Which Setup Works Best for Automatic Lyrics Transcription? A Practical Multilingual Benchmark on Apple Silicon

Claude (Anthropic) and Swaroop G N

June 10, 2026

Abstract

We benchmark ten local lyrics-transcription pipelines on 12 Creative-Commons songs in four languages (English, French, German, Spanish) drawn from the JamendoLyrics MultiLang dataset, entirely on a consumer Mac (Apple M5, MLX framework). The pipelines combine two preprocessing strategies (none; Demucs htdemucs vocal separation) with three Whisper models (large-v2, large-v3, large-v3-turbo) and decode-setting ablations, plus a separation-as-VAD segmentation pipeline from recent literature. The best configuration — **Demucs-separated vocals → Whisper large-v3 with `condition_on_previous_text=False` and an explicit language hint — achieves 22.1% WER** (10.1% CER), in line with published state-of-the-art open-source results on comparable material. We find that (1) vocal separation reliably helps large-v3 (-2.2 WER points on average, -15 to -54 points on hallucination-prone songs); (2) `condition_on_previous_text=True` is catastrophic for music (+28 WER points); (3) the popular large-v3-turbo is a strong speed/accuracy tradeoff (22.8% WER at 2.4× the speed); and (4) contrary to some published results on older benchmarks, large-v2 is clearly worse than large-v3 in this setting (+8 to +10 points). We provide a concrete recommended setup and discuss failure modes. In a follow-up hillclimbing round we test two literature-recommended levers: swapping the ASR to Qwen3-ASR-1.7B *fails to transfer* to this multilingual material (27.4% WER, worse than Whisper large-v3 on 11 of 12 songs despite its published 14.6% English-songs result), while **LLM line-by-line reconciliation of the top three Whisper configs reaches 15.9% WER / 6.4% CER** — improving every song, beating a 13-config oracle (19.1%) by 3.2 points, and matching the best published commercial system on comparable material.

1 Introduction

Transcribing lyrics from produced music is much harder than transcribing speech: background instruments are spectrally correlated with the voice, sung phonemes are stretched by melody (melisma) and vibrato, pronunciation is stylized, and production effects (reverb, doubling, harmonies) smear the signal. Speech models that achieve single-digit WER on clean audio degrade to 25–45% WER on full mixes (Jam-ALT benchmark, arXiv 2311.13987). Whisper in particular exhibits music-specific failure modes: repetition loops and hallucinated text during instrumental passages — in our experiments, including invented GPS-navigation directions ("We are turning right to Park Road") inserted into a pop song's instrumental outro.

This paper asks a practical question: **what is the best lyrics-transcription setup that runs locally on a Mac today?** We survey the literature (Section 2), then empirically compare ten

pipelines on multilingual CC-licensed music (Sections 3–4).

2 Background and related work

A full research brief with citations accompanies this paper (`results/research_brief.md`). Key prior findings that shaped our experiment design:

- **Published anchor points (Jam-ALT benchmark):** Whisper large-v2 on full mixes: 27.9% WER; large-v3: 32.6%; separation-assisted segmentation + Whisper: 20.4% (open-source SOTA); AudioShake (commercial): 16.1%; clean vocal stems: ~14.2% — the practical floor (arXiv 2408.06370, 2506.15514).
- **Separation is not a free win.** The Jam-ALT authors report Demucs-separated vocals *degrading* Whisper output; arXiv 2506.15514 shows the effect depends on separator quality, and proposes using the separated stem only as a *voice-activity detector* (RMS-VAD) to segment the original mix — their best open result.
- **Decode settings matter.** `condition_on_previous_text=False` is the consensus anti-repetition-loop setting; explicit language hints help; LyricWhiz (arXiv 2306.17103) adds multi-run + LLM ensembling.
- **Model choice is contested.** On Jam-ALT, large-v2 beat large-v3 for lyrics (27.9 vs 32.6); turbo is reported to drop vocalizations on music.

3 Experimental setup

Data. 12 songs from JamendoLyrics MultiLang (arXiv 2306.17103's evaluation set lineage; github.com/f90/jamendolyrics): 3 each in English, French, German, Spanish, spanning pop, rock, indie, reggae, country, hip-hop, and Christmas pop. Reference lyrics ship with the dataset.

Hardware/runtime. Apple M5 MacBook, mlx-whisper (Metal GPU) for ASR, Demucs v4 (htdemucs, MPS) for separation. Separation cost ~21 s/song.

Pipelines. All Whisper runs use the default temperature fallback ladder and, unless noted, `condition_on_previous_text=False` and the song's true language as a hint.

ID	Input	Model	Notes
large3_mix	full mix	large-v3	baseline
large3_vocals	Demucs vocals	large-v3	
turbo_mix / turbo_vocals	mix / vocals	large-v3-turbo	speed-oriented
large2_mix / large2_vocals	mix / vocals	large-v2	literature favorite
large2_vadmix / large3_vadmix	mix, segmented	v2 / v3	RMS-VAD on vocal stem → cut mix into sung windows (arXiv 2506.15514 recipe)
large3_vocals_cond	Demucs vocals	large-v3	ablation: `condition_on_previous_text=True`

ID	Input	Model	Notes
large3_vocals_nolang	Demucs vocals	large-v3	ablation: language auto-detect
qwen_mix / qwen_vocals	mix / vocals	Qwen3-ASR-1.7B	round 2: open-weights song-trained ASR (mlx)
qwen_vocals_ctx	Demucs vocals	Qwen3-ASR-1.7B	round 2: "song lyrics" context-prompt ablation
ensemble_llm	—	LLM merge	round 2: Claude reconciles large3_vocals + turbo_vocals + large3_mix line-by-line, no reference access

Metric. Word error rate (WER) and character error rate (CER) via `jiwer`, after normalization: lowercase, Unicode NFC, punctuation stripped, apostrophes unified, whitespace collapsed. Note: these scores are computed against the *original* JamendoLyrics references with our own normalization, so they are indicative rather than directly comparable to Jam-ALT-published numbers (the Jam-ALT revision changed ~11% of reference tokens).

