Scalable Music Cover Retrieval Using Lyrics-Aligned Audio Embeddings

J. Affolter[1,2], B. Martin[1], E. V. Epure[1], G. Meseguer-Brocal[1], and F. Kaplan[2]

[1] Deezer Research, Paris, France research@deezer.com
[2] EPFL, Lausanne, Switzerland

Abstract. Music Cover Retrieval, also known as Version Identification, aims to recognize distinct renditions of the same underlying musical work, a task central to catalog management, copyright enforcement, and music retrieval. State-of-the-art approaches have largely focused on harmonic and melodic features, employing increasingly complex audio pipelines designed to be invariant to musical attributes that often vary widely across covers. While effective, these methods demand substantial training time and computational resources. By contrast, lyrics constitute a strong invariant across covers, though their use has been limited by the difficulty of extracting them accurately and efficiently from polyphonic audio. Early methods relied on simple frameworks that limited downstream performance, while more recent systems deliver stronger results but require large models integrated within complex multimodal architectures. We introduce LIVI (Lyrics-Informed Version Identification), an approach that seeks to balance retrieval accuracy with computational efficiency. First, LIVI leverages supervision from state-of-the-art transcription and text embedding models during training to achieve retrieval accuracy on par with—or superior to—harmonic-based systems. Second, LIVI remains lightweight and efficient by removing the transcription step at inference, challenging the dominance of complexity-heavy pipelines.

Keywords: Music Cover Retrieval · Representation Learning · Audio to Text Alignment

1 Introduction

In information retrieval, tasks such as Near-Duplicate Detection [40, 45] and Entity Resolution (or Record Linkage) [57] aim to identify and link semantically equivalent entities. In the music information retrieval domain, an analogous task is Music Cover Retrieval—also known as Version Identification or Cover Detection—where the goal is to recognize distinct renditions of the same underlying composition [49]. Robust systems are critical for catalog management, copyright enforcement, cross-platform track linking, and music retrieval [49].

Defining similarity between covers, however, is challenging: models must account for wide variations in tempo, pitch, structure, lyrics or recording conditions [49]. State-of-the-art approaches have largely focused on harmonic and melodic features, relying on complex pipelines that aim to achieve invariance

to these attributes [18, 26, 30]. While effective, such models demand significant training time and computational resources, and their growing reliance on deep or multimodal architectures further limits scalability and reproducibility [1].

Unlike melodic or harmonic features, which may vary widely across renditions, prior work has shown that lyrics constitute a strong invariant [1, 10, 15, 31, 46]: (i) they are typically preserved across renditions, (ii) they largely retain their semantic content even under translation or minor rewrites, and (iii) they provide decisive cues for distinguishing works that share similar harmonic or melodic profiles. A well-known illustration is Jimi Hendrix's cover of "All Along the Watchtower" from Bob Dylan, where harmony and melody diverge strongly from the original, yet the lyrics remain largely intact. Despite this, their potential has been underexplored, mainly due to two obstacles: the availability of editorial lyrics at scale—often restricted by third-party licensing—and the challenge of extracting lyrics from polyphonic audio [49], a task where recent advances have improved accuracy but remain computationally demanding. While early works that use lyrics [1, 46] relied on relatively simple approaches to derive lyric representations, resulting in limited downstream performance, more recent work [15, 31] achieved stronger results but integrates transcription into a complex multimodal architecture, increasing model size and computational cost.

Our work builds on the hypothesis that songs with semantically similar lyrics are likely to be covers [10]. To test this idea, we first construct a pipeline that represents songs in a lyric-informed embedding space, obtained by applying an Automatic Speech Recognition (ASR) system followed by a multilingual text encoder. This design is motivated by two factors: (i) clean editorial lyrics are rarely available at scale, making transcription a necessary step, and (ii) modern multilingual text encoders, pretrained for semantic similarity, provide a powerful and readily applicable representation space. While this pipeline achieves strong performance, its reliance on full transcription makes it computationally costly. Motivated by the need for efficiency in real-world deployment, we introduce LIVI (Lyrics-Informed Version Identification), a model that learns to project latent audio representations directly into the lyric embedding space defined by the pipeline. In doing so, LIVI removes the transcription step, reducing inference cost while preserving retrieval accuracy.

Despite its relative simplicity and the absence of explicit fine-tuning for the downstream task, LIVI achieves performance on par with—or superior to—state-of-the-art systems. It delivers an efficient, reproducible[3], and domain-grounded alternative, challenging the dominance of complexity-heavy multimodal systems. By design, our method applies only to tracks with sufficient vocal content, with a preprocessing stage used to exclude those lacking it. While this restriction narrows the scope of applicability, its practical impact is limited given the predominance of vocal music in mainstream repertoires and its central role in industrial applications [11]. Moreover, a lyrics-informed approach such as LIVI could naturally be complemented by harmonic features, as in [46], forming part of a broader system that integrates both textual and musical cues.

[3] Code available at `https://github.com/deezer/LIVI-Lyrics-Informed-Version-Identification`

2 Related Works

Version Identification. Research on Music Cover Retrieval has undergone several stages of development over the past two decades. Early systems based on hand-crafted features achieved encouraging results on small benchmarks, but failed to scale due to their reliance on costly alignment techniques such as dynamic time warping (DTW) [33, 49]. The advent of deep learning marked a turning point, enabling data-driven feature learning from harmonic and melodic representations such as predominant melody, pitch class profiles (PCP), or the constant-Q transform (CQT) [50]. Since then, progress has followed two main directions: one line of work pushes towards increasingly deep architectures such as ResNet [2, 15–18, 26, 30, 41, 53, 54], while another emphasizes musically informed inductive biases to achieve invariance to transformations such as transposition or structural changes [1, 13, 14, 50, 51]. Beyond these, multimodal methods that integrate complementary features have demonstrated clear advantages over unimodal models [1, 14, 15, 46].

Although lyrics offer a strong discriminative signal for version identification [1, 15, 31, 46], their integration into Music Cover Retrieval systems is relatively recent, primarily due to the absence of large-scale datasets with clean, time-aligned lyrics and the difficulty of transcribing lyrics from polyphonic audio [49]. In early attempts, [46] adopted a Singing Voice Recognition (SVR) framework combining a TDNN-based acoustic model [22] with a language model to decode phoneme sequences compared via string matching. [1] proposed a lightweight Automatic Lyrics Recognition (ALR) that produced character posteriorgrams subsequently processed by a second, independently trained model fine-tuned for the retrieval task. Although these approaches marked important first steps, they relied on acoustic models to represent lyrics, yielding limited retrieval performance. Moreover, transcription models were trained from scratch on English-only datasets [32], restricting multilingual generalization and adding the overhead of training a separate retrieval model.

