Zeng and Lau
EURASIP Journal on Audio, Speech, and Music Processing (2023) 2023:31
https://doi.org/10.1186/s13636-023-00297-4

EURASIP Journal on Audio, Speech, and Music Processing

METHODOLOGY

Open Access

Training audio transformers for cover song identification

Te Zeng[1*] and Francis C. M. Lau[1]

Abstract

In the past decades, convolutional neural networks (CNNs) have been commonly adopted in audio perception tasks, which aim to learn latent representations. However, for audio analysis, CNNs may exhibit limitations in effectively modeling temporal contextual information. Analogous to the successes of transformer architecture used in the fields of computer vision and audio classification, to capture long-range global contexts better, we here extend this line of work and propose an *Audio Similarity Transformer* (ASimT), a convolution-free, purely transformer network-based architecture for learning effective representations of audio signals. Furthermore, we introduce a novel loss *MAPLoss*, used in tandem with classification loss, to directly enhance the mean average precision. In the experiments, ASimT demonstrates its state-of-the-art performance in cover song identification on public datasets.

Keywords Cover song identification, Transformer, Music representation learning

1 Introduction

Cover song identification (CSI), referred to as the task which identifies alternative versions of a given song from a music collection, is an important task in the field of music information retrieval (MIR). Various downstream applications can benefit from CSI, such as music rights management, music retrieval, and song recommendation. Despite numerous research efforts behind, CSI remains a challenging task due to its complexity in the presence of variations in timbre, rhythm, key signature, song structure, and lyrics across different song versions [1, 2]. While humans can easily discern these variations among versions, it is difficult for machines to perform similarity matching between distinct renditions, which makes CSI still an intriguing and demanding task within the MIR domain.

In addressing the issue of version identification, a diverse range of approaches have been proposed, which can be broadly classified into two main categories. The first category follows a more traditional methodology, while the second category employs data-driven methods. Specifically, the first category implements a three-stage process: feature extraction, optional post-processing, and similarity estimation. The initial stage focuses on the extraction of relevant features from high-dimensional audio signals. Considering the existence of keys, tempo, and structural variations among different song versions, some studies incorporate a second step, which adopts various post-processing techniques to achieve transposition, tempo, timing, and structure invariance in the version identification problem. In the final stage, an array of segmentation schemes and local alignment algorithms are leveraged to measure the similarity between sequences processed during the preceding stages.

For instance, Bello introduced a CSI system [3] that characterized audio signal in harmonic content using the Chroma [4] feature, which represents the intensity of twelve pitch classes. Subsequently, their system employed Needleman-Wunsch-Sellers (NWS) algorithm [5] to estimate the similarity between approximated chord sequences in order to identify possible cover songs. In [1], the authors used an enhanced chroma feature called

*Correspondence:
Te Zeng
tzeng@cs.hku.hk
[1] Department of Computer Science, The University of Hong Kong, Hong Kong, Hong Kong

harmonic pitch class profiles (HPCP) [6] to describe music audio. Besides, they introduced a second stage in order to attain transposition invariance by transposing the tonality of the target HPCP feature sequence to that of the other songs. In their proposed system, dynamic time warping (DTW) [7] was utilized to measure the similarity between extracted feature sequences.

Following the studies that focused on individual features and alignment methods, Foucard et al. [8] were the first to demonstrate that the combination of melody and its accompaniment as distinct modalities, employing various fusion schemes, could enhance performance in cover song identification. Their work laid the groundwork for a line of approaches that investigated the combination of different features and/or alignment methods to further improve accuracy. Tralie [9] explored the combination of complementary features—namely HPCP, Mel-frequency cepstral coefficients (MFCC), and self-similarity MFCC—using a similarity network fusion (SNF) technique prior to implementing alignment for version matching. In a similar vein, Chen et al. [10] utilized SNF to fuse two types of similarities, Qmax and Dmax. The fused similarity was subsequently fed into a classifier to identify cover pairs.

With the advent of deep learning techniques, over the last few decades, cover song identification tasks have gradually transitioned from methods that heavily relied on sequence alignment to end-to-end models that learn representations towards improved efficiency in accomplishing the task. Convolutional neural networks (CNNs) have become particularly popular in the second category. For instance, in [11–15], CNNs play a critical role to detect cover songs. While CNNs are widely used to learn audio representations by exploring spatial locality, we believe that incorporating long-range global context could help improve the performance of the CSI task, as the original song may be restructured (e.g., a main verse might be placed after the chorus in the cover version). However, few attempts have been made to capture long-range dependencies among audio frames in CSI tasks [16]. Ye et al. implemented an LSTM-based Siamese network in the CSI problem [17], which revealed the potential of investigating long-term contexts in CSI tasks. Despite this, the exploration of long-term dependencies in this field remains relatively uncharted.

The transformer architecture proposed in [18] has successfully demonstrated its ability to model sequential data with long-range dependencies in numerous NLP tasks (e.g., text generation and classification) and, more recently, in the computer vision field (e.g., image retrieval and classification). Additionally, an exciting extension of transformer-based models [19, 20] in audio classification suggests that transformer-based approaches may find alternative solutions and avoid typical errors caused by convolution backbones.