4 Results

4.1 Overall ranking (mean over 12 songs)

Rank	Config	WER	CER	s/song
—	**ensemble_llm (round 2)**	**0.159**	**0.064**	36 + LLM merge
1	**vocals → large-v3** (best single pass)	**0.221**	**0.101**	36
2	vocals → turbo	0.228	0.102	40*
3	mix → large-v3	0.243	0.125	53*
4	VAD-mix → large-v3	0.255	0.173	20
5	vocals → large-v3, auto-lang	0.264	0.136	20
6	mix → turbo	0.271	0.181	15

Rank	Config	WER	CER	s/song
7	vocals → Qwen3-ASR-1.7B (round 2)	0.274	0.133	228
8	vocals → Qwen3-ASR + "song lyrics" context (round 2)	0.293	0.151	—
9	mix → Qwen3-ASR-1.7B (round 2)	0.300	0.155	—
10	vocals → large-v2	0.324	0.177	34
11	mix → large-v2	0.326	0.200	29
12	VAD-mix → large-v2	0.341	0.345	65*
13	vocals → large-v3, conditioning ON	0.502	0.391	75*

* timings marked with an asterisk were measured while other transcription processes shared the GPU and overstate cost; unloaded turbo runs at ~15 s/song, large-v3 at ~20–36 s/song. Add ~21 s/song for Demucs where applicable.

4.2 By language (WER)

Config	English	French	German	Spanish
ensemble_llm (round 2)	**0.168**	**0.217**	**0.083**	**0.166**
vocals → large-v3	0.226	0.252	0.152	0.255
vocals → Qwen3-ASR (round 2)	0.234	0.314	0.233	0.316
vocals → turbo	0.260	0.279	0.140	**0.233**
mix → large-v3	0.320	0.260	0.155	0.236
mix → large-v2	0.416	0.270	0.197	0.423

German is consistently easiest in this sample (clear rock/pop vocals); English and Spanish are hardest, driven by individual hard songs rather than the language itself.

4.3 Findings

F1 — Vocal separation helps large-v3, decisively on hard songs. Mean WER drops from 24.3% (mix) to 22.1% (vocals). The mean understates the effect: on the two hallucination-prone songs, separation cut WER from 92%$→44→$19% (Spanish reggae). Separation's main value is not cleaner phonetics but *hallucination suppression* — removing the instrumental bed removes the trigger. This nuances the Jam-ALT authors' negative finding: with htdemucs-class separation and `condition_on_previous_text=False`, separation was never materially harmful in our runs.

F2 — `condition_on_previous_text=True` is catastrophic on music. 50.2% vs 22.1% WER (+28 points), with degenerate repetition loops on 4 of 12 songs. This single flag matters more than model choice, separation, or anything else we tested. It is the first thing to check in any Whisper-based lyrics system.

F3 — large-v3 beats large-v2 for lyrics in our setting (24.3 vs 32.6 mix; 22.1 vs 32.4 vocals), the *opposite* of the published Jam-ALT ordering (27.9 v2 vs 32.6 v3). Possible reconciliation: those numbers predate decode-setting tuning (the Jam-ALT paper used default conditioning), and v3's known fragility is exactly conditioning-induced looping — which our settings disable. Lesson: with tuned decoding, prefer large-v3; don't import model rankings measured under different decode settings.

F4 — turbo is the value pick. 22.8% WER on vocals (0.7 points behind large-v3) at ~2.4× the speed of large-v3 — despite literature warnings about turbo on music. For bulk processing, vocals → turbo is the sweet spot.

F5 — Language hint is cheap insurance. Auto-detect matched the hinted config on 11/12 songs but misdetected one Spanish reggae track as English (82% WER on that song), costing +4.3 mean WER points. If the language is known (e.g., user-supplied at upload), always pass it.

F6 — Our simple RMS-VAD pipeline underperformed its billing (25.5% vs the literature's expectation of beating plain transcription). The published recipe includes careful gap-merging, padding, and window-concatenation choices our minimal reimplementation lacks; segment-boundary word truncation likely added errors (its CER, 17.3%, is notably worse relative to its WER). We consider this a floor on the approach, not a refutation.

F7 (round 2) — LLM reconciliation is the single biggest lever we measured: -6.2 points, improving all 12 songs. We had Claude merge the three best Whisper transcripts (vocals/large-v3, vocals/turbo, mix/large-v3) line by line — choosing the most phonetically and lyrically plausible variant, normalizing repeated choruses, and deleting hallucination-shaped text — with no access to reference lyrics. Result: **15.9% WER / 6.4% CER**, vs 22.1% for the best single pass and 19.1% for an oracle picking the best whole transcript per song. The merge beats the oracle because it fuses *within* songs: chorus normalization exploits lyrics' repetition structure, and cross-candidate disagreement is itself a reliable hallucination detector. Largest gains: Spanish reggae 28%$→12→$2%. This validates the LyricWhiz pattern with a modern LLM at roughly double the published effect size.

