Double Entendre:
Robust Audio-Based AI-Generated Lyrics Detection via Multi-View Fusion

Markus Frohmann [1,2] **Gabriel Meseguer Brocal** [1] **Markus Schedl** [2,3] **Elena V. Epure** [1]
[1] Deezer Research, Paris, France
[2] Johannes Kepler University Linz, Austria [3] Linz Institute of Technology, AI Lab, Austria
research@deezer.com

Abstract

The rapid advancement of AI-based music generation tools is revolutionizing the music industry but also posing challenges to artists, copyright holders, and providers alike. This necessitates reliable methods for detecting such AI-generated content. However, existing detectors, relying on either audio or lyrics, face key practical limitations: audio-based detectors fail to generalize to new or unseen generators and are vulnerable to audio perturbations; lyrics-based methods require cleanly formatted and accurate lyrics, unavailable in practice. To overcome these limitations, we propose a novel, practically grounded approach: a multimodal, modular late-fusion pipeline that combines automatically transcribed sung lyrics and speech features capturing lyrics-related information within the audio. By relying on lyrical aspects directly from audio, our method enhances robustness, mitigates susceptibility to low-level artifacts, and enables practical applicability. Experiments show that our method, DE-DETECT, outperforms existing lyrics-based detectors while also being more robust to audio perturbations. Thus, it offers an effective, robust solution for detecting AI-generated music in real-world scenarios.[1]

1 Introduction and Background

The advent of AI-generated music (AIGM) has recently been transformative for the music industry, mainly driven by music generation tools such as Suno[2] or Udio[3]. While such tools can enhance creativity by aiding in composition and arrangement (Li et al., 2024b; Parada-Cabaleiro et al., 2024), they also raise concerns regarding copyright, artistic value, and the potential for AI-created works to overshadow human musicians (Afchar et al., 2024; Micalizzi, 2024; Henry et al., 2024). The divergent responses from music streaming services, with some ceasing to recommend AI-flagged songs[4] and others embracing them[5], underscore the increasingly critical need for robust and reliable AIGM detection methods.

Existing work on AIGM detection has mostly focused on AI-generated audio, whether with or without vocals (Afchar et al., 2024; Cooke et al., 2024; Rahman et al., 2024). While such detectors have been shown to achieve high (>99%) accuracy within their training domain, they fail to generalize to unseen AIGM models and are highly vulnerable to audio attacks such as adding noise or changing pitch (Afchar et al., 2024). This highly limits their usability in practice.

Beyond audio, for songs with vocals, lyrics (represented as text) are an essential medium of conveying a song's content (Li et al., 2024b). In most AIGM, lyrics are also generated by AI; thus, determining lyrics authorship (human or AI) could be a proxy for flagging a track as AI-generated. To detect AI-generated *lyrics*, Labrak et al. (2024) introduce a dataset of synthetic lyrics generated using several LLMs, based on prompts informed by lyric examples from diverse language–music genre pairs. They evaluate various text-based detectors, showing promising results. However, their methods rely on clean, perfectly formatted lyrics; but in practice, only audio is available, making this requirement impractical.[6]

Contributions. To overcome these limitations, we propose a novel multi-view pipeline for detecting AI-generated lyrics that is both robust and practically applicable, relying solely on audio as input.

[1] Our code is available at `https://github.com/deezer/robust-AI-lyrics-detection`.

[2] www.suno.com

[3] www.udio.com

[4] www.billboard.com/pro/deezer-ai-detection-tool-10-percent-music-tracks-ai-generated

[5] www.bigtechnology.com/p/spotifys-plans-for-ai-generated-music

[6] In practice, lyrics metadata is often unavailable for newly ingested music in industrial settings.

Findings of the Association for Computational Linguistics: ACL 2025, pages 1914–1926
July 27 - August 1, 2025

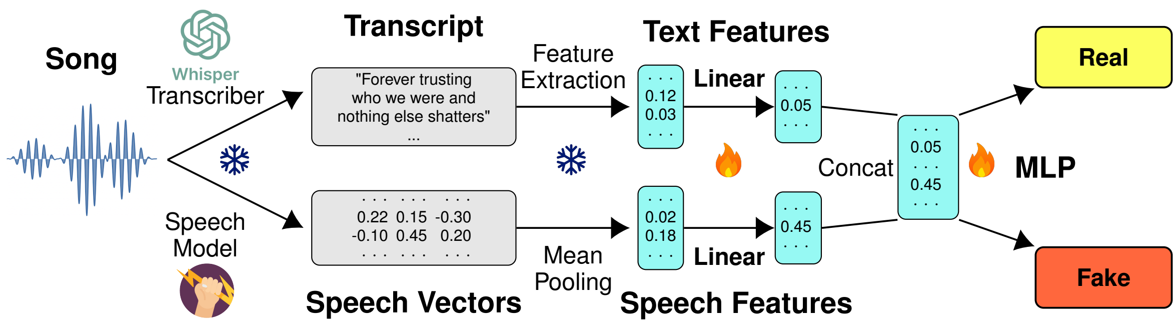

Figure 1: Overview of our pipeline to robustly detect AI-generated lyrics when only audio is available. In the top branch, we transcribe audio to lyrics via a *transcriber*. This transcript is then used to get *text features*. In the bottom branch, we use a speech model to get lyrics-related information only present in audio – *speech features*. Finally, we *linearly project* and *concatenate* both and feed them into an *MLP* detector to classify the input song as real or fake.

As Figure 1 shows, it robustly leverages this input as two different modalities: (i) automatically transcribed lyrics to eliminate reliance on perfectly formatted lyrics, and (ii) using speech models to capture lyrics-related information present only in singing voice. Experiments show that our method exhibits improved, more robust performance than unimodal ones, especially out-of-domain. This results in a practical solution for robust AI-generated lyrics detection, paving the way for greater transparency in the rapidly evolving AIGM landscape.

