfer / reference conditioning:** an in-context-learning mode enabling style transfer (e.g. genre conversion while preserving accompaniment) and reference-based generation.
- **Multilingual coverage:** checkpoints for English, Mandarin, Cantonese, Japanese, and Korean.

The YuE paper reports that the system “matches or even surpasses some of the proprietary systems in musicality and vocal agility,” and evaluates music *understanding* on the MARBLE benchmark [1]. We reproduce this as an attributed upstream claim only; no numeric scores are asserted here, and none should be attributed to Zen.

4 Integration in the Zen Stack

Zen-Musician’s contribution is purely **packaging and integration**, not modeling:

- **Distribution:** the YuE Stage-1 and Stage-2 weights are redistributed under Apache-2.0 with upstream attribution and license/notice files preserved.
- **Loader and runtime:** a consistent entry point so YuE can be invoked the same way as other Zen models.
- **API surface:** a thin wrapper exposing YuE’s lyrics+genre conditioning and two-stage pipeline.

4.1 Resource Notes

YuE is a 7B-class generator over long audio-token contexts; full-song generation is GPU-memory intensive, and the upstream project documents substantial VRAM requirements for long outputs as well as multi-stage (Stage-1 then Stage-2) inference [2]. Integrators should budget accordingly rather than expecting real-time, low-memory operation.

5 Usage

The snippet below is illustrative of the lyrics-to-song interface. Refer to the upstream YuE repository [3] for the authoritative inference pipeline, prompt format, and Stage-1/Stage-2 invocation.

```
1 from zen import ZenMusician
2
3 # Loads the packaged YuE Stage-1 (7B) generator; Stage-2 refinement
4 # runs as part of the pipeline.
5 model = ZenMusician.from_pretrained("zenlm/zen-musician") # wraps m-a-
6                 p/YuE-s1-7B-*
7
8 genre = "inspiring_pop_female_vocal_bright_piano_uplifting"
9
10 lyrics = """
11 [verse]
12 Lines of the verse go here
13 Another line that scans and rhymes
14 [chorus]
15 The hook that repeats and stays
16 Carrying the song through its days
17 """
18
19 song = model.generate(
20     genre_tags=genre,
21     lyrics=lyrics,
22     # optional reference audio enables ICL style/voice conditioning
23     # reference_audio="reference.wav",
24 )
25 song.save("song.wav")
```

Listing 1: Zen-Musician (packaged YuE) lyrics-to-song generation

6 Provenance and Attribution

Zen-Musician redistributes YuE. Required and recommended attribution:

- **Model:** YuE — Open Full-Song Music Generation Foundation Model.
- **Authors / affiliations:** M-A-P (Multimodal Art Projection) and HKUST (R. Yuan, H. Lin, S. Guo, G. Zhang, et al.; Y. Guo) [1].
- **License:** Apache-2.0 [2].
- **Suggested credit:** “YuE by HKUST/M-A-P.”
- **Source:** github.com/multimodal-art-projection/YuE; weights at huggingface.co/m-a-p.

7 Conclusion

Zen-Musician is the Zen stack’s packaging of YuE, an Apache-2.0 long-form lyrics-to-song model from M-A-P and HKUST. The real system is a decoder-only, LLaMA-2-style autoregressive transformer (Stage-1 $\approx 7B$) that performs track-decoupled next-token prediction over discrete audio tokens to generate vocals and accompaniment, followed by a $\approx 1B$ Stage-2 refinement/up-sampling model. Conditioning is by lyrics and genre/style text, with an optional ICL mode for

reference-based style transfer. All credit for the architecture, training, and reported capabilities belongs to YuE’s authors; Zen’s contribution is integration and distribution under the upstream license, with attribution. The from-scratch architecture and benchmark claims in the earlier draft were fabricated and have been removed.

References

- [1] R. Yuan, H. Lin, S. Guo, G. Zhang, et al., “YuE: Scaling Open Foundation Models for Long-Form Music Generation,” arXiv:2503.08638, 2025. <https://arxiv.org/abs/2503.08638>
- [2] M-A-P, “YuE model cards (Apache-2.0),” Hugging Face. <https://huggingface.co/m-a-p/YuE-s1-7B-anneal-en-cot>
- [3] Multimodal Art Projection, “YuE: Open Full-song Music Generation Foundation Model,” GitHub. <https://github.com/multimodal-art-projection/YuE>
- [4] H. Touvron et al., “Llama 2: Open Foundation and Fine-Tuned Chat Models,” arXiv:2307.09288, 2023.