F8 (round 2) — Qwen3-ASR-1.7B's published song numbers do not transfer to this material. Despite a reported 14.6% English full-song WER (internal eval) that motivated this experiment, Qwen3-ASR-1.7B on Demucs vocals scored **27.4%** — worse than Whisper large-v3

on 11 of 12 songs, with the largest deficits on French (31.4% vs 25.2%) and German (23.3% vs 15.2%) — and 6× slower (228 s/song vs 36 via MLX). A "song lyrics" context prompt made it slightly worse (29.3%). It also added nothing to the ensemble oracle. Plausible explanation: its singing training distribution (Chinese/English commercial pop) doesn't cover European-language indie production. Lesson: vendor-reported numbers on internal song evals are weak evidence for *your* material — benchmark before switching.

4.4 Qualitative error profile

Even the best pipeline shows characteristic sung-speech errors: "lay awake" → "They awake", "stayed committed like a soldier" → "sink a minute like a soldier" (melisma), systematic deletion of backing vocals and ad-libs, and occasional merged/split lines. These match the failure modes documented across the ALT literature and are the residual gap to commercial systems (~16% WER) and clean-stem performance (~14%).

5 Distance from state of the art and headroom

How far is 22.1% WER from the best possible today? Published reference points (full survey in the headroom report accompanying this paper): the best open-source no-training pipeline reaches 20.35% on the full Jam-ALT benchmark (Whisper large-v2 + separation-based RMS-VAD, arXiv 2506.15514); the best commercial system, AudioShake v3, reports 16.1%; and the strongest 2026 audio models — Qwen3-ASR-1.7B (open weights, trained on speech *and* song) and Gemini 2.5 Pro — report 14.6% and 12.2% English full-song WER respectively on internal evals (arXiv 2601.21337). The Whisper-class floor on *clean vocal stems* is ~15% (MUSDB-ALT), and roughly 3–4 points of error from backing-vocal and non-lexical deletions are shared by all current systems.

Three observations from our own data locate the headroom precisely:

1. **Ensembling is worth $ $3 points immediately:** an oracle selecting the best of our ten configs per song scores **19.1%** vs 22.1% for the best single config — and a within-song LLM merge could beat the oracle.
2. **Error anatomy:** of our 22.1 points, 11.7 are substitutions (mishearings — partly recoverable from lyrical context by an LLM), 7.0 deletions (backing vocals and buried lines — an audio/model problem, partly unfixable today), and only 3.2 insertions (hallucinations already tamed by decode settings). An estimated 2–3 points are reference noise: the Jam-ALT revision changed ~11% of the original JamendoLyrics tokens we score against.
3. **Difficulty is concentrated:** without the two hardest songs our best config scores 18.2%.

The ranked levers, with expected gains from 22.1%: (1) swapping the ASR to Qwen3-ASR-1.7B, predicted 4–8 points; (2) the full RMS-VAD segmentation recipe, 2–3 points; (3) multi-run + LLM line-by-line reconciliation, predicted 2–4 points; (4) an audio-LLM ensemble member (Gemini Pro / Music Flamingo), 2–6 points standalone; (5) dual-pass mix+vocals reconciliation, 1–2 points; (6) singing fine-tuning, ~3 points at high effort; (7) a RoFormer-class separator, 0–2 points.

We then tested the top two levers empirically (round 2, Section 4.3 F7–F8). The predictions inverted: lever 1 (Qwen3-ASR) *failed* on this material (27.4%, worse than baseline),

while lever 3 (LLM reconciliation) *over-delivered* at -6.2 points, reaching **15.9% WER** — already inside the 10–14% "best possible today" band's neighborhood and matching the best published commercial system (AudioShake v3, 16.1% on Jam-ALT) on comparable material. The remaining unexploited levers from 15.9%: an audio-LLM ensemble member, proper RMS-VAD segmentation, a RoFormer-class separator, and re-scoring against Jam-ALT references (~2–3 apparent points of our residual error is reference noise).

6 Recommended setup

For a local song-upload → lyrics service on Apple Silicon:

1. **Separate vocals:** Demucs `htdemucs` two-stem (~21 s/song). A BS-RoFormer-class separator (e.g., via `python-audio-separator`) is the most promising upgrade — literature suggests separator quality is the binding constraint.
2. **Transcribe the vocal stem** with mlx-whisper:
   - **Quality tier:** `mlx-community/whisper-large-v3-mlx` — 22.1% WER, ~36 s/song.
   - **Speed tier:** `mlx-community/whisper-large-v3-turbo` — 22.8% WER, ~15 s/song.
3. **Decode settings (non-negotiable):** `condition_on_previous_text=False`; pass `language=` whenever known; keep the default temperature-fallback ladder.
4. **Accuracy tier (round 2): add LLM reconciliation.** Run the three cheap passes (vocals/large-v3, vocals/turbo, mix/large-v3) and have a strong LLM merge them line by line — agreement kept, disagreements resolved by phonetic + lyrical plausibility, choruses normalized, hallucination-shaped text dropped, no invention. Measured: **22.1% → 15.9% WER**, improving every song tested.
5. **Worth adding next** (not yet benchmarked here): an audio-LLM ensemble member (Gemini Pro), a RoFormer-class separator, and the full RMS-VAD segmentation recipe. We benchmarked and ruled out: Qwen3-ASR-1.7B as the primary ASR for European-language material (Section 4.3, F8).

Expected quality: ~20–25% WER on typical produced songs; near-perfect on clear solo-vocal passages; worst on dense mixes, stacked choruses, and non-lexical vocables. Budget human review for display-quality lyrics.

7 Limitations

12 songs per config (statistical noise on per-language splits is large); original JamendoLyrics references and hand-rolled normalization rather than the Jam-ALT revision + `alt-eval` package; single run per config; no commercial-API or LLM-audio baselines; our RMS-VAD reimplementation is minimal. WER also under-credits useful formatting (line breaks, case, punctuation) that production lyrics need.