2 Method

Existing unimodal approaches – whether audio-based detectors that are sensitive to perturbations and generalize poorly, or lyrics-based detectors that require clean, often inaccessible lyrics – tend to falter in real-world scenarios. To address the impracticality of relying on perfectly clean lyrics, we turn to automatically transcribed lyrics. However, transcripts capture *what* (the semantic content), but they may miss *how* (subtle audio cues indicative of AI generation). We hypothesize speech embeddings capture this *how* – lyrics-related cues present in audio but not in lyrics themselves.

To combine *what* and *how*, our method employs late fusion and synergistically integrates features from transcribed lyrics (semantic content) and speech (lyrics-related audio cues). This multi-view fusion aims to overcome the limitations of text-only methods, enabling accurate detection resilient to audio attacks, as detailed in Section 4. We provide an overview of our method in Figure 1.

(i) Text Branch. We use a transcription model (ASR model) to transcribe audio to lyrics. To represent the semantic content of these lyrics for downstream processing, we feed the entire lyrics transcript into a text embedding model. This model captures rich semantic information and generates a single, contextualized *lyrics text embedding* (top branch in Fig. 1).

(ii) Speech Branch. To capture the *how* of lyrics (complementary audio cues indicative of AI generation) we use a speech model. Unlike general audio embeddings, or the ASR models used for transcription, speech embedding models are specifically designed to capture rich acoustic and paralinguistic information from speech signals, such as prosody, intonation, and speaker characteristics, which can be indicative of AI generation even if not present in the transcribed text. To our knowledge, this is the first application of dedicated speech embeddings for AI-generated lyrics detection in the music domain. This model extracts lyrics-related audio features: phonetic and contextual patterns like prosody and intonation from audio, resulting in a *lyrics speech embedding* (bottom branch in Fig. 1). We also conducted in-domain experiments with source separation to isolate instances of singing voice. However, this did not significantly improve performance, suggesting our method is already somewhat resilient to background music.

(iii) Late Fusion. We employ late fusion to synergistically combine lyrics and speech features, derived from audio alone. Its simple and modular design offers key benefits: independent component updates, preservation of each component's strengths (e.g., multilinguality), and robustness to component changes (cf. §4.1). In the face of the evolving AIGM landscape, we argue these characteristics are crucial for a practically applicable robust detection system. For fusion, features

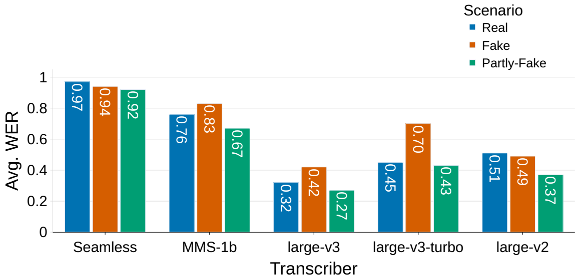

Figure 2: Word error rates (WER) of different transcription models across Real, Fake, and Partly-Fake song scenarios. Lower WER indicates better transcription.

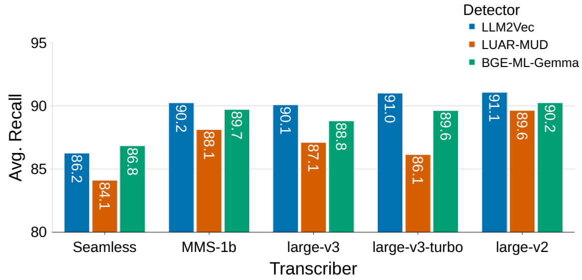

Figure 3: Recall of transcriber-feature combinations.

from both branches are linearly down-projected to 128, concatenated, and then classified using a lightweight MLP, trained with binary cross-entropy loss (details in Appendix A). Overall, our modular late-fusion design enables a robust, generalizable, and practically evolvable detection method.

3 Experimental Setup

3.1 Dataset

We start from the lyrics dataset of Labrak et al. (2024), which provides 3,655 real and 3,535 AI-generated lyrics from three LLM generators. Human lyrics spanning nine languages and the six most popular genres per language are used as seeds in the generation pipeline.

A key limitation of this dataset is that it provides lyrics only. Therefore, to enable realistic audio-based experiments representative of current AIGM, we generate corresponding audio for the AI-generated lyrics using state-of-the-art Suno v3.5, conditioned on lyrics and genre.[7] For songs with human-generated lyrics, we use their original audio. This results in a dataset of 7,190 songs, balanced between fully real songs and Suno-generated songs with AI lyrics generated by multiple LLMs. We follow the train/test split of Labrak et al. (2024).

Moreover, a key question is whether our model and its components detect AI-generated lyrics or just audio artifacts inherent in AI-generated audio (robustness to AI audio artifacts). To address this, we design a "partly-fake" experiment: generating Suno audio for *real* lyrics and evaluating performance compared to detecting fully fake songs. This mitigates the influence of audio artifacts that should be mostly similar for partly-fake and fully fake, allowing us to verify if each model relies on lyrics.

In addition, to test generalization, we generate 260 additional songs with synthetic lyrics from the test set of the lyrics dataset by Labrak et al. (2024) using *Udio*, another music generation tool,[8] and sample 260 real songs. Our model, trained on the Suno dataset, is then evaluated on these out-of-domain scenarios without further training. For details, we refer to Appendix A.

3.2 Evaluation Metrics

Following (Labrak et al., 2024; Nakov et al., 2013; Li et al., 2024a), we evaluate performance primarily via macro-recall as our main metric and additionally report AUROC scores. We provide a definition of these metrics in Appendix F.