Inspired by the success of adopting the transformer architecture for modeling long-term dependencies in audio classification tasks, we propose a transformer-based method to explore whether long-range global contexts can also enhance cover song detection. While there have been some efforts to explore audio comprehension with a transformer architecture [19–22], to our knowledge, the utilization of a plain transformer directly in cover song identification has not been studied before. To address this void, we propose the *Audio Similarity Transformer* (ASimT), which employs a Siamese architecture with a transformer backbone mapping each audio signal to a single embedding vector. Current deep learning-based approaches predominantly employ classification loss, triplet loss, or their variants or combinations during the training stage. However, these losses do not guarantee the optimization of mean average precision (MAP) [23], a critical evaluation metric in version identification tasks. Therefore, in this paper, we explore a rank loss named *MAPLoss* that directly optimizes MAP for an enhanced version identification performance. Given that version identification can also be regarded as a retrieval problem (i.e., retrieving all versions of a query song), our *MAPLoss* is adapted from SmoothAP Loss, which has achieved successes in image retrieval task [23–25]. To boost the learning efficiency and supply additional supervised information, we combine *MAPLoss* and cross-entropy loss for training our Siamese architecture. Experimental results demonstrate a competitive performance of our proposed method.

2 Related work

2.1 Audio feature

Audio feature extraction is necessary for both traditional and deep learning-based approaches, as it is a crucial element in the former and is used for further learning in the latter. The constant-Q transform (CQT) [26], a low-level descriptor, has been used in numerous CSI studies [13–15] since it was first introduced. Notably, it is found that cover versions tend to maintain similar melodic and harmonic contents while they may exhibit variations in style, instrumentation, and the arrangement [15]. Consequently, researchers have been motivated to adopt music descriptors representing melodic and harmonic information to tackle the CSI problem. Dominant melody has been studied in [12, 27] to describe melodic content for the CSI problem. Chroma [4], which captures the

intensity of twelve pitch classes, has been widely used as an essential audio feature in classical approaches [28–30]. The pitch class profile (PCP) [31] has emerged as a predominant representation for analyzing harmonic content in audio signals. Subsequently, HPCP was developed to enhance the robustness of tonal content summarization and has been extensively applied to the CSI problem [32, 33]. In particular, feature combinations with HPCP have been investigated in the CSI problem [9, 34]. Salamon et al. utilized HPCP to summarize harmonic content, subsequently integrating melody and bass content to enhance the performance of their CSI system. HPCP was employed in [9] alongside MFCC and self-MFCC [35] in a fusive manner to improve the CSI problem performance.

As a novel PCP variant, convolutional and recurrent estimators for music analysis (CREMA) [36] estimates pitch-class information required for chord sequence prediction and has contributed to superior performance in cover song analysis, as reported in [37]. This finding is plausible, given that cover versions often exhibit similar chord progression. The advancements achieved by CREMA have spurred a series of subsequent studies [38, 39], which further corroborate the validity of the CREMA feature in CSI problems. Consequently, we employ CREMA as the feature of our proposed method, each of which can be represented as $\mathbf{x} \in \mathbb{R}^{12\times W}$, where W denotes the number of frames. To bolster the performance of our system, we apply data augmentation to the original CREMA feature, yielding a processed CREMA with dimension of $23 \times W$, as detailed in Section 3.4.

2.2 Metric learning

Deep learning-based CSI systems can be further classified into two categories. The first category approaches the version identification problem as a multi-class classification task, with each version group being treated as a unique class [15, 40, 41]. However, due to the substantial number of version groups (i.e., the classes) and the limited number of versions within each group (i.e., the samples), version identification does not entirely fit within the framework of a classification problem. This observation gives rise to the second category, which leverages metric learning techniques to enhance intra-class similarity and inter-class discrimination [13, 14, 38, 39]. Loss functions employed in these metric learning-based methods typically consider triplets [13, 14, 38] to achieve the desirable results in the context of CSI—for instance, contrastive loss [42] and triplet loss [43]. The training procedure for these methods involves repeatedly sampling of random and different triplets of song versions and backpropagating the loss gradients. Nonetheless, as Burges et al. [44] highlight, the limited rank-positional awareness provided by the triplet loss may lead to inefficient use of a model's capacity, causing the model to focus on improving the rank order of positive instances at lower ranks, which is often to the detriment of those at high ranks. Consequently, there is no theoretical assurance that the process of minimizing the triplet loss would necessarily coincide with minimizing the actual ranking loss.

In this paper, we embark on a distinct path by directly optimizing the mean average precision (MAP) metric. While the average precision (AP) is a non-decomposable and non-differentiable function, recent advancements by He et al. have demonstrated that it can be approximated [45]. This method has yielded successful outcomes in the realm of image matching and retrieval tasks [23, 24]. Given that the task of version identification can be interpreted as a version retrieval problem, and considering that no previous attempts have been made to leverage a loss function to directly improve the MAP value, we introduce an adaption of the smoothAP loss in our model (we term it as *MAPLoss*), the efficacy of which in resolving image retrieval problems has been demonstrated before.

3 Methodology

3.1 Transformer architecture

This section describes the transformer architecture in a fashion similar to [18]. We have adopted this architecture in our work.