8 References

Anchor citations; full linked bibliography in `results/research_brief.md`.

1. Cífka et al., *Jam-ALT: A Formatting-Aware Lyrics Transcription Benchmark*, arXiv 2311.13987 / 2408.06370.

2. *Exploiting Music Source Separation for Automatic Lyrics Transcription with Whisper*, arXiv 2506.15514.
3. Zhuo et al., *LyricWhiz: Robust Multilingual Zero-shot Lyrics Transcription*, arXiv 2306.17103.
4. Défossez, *Hybrid Transformers for Music Source Separation* (Demucs v4), github.com/adefossez/demucs.
5. Lu et al., *Music Source Separation with Band-Split RoPE Transformer*, arXiv 2309.02612.
6. Stoller et al., JamendoLyrics MultiLang, github.com/f90/jamendolyrics.
7. *More than words: Advancements and Challenges in Speech Recognition for Singing*, arXiv 2403.09298.

A Research Brief: Automatic Lyrics Transcription, 2025–2026

This appendix reproduces the full literature survey conducted prior to the experiments.

A.1 Why lyrics transcription is harder than speech ASR

Singing breaks most assumptions baked into speech-trained ASR models. The survey *"More than words: Advancements and Challenges in Speech Recognition for Singing"* (arXiv 2403.09298) and the foundational *"Automatic Lyrics Alignment and Transcription in Polyphonic Music: Does Background Music Help?"* (arXiv 1909.10200) identify the core problems:

- **Background accompaniment:** harmonic and percussive instruments add spectral components that overlap and correlate with the vocal (they're in the same key and rhythm), unlike typical uncorrelated speech noise. This is the single largest degradation factor (arXiv 1909.10200).
- **Pitch and duration distortion:** singing spans far wider pitch ranges with vibrato; phonemes—especially vowels—are elongated or compressed to fit melody and rhythm (melisma stretches one syllable across many notes), confirmed by phoneme-duration analysis in the survey (arXiv 2403.09298). The same text sung to different melodies yields different acoustics, making training data effectively sparse (PDAugment, arXiv 2109.07940).
- **Pronunciation and lexicon shift:** singers deliberately alter pronunciation; lyrics use vocabulary, repetition structures, and non-lexical vocables ("oh-oh-oh", "na na") rare in speech corpora.
- **Production effects:** reverb, delay, doubling, backing vocals/harmonies, and compression smear the signal. Whisper systematically deletes backing vocals and non-lexical vocables regardless of preprocessing (arXiv 2506.15514).
- **Foundation-model degradation is measurable:** Whisper that achieves single-digit WER on clean speech lands at 25–45% WER on full mixes (Jam-ALT, arXiv 2311.13987).

A.2 Vocal source separation as preprocessing — does it help?

A.2.1 Separator quality landscape (MUSDB18-HQ vocals SDR)

- **Spleeter** (Deezer, 2019): obsolete for quality (~6.9 dB vocals); only relevant for speed.
- **Hybrid Demucs / htdemucs** (Meta, github.com/adefossez/demucs): `mdx` approx. 7.97 dB, `mdx_extra` approx. 8.76 dB vocals SDR (as measured in arXiv 2506.15514); `htdemucs_ft` is the fine-tuned v4 default.

- **MDX-Net / KUIELab** (arXiv 2111.12203): SDX-challenge-era frequency-domain models; many checkpoints live on in the UVR ecosystem.
- **BS-RoFormer** (ByteDance, arXiv 2309.02612): current SOTA — 9.80 dB average SDR on MUSDB18-HQ without extra data, ~+1.3 dB over htdemucs; won SDX'23. Community checkpoints (e.g., `model_bs_roformer_ep_317_sdr_12.97`) reach ~12.9 dB vocals SDR and are downloadable via python-audio-separator. **Mel-Band RoFormer** (arXiv 2310.01809) variants are similar; MVSEP's leaderboard tracks the practical SOTA.

A.2.2 Does separating before ASR reduce WER? The evidence is nuanced.

- **Pre-Whisper era:** arXiv 1909.10200 found that *training acoustic models on polyphonic mixtures beat extraction-based pipelines* — separation artifacts hurt more than music helps confuse.
- **Against (Whisper, naive use):** The Jam-ALT authors found that HTDemucs-isolated vocals **substantially degraded** Whisper results, especially v3 — on separated vocals "Whisper has a general tendency to omit parts of the lyrics (often the entire song) and instead produce generic or irrelevant text" (arXiv 2408.06370). Whisper was trained on noisy real-world audio; sparse, artifact-laden solo vocals are out-of-distribution.
- **For (with care):** The systematic study *"Exploiting Music Source Separation for ALT with Whisper"* (arXiv 2506.15514, Whisper large-v2, beam 5, language given) found:
  - Short-form, Jam-ALT: mix 20.99% vs separated 21.08–21.17% WER — **no real gain**.
  - Short-form, MUSDB-ALT: mix 23.59% → 20.00% with the better `mdx_extra` separator — **separation quality is the deciding variable**.
  - **Clean vocal stems: 14.19% WER** — the ~6-point gap between separated vocals and true stems shows separation artifacts (not the concept) are the bottleneck.
  - **Best use = separation as VAD:** their "RMS-VAD" — compute energy on the separated vocal to find sung segments, then transcribe — gives consistent gains and **20.35% WER on Jam-ALT long-form, SOTA for open-source, with zero training**.
- **For (with fine-tuned ASR):** SongPrep reports Whisper improving from ~47% on mixtures to ~28% on separated vocals on their SSLD-200 set, and pipelines like VietLyrics separate first then fine-tune.