4 Experiments

4.1 Component Instantiation

We first evaluate several unimodal features to select as components in our multimodal pipeline, starting with text-based detectors. To provide text for text-based detectors, we transcribe real and synthetic audio to lyrics using five recent multilingual transcription models: Whisper in variations *large-v2*, *large-v3*, and *large-v3-turbo* (Radford et al., 2022), *mms-1b* (Pratap et al., 2023), and *Seamless-large* (Communication et al., 2023).

Transcription Quality (WER). To first assess the intrinsic quality of these transcribers, we calculate their Word Error Rate (WER) against ground truth lyrics for different song types: human-generated (*real*), AI-generated (*fake*), and AI-generated audio with human lyrics (*partly-fake*). Figure 2 illustrates these WERs. Overall, Whisper-based models demonstrate the highest transcription quality, with *Whisper-large-v3* generally achieving the lowest WERs among all tested transcribers. In

[7] While tools such as Suno can also generate lyrics, we use the provided lyrics to ensure control over the lyrical content.

[8] We searched for other models or tools capable of conditioning on lyrics but found no other suitable ones.

Model	Recall
Text-based Detectors via Whisper large-v2 Transcripts	
UAR-MUD	89.6
BGE-ML-GEMMA	90.2
LLAMA3 $8B_{\text{LLM2Vec}}$	**90.7**
Speech-based Detectors	
WAV2VEC 2.0	83.1
MMS-1B	88.8
XEUS	**92.2**

Table 1: Recall scores, macro-averaged over multilingual lyrics, for several unimodal detectors.

Model	REAL VS. PARTLY-FAKE	FAKE VS. PARTLY-FAKE
UAR-MUD	**66.9**	86.1
LLAMA3 8B	64.9	90.0
BGE-ML-GEMMA	67.7	89.0
WAV2VEC 2.0	**50.9**	83.1
MMS-1B	50.7	88.5
XEUS	50.5	**92.0**

Table 2: Recall scores on detecting *partly-fake* songs with human-generated lyrics but synthetic audio.

contrast, *MMS-1b* and *Seamless* exhibit substantially higher WERs across all scenarios, indicating more transcription errors. A notable pattern across Whisper models is that WERs on *partly-fake* songs are often among the lowest, potentially due to the AI-generated audio in these cases being clearer or more consistently enunciated than some human recordings. Moreover, we also observe that *real* songs were generally transcribed slightly worse than *fake* songs across Whisper models.

Impact on Downstream Detection. While the raw WER provides insights into transcriber quality, the key consideration is how these transcriptions affect performance on our downstream task of AI-generated lyrics detection. Therefore, we next evaluate the impact of using different transcribers when combined with various text-based detection features. For text, we use the two best-performing detection features from Labrak et al. (2024): *UAR-MUD* (Rivera-Soto et al., 2021), and *LLM2Vec* with Llama3 8B as base model (BehnamGhader et al., 2024), and a recent multilingual general-purpose embeddings model, *BGE-Multilingual-Gemma2* (Chen et al., 2024a).

Figure 3 compares the average detection performance of these transcriber-feature combinations, where we train an MLP classifier on each. Results reveal that while Whisper-large-v3 exhibits slightly lower WERs (Figure 2), Whisper-large-v2 achieves the best average recall at 90.2%. This shows that lower raw WER does not necessarily correspond to improved detection performance. Further, it also indicates that our feature extraction and classification pipeline can effectively handle the moderate level of transcription errors from models such as Whisper large-v2. Similarly, text detector choice shows minimal difference: UAR-MUD performs slightly lower, while LLM2Vec shows the highest average recall. Thus, robust detection performance is not tied to a single transcriber or text feature, indicating robustness of our approach to variations in unimodal components, even with the observed differences in raw WER.

Next, we evaluate speech embeddings from three strong multilingual models: *XEUS* (Chen et al., 2024b), *Wav2Vec 2.0* (Baevski et al., 2020), and *MMS-1b* (using the ASR-finetuned variant) (Pratap et al., 2023). For each, we apply mean-pooling to obtain a single vector. As with text features, we train an MLP using the features.

Table 1 shows results for each, and for comparison, includes text-based detector results with Whisper large-v2. Comparing speech embeddings, performance margins are slightly larger, with XEUS performing best at 92.2% average recall. We attribute its performance to a large and diverse training dataset that includes not only spoken dialogue but also instances of singing voice. This may enable the model to capture richer vocal characteristics relevant to distinguishing AI-generated sung lyrics, such as prosody and timbre. However, its training data lacks AI-generated voice, crucial for fair evaluation. Given these findings, we use features from LLM2VEC with transcripts from Whisper large-v2 and XEUS to instantiate our multimodal pipeline. In addition, Appendix D shows results using various other text-based features.

Sensitivity to Audio Artifacts. We further analyze artifact influence using *partly-fake* songs: Suno-generated audio with real lyrics. Table 2 shows results for two scenarios: *real vs. partly-fake* (differentiating human-generated vs. synthetic audio, both with human-generated lyrics) and *fake vs. partly-fake* (differentiating synthetic vs. human-generated lyrics, both with AI-generated audio).

In the *real vs. partly-fake* scenario, the speech-based XEUS performs at a level consistent with random chance, indicating its features are not primarily driven by AI audio artifacts. Transcription-

Model	Recall		AUROC	
	en	all	en	all
GT LYRICS$_{\text{LLM2Vec}}$†	91.3	94.3	99.0	97.3
CNN$_{\text{Spectrogram}}$‡	**97.5**	**97.4**	**99.9**	**99.8**
XEUS	89.1	92.2	94.5	97.0
LLAMA3 8B$_{\text{LLM2Vec}}$	90.6	90.7	97.6	94.8
DE-DETECT	**93.9**	**94.9**	**98.2**	**98.5**

Table 3: Recall and AUROC scores on English-language and macro-averaged over multilingual lyrics. For transcription, we use Whisper large-v2. For OURS, we combine embeddings from LLM2Vec and XEUS. † denotes the best-performing baseline by Labrak et al. (2024), using non-transcribed ground truth (GT) lyrics with LLAMA3 8B$_{\text{LLM2Vec}}$. ‡ uses the amplitude spectrogram to train a CNN on the task as in (Afchar et al., 2024).