Recent advances in lyrics-based cover detection include dedicated datasets, such as LyricsCovers 2.0 [4], and new methods [5, 15, 31, 38]. Among these, [15] achieved stronger results by adapting Whisper [36] with lightweight prefix- and suffix-tuning, enabling faster inference while keeping most parameters frozen. However, the approach remains integrated into a complex multimodal pipeline, increasing model size and computational cost, and making it difficult to isolate the impact of lyrics. Reproducibility is further constrained by the lack of open-source implementations and technical details [1].

Automatic Lyrics Recognition. Modern Automatic Speech Recognition (ASR) systems are based on end-to-end neural architectures that map raw audio directly to text, in contrast to older hybrid pipelines combining hidden Markov models with deep neural networks (HMM-DNN) [29]. Recent advances in Automatic Lyrics Transcription have been driven by systems such as AudioShake v3 [9], which achieves state-of-the-art accuracy but remains proprietary. As an open-source alternative, Whisper [36] has been widely adopted for lyric transcrip-

tion [43]. Its robustness to noise, accents, and other real-world variability makes it particularly suitable for the heterogeneous conditions of music audio [9, 37]. As an encoder–decoder Transformer, Whisper encodes audio into latent representations, which the decoder attends to via cross-attention to generate transcriptions autoregressively.

Audio to Text Alignment. Recent advances in audio–text modeling focus on learning aligned representations across modalities, typically through contrastive pretraining in a shared embedding space. Inspired by CLIP [35], CLAP-style models [12, 20, 25, 27, 55] jointly train audio and text encoders to maximize similarity between paired data, achieving strong zero-shot performance in tagging, retrieval, and captioning. However, these methods typically rely on high-level textual descriptors rather than structured content such as lyrics.

Several works have instead explored aligning audio with lyrics, though with different objectives. Durand et al. [19] proposed a contrastive learning framework in which singing audio and lyric transcripts are encoded as sequences of frame- and token-level embeddings. A similarity matrix is then computed to determine the optimal alignment path, enabling precise word-level synchronization. Yu et al. [52] addressed cross-modal retrieval by training parallel audio and lyric encoders with a Deep Canonical Correlation Analysis (DCCA) loss. Their approach projects spectrogram-based audio features and text embeddings into a joint space, but the reliance on correlation objectives makes training computationally expensive. In contrast, our method learns global song-level embeddings that integrate lyric semantics directly into the audio representation.

3 Methodology

We present LIVI (Lyrics-Informed Version Identification), an approach that leverages the invariance of lyrics across renditions to identify covers (Figure 1). Our starting point is a lyrics-informed pipeline designed to maximize retrieval accuracy, without regard to efficiency. Audio is first transcribed using an encoder–decoder ASR model, and the resulting text is then embedded with a multilingual model fine-tuned for semantic similarity. This produces an embedding space in which semantically similar lyrics—even when expressed in different languages—cluster closely, while unrelated text lies further apart.

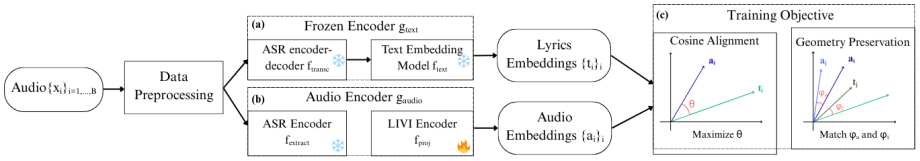

Fig. 1: Overview of the proposed LIVI framework. (a) A frozen text encoder (g_{text}) combines an ASR model with a pre-trained text embedding model to produce lyrics embeddings t_i. (b) An audio encoder (g_{audio}) projects ASR encoder latent representations into the same embedding space. (c) Training optimizes a combined objective: pointwise alignment of a_i with t_i under cosine similarity, and geometry preservation ensuring that pairwise similarities between audio embeddings mirror those of their corresponding lyric embeddings.

Its key drawback lies in the reliance on transcription, where the ASR autoregressive decoder introduces considerable computational overhead. To overcome this, LIVI discards the decoder and trains an audio encoder to map latent ASR states directly into the lyric-informed embedding space derived from the pipeline. This removes the need for full transcription while preserving retrieval accuracy, resulting in a more efficient and scalable solution.

3.1 Problem Formulation

We formulate Music Cover Retrieval as a similarity ranking problem over embeddings produced by an encoder g. Let $\mathcal{C}$ be a catalog of music tracks and $g : \mathcal{C} \to \mathbb{R}^d$ map each track $x \in \mathcal{C}$ to an embedding $\mathbf{e}_x \in \mathbb{R}^d$. Given a query $q \in \mathcal{C}$ (i.e., a song), the system assigns to each $x \in \mathcal{C} \setminus \{q\}$ a cosine similarity score

$$s(q, x) = \cos(\mathbf{e}_q, \mathbf{e}_x) = \frac{\mathbf{e}_q^\top \mathbf{e}_x}{\|\mathbf{e}_q\|_2 \, \|\mathbf{e}_x\|_2}$$

and returns the catalog ordered in descending order by $s(q, \cdot)$. Let $\mathcal{V}(q) \subset \mathcal{C}$ denote the set of versions of q. The desired ranking property is

$$s(q, v^+) > s(q, v^-) \qquad \forall v^+ \in \mathcal{V}(q), \;\; v^- \in \mathcal{C} \setminus \mathcal{V}(q)$$

Accordingly, the encoder must learn an embedding space in which versions are embedded more closely than to non-versions.