Since ASimT is designed for similarity metric learning, we utilized only the encoder component of the transformer architecture. The transformer backbone, acting as the encoder, takes as an input a sequence of pre-processed CREMA features (the detailed processing procedure is explained in Section 3.4) and produces the corresponding learned latent representation. Given that the standard transformer processes 1D sequences of token embeddings, it is necessary to reshape the processed CREMA features into a sequence of flattened 2D patches. Following the method employed in vision transformer (ViT) [46], we reshape the processed CREMA sequence $\mathbf{x} \in \mathbb{R}^{H\times W}$ into flattened 2D patches $\mathbf{x}_p \in \mathbb{R}^{N\times P^2}$, where (H, W) represents the resolution of our processed CREMA feature. In contrast to ViT, audio feature is a single-channel spectrogram whereas an image feature comprises 3 channels. (P, P) denotes the resolution of each processed CREMA feature patch with an overlap of L in both the time and frequency dimensions. Consequently, the number of patches, which is the input sequence length for the standard transformer encoder, would be $N = 2L\lfloor (W - L)/(P - L)\rfloor$. In our case, $H = 23$ is the frequency dimension and W is the time dimension. Because we use the SHS_{5+} dataset [12] (details of which will be given later), where the CREMA representation

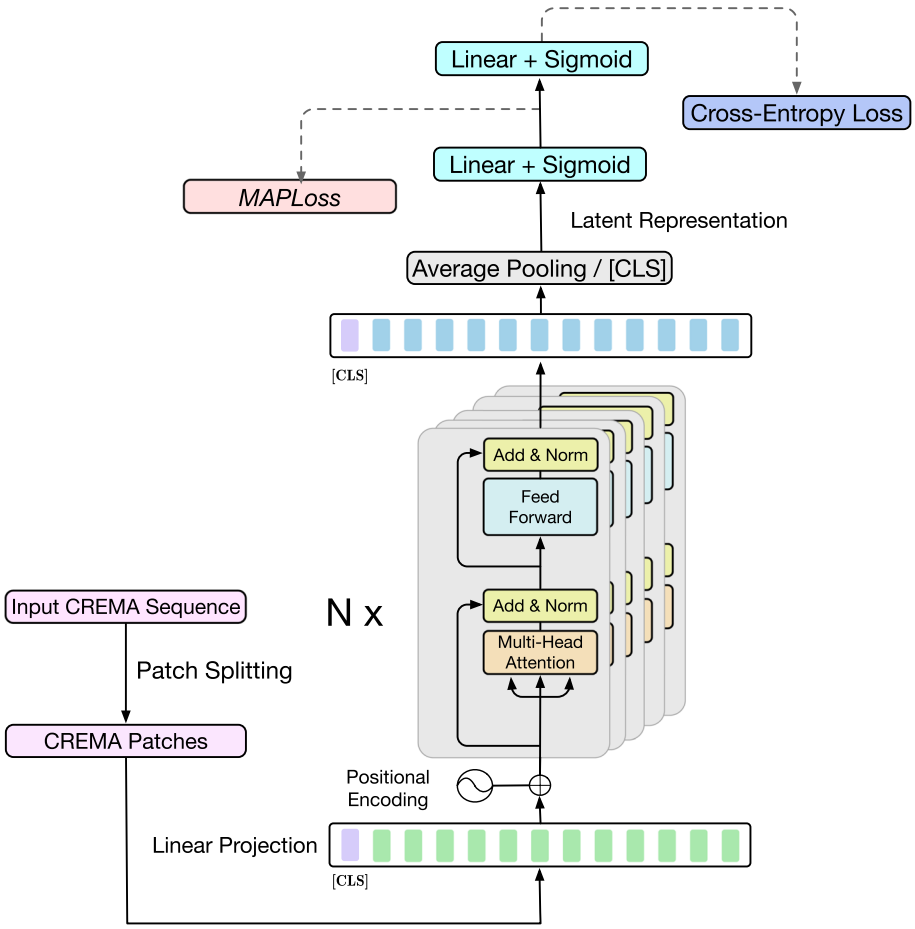

Fig. 1 The overall framework of our proposed ASimT for CSI

spans the first 3 min of the audio of each track, the time dimension has the value of 1937. Following the settings in ViT, we set the patch resolution as $(P, P) = (16, 16)$. Similar to Audio Spectrogram Transformer (AST), we have an overlap of $L = 6$.

Analogous to the bidirectional encoder representations from transformers (BERT), we introduce a learnable `[CLS]` token at the beginning of the input sequence. More specifically, the input sequence for our framework can be expressed as $\mathbf{x}_r = (\mathbf{x}_{\text{class}}, \mathbf{x}_p^1, \mathbf{x}_p^2, \ldots, \mathbf{x}_p^N)$. This sequence is then mapped into a D-dimensional embedding via a trainable linear projection. To preserve positional information, we utilize position embeddings employed in the standard transformer. These embeddings are added to patch embeddings, obtaining the input sequence for the transformer backbone, as illustrated in Eq. 1.