Model	AUDIO ATTACKS					UDIO
	Stretch	Pitch	EQ	Noise	Reverb	
CNN	**98.1**	59.0	79.4	77.4	80.7	56.9
XEUS	92.5	92.3	92.3	92.4	92.4	85.9
UAR-MUD	86.7	88.8	88.8	88.6	88.5	85.6
LLAMA3 8B	90.0	89.7	89.6	89.3	89.6	85.9
DE-DETECT	94.1	**93.9**	**94.0**	**93.9**	**94.1**	**87.9**

Table 4: Recall scores on out-of-distribution data (Udio) and when fake songs are perturbed (attacked) in five different ways. We report average scores over languages.

based detectors, however, perform above random. This may be due to the transcription process capturing subtle audio generation artifacts (e.g., ASR training bias or differing distributions of non-lyrical tokens like "[Outro]"). Nevertheless, in the *fake* vs. *partly-fake* scenario (both audio types AI-generated, lyrics differ), performance is higher for all methods, with XEUS achieving 92.0% recall. This suggests models primarily distinguish lyrical content even when audio is AI-generated, highlighting the resilience of our multi-view approach.

4.2 In-domain Evaluation

Table 3 shows our main evaluation results on detecting AI-generated songs. We compare our multi-view model (XEUS+LLM2VEC late fusion), which we term *Double Entendre detect* (DE-DETECT), against the best unimodal baselines, and two additional strong baselines: LLM2Vec (with Llama3 8B) using ground truth, non-transcribed lyrics, which was reported with high performance by Labrak et al. (2024), and a CNN trained on amplitude spectrograms to detect audio artifacts, following Afchar et al. (2024).[9]

We first observe that LLAMA3 8B$_{\text{LLM2Vec}}$ using transcripts performs closely to GT LYRICS$_{\text{LLM2Vec}}$ (using clean, non-transcribed lyrics), reaching recall scores of 90.7% and 94.3%, respectively. This indicates transcription effectively retains AI-generated lyric characteristics for detection. Moreover, our multi-view model achieves higher scores than methods using audio-derived lyrics, reaching a recall of 94.9% (and an AUROC score of 98.5%), and even improves upon the clean ground truth lyrics baseline despite audio-only input. Only CNN slightly outperforms our method in-domain.

4.3 Out-of-domain Evaluation

We now evaluate robustness to (i) audio perturbations/attacks and (ii) out-of-domain generalization to Udio. The former simulates real-world audio variations and potential adversarial attacks, while the latter tests generalization w.r.t. audio generators. Results are shown in Table 4, painting a contrasting picture to in-domain findings: The CNN shows large performance drops in attacks, especially pitch, and poor generalization to Udio (56.9% recall), revealing its artifact sensitivity. In contrast, models relying on lyrics-related information are much more stable, showing they are less prone to artifacts. Finally, our multi-view model, DE-DETECT, shows recall scores 1.5-2% higher than the unimodal ones across these settings, suggesting consistently more robust performance, crucial for practical, real-world applications. We also ablate different fusion components in Appendix C.

5 Conclusion

In this work, we proposed a novel modular multimodal approach – *Double Entendre detect* (DE-DETECT) – for robust AI-generated lyrics detection, late-fusing lyrics and speech representations. DE-DETECT consistently outperformed text-based baselines in-domain. We also stressed the importance of robustness for practical AIGM detection and showed that our method is more robust than all unimodal ones. Our findings underscore the importance of considering both lyrical and speech features for reliable detection, offering a more resilient and forward-looking solution with significant implications for copyright, music industry transparency, and the evolving relationship between humans and AI in creative domains.

[9]Such models could also be trained on other input representations, but the findings of Afchar et al. (2024) are consistent across them, so we resort to the best-performing one.

Limitations

While our multi-view method demonstrates promising results in AI-generated lyrics detection, we acknowledge several limitations that warrant further investigation in future work.

First, our model's training and evaluation are primarily based on the dataset by Labrak et al. (2024). This introduces potential biases related to the dataset's distribution, despite its inspiration from multiple language and music genres pairs. We thus encourage future work to introduce and explore larger, more diverse datasets encompassing a wider range of music styles and languages.

Furthermore, relying on Suno v3.5 for generating AI-generated audio for training introduces a potential bias toward this specific tool's artifacts and stylistic characteristics. Although we evaluated our method on Udio as an out-of-domain generator, our core training remains Suno-centric. Once other music-generation tools that support lyrics conditioning are available, future research should investigate training and evaluating audio from a more diverse set of AI music-generation tools to reduce tool-specific biases.

We also acknowledge that our robustness evaluation does not cover every potential attack; for instance, attacks that combine two or more audio perturbations (e.g., changing pitch and time stretching). We leave this to future work.

Ethical Considerations

While intended for positive applications like copyright protection and transparency, revealing vulnerabilities in detection systems carries a dual-use risk. Malicious actors could exploit these weaknesses to create AI music designed to evade detection, potentially enabling further copyright infringement and music streaming platform manipulation. This risk is compounded by the potential for bias in our approach since our model may inherit biases from the training data, leading to unfair or inaccurate detection (Barocas et al., 2017). This could result in unjust content takedown or censorship, disproportionately impacting certain artists (Henry et al., 2024).