3.2 Framework Overview

We first define the lyrics-informed embedding space, demonstrating the effectiveness of lyric semantic similarity for cover song retrieval. This embedding space then serves as supervision for training the audio encoder. Given an audio excerpt x_i from a track $x \in \mathcal{C}$, its lyrics embedding $t_i \in \mathbb{R}^d$ is obtained by composing an encoder-decoder ASR model f_{transc} with a pre-trained text embedding model f_{text}. This composition can be seen as a fixed encoder:

$$g_{\text{text}} = f_{\text{text}} \circ f_{\text{transc}} : \begin{cases} \mathcal{C} \;\to\; \mathbb{R}^d \\ x_i \;\mapsto\; t_i \end{cases}$$

Next, we define an audio encoder that projects raw audio into the lyrics-informed embedding space. Given the same audio excerpt x_i, latent features are extracted from the ASR encoder via f_{extract} and projected by f_{proj} to yield the audio embedding:

$$g_{\text{audio}} = f_{\text{proj}} \circ f_{\text{extract}} : \begin{cases} \mathcal{C} \;\to\; \mathbb{R}^d \\ x_i \;\mapsto\; a_i \end{cases}$$

The objective is to learn g_{audio} such that audio embeddings a_i are aligned with their corresponding lyric embeddings t_i under cosine similarity. Formally, this corresponds to minimizing the loss:

$$\mathcal{L}_{\cos} = \sum_{x_i \in \mathcal{C}} \Big(1 - s\big(g_{\text{audio}}(x_i), g_{\text{text}}(x_i)\big)\Big)$$

Yet the training objective can be pushed further given the data available: since the lyrics-informed space is fixed and the target lyrics embeddings are accessible during training, one can leverage not only pointwise alignment but also the geometry of the target space. More specifically, the inter-sample distances between lyrics embeddings can serve as an additional supervision signal to guide the training of the audio encoder. This is achieved by enforcing that pairwise similarities between audio embeddings, $s(a_i, a_j)$, match those of the corresponding lyrics embeddings, $s(t_i, t_j)$. Formally, given a batch $\{(x_i, t_i)\}_{i=1}^{B}$, this component of the training objective takes the form:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{B^2} \sum_{i,j=1}^{B} \Big(s(a_i, a_j) - s(t_i, t_j) \Big)^2 .$$

This yields the final objective optimized during training, combining a pointwise alignment term $\mathcal{L}_{\cos}$ with a geometry-preservation term $\mathcal{L}_{\text{MSE}}$:

$$\mathcal{L}_{\text{total}} = \alpha \, \mathcal{L}_{\cos} + (1 - \alpha) \, \mathcal{L}_{\text{MSE}}, \qquad \alpha \in [0, 1]$$

Unlike standard multimodal representation learning [20, 27, 48], which jointly trains both audio and text encoders to construct a joint embedding space, our method fixes the textual space and adapts only the audio encoder. This subtle difference allows the training process to exploit the geometry of the lyrics-informed embedding space directly, rather than relying on implicit structure induced by contrastive objectives and in-batch negatives.

4 Implementation Details of LIVI

4.1 Data Preprocessing

Because transcription models are trained primarily on speech, they tend to hallucinate in non-vocal sections, generating spurious outputs despite the absence of linguistic content [6, 23, 28, 47]. A dedicated preprocessing stage is therefore introduced to filter out tracks with insufficient lyrical content and extract vocal-only segments. We employ a proprietary model deep learning model to estimate vocalness, as it yields more reliable predictions than Whisper's integrated voice activity detection, which we found to suffer from lower transcription accuracy and frequent hallucinations (see Section 5.2). This model consists of the Musicnn architecture[4] [34], augmented with a single linear layer of dimension 2 for binary classification. It estimates a vocalness probability v for each non-overlapping 3s audio window, and a global vocalness score is obtained by averaging across all windows. Tracks with a score below a threshold $\lambda \in [0, 1]$ are excluded to ensure sufficient lyrical content. Second, windows with $v \geq 0.5$ are retained as vocal segments, concatenated into contiguous regions, and symmetrically padded by up to 10s to smooth temporal boundaries. The resulting segments are then truncated or zero-padded to a fixed length of 30s to match the input requirements

[4] Model architecture and pretrained weights are available at `https://github.com/jordipons/musicnn`.

of the ASR model. For a given track $x \in \mathcal{C}$, this process produces multiple audio segments x_i, which are treated independently from one another and used as inputs to both the frozen text encoder g_{text} and the audio encoder g_{audio}.

4.2 Lyrics-Informed Embedding Space

The lyrics-informed embedding space (Figure 1.a) is defined as $\mathcal{T} = \{g_{\text{text}}(x_i) \mid x_i \in \mathcal{C}\}$, where $g_{\text{text}} = f_{\text{text}} \circ f_{\text{transc}}$ maps an audio excerpt x_i to its lyric embedding $t_i \in \mathbb{R}^d$. This space clusters semantically similar lyrics— assumed to represent versions—closer together than unrelated ones, and serves as the target structure for training the audio encoder g_{audio}.

Transcription model f_{transc} Given its suitability for this task (more details in Section 2), we use *whisper-large-v3-turbo*, an optimized variant of *whisper-large-v3* that offers comparable transcription accuracy with up to 8× faster inference. Its architecture combines a convolutional front-end, 32 Transformer encoder layers, and a 4-layer Transformer decoder generating transcriptions autoregressively.

Text embedding model f_{text} As mentioned in 3, the objective is to construct a semantically meaningful space for lyric transcriptions, providing a robust structure for the audio encoder to align with. Recent progress in Natural Language Processing has produced text embedding models especially well suited to this purpose. Most are derived from large pre-trained language models and fine-tuned for semantic similarity, typically within the Sentence-BERT framework [39], which maps full sentences into fixed-size embeddings that preserve semantic proximity—even in multilingual settings. Based on an evaluation of six multilingual text embedding models for the downstream task (see 5.1), we select *gte-multilingual-base* [56], an encoder-only Transformer that produces 768-dimensional embeddings across more than 70 languages. It achieves SOTA results in multilingual retrieval for models of comparable size [56] and outperforms alternatives in our evaluations. In our case, we rely on this off-the-shelf model without additional fine-tuning, as it already provides strong results. We leave task-specific fine-tuning for future work, where it could further enhance performance.

4.3 Audio Encoder

The audio encoder g_{audio} (Figure 2) maps an audio excerpt $x_i \in \mathcal{C}$ to an embedding $a_i \in \mathbb{R}^d$ aligned with its lyric-based counterpart $t_i = g_{\text{text}}(x_i)$. Built on top of Whisper's frozen encoder, which provides frame-level representations subsequently aggregated by an attention-based pooling mechanism, the projection head refines Whisper's latent space rather than learning cross-modal alignment from scratch. This design keeps the model compact and computationally efficient during both training and inference.

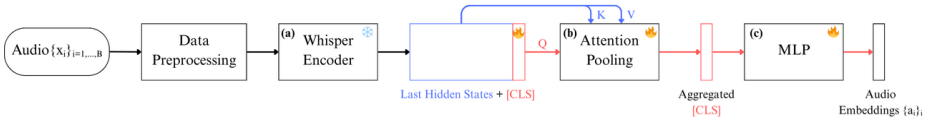

Fig. 2: Architecture of the audio encoder g_{audio}. (a) Raw audio is first processed by the Whisper encoder to obtain hidden representations. (b) A [CLS] token is appended to aggregate frame-level features using an attention pooling mechanism. (c) A multi-layer perceptron projects the pooled representation into the lyrics-informed embedding space, yielding the final audio embedding a_i.