As depicted in Fig. 1, the transformer module consists of a multi-headed self-attention block and a feed-forward block. Layer normalization is applied prior to each block, while residual connections are implemented following each block [47]. The multi-headed self-attention block calculates a probabilistic score that indicates the importance of each embedding. Each multi-headed attention layer projects the input sequence to query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$, through three learnable matrices $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathrm{R} \in \mathrm{D} \times \mathrm{D_k}$, where D_k represents the dimension of each attention head. We employ the scaled-dot production attention as the type of attention mechanism. More specifically, for the layer representation at the l-th transformer layer, $\mathbf{z}_l = [\mathbf{h}_l^1, \mathbf{h}_l^2, \ldots, \mathbf{h}_l^n]$ is utilized to compute the l-th layer self-attention head $\mathbf{A}_l$:

$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}}; \mathbf{x}_p^1\mathbf{E}; \mathbf{x}_p^2\mathbf{E}; \ldots; \mathbf{x}_p^N\mathbf{E}] + E_{pos}, \quad \mathbf{E} \in \mathbb{R}^{P^2 \times D}, \mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D} \tag{1}$$

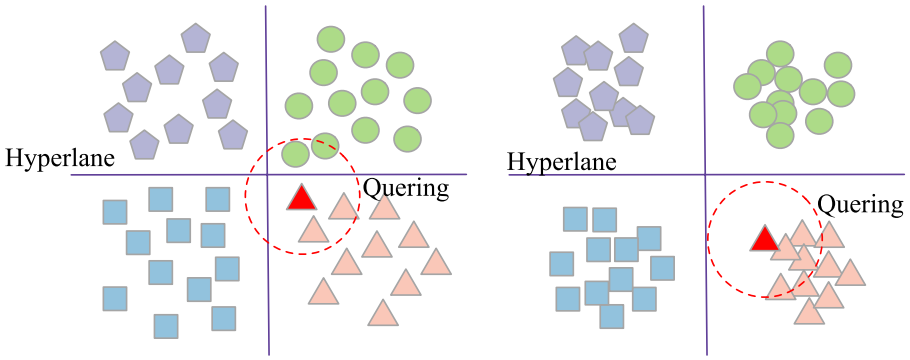

Fig. 2 Motivation of using MAPLoss. From **a**, we can see that while purely classification loss can facilitate successful identification of each class, the minimal inter-class distance may result in erroneous version retrievals. The integration of *MAPLoss*, as depicted in **b**, aims to enhance intra-class compactness and inter-class separation. As a result, this approach can achieve better retrieval performance, i.e., MAP in this work

$$\begin{aligned}\mathbf{Q}_l &= \mathbf{z}_{l-1}\mathbf{W}_Q^l,\\ \mathbf{K}_l &= \mathbf{z}_{l-1}\mathbf{W}_K^l,\\ \mathbf{V}_l &= \mathbf{z}_{l-1}\mathbf{W}_V^l,\end{aligned} \tag{2}$$

$$\mathbf{A}_l = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{D_k}}\mathbf{V}\right), \tag{3}$$

The feed-forward block comprises two linear layers. The first linear layer is followed by a GELU activation and a dropout layer, while after the second layer, only a dropout layer is applied. In an effort to optimize the classification task and enhance the learning of efficient representations for *MAPLoss*, we incorporate two linear layers after the transformer backbone. Each of these layers is followed by a Sigmoid activation function.

3.2 ImageNet pretraining

Although the transformer is capable of modeling long-range contexts compared to CNN models, it requires more data during the training stage, which can be quite resource-intensive. Therefore, akin to the transformer AST [19], we also adopt an off-the-shelf ImageNet-pretrained ViT in our proposed ASimT with a few modifications. First, the input sequence of ViT has three channels, whereas the input sequence of our ASimT is a single-channel spectrogram. Thus, we average the weights along the three channels of the ViT patch embedding layer and make use of them as the weights of the ASimT patch embedding layer. Second, we adopt the cut and bi-linear method proposed in [19] for positional embedding adaptation. For instance, consider a ViT that takes an image input of size 384×384 and employs a patch size of 16×16. This results in $24 \times 24 = 576$ patches and corresponding positional embeddings, given that there is no overlap between patches. Conversely, our ASimT processes the CREMA feature of size 23×1937, generating 1×193 patches, each requiring a positional embedding. To adapt the ViT's positional embedding for our ASimT method, we truncate the first dimension and interpolate the second dimension of the 24×24 ViT positional embedding, resizing it to 1×193, which serves as the positional embedding of our ASimT. Third, slightly different from ViT and AST, we are not addressing a classification problem. ASimT is designed to learn efficient latent representations for input audio. Hence, we conduct experiments using average-pooling and global feature vector `[CLS]` independently after the ASimT patch embeddings to investigate the viability of the global feature vector `[CLS]` for our specific problem of CSI.

3.3 Loss functions

As we discussed in Section 2.2, the classification accuracy alone cannot guarantee good mean average precision for the version identification problem. Upon a deeper exploration of various metric learning techniques in Section 2.2, we decide to utilize both the classification loss and a novel *MAPLoss* to optimize the version identification/retrieval task. The *MAPLoss* could directly improve the MAP value during the training stage, resulting a more effective latent representation produced by our Siamese network (Fig. 2).

3.3.1 A. Cross-entropy loss

To provide more supervised information for the training signals, we also consider a cross-entropy loss during the training phase. The cross-entropy loss is computed as follows:

$$\mathcal{L}_{CE} = -\frac{1}{N}\sum_{i}^{N}\hat{p}(y,\mathbf{x})\log P(y \mid \mathbf{z}), \tag{4}$$

where $\hat{p}(y,\mathbf{x})$ is the ground-truth one-hot distribution of sample $\mathbf{x}$ and $P(y \mid \mathbf{z})$ is the predicted distribution by our ASimT encoder and linear classifier.