Engineering takeaway: with stock Whisper, (a) always use separation to *segment* (VAD); (b) only feed separated vocals to Whisper if your separator is BS-RoFormer-class (≥~9–12 dB vocals SDR); (c) consider transcribing **both** mix and separated vocals and reconciling — that's effectively what commercial systems and LyricWhiz-style ensembles exploit.

A.3 ASR models for singing voice

A.3.1 Whisper family behavior on sung vocals

- **Hallucinations/loops:** instrumental intros, solos, and outros trigger ghost text, repetition loops ("burning through the sky" repeated), spurious "Thank you"/"(singing in foreign language)" tokens. Documented across openai/whisper #679, whisper.cpp #2286, and a music-specific comparison on Queen's "Don't Stop Me Now" (whisper.cpp #3074).
- **large-v2 vs large-v3:** counterintuitively, **v3 is not better for lyrics**. On Jam-ALT with language hints, v2 = 27.9% WER vs v3 = 32.6% (arXiv 2408.06370); the earlier

benchmark run had them near-tied (35.7 vs 35.5, arXiv 2311.13987). v3 is also more fragile on separated vocals. **Benchmark both; default to large-v2 for lyrics.**

- **large-v3-turbo:** 4 decoder layers (vs 32) — fine for speech (within ~0.4 pt of v3, HF model card), but on music it misses vocalizations and flubs endings; the Queen test ranked large-v3 best for music fidelity (whisper.cpp #3074).
- **Runtimes:** faster-whisper (CTranslate2, built-in Silero VAD, ~4x faster, default beam 5); whisper.cpp (portable, GGML quantization); mlx-whisper (Apple MLX, Metal GPU) — ~2.0x faster than whisper.cpp on Apple Silicon with large-v3-turbo (billmill benchmark, Jan 2026); lightning-whisper-mlx claims further speedups; mac-whisper-speedtest lets you benchmark locally.

A.3.2 Singing-specific systems

- **LyricWhiz** (arXiv 2306.17103, HTML v4): training-free — Whisper-large run 3–5×, prompt `"lyrics:"`, drop outputs with no-speech prob > 0.9 (plus PANNs audio-event filtering), then GPT-4 ensembles/corrects the candidates. Jamendo 24.25% WER (prior SOTA W2V2-ALT: 33.13%), Hansen 7.85%, MUSDB18 26.29%. No source separation used. The "Whisper-ear + LLM-brain, multi-run voting" pattern remains the best zero-training accuracy trick.
- **Fine-tunes:** no widely-adopted general-purpose singing Whisper checkpoint exists on HuggingFace; published fine-tunes are language-specific (e.g., VietLyrics fine-tuned large-v2: 20.5% lowercase WER on Vietnamese songs). Several papers deliberately skip fine-tuning because Whisper generalizes well (arXiv 2506.15514).
- **End-to-end research models: SongPrep/SongPrepE2E** (arXiv 2509.17404) — full-song structure + lyrics, 24.3% WER on SSLD-200 (vs Whisper 27.7%), dataset released but not weights. **OWSM v3.1** does badly on lyrics (69.3% WER, arXiv 2408.06370).
- **Commercial: AudioShake** is the only vendor publishing lyrics WER: v3 = **16.1% WER on Jam-ALT** (case-sensitive 20.1%), a 57% reduction vs Whisper v2, best on every formatting metric (AudioShake blog, arXiv 2408.06370). **AssemblyAI**'s own blog testing 15 songs found WERs of **0.47–0.88** — i.e., not usable for lyrics (AssemblyAI blog). Deepgram publishes nothing on music. **Gemini 2.5 Pro / GPT-4o-audio** have no published Jam-ALT numbers; NVIDIA's **Music Flamingo** (arXiv 2511.10289) reports beating both on lyrics transcription (19.6% WER English, 12.9% Chinese on its eval). Anecdotally Gemini handles full songs well, but treat it as unbenchmarked; it's worth including in your own bake-off.

A.4 Known best practices (decode-time)

1. `condition_on_previous_text=False` — the single most important anti-repetition-loop setting for music; instrumental gaps otherwise poison subsequent windows (whisper discussion #679, Memo AI writeup). Cost: loses cross-segment context (slightly worse consistency).
2. **Language hint** — always pass `language=`; Jam-ALT shows `+lang` improves Whisper and avoids per-segment misdetection on sung vowels (arXiv 2408.06370).
3. **VAD before Whisper** — don't let Whisper see instrumental-only audio. Either faster-whisper's built-in Silero `vad_filter=True` (faster-whisper) or, better for music, **energy/RMS-VAD computed on the separated vocal stem**, then cut the *original*

mix at those boundaries — this is the published SOTA open recipe (20.35% Jam-ALT WER, arXiv 2506.15514). Note Silero is speech-trained and can be unreliable on singing; vocal-stem RMS sidesteps that.