Therefore, we advocate for the responsible development and deployment of AIGM detection technologies, emphasizing transparency, fairness, and human-in-the-loop approaches to maximize benefits while mitigating possible harms to artists, creators, and the broader music ecosystem.

6 Acknowledgements

This research was funded in whole or in part by the Austrian Science Fund (FWF): https://doi.org/10.55776/COE12, https://doi.org/10.55776/DFH23, https://doi.org/10.55776/P36413. The authors would like to thank Aurelien Herault, Manuel Moussallam, Romain Hennequin, Yanis Labrak, and Gaspard Michel for their invaluable feedback on this work.

References

Darius Afchar, Gabriel Meseguer-Brocal, and Romain Hennequin. 2024. Detecting music deepfakes is easy but actually hard. *ArXiv*, abs/2405.04181.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460. Curran Associates, Inc.

Solon Barocas, Kate Crawford, Aaron Shapiro, and Hanna Wallach. 2017. The problem with bias: Allocative versus representational harms in machine learning. In *9th Annual Conference of the Special Interest Group for Computing, Information and Society*.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. LLM2Vec: Large language models are secretly powerful text encoders. In *First Conference on Language Modeling*.

Daria Beresneva. 2016. Computer-generated text detection using machine learning: A systematic review. In *Natural Language Processing and Information Systems*, pages 421–426, Cham. Springer International Publishing.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024a. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.

William Chen, Wangyou Zhang, Yifan Peng, Xinjian Li, Jinchuan Tian, Jiatong Shi, Xuankai Chang, Soumi Maiti, Karen Livescu, and Shinji Watanabe. 2024b. Towards robust speech representation learning for thousands of languages. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10205–10224, Miami, Florida, USA. Association for Computational Linguistics.

Ondrej Cífka, Hendrik Schreiber, Luke Miner, and Fabian-Robert Stöter. 2024. Lyrics transcription for humans: A readability-aware benchmark. *ArXiv*, abs/2408.06370.

Seamless Communication, Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenthaler, Paul-Ambroise Duquenne, Brian Ellis, Hady ElSahar, Justin Haaheim, John Hoffman, Min-Jae Hwang, Hirofumi Inaguma, Christopher Klaiber, Ilia Kulikov, Pengwei Li, Daniel Licht, Jean Maillard, Ruslan Mavlyutov, Alice Rakotoarison, Kaushik Ram Sadagopan, Abinesh Ramakrishnan, Tuan Tran, Guillaume Wenzek, Yilin Yang, Ethan Ye, Ivan Evtimov, Pierre Fernandez, Cynthia Gao, Prangthip Hansanti, Elahe Kalbassi, Amanda Kallet, Artyom Kozhevnikov, Gabriel Mejia Gonzalez, Robin San Roman, Christophe Touret, Corinne Wong, Carleigh Wood, Bokai Yu, Pierre Andrews, Can Balioglu, Peng-Jen Chen, Marta Ruiz Costa-jussà, Maha Elbayad, Hongyu Gong, Francisco Guzm'an, Kevin Heffernan, Somya Jain, Justine T. Kao, Ann Lee, Xutai Ma, Alexandre Mourachko, Benjamin Peloquin, Juan Pino, Sravya Popuri, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, Anna Y. Sun, Paden Tomasello, Changhan Wang, Jeff Wang, Skyler Wang, and Mary Williamson. 2023. Seamless: Multilingual expressive and streaming speech translation. *ArXiv*, abs/2312.05187.

Di Cooke, Abigail Edwards, Sophia Barkoff, and Kathryn Kelly. 2024. As good as a coin toss: Human detection of ai-generated images, videos, audio, and audiovisual stimuli. *ArXiv*, abs/2403.16760.

William Falcon and The PyTorch Lightning team. 2019. PyTorch Lightning.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116, Florence, Italy. Association for Computational Linguistics.

Antoine Henry, Valdy Wiratama, Adelaida Afilipoaie, Heritiana Ranaivoson, and Eric Arrivé. 2024. Impacts of ai on music consumption and fairness. *Emerging Media*, 2(3):382–396.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822, Online. Association for Computational Linguistics.

Guillaume Klein, Jong Wook Kim, Yoon Kim, and Clement Delangue. 2023. faster-whisper: A reimplementation of openai's whisper model using ctranslate2.

Yanis Labrak, Markus Frohmann, Gabriel Meseguer-Brocal, and Elena V. Epure. 2024. Synthetic lyrics detection across languages and genres. *CoRR*, abs/2406.15231v2.

Thomas Lavergne, Tanguy Urvoy, and François Yvon. 2008. Detecting fake content with relative entropy scoring. In *Proceedings of the 2008 International Conference on Uncovering Plagiarism, Authorship and Social Software Misuse - Volume 377*, PAN'08, page 27–31, Aachen, DEU. CEUR-WS.org.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2024a. MAGE: Machine-generated text detection in the wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–53, Bangkok, Thailand. Association for Computational Linguistics.

Yupei Li, Manuel Milling, Lucia Specia, and Björn W. Schuller. 2024b. From audio deepfake detection to ai-generated music detection - A pathway and overview. *CoRR*, abs/2412.00571.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Brian McFee, Colin Raffel, Dawen Liang, Daniel P. W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. 2015. librosa: Audio and music signal analysis in python. In *SciPy*.

Alessandra Micalizzi. 2024. *Artificial Creativity. Perceptions and Prejudices on AI Music Production*, pages 481–491. Springer Nature Singapore, Singapore.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. SemEval-2013 task 2: Sentiment analysis in Twitter. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 312–320, Atlanta, Georgia, USA. Association for Computational Linguistics.