Feature Extractor (Fig. 2.a). We adopt the encoder of the Whisper model used in the lyrics-informed embedding space as our audio backbone. This choice is motivated by two considerations: (i) its internal representations, shaped by the ASR training objective, are expected to capture phonetic and linguistic information [24] that makes them suitable for alignment with lyrics embeddings; and (ii) reusing the encoder from the same ASR model enables our model to specifically target the decoder in order to have an efficient alternative. Consequently, we keep the encoder frozen to preserve this alignment and maintain the latent structure learned during Whisper's large-scale training. Given an 80-channel log-Mel spectrogram, the encoder produces a sequence of hidden states $H \in \mathbb{R}^{L \times d_w}$, where $d_w = 1280$ is the latent dimension and $L = 1500$ the number of frames for a 30s input.

Attention-based Temporal Pooling (Fig. 2.b). To reduce frame-level representations into a fixed-dimensional vector suitable for projection into the lyrics-informed embedding space, we adopt an attention-based pooling mechanism inspired by [44]. A learnable [CLS] token $q_{\text{cls}} \in \mathbb{R}^{d_w}$ is appended to the hidden states H and acts as the sole query in a single-head attention mechanism with Rotary positional embeddings (RoPE) [42]. Formally:

$$\text{Attention(Q, K, V)} = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \qquad Q = q_{\text{cls}}W_Q,\ K = HW_K,\ V = HW_V$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d_w \times d_k}$ are learnable projection matrices, and d_k denotes the dimensionality of the key vector, which we set to d_w in our implementation. The attention weights determine the relative importance of each frame, guiding how information from the sequence is aggregated into the [CLS] representation. This representation is then passed through a residual feed-forward block with LayerNorm to yield the final pooled embedding $h \in \mathbb{R}^{d_w}$.

Projection Head (Fig. 2.c). The pooled vector h is finally projected into the 768-dimensional lyric-informed embedding space through a four-layer MLP with hidden sizes [3072, 2048, 2048, 1536]—a configuration selected based on empirical validation, totaling 13.6M trainable parameters. Each intermediate layer is followed by LayerNorm and a ReLU activation, while the final layer outputs the audio embedding a_i used in the downstream retrieval task.

5 Experiments

Training Configuration. Training runs for three epochs with batch size 128 on a single NVIDIA RTX A5000 (24GB), taking about 33 hours. We use AdamW (weight decay 0.01, $\beta = (0.9, 0.98)$) with a fixed learning rate of 10^{-4} and linear warmup over the first 10k steps. Mixed precision is enabled for the Whisper encoder, while the rest of the model is trained in full precision. Early stopping is applied based on the average cosine similarity between audio and text embeddings on the validation set. For data, we use a subset of 679,692 entries from Discogs-VI [3], linked to a proprietary catalog to recover the corresponding .mp3 audio. Training pairs (z_i, t_i) are formed by (i) precomputing log-Mel spectrograms z_i with Whisper's feature extractor on extracted 30s vocal segments x_i (Section 4.1), and (ii) deriving lyrics embeddings t_i through the pipeline described in Section 4.2. We retain a subset of 1.5M pairs, of which 1.2M are used for training, 170k for validation and 170k for testing (80/10/10 split).

Benchmarks. We use the standard Covers80 [21] and SHS100k-TEST [7], along with the test set of Discogs-VI [3], restricted to entries with an available YouTube link. All datasets follow the same preprocessing pipeline: (1) tracks are linked to a proprietary catalog via fingerprinting after matching against YouTube audio with the provided links; and (2) tracks with insufficient vocal content (see Section 4.1) are discarded. At this point, it should be acknowledged that due to our lyrics-centered method, we exclude some of the tracks in the reference datasets. Consequently, 82.76% of Covers80 (116 tracks in 58 cliques of average size 2), 81.95% of SHS100k (890 tracks in 105 cliques of average size 7.28 ± 6.29), and 85.29% of Discogs-VI (72,316 tracks in 33,660 cliques of average size 3.04 ± 2.34) are retained for evaluation. We leave as future work the use of different musical modalities to retrieve these instances. Nevertheless, our experiments operate at a large scale of 72,316 tracks for Discogs-VI, yielding an evaluation setting that reflects real-world conditions.

Evaluation. We follow the standard evaluation setting for the retrieval task [50]. Given a query track, the system ranks all other tracks in the dataset according to their cosine similarity with the query in the embedding space (see 3.1), and retrieval performance is assessed using standard metrics: MR1, the mean rank of the first true positive; HR@1, the fraction of queries with the correct cover ranked first; and MAP@10, which evaluates precision within the top 10 results. Note that a query may correspond to multiple covers, with an average of 2 covers per clique in Covers80, 12 in SHS100k, and 6 in Discogs-VI.

5.1 Validation of the Lyrics-Informed Embedding Space

We evaluate the lyrics embeddings $t_i = g_{\text{text}}(x_i)$ in the downstream retrieval task to assess the performance of the lyrics-informed embedding space. Results are reported in Table 1 across six multilingual text embedding models f_{text}, supporting the choice of *gte-multilingual-base* in our pipeline. To approximate

	Metric		gte-b		e5-s		e5-l		e5-l-inst		jina		mpnet	
C80	MR1	↓	1.101	1.000	1.025	1.000	1.051	1.000	1.013	1.000	1.089	1.000	3.367	4.139
	HR1	↑	0.975	1.000	0.975	1.000	0.975	1.000	0.987	1.000	0.975	1.000	0.899	0.924
	MAP	↑	0.987	1.000	0.987	1.000	0.983	1.000	0.994	1.000	0.977	1.000	0.914	0.921
SHS	MR1	↓	2.810	4.033	5.612	4.826	3.446	2.190	3.413	2.620	3.934	4.851	4.702	5.050
	HR1	↑	0.909	0.917	0.909	0.909	0.926	0.934	0.926	0.917	0.909	0.926	0.884	0.909
	MAP	↑	0.852	0.863	0.829	0.842	0.858	0.866	0.867	0.863	0.863	0.880	0.836	0.848
D-VI	MR1	↓	13.21	12.16	14.56	21.22	15.33	19.53	13.44	16.96	13.81	17.44	40.95	39.96
	HR1	↑	0.929	0.934	0.927	0.924	0.922	0.919	0.926	0.923	0.925	0.924	0.878	0.902
	MAP	↑	0.893	0.913	0.890	0.895	0.891	0.897	0.897	0.901	0.886	0.905	0.800	0.861