4. **Temperature fallback** — keep the default ladder `(0.0, 0.2, …, 1.0)` with `compression_ratio_threshold=2.4` and `logprob_threshold=-1.0`; these gate the retry that escapes degenerate beams. Greedy/beam at T=0 plus fallback beats fixed nonzero temperature.
5. `no_speech_threshold` — lower it (e.g., 0.4–0.6) and drop high no-speech-probability segments; LyricWhiz dropped predictions with no-speech prob > 0.9 (arXiv 2306.17103v4).
6. `initial_prompt` — LyricWhiz's `"lyrics:"` prompt (localized: "paroles", "Liedtext") nudges the decoder toward lyric register; you can also seed artist/title or known chorus lines. Beware: prompts can also seed hallucinations on silent stretches.
7. **Chunking** — prefer segment boundaries at vocal-activity gaps (from VAD above) padded to $ $30 s windows; for short-form scoring, the concatenation approach in arXiv 2506.15514 reduced WER consistently. `hallucination_silence_threshold` (openai/mlx implementations) helps skip long silences when using word timestamps.
8. **Multi-run + LLM rerank** (optional, biggest accuracy lever after separation): run 3–5 decodes (vary temperature/seed), have an LLM merge — LyricWhiz pattern, worth ~5–10 WER points on hard genres (arXiv 2306.17103).

A.5 Evaluation methodology

- **Metrics:** word-level WER is standard; CER for logographic languages (Mandarin lyrics work uses CER). Lyrics raise normalization landmines: case, punctuation, line/section breaks, parenthesized backing vocals, and non-lexical vocables are all meaningful content for display but destroy naive WER comparability.
- **Jam-ALT** (arXiv 2311.13987, project page, HF dataset) is the de-facto 2024–2026 benchmark: a *revision* of JamendoLyrics MultiLang with an industry-style annotation guide (Apple/Musixmatch/LyricFind conventions) and an eval package (`alt-eval`) that defines WER, case-sensitive WER', case error rate, punctuation-F, parentheses-F (backing vocals), line-break-F and section-break-F. The revision alone amounts to 11.1% WER-worth of changes vs the original JamendoLyrics — i.e., **numbers on "JamendoLyrics" vs "Jam-ALT" are not comparable**.
- **JamendoLyrics MultiLang** (github f90/jamendolyrics, now deprecated in favor of HuggingFace hosting): 79–80 songs, EN/FR/DE/ES, word-level timing — primarily an *alignment* benchmark; Jam-ALT is its transcription-focused descendant.
- **MUSDB-ALT** (arXiv 2506.15514): new long-form lyrics benchmark where **true vocal stems are public** — uniquely lets you measure separation-artifact cost.
- **Published WER anchor points (Jam-ALT, overall unless noted):**

System	WER	Source
Whisper large-v2 (+lang)	27.9%	2408.06370
Whisper large-v3 (+lang)	32.6%	2408.06370
Whisper large-v2, sep-assisted RMS-VAD long-form	**20.35%** (open SOTA)	2506.15514
LyricWhiz (Whisper+GPT-4)	24.6%	2408.06370

System	WER	Source
OWSM v3.1	69.3%	2408.06370
AudioShake v3 (commercial)	**16.1%**	2408.06370
Whisper large-v2 on clean vocal *stems* (MUSDB-ALT)	~14.2%	2506.15514

Per-language (Whisper v2 +lang): EN 39.7, ES 21.9, DE 19.9, FR 27.1 — English is paradoxically hardest in this set (genre mix). Expect 20–25% WER from a good open pipeline, 35–50%+ on rock/metal/heavy mixes, near-15% only with clean stems or commercial systems.

A.6 Concrete recommendation: Mac (Apple Silicon) pipeline

Primary recommendation (best published-evidence open pipeline):

1. **Separate vocals with a RoFormer-class model** via python-audio-separator (`pip install "audio-separator[cpu]"` — uses CoreML execution provider on macOS; e.g. `model_bs_roformer_ep_317_sdr_12.97.ckpt`), or the MLX-native fork mlx-audio-separator. Fallback: `demucs -n htdemucs_ft` (demucs).
2. **Segment using the separated vocal as a VAD**: RMS-energy gate on the vocal stem → sung-region boundaries, merge gaps < ~1 s, pad ~0.3 s, cap windows at 30 s (recipe per arXiv 2506.15514).
3. **Transcribe with mlx-whisper**, model `mlx-community/whisper-large-v2-mlx` (benchmark v3 too; v2 is the safer lyrics default), per segment:

```
mlx_whisper.transcribe(
    seg_audio,
    path_or_hf_repo="mlx-community/whisper-large-v2-mlx",
    language="en",                        # always hint
    condition_on_previous_text=False,     # kills repetition loops
    temperature=(0.0, 0.2, 0.4, 0.6, 0.8, 1.0),  # fallback ladder
    compression_ratio_threshold=2.4,
    logprob_threshold=-1.0,
    no_speech_threshold=0.5,
    initial_prompt="lyrics:",             # LyricWhiz trick; A/B it
)
```

4. **Feed segments from the original mix by default**; if your separator is BS-RoFormer-class, also transcribe the separated-vocal segments and **reconcile the two passes with an LLM** (LyricWhiz pattern: present both candidates, ask for the most plausible lyric line). Expected: ~20% WER territory on Jam-ALT-like material; the LLM merge is your path below that.

Runner-up configurations to benchmark against: - **faster-whisper large-v2/v3** with `vad_filter=True`, `beam_size=5`, `condition_on_previous_text=False` on the mix — simplest credible baseline (faster-whisper). - **whisper.cpp `large-v3`** (not turbo — turbo demonstrably drops vocalizations on music, whisper.cpp #3074) if you want a zero-Python binary; expect ~2x

slower than MLX on M-series (benchmark). - **Gemini 2.5/3 Pro full-song audio prompt** — unbenchmarked on Jam-ALT but cheap to test and strong anecdotally; include in your bake-off. - **AudioShake API** if budget allows — only system with published ~16% Jam-ALT WER plus correct line breaks/casing (AudioShake).

Evaluate with `alt-eval` on the jam-alt dataset (and MUSDB-ALT if you care about long-form), so your numbers are comparable to the literature; don't hand-roll WER normalization.