Emilia Parada-Cabaleiro, Maximilian Mayerl, Stefan Brandl, Marcin Skowron, Markus Schedl, Elisabeth Lex, and Eva Zangerle. 2024. Song lyrics have become simpler and more repetitive over the last five decades. *Scientific Reports*, 14(1):5531.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Mamdouh Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, Alexei Baevski, Yossi Adi, Xiaohui Zhang, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. 2023. Scaling speech technology to 1, 000+ languages. *ArXiv*, abs/2305.13516.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *ArXiv*, abs/2212.04356.

Md Awsafur Rahman, Zaber Ibn Abdul Hakim, Najibul Haque Sarker, Bishmoy Paul, and Shaikh Anowarul Fattah. 2024. Sonics: Synthetic or not - identifying counterfeit songs. *ArXiv*, abs/2408.14080.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Rafael A. Rivera-Soto, Olivia Elizabeth Miano, Juanita Ordonez, Barry Y. Chen, Aleem Khan, Marcus Bishop, and Nicholas Andrews. 2021. Learning universal authorship representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 913–919, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Gemma Team Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, L'eonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ram'e, Johan Ferret, Peter Liu, Pouya Dehghani Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stańczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Boxi Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Christoper A. Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozi'nska, D. Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Pluci'nska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost R. van Amersfoort, Josh Gordon, Josh Lipschultz, Joshua Newlan, Junsong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, L. Sifre, L. Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Gorner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, S. Mc Carthy, Sarah Perrin, S'ebastien M. R. Arnold, Se bastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomás Kociský, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeffrey Dean, Demis Hassabis, Koray Kavukcuoglu, Clément Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. 2024. Gemma 2: Improving open language models at a practical size. *ArXiv*, abs/2408.00118.

C. E. Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2024. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations*.

Peter Sobot. 2021. Pedalboard.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *Preprint*, arXiv:1908.09203.

Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2021. MiniLMv2: Multi-head self-attention relation distillation for compressing pretrained transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2140–2151, Online. Association for Computational Linguistics.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Yalta, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. 2018. Espnet: End-to-end speech processing toolkit. In *Interspeech*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

A Complete Experiment Details

Training Overview. For clarity, we provide an overview of the training pipeline for each model type in Table 5.

Computing Infrastructure. We transcribe lyrics and compute their features on a server with a single Nvidia RTX A5000 GPU and Intel Xeon Gold 6244 CPUs. We also use it for training lightweight MLPs and the $\text{CNN}_{\text{Spectrogram}}$ baseline.

Implementation Details. We use the `PyTorch` (Paszke et al., 2019) and `transformers` (Wolf et al., 2020) libraries and use models in fp16 for all experiments. To make sure no encoding-specific patterns are picked up, we convert all audio to mp3 with 128kbps.

Transcription. To transcribe audio to lyrics via Whisper models, we use the `faster-whisper` (Klein et al., 2023). For *mms-1b* and *Seamless*, we use the transformers library with model versions `facebook/mms-1b-all` and `facebook/hf-seamless-m4t-large`, respectively. Since both models require language codes and utilize language adapters, we use the language identification module of Whisper large-v3 to provide the required language code. We also transcribed when using the ground truth language code (which, however, is unrealistic in practical scenarios), but did not find it to consistently improve transcription performance. Additionally, we experimented with applying source separation but did not find it improves performance, which is in line with the findings of Cífka et al. (2024).

Audio-based Baselines. We first convert the waveform to mono in 16kHz, the input format of our speech embedding models, to extract speech features. To use XEUS, we use the ESPnet library (Watanabe et al., 2018) and disable masking. For WAV2VEC 2.0 and MMS-1B, we use the `transformers` (Wolf et al., 2020) library with model versions `facebook/wav2vec2-large-960` and `facebook/mms-1b-all`, respectively. Since all models extract several feature vectors whose size depends on the duration of the audio sample, we apply mean-pooling to aggregate these features into a single, fixed-length *speech embedding*. We also experimented with source separation but observed that it resulted in similar detection performance with worse generalization. This indicates that, indeed, semantics of sung lyrics are being captured, and that source separation is not robust w.r.t. audio artifacts.

Text-based Baselines. For LLM2VEC, we use `McGill-NLP/LLM2Vec-Meta-Llama-3-8B-Instruct-mntp`, i.e., the mntp-tuned (masked next token prediction) of Llama3 8B, following Labrak et al. (2024). For MINILMV2, BGE-M3, and BGE-ML-GEMMA, we utilize the `sentence-transformers` (Reimers and Gurevych, 2019) library with model versions `sentence-transformers/all-MiniLM-L6-v2`, `BAAI/bge-m3`, and `BAAI-bge-multilingual-gemma2`, respectively. Finally, for UAR models, we use `UAR-MUD` and `UAR-CRUD`, respectively. To stay within memory constraints, we truncate the input to each model to a maximum of 512 tokens. Note that this only affects a handful of songs.

Audio generation. To generate songs with Suno, we use their latest stable audio generation model, v3.5. Crucially, unlike previous versions that can only generate relatively short songs, it can create songs with up to 4 minutes, making them much more realistic. Specifically, we copy the LLM-generated lyrics into the *Lyrics* field and the song's corresponding genre into the *Style of Music* field. We follow this process using both synthetic and human-written lyrics. For the latter, a few songs were blocked during generation, making our *Partly-Fake* subset slightly smaller than the human-written one.

For our Udio subset used to test generalization, we use the latest and highest-quality udio-130 v1.5 model. We copy the LLM-generated lyrics for the stratified subset of 260 samples from the test set of lyrics into the *Lyrics Editor* field and fill the song's genre to *Describe your Song*. For controllability, we set *Lyrics Strength* to 100% and the seed to 42, leaving the rest unchanged. Since Udio does not support generating songs with real lyrics (i.e., *Partly-Fake*), we could not consider this scenario.

Since both Suno and Udio generate two songs with different audio per generation requests, we compute features, train models, and evaluate both independently, and then average over them.