Table 1: Music Cover Retrieval results across six text embedding models[5]. Each cell reports x|y, where x corresponds to results using transcribed lyrics embeddings $t_i = f_{\text{transc}}(f_{\text{text}}(x_i))$, and y to editorial lyrics embeddings $\tilde{t}_i = f_{\text{text}}(\ell_i)$. Underline indicates the best result within each row.

the upper bound of text embedding performance, we further assess the models on editorial lyrics ℓ_i rather than transcriptions, thus eliminating transcription noise. This evaluation is limited to subsets of the datasets for which editorial lyrics are available in our proprietary catalog, yielding 116, 167, and 4,623 tracks for Covers80, SHS100k, and Discogs-VI.

Among the six candidates, *gte-multilingual-base* emerges as the most effective since it ranks first across nearly all metrics on Discogs-VI, the largest and most representative benchmark of real-world conditions. Its performance approaches the ceiling defined by editorial lyrics, underscoring both its robustness to transcription noise and its suitability as the backbone of our pipeline. Additionally, results show that most multilingual encoders achieve competitive scores and, notably, reach ceiling performance on Covers80 when provided with editorial lyrics. This underscores the role of lyrics as a stable and discriminative signal for Music Cover Retrieval and validates the lyrics-informed embedding space as a robust supervisory signal: it captures the semantic structure necessary to distinguish versions. In our case, we rely on off-the-shelf multilingual models without additional fine-tuning, as they already provide strong results. We leave task-specific fine-tuning for future work, where it could further enhance performance.

5.2 Validation of the Vocal Detection Model

To assess the gains of the proprietary vocal detection model over Whisper's integrated Vocal Activity Detection (VAD) module, we evaluate transcription quality on the Discogs-VI test set restricted to tracks with editorial lyrics (4,623 tracks). We compare Word Error Rate (WER) between editorial lyrics and Whisper transcriptions obtained either from vocal segments extracted with the proprietary model or from non-overlapping 30-second segments processed with Whisper's VAD. In addition, we analyze hallucinated outputs commonly observed in Whisper transcriptions of non-spech audio (e.g., "thank you", "music", "subtitle") [6].

[5] gte-multilingual-base (gte-b), multilingual-e5-{small, large, large-instruct} (e5-{s, l, l-inst}), jina-embeddings-v3 (jina), multilingual-mpnet-base-v2 (mpnet).

While WER differences are not statistically significant, the average number of hallucinations per track is significantly lower with the proprietary model ($p = 1.24 \times 10^{-6}$), decreasing from 0.51 to 0.25. The total number of hallucinations across all transcriptions is likewise reduced (1,023 vs. 509).

5.3 Alignment of Audio with Lyrics-Informed Embeddings

We evaluate the performance of the audio encoder g_{audio} by examining whether it fulfills its training objective, namely aligning audio embeddings with their lyric-based counterparts. Alignment is assessed at both the *segment* and *track* levels under cosine similarity. At the segment level, cosine similarity is computed for each of the 167,484 audio–lyrics embedding pairs (a_i, t_i) taken from the test split of the audio encoder training phase, with $a_i = g_{\text{audio}}(x_i)$ and $t_i = g_{\text{text}}(x_i)$. At the track level, segment embeddings from the same recording are averaged to form a global representation, which is then compared against the lyrics embedding derived from the full transcription (60,524 tracks). Segment-level embeddings yield a mean similarity of 0.8574 (std: 0.0757), while aggregated track-level embeddings reach 0.9109 (std: 0.0379). While the higher mean and lower variance at the track level suggest that the encoder integrates local cues into stable global representations, the results further demonstrate that our approach achieves tight audio-lyric alignment.

5.4 Application to Music Cover Retrieval

We evaluate in Table 2 the audio encoder g_{audio} on the retrieval task by computing segment-level embeddings $a_i = g_{\text{audio}}(x_i)$ for all vocal segments x_i of each track (see Section 4.1) and aggregating them into a global representation. To disentangle the contributions of our method, we first use lyrics embeddings as an approximate upper bound, since LIVI is trained to project audio into this space: local embeddings $t_i = g_{\text{text}}(x_i)$ are averaged into t_{local}, while global embeddings t_{global} are computed from full transcriptions. As a second ablation, we introduce a Whisper baseline obtained by mean-pooling frame-level encoder states (Figure 1.a); this removes the projection module and training stage, thereby isolating the effect of LIVI's alignment beyond raw ASR features. Finally, we compare against state-of-the-art systems for version identification [2, 16, 41, 54][6], using official implementations and pretrained checkpoints from [41].

LIVI consistently outperforms lyrics embeddings derived from averaged local segments and approaches—or even surpasses—the upper bound defined by global embeddings. Whisper embeddings, by contrast, yield poor performance across all datasets, underscoring the gains from LIVI's architecture and training strategy. Through attention-based pooling and a projection network, Whisper's outputs are effectively mapped into the lyrics-informed embedding space, producing representations sufficiently discriminative to recognize versions. Compared to audio baselines, LIVI delivers competitive or superior performance, with particularly

[6] An evaluation against [31] was not performed due to concurrent publication timelines.

	Metric		LIVI	t_{global}	t_{local}	Whisper	Bytecover2	CLEWS	CQTNet	DViNet
C80	MR1	↓	1.51	**1.10**	1.92	7.67†	1.57	2.24	3.43	3.05
	HR1	↑	0.949	**0.975**	0.937	0.632†	0.865	0.835	0.848	0.861
	MAP	↑	0.966	**0.979**	0.945	0.691†	0.877	0.880	0.856	0.886
SHS	MR1	↓	**3.25**	6.05	5.52	6.56†	4.66	3.97	5.59	7.63
	HR1	↑	0.935	**0.954**	0.925	0.777†	0.953	0.931	0.900	0.931
	MAP	↑	0.875	**0.910**	0.870†	0.558†	0.884	0.847†	0.789†	0.859†
D-VI	MR1	↓	**232.21**	275.77	360.21†	1051.36†	312.32†	410.39†	810.89†	507.04†
	HR1	↑	0.853	**0.856**†	0.843	0.524†	0.843†	0.816†	0.641†	0.751†
	MAP	↑	**0.923**	0.832†	0.817†	0.406†	0.812†	0.790†	0.568†	0.719†