3.3.2 B. Mean average precision loss

Mean average precision is a prevalent metric in the field of cover version identification, which is used in the Mirex Audio Cover Song Identification contest.[1] Given an input query song, the task is to rank all instances in the retrieval set, denoted as $\Omega = \{I_i, i = 0, \ldots, k\}$. For each query song I_q, the retrieval set can be split into positive and negative sets premised upon the relevance score. Suppose the set with positive relevance scores is represented by R_P and the set with negative relevance scores is by R_N. Therefore, the complete relevance score set is manifested as $R_\Omega = R_P \cup R_N$. Subsequently, for a query song I_q, the approximated AP can be expressed as follows:

$$AP_q = \frac{1}{\mid R_P \mid}\sum_{i\in R_P}\frac{1+\sum_{j\in R_P}\sigma(d_{ij};\tau)}{1+\sum_{j\in R_\Omega}\sigma(d_{ij};\tau)}, \tag{5}$$

where $\sigma(\cdot;\tau)$ is a sigmoid function. $\sigma(x;\tau) = \frac{1}{1+e^{\frac{-x}{\tau}}}$, in which τ is the temperature to parameterize the margin. $d_{ij} = [s(q,i) - s(q,j)]$ is the difference matrix and $s(\cdot,\cdot)$ denotes the cosine similarity. It can be computed as $s(q,i) = \frac{\mathrm{v_q}^T \mathrm{v_j}}{\|\mathrm{v_q}\|^2\|\mathrm{v_j}\|2}$, where $\mathrm{v_q}$ is the vectorial latent representation obtained from our Siamese model. Then, the MAP of a batch input can be computed as:

$$MAP = \frac{1}{k}\sum_{t=1}^{k}AP_t, \tag{6}$$

where m is the number of instances in the batch, AP_t is the average precision of the t-th query. Subsequently, we can formulate the *MAPLoss* as follows:

$$\mathcal{L}_{MAP} = 1 - \frac{1}{k}\sum_{t=1}^{k}AP_t, \tag{7}$$

Hence, our final loss function can be written as:

$$\mathcal{L} = \mathcal{L}_{MAP} + \mathcal{L}_{CE} \tag{8}$$

3.3.3 C. Contrastive loss

To further underscore the efficacy of *MAPLoss*, we conduct additional experiments with a commonly used metric learning technique, contrastive loss. The contrastive loss maximizes the similarity between encoded low-dimensional representations with the same labels which are referred to as positives and minimizes the similarity between learned representations with unmatched labels, by defining a negative. Given a set $\{\mathbf{z}_k\}$ of learned representations and a set of $\{y_k\}$ including positive and negative samples, the contrative loss can be computed as:

$$\mathcal{L}_{CON} = \frac{1}{N}\sum_{i}^{N}\left[\sum_{j:y_i=y_j}\left[1-\mathbf{z}_i^\top\mathbf{z}_j\right] + \sum_{j:y_i\neq y_j}\left[\mathbf{z}_i^\top\mathbf{z}_j - \alpha\right]\right], \tag{9}$$

where β is a constant margin. The constant margin is designed to prevent the model from being overwhelmed. With the constant margin, only negative pairs whose similarity is higher than the tolerance will contribute to the contrastive loss.

3.4 Data augmentation

In order to enhance the learning of ASimT and prevent it from overfitting to the training data, we adopt two data augmentation functions. The first function considers key transposition and tempo variation which are common in cover songs. Following the strategy proposed in [41] and [38], we expand the dimension of the CREMA feature from $12 \times T$ to $23 \times T$. To bolster the robustness of ASimT in dealing with potential key transpositions, we randomly roll the input CREMA feature $\mathbf{x}$ in the pitch dimension between 0 and 11 bins. For tempo variation, we adopt the strategy used in [38], stretching the temporal dimension with a random factor ranging from 0.7 to 1.5. Additionally, time warping is also incorporated into our first augmentation function, involving the duplication, silence, or removal of frames with respective probabilities of 0.3, 0.4, and 0.3. The second function focuses on addressing variable lengths of the input audio signals. For training data exceeding the predefined length of 1800 in this work, we randomly truncate it at any point for further data augmentation. If the resulting sequence falls short of the predefined length, zero-padding is applied. For testing data longer than the predefined length, it will be trimmed from the very beginning.

[1] https://www.musicir.org/mirex/wiki/2021:Audio Cover Song Identification

An overview of the ASimT training process is provided in Algorithm 1.

