Ruben: Multimodal Music Composition via Tree-Structured Semantic Representation

David Korcak, Frantisek Kmjec

Abstract. Ruben transforms multimodal signals—text, images, video, audio—into music through a novel tree-structured semantic representation. A multimodal LLM (Kimi K2.5) decomposes creative intent into an editable hierarchical tree, which is then assembled into a coherent prompt for ACE-Step 1.5 (text-to-audio diffusion). Key contributions: (1) a two-pass pipeline (analysis → human editing → assembly) that separates creative decomposition from prompt synthesis, (2) an audio-to-text feedback loop enabling style transfer through natural language rather than learned embeddings, and (3) an interactive tree editor providing interpretability and fine-grained control over generation.

1 Architecture

Pipeline. React frontend → FastAPI backend → ACE-Step 1.5 on remote GPU. Async job-based; end-to-end latency 45–90s (generation dominates at 30–60s).

Inputs. Text (direct prompt). Images (base64, Kimi vision extracts color palette, mood, spatial qualities → musical characteristics). Video (6 keyframes via OpenCV with temporal annotations). Audio references (ACE-Step `/lm/understand` extracts caption, BPM, key, lyrics → text representation for Kimi).

Tree representation. The core abstraction is a hierarchical JSON tree of musical characteristics with flexible, LLM-decided schema. Typical branches include Emotional Landscape, Instrumentation, Sonic Production, Temporal Dynamics, Harmonic Language, and Narrative Arc. Internally represented as recursive `SongNode` objects (name, value, metadata, children).

1.1 Two-Pass Generation

The critical insight: mechanical flattening of a tree into comma-separated tags produces incoherent prompts. Ruben instead uses two distinct LLM passes:

Pass 1 — Analysis. All inputs → Kimi + `SYSTEM_PROMPT` → structured tree. The tree is rendered in the frontend for human inspection and editing.

Pass 2 — Assembly. Edited tree → Kimi + `ASSEMBLY_PROMPT` → coherent caption + lyrics optimized for ACE-Step. The assembly pass resolves cross-branch convergences and tensions (e.g., when mood and genre imply conflicting tempos), specifies instruments with production-quality descriptors, and outputs a unified musical narrative.

1.2 Audio-to-Text Feedback

Reference audio → ACE-Step audio understanding → text caption + metadata → injected into Kimi's analysis context. This enables style transfer through the text-to-audio pathway: a lo-fi hip-hop reference produces trees with tape saturation, vinyl crackle, and relaxed swing—without requiring audio embeddings or fine-tuning.

1.3 Models

Kimi K2.5 (via OpenRouter): long-context (128k+) multimodal LLM with text + vision, structured JSON output, and chain-of-thought reasoning. Serves two roles: tree analysis and prompt assembly.

ACE-Step 1.5 [?]: open-source music diffusion model (DiT + LM). Turbo variant, 8-step inference, <2s per song on A100. Supports text-to-music, audio understanding, and metadata control (BPM, key, time signature, duration up to 600s).

2 Interface

Input panel: 2×2 grid (text, audio, images, video) with drag-and-drop. Tree editor: tabbed `TreeStack` with hover toolbars, color-coded nodes, diff tracking against previous generations, and markdown export. Controls: duration (10–240s), BPM, key, time signature. History panel: previous generations with audio player, tree diffs, and restore.

3 Results

Multimodal inputs produce richer trees than any single modality. A frozen fjord photograph combined with "lonely but hopeful" text generated a D Dorian ambient piece with felt piano, glacial pads, and a narrative arc from isolation to warmth—detail that neither input alone would produce. Reference audio measurably influences generation: a lo-fi hip-hop clip shifted instrumentation toward tape-saturated Rhodes and dusty drums. Tree editing propagates to audio: replacing "felt piano" with "music box" changed timbral character while preserving mood and structure. Assembly consistently outperforms mechanical flattening, which produces tag-soup prompts with no narrative coherence.

4 Discussion

Contributions. (1) Tree-structured semantic representation as a reusable creative primitive. (2) Two-pass generation separating decomposition from synthesis. (3) Audio-to-text feedback enabling style transfer via natural language. (4) Multimodal orchestration through a single LLM. (5) Human-in-the-loop editing at arbitrary granularity.

Limitations. Duration capped at 240s for reliability (ACE-Step supports 600s). Async polling adds latency. Internet-dependent (Kimi via OpenRouter). Assembly can fail on contradictory inputs. Audio understanding limited for experimental music.

Scale vision. Trees become shareable templates, recommendation primitives (cluster by topology), and A/B testing units. Multi-gen stitching extends duration. Tree merging enables collaborative composition.

5 Conclusion

Ruben demonstrates that multimodal → semantic tree → audio is a viable paradigm for controllable AI music generation. The tree provides interpretability and editability absent from flat-prompt systems. Two feedback loops—audio-to-text style transfer and two-pass assembly—point toward a general architecture for human-AI creative collaboration.

References

[1] Gong et al. ACE-Step 1.5. `https://github.com/ace-step/ACE-Step-1.5`, 2026.
[2] Moonshot AI. Kimi K2.5. `https://kimi.moonshot.cn`, 2025.
[3] OpenRouter. `https://openrouter.ai`, 2025.
[4] Ho et al. Denoising Diffusion Probabilistic Models. NeurIPS, 2020.
[5] OpenAI. GPT-4 Technical Report. arXiv:2303.08774, 2023.
[6] Agostinelli et al. MusicLM. arXiv:2301.11325, 2023.
[7] Kong et al. HiFi-GAN. TASLP, 2020.
[8] Verma & Chetty. Affective Music. ISMIR, 2018.