Table 2: Comparison of LIVI audio encoder against transcription-, Whisper-, and audio-based baselines for Music Cover Retrieval. t_{global} denotes lyrics embeddings from the *full transcription*, while t_{local} and LIVI correspond to the mean of 30s segment-level embeddings (lyrics and audio). Bold numbers indicate the best result and underlined numbers the second-best within each row. † denotes a significant difference ($p<0.05$, Holm-Bonferroni-corrected - Wilcoxon Signed-Rank Test for MR1 and MAP, McNemar's Test for HR1) to LIVI.

strong results on Covers80 and Discogs-VI, where it outperforms all in HR@1 and MAP@10. On SHS100k, Bytecover2 slightly outperforms LIVI, but this difference can be partially explained by dataset characteristics: SHS100k includes a notable fraction of parodies, covers that preserve the melody but replace lyrics with ironic content, which are challenging for our lyrics-centered approach. While differences on Covers80 and SHS100k are often not statistically significant, likely due to limited dataset sizes and ceiling effects across models, LIVI achieves statistically significant improvements over all audio baselines across all metrics on the largest and most diverse benchmark Discogs-VI.

Together, these results show that LIVI generalizes effectively to Music Cover Retrieval. It nearly matches the lyrics-based upper bound, clearly outperforms raw Whisper representations, and competes with or surpasses state-of-the-art audio baselines on vocal tracks. Its relatively simple and reproducible design contrasts with the complexity of models like ByteCover2, yet its performance across datasets establishes LIVI as a compact and powerful alternative.

5.5 Model Size and Inference

A central motivation for LIVI was to circumvent the computational costs of Whisper's autogressive decoding process. To quantify efficiency gains, we measure end-to-end latency on 200 randomly sampled tracks from Discogs-VI, com-

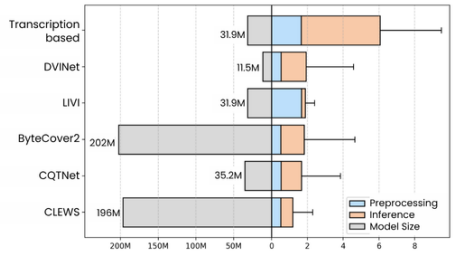

Fig. 3: Runtime and model size comparison. Average preprocessing and inference times are shown alongside model sizes for LIVI and baseline models. Error bars denote std across runs.

paring LIVI with the transcription-based and audio baselines. For all models, we separate preprocessing (audio loading for audio baselines, plus vocal detection for LIVI) from model inference (forward pass). We also report model size in terms of trainable parameters (Figure 3). The transcription pipeline requires on average 6.07s per track, with Whisper alone contributing 4.41s. LIVI reduces total latency to 1.90s, corresponding to a 3.2× end-to-end speed-up. Where LIVI truly stands out is at the inference stage: its forward pass completes in 0.22s, nearly 20× faster than Whisper and 3–6× faster than audio baselines (0.66–1.40s). In addition, its latency variance is much lower (std. 0.12 vs. 1.03–2.66), ensuring more predictable and stable performance. The main overhead instead arises from the preprocessing stage, where vocal detection and segmentation remain relatively costly compared to the inference of LIVI itself. Reducing this cost therefore represents an important direction for future work. Yet LIVI offers a favorable trade-off between complexity and accuracy when compared to baselines. With 31.9M parameters, it is substantially lighter than large systems such as ByteCover2 (202.3M) and CLEWS (196.8M) while remaining competitive in accuracy, and it surpasses similarly sized models like CQTNet and DViNet, particularly on large-scale benchmarks such as Discogs-VI.

6 Conclusion

This work introduced LIVI, an approach for Music Cover Retrieval that balances retrieval accuracy with computational efficiency. With no task-specific fine-tuning and a relatively simple architecture, it stands in contrast to the prevailing trend toward increasingly complex and resource-intensive models. Yet, when tracks have lyrics, LIVI achieves performance on par with these systems, demonstrating that musically informed inputs—rather than architectural escalation—can deliver state-of-the-art results while remaining computationally efficient at both training and inference.

Several limitations of the proposed approach should be acknowledged. First, the text embedding model is used off-the-shelf and is not fine-tuned specifically for the retrieval task, leaving room for further performance gains. Second, LIVI is inherently restricted to musical content with sufficient vocal material; purely instrumental tracks or songs with minimal vocals are therefore excluded, limiting the universality of the method and making its applicability dependent on the availability and quality of lyrics information. Incorporating complementary harmonic features could help mitigate this limitation, particularly for cover versions that preserve melodic structure while substantially altering lyrics or falling outside the scope of the current model. Third, the vocal detection component used during preprocessing is proprietary, which impacts full reproducibility; however, open-source alternatives [8, 37] for vocal activity detection could be explored with limited changes to the pipeline. Finally, although the vocal detection and segmentation stages improve transcription quality, they introduce additional computational overhead that one might want to reduce to gain a substantial runtime improvement compared to audio-only systems such as ByteCover.

Bibliography

[1] Abrassart, M., Doras, G.: And what if two musical versions don't share melody, harmony, rhythm, or lyrics? In: Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR). pp. 677–684. Bengaluru, India (2022), https://archives.ismir.net/ismir2022/paper/000081.pdf

[2] Araz, R.O., Serrà, J., Serra, X., Mitsufuji, Y., Bogdanov, D.: Discogs-vinet-mirex. In: Proceedings of the 25th International Society for Music Information Retrieval Conference (ISMIR) (2024), technical report submitted as MIREX 2024 entry

[3] Araz, R.O., Serra, X., Bogdanov, D.: Discogs-vi: A musical version identification dataset based on public editorial metadata. In: Proceedings of the 25th International Society for Music Information Retrieval Conference (ISMIR). pp. 541–548. San Francisco, USA (2024)

[4] Balluff, M., Auch, M., Mandl, P., Wolff, C.: Lyriccovers 2.0: An enhanced dataset for cover song analysis. IADIS International Journal on WWW/Internet **22**(2), 75–92 (2024)

[5] Balluff, M., Mandl, P., Wolff, C.: Innovations in cover song detection: A lyrics-based approach (2024), https://arxiv.org/abs/2406.04384