Input: D batch size M, constant τ, structure of $f, g, \mathcal{T}, \mathcal{V}$.
for sampled mini-batch $B \in D$ **do**
  for all each audio signal CREMA feature $\mathbf{x} \in B$ **do**
    draw two augmentation functions $t \sim \mathcal{T}$, $t' \sim \mathcal{T}$;
    draw patch splitting function $v \sim \mathcal{V}$;
    $\mathbf{x}_{a1} = t(\mathbf{x})$; ▷ the first augmentation
    $\mathbf{x}_{a2} = t'(\mathbf{x}_{a1})$; ▷ the second augmentation
    $\mathbf{x}_p = v(\mathbf{x}_{a2})$; ▷ Reshape $\mathbf{x}_a 2$ into flattened 2D patches $\mathbf{x}_p$
    $\mathbf{z} = f(\mathbf{x}_p)$; ▷ latent representation by ASimT encoder
    $y = g(\mathbf{z})$; ▷ linear projection for classification
  end for
  Compute loss $\mathcal{L}$ by Equation (8);
  Update encoder f and classifier g to minimize loss $\mathcal{L}$
end for

Algorithm 1 ASimT's learning algorithm

4 Experiments

SHS_{5+} and SHS_{4-} are built with the *SecondHandSongs* API by [27] to train and evaluate CSI systems. Specifically, SHS_{5+} is utilized as the training set, whereas SHS_{4-} is employed for testing. The splitting of the two datasets is founded on the number of cover versions of each collected song to counteract data imbalance. For optimal data availability during the training phase, SHS_{5+} exclusively comprises songs with at least five versions, culminating in a total of 62,311 tracks from 7460 unique original works.

However, in practical scenarios, most songs usually have 2 or 3 covers [27]. Consequently, SHS_{4-}, serving as the test set, consists of 19,455 original works, with each work only incorporating songs with up to four versions, totaling 48,483 tracks. This makes SHS_{4-} more representative of real-world conditions compared to SHS_{5+} when performing the cover song identification task with normal query audio. Therefore, we employ SHS_{4-} as the test set to assess the performance of our proposed ASimT.

We use ImageNet in our experiments. More specifically, our ASimT is trained on the pretrained weights of a data-efficient image transformer (DeiT) [48]. The configuration for our transformer encoder is set with an embedding dimension of 768 and consists of 12 transformer layers. Furthermore, the multi-head attention block is configured with 12 heads. The first linear layer transforms the 768-dimensional outputs from the transformer backbone into a 256-dimensional space. Subsequently, the second linear layer maps this 256-dimensional output from the first layer into the number of classes present in the training dataset, which in this study, amounts to 7460 classes. In order to facilitate the application of *MAPLoss*, we set the batch size to be 350, which includes 70 classes, each containing 5 instances (equivalent to the minimal number of versions for each song in the SHS_{5+}). The learning rate is initially set at $2e^{-3}$ with cosine learning rate decay. The model is coupled with a stochastic gradient descent (SGD) [49] optimizer with a weight decay of $1e^{-4}$.

Table 1 The results of experiments

Results		
Methods	**MAP**	**MR1**
Standard Triplet Loss(F0) [12]	0.222	-
Standard Triplet Loss(Multi-F0) [12]	0.280	-
RE-MOVE [39]	0.457	1388
ASimT ($\mathcal{L}_{CE} + \mathcal{L}_{CON}$)	0.301	1234
ASimT ($\mathcal{L}_{CE} + \mathcal{L}_{MAP}$)	**0.466**	**1048**

4.1 Evaluation on large dataset

For the purpose of evaluation, we have chosen to utilize the MAP and the mean rank of the first correctly identified cover (MR1), commonly accepted metrics in the Mirex Audio Cover Song Identification contest. This paper utilizes the SHS_{5+} and SHS_{4-} datasets for the purpose of training and testing, respectively. As a result, we evaluate our proposed method in comparison to existing studies that make use of these two datasets.

Triplet loss is explored in [12] for the CSI problem. Notably, they compiled the SHS_{4-} dataset and conducted an evaluation of their methodology using this collection. Hence, we select this work as the baseline method to compare with our method. Yesiler et al. introduced a data distillation method to address the CSI problem, which included reducing the embedding size [39]. Given that their model also employs the CREMA feature, we include it as a baseline for comparison with our method. As their original research was trained on *Da-TACOS*, we implement their method using our SHS_{5+} dataset for training and evaluate it on the SHS_{4-} dataset. We also conduct experiments training with classification and contrastive loss as part of the baselines to explore the validity of *MAPLoss*.

Table 1 presents the results of our ASimT. Evidently, our proposed method surpasses all the baselines in terms of MAP and MR1. This implies that our approach sets a new benchmark in performance, demonstrating how a standard transformer backbone can be effectively adapted for audio understanding and cover song analysis. Furthermore, our experiments show that the combination of classification loss and our proposed *MAPLoss* outperforms that of classification loss and contrastive loss. This indicates that directly optimizing the MAP value during the training stage can significantly improve the performance of version indentification.