Known residual failure modes no pipeline fixes today: backing vocals/harmonies (systematically deleted), non-lexical vocables, heavily distorted genres (rock/metal), and choruses with stacked vocals — budget for human review there.

B Full Evaluation Scores

B.1 Mean WER/CER by configuration (percent)

config	wer	cer
ensemble_llm	15.9	6.4
large3_vocals	22.1	10.1
turbo_vocals	22.8	10.2
large3_mix	24.3	12.5
large3_vadmix	25.5	17.3
large3_vocals_nolang	26.4	13.6
turbo_mix	27.1	18.1
qwen_vocals	27.4	13.3
qwen_vocals_ctx	29.3	15.1
qwen_mix	30.0	15.5
large2_vocals	32.4	17.7
large2_mix	32.6	20.0
large2_vadmix	34.1	34.5
large3_vocals_cond	50.2	39.1

B.2 Per-song WER by configuration (percent)

song	ens	v2 mix	v2 VAD	v2 voc	v3 mix	v3 VAD	v3 voc
1 Freak - Automati	11.2	27.2	30.0	16.9	17.2	39.2	18.5
Baila - Alfonso Lu	18.8	27.3	31.2	23.8	23.0	20.2	25.2
Besando Sapos - Dr	19.3	27.6	28.1	34.6	24.1	25.0	22.8
Bitte beweg dich n	2.0	18.4	30.3	11.3	13.9	21.2	13.9
Burn Out Man - Abe	11.8	13.3	55.5	13.3	15.5	20.9	13.3
CHRISTMAS AVEC TOI	36.6	46.9	50.0	45.4	43.7	54.3	38.6
Capotes à un Franc	4.5	11.4	10.6	11.4	8.2	8.2	10.6
Caralibro - Vagos	11.8	72.0	22.7	119.4	23.7	18.5	28.4
Confession - Quesa	24.1	22.6	28.9	38.7	26.2	24.7	26.5
HILA - Give Me the	8.4	10.6	53.4	18.6	10.6	13.0	9.9
Quentin Hannappe -	6.9	21.7	20.0	9.1	12.0	17.7	12.6
Songwriterz - Back	35.3	92.4	48.3	46.2	73.5	42.4	45.4

song	ens	v2 mix	v2 VAD	v2 voc	v3 mix	v3 VAD	v3 voc

song	v3 cond	v3 auto	qw mix	qw voc	qw ctx	tu mix	tu voc
1 Freak - Automati	22.3	18.5	24.0	23.2	23.2	16.1	15.3
Baila - Alfonso Lu	64.9	25.2	32.6	34.8	39.0	18.4	28.7
Besando Sapos - Dr	100.0	22.8	40.8	26.8	33.3	24.1	22.8
Bitte beweg dich n	17.8	13.9	28.9	28.9	24.4	19.5	12.2
Burn Out Man - Abe	37.3	13.3	20.9	17.9	17.0	20.0	14.5
CHRISTMAS AVEC TOI	39.4	40.3	48.6	42.0	41.7	50.0	40.9
Capotes à un Franc	17.6	10.6	15.1	22.4	16.7	12.2	10.6
Caralibro - Vagos	100.0	82.0	48.3	33.2	54.5	39.3	18.5
Confession - Quesa	30.1	26.5	28.0	29.8	29.8	24.1	32.1
HILA - Give Me the	44.4	9.9	13.4	10.9	11.2	10.2	15.8
Quentin Hannappe -	41.7	12.6	18.3	18.3	17.7	12.0	17.7
Songwriterz - Back	87.0	40.8	41.6	41.2	42.9	79.4	44.5

Config key: v2/v3/tu = whisper large-v2 / large-v3 / large-v3-turbo, qw = Qwen3-ASR-1.7B; mix = full mix input, voc = Demucs vocal stem, VAD = RMS-VAD-segmented mix, cond = conditioning ON ablation, auto = language auto-detect ablation, ctx = context-prompt ablation, ens = LLM reconciliation of v3 voc + tu voc + v3 mix.

Raw per-run data: `results/scores.csv`; full transcripts under `results/transcripts/`.

C SOTA Gap and Headroom Report

This appendix reproduces the post-benchmark state-of-the-art gap analysis in full.

Date: June 10, 2026 | Companion to `paper.md` (our result: 22.1% WER, htdemucs → Whisper large-v3)

C.1 Bottom line

Our 22.1% WER is ~**2 points off the best published open-source no-training result** (20.35%, Whisper large-v2 + separation-based RMS-VAD on the full 79-song Jam-ALT set), ~**6 points off the best commercial system** (AudioShake v3, 16.1% Jam-ALT), and ~**8–10 points off the best audio-LLM numbers** on full songs (Gemini 2.5 Pro 12.2% EN; Qwen3-ASR 14.6% EN, internal evals). Roughly 2–4 of our 22.1 points are likely not model error at all: reference noise in the original JamendoLyrics annotations plus systematic backing-vocal/non-lexical deletions that all current systems share. A realistic "best possible today" on our material is ~**12–16% WER**, with ~10% as the practical floor for full mixes.

C.2 Where our 22.1 points of error actually go

Combining our own error decomposition with published analyses:

Error bucket	Points (approx.)	Fixable by
Substitutions (mishearings)	11.7	better acoustic model, LLM correction
Deletions (backing vocals, buried/non-lexical lines)	7.0	partly unfixable (~3–4 pts systematic across all systems); rest: better separation/model
Insertions (residual hallucination)	3.2	ensembling, VAD
— of which reference noise (original JamendoLyrics vs Jam-ALT revision changed ~11% of tokens)	~2–3	re-scoring against Jam-ALT refs with `alt-eval`

Two more facts from our own runs sharpen this: an **oracle picking the best of our 10 configs per song scores 19.1%** (so pass-ensembling is worth $ $3 points before any new model), and our best config without the two hardest songs scores 18.2% — the headroom is concentrated in a minority of hard songs.