[6] Barański, M., Jasiński, J., Bartolewska, J., Kacprzak, S., Witkowski, M., Kowalczyk, K.: Investigation of whisper asr hallucinations induced by non-speech audio. In: ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). p. 1–5. IEEE (Apr 2025). https://doi.org/10.1109/icassp49660.2025.10890105, http://dx.doi.org/10.1109/ICASSP49660.2025.10890105

[7] Bertin-Mahieux, T., Ellis, D.P.W., Whitman, B., Lamere, P.: The million song dataset. In: Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR). pp. 591–596. Miami, Florida, USA (2011), https://ismir2011.ismir.net/papers/OS6-1.pdf

[8] Bredin, H.: pyannote.audio 2.1 speaker diarization pipeline: principle, benchmark, and recipe. In: Proc. INTERSPEECH 2023 (2023)

[9] Cífka, O., Schreiber, H., Miner, L., Stöter, F.R.: Lyrics transcription for humans: A readability-aware benchmark. In: Proceedings of the 25th International Society for Music Information Retrieval Conference (ISMIR). pp. 145–152. San Francisco, USA (2024)

[10] Correya, A., Hennequin, R., Arcos, M.: Large-scale cover song detection in digital music libraries using metadata, lyrics and audio features. In: Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR). pp. 237–243. Paris, France (2018)

[11] Demetriou, A.M., Jansson, A., Kumar, A., Bittner, R.M.: Vocals in music matter: the relevance of vocals in the minds of listeners. In: Proceedings of the 19th International Society for Music Information Retrieval Con-

ference. pp. 514–520. Paris, France (2018), http://ismir2018.ircam.fr/doc/pdfs/98_Paper.pdf

[12] Dinkel, H., Yan, Z., Wang, T., Wang, Y., Sun, X., Niu, Y., Liu, J., Li, G., Zhang, J., Luan, J.: Glap: General contrastive audio-text pretraining across domains and languages (2025)

[13] Doras, G., Peeters, G.: Cover detection using dominant melody embeddings. In: Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR). pp. 107–114. Delft, The Netherlands (2019), https://archives.ismir.net/ismir2019/paper/000010.pdf

[14] Doras, G., Yesiler, F., Serrà, J., Gómez, E., Peeters, G.: Combining musical features for cover detection. In: Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR). Montreal, Canada (2020), https://archives.ismir.net/ismir2020/paper/000239.pdf

[15] Du, X.: X-cover: Better music version identification system by integrating pretrained asr model. In: Proceedings of the 25th International Society for Music Information Retrieval Conference (ISMIR). San Francisco, USA (2024). https://doi.org/10.5281/zenodo.14877280

[16] Du, X., Chen, K., Wang, Z., Zhu, B., Ma, Z.: Bytecover2: Towards dimensionality reduction of latent embedding for efficient cover song identification. In: ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 616–620 (2022). https://doi.org/10.1109/ICASSP43922.2022.9747630

[17] Du, X., Wang, Z., Liang, X., Liang, H., Zhu, B., Ma, Z.: Bytecover3: Accurate cover song identification on short queries. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5. Toronto, Canada (2023). https://doi.org/10.1109/ICASSP49357.2023.10095389

[18] Du, X., Yu, Z., Zhu, B., Chen, X., Ma, Z.: Bytecover: Cover song identification via multi-loss training. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Toronto, Canada (2021). https://doi.org/10.1109/ICASSP39728.2021.9414128

[19] Durand, S., Stoller, D., Ewert, S.: Contrastive learning-based audio to lyrics alignment for multiple languages. In: ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). p. 1–5. IEEE (Jun 2023). https://doi.org/10.1109/icassp49357.2023.10096725

[20] Elizalde, B., Deshmukh, S., Ismail, M.A., Wang, H.: Clap learning audio concepts from natural language supervision. In: ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5 (2023). https://doi.org/10.1109/ICASSP49357.2023.10095889

[21] Ellis, D.P.W.: The "covers80" cover song data set (2007), http://labrosa.ee.columbia.edu/projects/coversongs/covers80, accessed: 2025-03-27

[22] Fan, C., Liu, B., Tao, J., Yi, J., Wen, Z., Song, L.: Deep time delay neural network for speech enhancement with full data learning. In: 2021 12th In-

ternational Symposium on Chinese Spoken Language Processing (ISCSLP). pp. 1–5 (2021). https://doi.org/10.1109/ISCSLP49672.2021.9362059

[23] Frieske, R., Shi, B.E.: Hallucinations in neural automatic speech recognition: Identifying errors and hallucinatory models (2024), https://arxiv.org/abs/2401.01572

[24] Glazer, N., Segal-Feldman, Y., Segev, H., Shamsian, A., Buchnick, A., Hetz, G., Fetaya, E., Keshet, J., Navon, A.: Beyond transcription: Mechanistic interpretability in asr (08 2025). https://doi.org/10.48550/arXiv.2508.15882

[25] Guzhov, A., Raue, F., Hees, J., Dengel, A.: Audioclip: Extending clip to image, text and audio. In: ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 976–980 (2022). https://doi.org/10.1109/ICASSP43922.2022.9747631

[26] Hu, S., Zhang, B., Lu, J., Jiang, Y., Wang, W., Kong, L., Zhao, W., Jiang, T.: Wideresnet with joint representation learning and data augmentation for cover song identification. In: Interspeech 2022. pp. 4187–4191 (2022). https://doi.org/10.21437/Interspeech.2022-10600

[27] Huang, Q., Jansen, A., Lee, J., Ganti, R., Li, J.Y., Ellis, D.P.W.: Mulan: A joint embedding of music audio and natural language. In: Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR) (2022), https://archives.ismir.net/ismir2022/paper/000067.pdf

[28] Koenecke, A., Choi, A.S.G., Mei, K.X., Schellmann, H., Sloane, M.: Careless whisper: Speech-to-text hallucination harms. In: The 2024 ACM Conference on Fairness, Accountability, and Transparency. p. 1672–1681. FAccT '24, ACM (Jun 2024). https://doi.org/10.1145/3630106.3658996, http://dx.doi.org/10.1145/3630106.3658996

[29] Li, L., Zhao, Y., Jiang, D., Zhang, Y., Wang, F., Gonzalez, I., Valentin, E., Sahli, H.: Hybrid deep neural network–hidden markov model (dnn-hmm) based speech emotion recognition. In: 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction. pp. 312–317 (2013). https://doi.org/10.1109/ACII.2013.58

[30] Liu, F., Tuo, D., Xu, Y., Han, X.: Coverhunter: Cover song identification with refined attention and alignments. In: 2023 IEEE International Conference on Multimedia and Expo (ICME). pp. 1080–1085 (2023). https://doi.org/10.1109/ICME55011.2023.00189