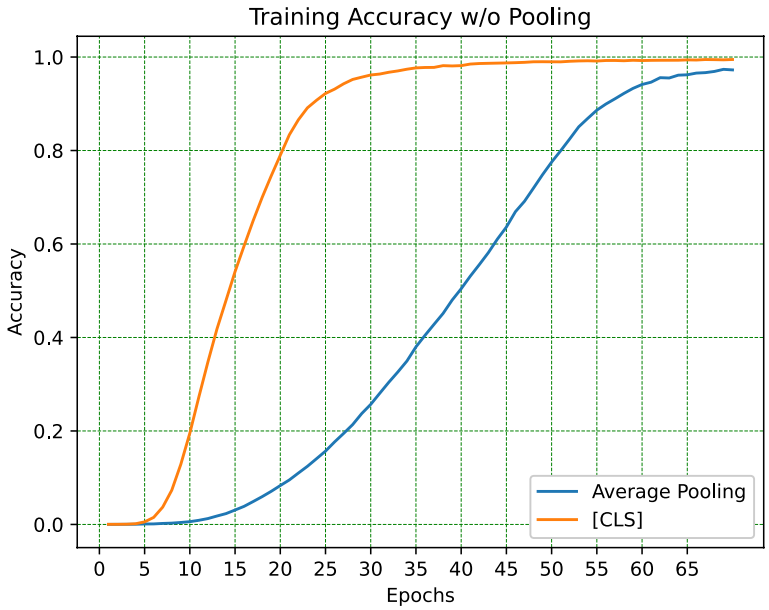

Fig. 3 The training accuracy with or without average pooling

4.2 Evaluation on small dataset

In real-world scenarios, a vast number of songs are available on the Internet, including original songs and their cover versions. This abundance of data can be utilized to train CSI models more effectively, thereby improving the accuracy of version identification for practical use. In many cases, the original or alternate versions of a query song may be present in the training collection. To simulate this situation, we create a small dataset following the approach employed in [27]. We randomly select 350 tracks from the training dataset, comprising works with 7 covers each. Out of these, 100 tracks are included in the training stage. For testing, we compute the similarity between all pairs of the 350 tracks, resulting in a 350×349 similarity matrix. As a result, we achieve a MAP of 74.68 %. This approach is practical in real-world applications, as companies like *Shazam*[2] typically train their models on millions of songs to achieve high accuracy. The results obtained on the small dataset significantly surpass those achieved on SHS_{4-} dataset. Such observations align with findings from previous research. For instance, [50] reported a MAP of 0.09475 on a large dataset containing 12,960 tracks. In a similar vein, [15] noted a decrease in accuracy as the dataset size increased. They speculated that this could be attributed to larger datasets have the tendency to contain songs with similar melodic structures, chord patterns, and accompaniments, thus complicating the task of identifying cover versions. Our large evaluation set, SHS_{4-}, consisting of a total of 48,483 tracks, presents a comparable challenge for version identification tasks.

4.3 The impact of pooling

In the previous work on image classification [46] and retrieval [51], the output of the `[CLS]` token was used as the latent representation for subsequent classification or metric learning tasks. In a similar manner, AST, like [46], transforms the output of the `[CLS]` token into a class prediction linear layer. As AST employs DeiT as the pretrained model, and given that DeiT incorporates two `[CLS]` tokens, AST averages the outputs of these two tokens for the purpose of audio event classification. To further explore the impact of pooling in version identification problems, we conduct experiments comparing the training curves of both average pooling and the sole use of the global feature vector `[CLS]` (Figs. 3 and 4). This experiment, conducted with only the classification loss, guides our decision to use either average pooling or `[CLS]` in our final training. Interestingly, the final performance using average pooling and that of solely the `[CLS]` token proved similar. However, the accuracy increased more rapidly when using the global feature vector `[CLS]`, and similarly, the loss declined more rapidly when using `[CLS]` compared to using average pooling. As a result, we adopted `[CLS]` as the output of our transformer backbone for further training.

5 Conclusion

In this work, we venture to explore how a convolution-free, purely attention-based transformer architecture can be adapted for cover song analysis. We introduce

[2] https://www.shazam.com

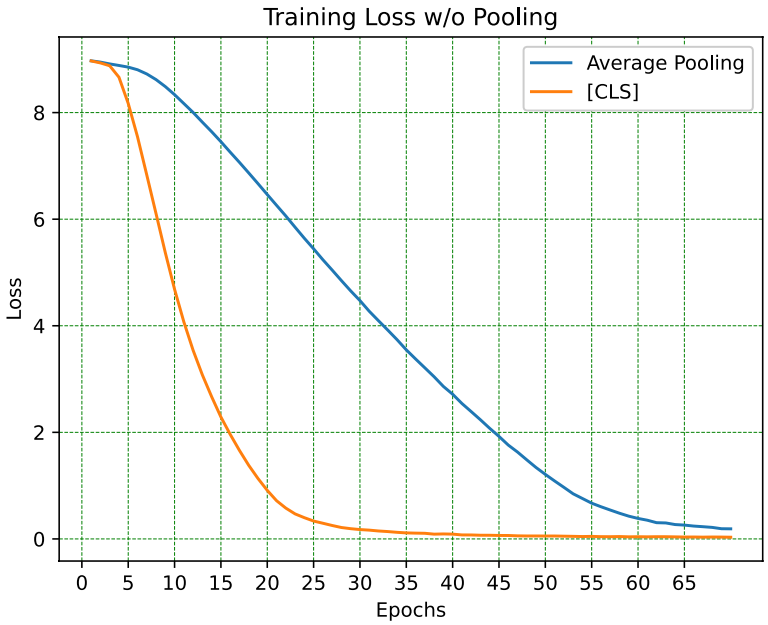

Fig. 4 The training loss with or without average pooling

the novel *MAPLoss*, used in tandem with classification loss, to directly optimize the mean average precision. Our experiments demonstrated that our *MAPLoss* could deliver competitive results and also illustrated the potential utility of the transformer model in cover song identification tasks. Nonetheless, when evaluated on the large dataset, both in this work and in related research, the mean average precision was found to be relatively low. This could be due, in part, to that large datasets having the tendency to contain music works sharing similar chord sequences. Given that our CREMA feature mainly encapsulates the harmonic context in audio signals, in the future, to further enhance the performance, we plan to take the melodic context into account as well. By integrating these two musical dimensions in a fusive approach, we anticipate that we can more effectively identify cover versions, even when dealing with large datasets.