Audio perturbations. We use pedalboard (Sobot, 2021) and librosa (McFee et al., 2015) to perturb audio. To simulate real-life audio attacks, we only perturb AI-generated audio and base our implementation on Afchar et al. (2024).

MLP training. To evaluate unimodal features, we

Model Type	Input	Processing Pipeline	Classifier
CNN Baseline	Audio waveform	Amplitude spectrogram ⇒ CNN	-
GT Lyrics Baseline	Ground truth lyrics	Text embedding model (LLM2Vec)	MLP (256, ReLU, 128, 2)
Unimodal (Text)	Audio waveform	Transcriber ⇒ Text embedding model	MLP (256, ReLU, 128, 2)
Unimodal (Speech)	Audio waveform	Speech embedding model ⇒ Mean pooling	MLP (256, ReLU, 128, 2)
DE-DETECT (multi-view)	Audio waveform	Transcriber (Whisper) ⇒ Text Embedding Speech embedding ⇒ Mean Pooling Linearly project both to 128, concatenation	MLP (128, ReLU, 128, 2)

Table 5: Training overview for each model type. Each is trained on the same set of songs in each scenario. For models other than CNN, only the MLP classifier is trained, while the rest of the processing pipeline remains frozen.

train a multi-layer perceptron (MLP) with two hidden layers of size 256 and 128, respectively, and ReLU activation function. For the multimodal fusion MLP, we first project each feature to an intermediate representation of size 128. After concatenation, we apply a ReLU activation function and a linear layer with size 128 before the final classification layer. They are each optimized with AdamW (Loshchilov and Hutter, 2017) optimizer with a learning rate of 1e-3, scaled down by a factor of 0.1 if the training loss does not increase for five consecutive epochs. We also experimented with different settings and classifiers, such as kNN, but noticed that the specific configuration of both unimodal and multimodal MLPs does not make a significant difference in detection results. For MLP training, we use `pytorch-lightning` (Falcon and The PyTorch Lightning team, 2019), a wrapper of `PyTorch` (Paszke et al., 2019).

B Information about Feature Extractors

We distinguish between models employed for *Automatic Speech Recognition (ASR)* to obtain lyric transcripts (for the text branch) and models used to extract *speech embeddings* that capture acoustic and paralinguistic cues directly from the audio (for the speech branch).

Text Features. LLM2VEC (BehnamGhader et al., 2024) is an unsupervised method transforming autoregressive LLMs into text encoders in a three-step process. First, bidirectional attention is enabled by modifying the causal attention mask to a bidirectional one. The next step is masked next-token prediction (MNTP), where the model is trained on a small dataset to adapt it to this new attention mask. The final, optional step consists of SimCSE (Gao et al., 2021) learning, where the model is adapted on larger, more diverse datasets to improve sequence representation for downstream tasks.

Universal Authorship Attribution models (UAR; Rivera-Soto et al., 2021) capture capture authorial writing style. They exist in variants MUD (UAR-MUD) and CRUD (UAR-CRUD), trained on texts from 1 million and 5 Reddit users, respectively.

Finally, BGE-ML-GEMMA adapts Gemma2 9B (Riviere et al., 2024) to a multilingual text embedded using the M3-Embedding methodology by Chen et al. (2024a) on diverse multilingual datasets, resulting in a strong general text embedding model, particularly excelling in multilingual tasks.

Speech Features. WAV2VEC 2.0 (Baevski et al., 2020) uses self-supervised learning to learn speech representations from raw audio. It uses a convolutional network to create latent representations and a Transformer to build contextualized representations. Pre-training involves identifying masked quantized latent representations, enabling powerful representations from unlabeled data for downstream speech tasks.

Next, MMS-1B (Pratap et al., 2023) is a multilingual speech model that supports speech in over 1,000 languages. It expands the number of supported languages by over 40x, trained using self-supervised learning with Wav2vec 2.0 using data unlabeled from publicly available religious texts. For its use in our *speech branch* (i.e., for extracting embeddings), we utilize an *ASR-finetuned variant* of MMS-1B. We leverage the encoder outputs from this variant to capture rich acoustic and paralinguistic features relevant to sung speech, rather than its final transcribed text output. This application is distinct from using MMS-1B as a full ASR system for transcription, a role in which we also evaluate it (c.f. Section 4.1).

Finally, XEUS (Chen et al., 2024b) represents the current state-of-the-art in multilingual speech representation learning, extending language coverage four-fold by combining speech from publicly accessible corpora with a newly created corpus of 7400+ hours from 4,057 languages. Moreover, a

Model	en	all
SPEECH EMBEDDINGS		
WAV2VEC 2.0	78.2	83.1
MMS-1B	80.7	88.8
XEUS	89.1	92.2
WAV2VEC 2.0 + MMS-1B	87.8	93.2
WAV2VEC 2.0 + XEUS	87.7	92.2
XEUS + MMS-1B	87.8	93.2
TEXT-BASED DETECTORS (LYRICS TRANSCRIPTION)		
UAR-MUD	85.2	89.6
BGE-ML-GEMMA	84.4	90.2
LLAMA3 $8\text{B}_{\text{LLM2Vec}}$	90.6	90.7
UAR-MUD + BGE-ML-GEMMA	84.0	90.0
UAR-MUD + LLM2VEC	91.2	92.2
BGE-ML-GEMMA + LLM2VEC	87.4	91.7
MULTIMODAL		
XEUS+LLM2VEC (OURS)	**93.9**	**94.9**
XEUS+UAR-MUD	92.0	94.3
XEUS+BGE-ML-GEMMA	91.8	94.0
WAV2VEC 2.0+LLM2VEC	92.2	92.9
WAV2VEC 2.0+UAR-MUD	85.9	90.5
WAV2VEC 2.0+BGE-ML-GEMMA	88.5	92.0
MMS-1B+LLM2VEC	91.8	93.1
MMS-1B+UAR-MUD	88.1	91.1
MMS-1B+BGE-ML-GEMMA	87.2	92.1

Table 6: Recall scores on English-language songs and macro-averaged over multilingual lyrics using different unimodal and multimodal feature combinations. For transcription, we use Whisper-large-v2.

novel joint dereverberation task is introduced to improve robustness.