[31] Mancini, E., Serrà, J., Torroni, P., Mitsufuji, Y.: Leveraging whisper embeddings for audio-based lyrics matching (2025), https://arxiv.org/abs/2510.08176

[32] Meseguer-Brocal, G., Cohen-Hadria, A., Peeters, G.: Dali: A large dataset of synchronized audio, lyrics and notes, automatically created using teacher-student machine learning paradigm. (2018). https://doi.org/10.5281/ZENODO.1492443

[33] Müller, M.: Dynamic Time Warping, pp. 69–84. Springer Berlin Heidelberg, Berlin, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74048-3_4

[34] Pons, J., Serra, X.: musicnn: Pre-trained convolutional neural networks for music audio tagging (2019), https://archives.ismir.net/ismir2019/latebreaking/000038.pdf

[35] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning (2021), https://api.semanticscholar.org/CorpusID:231591445

[36] Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I.: Robust speech recognition via large-scale weak supervision (2022), https://arxiv.org/abs/2212.04356

[37] Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I.: Robust speech recognition via large-scale weak supervision. In: Proceedings of the 40th International Conference on Machine Learning. ICML'23, JMLR.org (2023)

[38] Rathnaweera, A.H.: A language-independent method for lyrics-based cover song identification using phoneme transcriptions (May 2024), supervisor: Dr. M.I.E. Wickramasinghe, Co-Supervisor: Mr. W.R.N.S. Abeyweera

[39] Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Conference on Empirical Methods in Natural Language Processing (2019), https://api.semanticscholar.org/CorpusID:201646309

[40] Rodier, S., Carter, D.: Online near-duplicate detection of news articles. In: Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S. (eds.) Proceedings of the Twelfth Language Resources and Evaluation Conference. pp. 1242–1249. European Language Resources Association, Marseille, France (May 2020), https://aclanthology.org/2020.lrec-1.156/

[41] Serrà, J., Araz, R.O., Bogdanov, D., Mitsufuji, Y.: Supervised contrastive learning from weakly-labeled audio segments for musical version matching. In: ICML 2025 Workshop (2025), poster, in press

[42] Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding. Neurocomput. **568**(C) (Feb 2024). https://doi.org/10.1016/j.neucom.2023.127063

[43] Syed, J., Meresman Higgs, I., Cífka, O., Sandler, M.: Exploiting music source separation for automatic lyrics transcription with whisper. pp. 1–6 (06 2025). https://doi.org/10.1109/ICMEW68306.2025.11152264

[44] Touvron, H., Cord, M., El-Nouby, A., Bojanowski, P., Joulin, A., Synnaeve, G., Jégou, H.: Augmenting convolutional networks with attention-based aggregation (12 2021). https://doi.org/10.48550/arXiv.2112.13692

[45] Tumre, S., Patil, S., Kumar, A.: Improved near-duplicate detection for aggregated and paywalled news-feeds. In: Chen, W., Yang, Y., Kachuee, M., Fu, X.Y. (eds.) Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track). pp. 979–987. Associa-

tion for Computational Linguistics, Albuquerque, New Mexico (Apr 2025). https://doi.org/10.18653/v1/2025.naacl-industry.73

[46] Vaglio, A., Hennequin, R., Moussallam, M., Richard, G.: The words remain the same: Cover detection with lyrics transcription. In: 22nd International Society for Music Information Retrieval Conference ISMIR 2021 (2021), https://archives.ismir.net/ismir2021/paper/000089.pdf

[47] Wang, Alhmoud, Alsahly, Alqurishi, Ravanelli: Calm-whisper: Reduce whisper hallucination on non-speech by calming crazy heads down. In: Proceedings of Interspeech 2025 Conference. pp. 3414–3418. Rotterdam, The Netherlands (2025)

[48] Wu, Y., Chen, K., Zhang, T., Hui, Y., Berg-Kirkpatrick, T., Dubnov, S.: Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In: ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5 (2023). https://doi.org/10.1109/ICASSP49357.2023.10095969

[49] Yesiler, F., Doras, G., Bittner, R.M., Tralie, C.J., Serra, J.: Audio-based musical version identification: Elements and challenges. IEEE Signal Processing Magazine **38**(6), 115–136 (Nov 2021). https://doi.org/10.1109/msp.2021.3105941

[50] Yesiler, F., Serrà, J., Gómez, E.: Accurate and scalable version identification using musically-motivated embeddings. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 21–25. Barcelona, Spain (2020). https://doi.org/10.1109/ICASSP40776.2020.9053793

[51] Yesiler, F., Serrà, J., Gómez, E.: Less is more: Faster and better music version identification with embedding distillation. In: Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR). Montreal, Canada (2020), https://archives.ismir.net/ismir2020/paper/000244.pdf

[52] Yu, Y., Tang, S., Raposo, F.A., Chen, L.: Deep cross-modal correlation learning for audio and lyrics in music retrieval. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) **15**, 1 – 16 (2017), https://api.semanticscholar.org/CorpusID:28221141

[53] Yu, Z., Xu, X., Chen, X., Yang, D.: Temporal pyramid pooling convolutional neural network for cover song identification. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence. p. 4846–4852. IJCAI'19, AAAI Press (2019)

[54] Yu, Z., Xu, X., Chen, X., Yang, D.: Learning a representation for cover song identification using convolutional neural network. In: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 541–545 (2020). https://doi.org/10.1109/ICASSP40776.2020.9053839

[55] Yuan, Y., Chen, Z., Liu, X., Liu, H., Xu, X., Jia, D., Chen, Y., Plumbley, M.D., Wang, W.: T-clap: Temporal-enhanced contrastive language-audio pretraining. In: 2024 IEEE 34th International Workshop on Ma-

chine Learning for Signal Processing (MLSP). pp. 1–6 (2024). https://doi.org/10.1109/MLSP58920.2024.10734763

[56] Zhang, X., Zhang, Y., Long, D., Xie, W., Dai, Z., Tang, J., Lin, H., Yang, B., Xie, P., Huang, F., et al.: mgte: Generalized long-context text representation and reranking models for multilingual text retrieval. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track. pp. 1393–1412 (2024)

[57] Zhao, Z., Guo, Y., Wang, D., Huang, Y., He, X., Gu, B.: Graph-based aspect representation learning for entity resolution. In: Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs). pp. 15–23. Barcelona, Spain (Online) (2020). https://doi.org/10.18653/v1/2020.textgraphs-1.2