C.3 Current SOTA reference points (2025–2026)

System	WER	Where
Whisper large-v2 + RMS-VAD (open, no training)	20.35%	Jam-ALT, arXiv 2506.15514
AudioShake v3 (commercial API, separation-first)	**16.1%**	Jam-ALT, audioshake.ai
Qwen3-ASR-1.7B (open weights Apache 2.0, Jan 2026; trained on speech+song, 52 languages)	**14.6% EN** full songs	internal eval, arXiv 2601.21337

System	WER	Where
Gemini 2.5 Pro (full-song audio prompt)	12.2% EN	same eval
NVIDIA Music Flamingo (8B, open weights)	19.6% EN	MUSDB18, arXiv 2511.10289
GPT-4o-Transcribe	30.7% EN	same eval (poor on music)
Whisper-class floor on clean vocal stems	~15.0%	MUSDB-ALT, arXiv 2506.15514

Notable: no published paper has beaten 20.35% on Jam-ALT itself yet, but Qwen3-ASR has very likely leapfrogged it un-benchmarked; and generic 2026 speech models (Parakeet, Canary, Voxtral) top speech leaderboards but are *not* better on singing.

C.4 Ranked improvement levers (from our 22.1%)

#	Lever	Expected gain	Effort
1	**Swap ASR to Qwen3-ASR-1.7B** (local, Apache 2.0) for vocals + mix passes	**4–8 pts** → ~14–18%	Low
2	Proper RMS-VAD segmentation (full recipe from arXiv 2506.15514, not our minimal version)	2–3 pts	Low
3	Multi-run ensemble + LLM line-by-line reconciliation (LyricWhiz pattern, modern LLM)	2–4 pts	Medium
4	Audio-LLM pass (Gemini 2.5/3 Pro API, or local Music Flamingo) as ensemble member	2–6 pts standalone	Low–Medium
5	Dual-pass mix+vocals reconciliation (our data: per-song winners differ)	1–2 pts	Low
6	Fine-tune Whisper on singing data (DALI v2, MulJam)	~3 pts	High

#	Lever	Expected gain	Effort
7	RoFormer-class separator instead of htdemucs	0–2 pts	Low
8	Re-score against Jam-ALT references with `alt-eval` (measurement fix, not model fix)	~2–3 pts apparent	Trivial

Levers 1+2+5 are roughly additive at first; 3 and 4 overlap with each other and with 1.

C.5 Recommended "best result" stacks

API allowed (estimated 10–14% WER): AudioShake API + Gemini 2.5/3 Pro on the mix + Qwen3-ASR on RoFormer-separated vocals, merged line-by-line by an LLM with phonetic-plausibility instructions and explicit "keep backing vocals and non-lexical vocables" prompting.

Fully local (estimated 13–17%): Mel-RoFormer vocals → Qwen3-ASR-1.7B dual-pass (vocals + mix) with RMS-VAD segmentation → local LLM tie-breaker/corrector.

Single highest-value next experiment: run Qwen3-ASR-1.7B on our existing 12 songs (both the htdemucs vocals and raw mixes) — one afternoon of work, expected landing zone 14–18% WER, the largest single lever available.

C.6 Post-report experimental validation (round 2, same 12 songs)

We tested the top two levers from the table above. The predictions inverted:

Lever	Predicted	Measured
#1 Qwen3-ASR-1.7B (vocals input, mlx, language hint)	4–8 pts gain → 14–18%	**-5.3 pts (WORSE): 27.4%**, behind Whisper large-v3 on 11/12 songs; FR/DE hit hardest; 6× slower; "song lyrics" context prompt made it worse still (29.3%)
#3 LLM reconciliation (Claude merges vocals/large-v3 + vocals/turbo + mix/large-v3, line-by-line, no reference access)	2–4 pts gain	**+6.2 pts gain: 15.9% WER / 6.4% CER**, improving all 12 songs, beating the 13-config oracle (19.1%) and matching AudioShake v3's published 16.1% on comparable material

Two lessons: (1) vendor-published internal song WERs are weak evidence for your own material — Qwen3-ASR's singing distribution apparently does not cover European-language indie

production; benchmark before switching. (2) Lyrics' repetition structure makes LLM merging unusually effective — chorus normalization plus disagreement-as-hallucination-detector delivers roughly double the literature's reported ensemble gain.

Updated best local recipe: Demucs vocals → Whisper large-v3 (+turbo, +mix passes) → LLM line-by-line merge = 15.9% WER. Remaining unexploited levers from here: audio-LLM ensemble member (Gemini Pro), proper RMS-VAD segmentation, RoFormer-class separator, Jam-ALT re-scoring (~2–3 apparent points).

C.7 Sources

Jam-ALT benchmark: arXiv 2408.06370, audioshake.github.io/jam-alt | Separation/VAD study and MUSDB-ALT: arXiv 2506.15514 | Qwen3-ASR: arXiv 2601.21337, github.com/QwenLM/Qwen3-ASR | Music Flamingo: arXiv 2511.10289 | LyricWhiz: arXiv 2306.17103 | Singing fine-tuning: arXiv 2506.02339, 2510.22295 | AudioShake: audioshake.ai benchmark posts | SongPrep/SSLD-200: arXiv 2509.17404 | LLM over-correction caution: arXiv 2505.17410.