C Ablation Study

In Table 6, we ablate the choice of fusing speech and transcript-based lyrics embeddings from XEUS and LLAMA3 $8\text{B}_{\text{LLM2Vec}}$, respectively. We late-fuse two of each of the best-performing text and speech features both in unimodal and multimodal combinations, resulting in our model, DE-DETECT While some unimodal combinations improve performance compared to only using one feature, none reaches our multimodal model's performance. Moreover, other multimodal feature combinations get close to the performance of DE-DETECT (e.g., XEUS+UAR-MUD), none outperforms DE-DETECT. However, multimodal methods consistently outperform their unimodal counterparts. Overall, this further demonstrates the robustness of our pipeline with respect to different components.

Model	en	all
TEXT-BASED DETECTORS (LYRICS TRANSCRIPTION)		
Neural Embeddings		
UAR-CRUD	81.9	88.2
MINILMV2	80.8	87.3
BGE-M3	84.7	87.7
BGE-ML-GEMMA	84.4	90.2
Metrics based on Llama3 8B Per-Tokens Probabilities		
PERPLEXITY	53.4	34.9
MAX. NEG. LL	61.4	55.8
SHANNON ENTROPY	56.5	59.8
MIN-K%PROB (K=10)	66.0	54.0

Table 7: Recall scores on English songs and macro-averaged over multilingual lyrics using additional neural and probabilistic features based on Llama3 8B per-tokens probabilities using Whisper large-v2 transcripts.

D Results using Additional Features

Furthermore, we show results using additional neural features and several probabilistic features based on Llama3 8B per-tokens probabilities in Table 7. For neural features, we use another variation of UAR, UAR-CRUD, trained on a smaller dataset (Rivera-Soto et al., 2021). Moreover, we evaluate two more text embedding models, MINILM-L6-V2 (Wang et al., 2021), an efficient lightweight model, as well as another recent strong text embedders, BGE-M3 (Chen et al., 2024a). PERPLEXITY (Beresneva, 2016) corresponds to the overall likelihood of the lyrics based on an exponential average using the negative log-likelihood (NLL). Shannon ENTROPY (Shannon, 1948; Lavergne et al., 2008) measures the diversity of text leveraging token-level NLL. MIN-K% PROB (Shi et al., 2024) selects a subset of the lowest token-level NLL values, with the size of the subset being K%. We use $K = 10$, following Labrak et al. (2024). Finally, MAX. NEG. LL (Mitchell et al., 2023; Solaiman et al., 2019; Gehrmann et al., 2019; Ippolito et al., 2020) uses the maximum token-level NLL as a single feature.

E Effect of Different Transcribers

We show complete results on a non-Whisper transcriber, *MMS-1b*, in Table 8, demonstrating similar performance and patterns as with using *Whisper large-v2*. This further demonstrates our method is not reliant on any specific architecture for its transcription component.

Model	en	all
SPEECH EMBEDDINGS		
WAV2VEC 2.0	78.2	83.1
MMS-1B	80.7	88.8
XEUS	**89.1**	**92.2**
TEXT-BASED DETECTORS (LYRICS TRANSCRIPTION)		
UAR-CRUD	78.8	88.1
UAR-MUD	78.0	88.1
MINILMV2	81.1	87.7
BGE-M3	81.9	87.7
BGE-ML-GEMMA	**85.4**	89.7
LLAMA3 $8B_{\text{LLM2Vec}}$	**85.4**	**90.3**
MULTIMODAL		
DE-DETECT	**89.1**	**93.6**

Table 8: Recall scores on English-language songs and macro-averaged over multilingual lyrics using a different transcriber, *MMS-1b*. In this setting, DE-DETECT combines XEUS embeddings with LLAMA3 $8B_{\text{LLM2Vec}}$ embeddings from *MMS-1b* transcriptions. While speech embeddings' scores remain unchanged when changing the transcriber, we include them for completeness.

F Metrics definition

Macro-Recall. Given a binary classification task with classes $C = \{c_1, c_2\}$ (in our case, real and AI-generated), recall for a specific class c_i is defined as:

$$\text{Recall}(c_i) = \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i}$$

where TP_i is the number of true positives for class c_i (samples correctly identified as c_i), and FN_i is the number of false negatives for class c_i (samples of c_i incorrectly identified as belonging to another class). Macro-recall is then the unweighted arithmetic mean of the per-class recalls:

$$\text{Macro-recall} = \frac{1}{|C|} \sum_{i=1}^{|C|} \text{Recall}(c_i)$$

This metric is chosen as it gives equal weight to the performance on each class, which is crucial for tasks where misclassification costs might be similar for all classes or when class imbalance is present, ensuring that the performance on a minority class is not overshadowed.

AUROC. The AUROC quantifies the overall ability of a classifier to discriminate between positive and negative classes across various decision thresholds. It is the area under the ROC curve, which plots the true positive rate (TPR, equivalent to recall or sensitivity) against the false positive rate (FPR) at different threshold settings.

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

where TP, FN, FP (False Positives), and TN (True Negatives) are defined with respect to a designated positive class (e.g., AI-generated). An AUROC of 1.0 signifies a perfect classifier, correctly distinguishing all positive and negative instances, while an AUROC of 0.5 suggests performance no better than random guessing.