Abbreviations

CNNs	Convolutional neural networks
ASimT	Audio Similarity Transformer
CSI	Cover song identification
MIR	Music information retrieval
NWS	Needleman-Wunsch-Sellers
HPCP	Harmonic pitch class profiles
DTW	Dynamic time warping
MFCC	Mel-frequency cepstral coefficients
MAP	Mean average precision
CQT	Constant-Q transform
CREMA	Convolutional and recurrent estimators for music analysis
AP	Average precision
ViT	Vision transformer
AST	Audio spectrogram transformer
BERT	Bidirectional encoder representations from transformers
DeiT	Data-efficient image transformer
MR1	Mean rank of the first correctly identified cover

Acknowledgements
Not applicable.

Authors' contributions
Both authors contributed to this work, including the problem formalization, the idea development, and the manuscript writing. Te Zeng implemented the methods and conducted the experiments. Francis C.M. Lau polished the presentation.

Funding
Not applicable.

Availability of data and materials
Publicly accessable dataset SHS_{5+} and SHS_{4-} are used in out experiments. It is avaliable on https://gdoras.github.io/topics/coversdataset, accessed on 18 July 2023.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 4 February 2023 Accepted: 7 August 2023
Published online: 25 August 2023

References

1. J. Serra, E. Gómez, P. Herrera, X. Serra, Chroma binary similarity and local alignment applied to cover song identification. IEEE Trans. Audio Speech Lang. Process. **16**(6), 1138–1151 (2008)
2. J.S. Seo, Improving cover song search accuracy by extracting salient chromagram components. J. Korea Multimed. Soc. **22**(6), 639–645 (2019)

3. J.P. Bello, in *Proceedings of ISMIR (International Society for Music Information Retrieval)*, Audio-based cover song retrieval using approximate chord sequences: Testing shifts, gaps, swaps and beats (Vienna, 2007), pp. 239–244
4. T. FUJISHIMA. Realtime chord recognition of musical sound: a system using common lisp music (CCRMA, Stanford University, 1999)
5. S.B. Needleman, C.D. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol. **48**(3), 443–453 (1970)
6. E. Gómez, Tonal description of polyphonic audio for music content processing. INFORMS J. Comput. **18**(3), 294–304 (2006)
7. M. Müller, Dynamic time warping. *Information retrieval for music and motion*, pp. 69–84 (Springer, 2007)
8. R. Foucard, J.L. Durrieu, M. Lagrange, G. Richard, in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, Multimodal similarity between musical streams for cover version detection (IEEE, 2010), pp. 5514–5517
9. C.J. Tralie, Early mfcc and hpcp fusion for robust cover song identification. arXiv preprint arXiv:1707.04680 (2017)
10. N. Chen, W. Li, H. Xiao, Fusing similarity functions for cover song identification. Multimedia Tools Appl. **77**, 2629–2652 (2018)
11. X. Qi, D. Yang, X. Chen, in *International Conference on Multimedia Modeling*, Triplet convolutional network for music version identification (Springer, 2018), pp. 544–555
12. G. Doras, G. Peeters, in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, A prototypical triplet loss for cover detection (IEEE, 2020)
13. X. Du, Z. Yu, B. Zhu, X. Chen, Z. Ma, in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Bytecover: Cover song identification via multi-loss training (IEEE, 2021), pp. 551–555
14. X. Du, K. Chen, Z. Wang, B. Zhu, Z. Ma, in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Bytecover2: Towards dimensionality reduction of latent embedding for efficient cover song identification (IEEE, 2022), pp. 616–620
15. Z. Yu, X. Xu, X. Chen, D. Yang, in *Proceedings of the Int. Joint Conference on Artificial Intelligence (IJCAI)*, Temporal pyramid pooling convolutional neural network for cover song identification (Macao, 2019), pp. 4846–4852
16. F. Yesiler, G. Doras, R.M. Bittner, C.J. Tralie, J. Serrà, Audio-based musical version identification: Elements and challenges. IEEE Signal Proc. Mag. **38**(6), 115–136 (2021)
17. Z. Ye, J. Choi, G. Friedland, in *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, Supervised deep hashing for highly efficient cover song detection (IEEE, 2019), pp. 234–239
18. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*, Attention is all you need (New York, 2017)
19. Y. Gong, Y.A. Chung, J. Glass, Ast: Audio spectrogram transformer. arXiv preprint arXiv:2104.01778 (2021)
20. Y. Gong, C.I. Lai, Y.A. Chung, J. Glass, in *Proceedings of the AAAI Conference on Artificial Intelligence*, Ssast: Self-supervised audio spectrogram transformer, **36**(10), 10699–10709 (virtual, 2022)
21. D. Yao, Z. Zhao, S. Zhang, J. Zhu, Y. Zhu, R. Zhang, X. He, in *Proceedings of the ACM Web Conference 2022 (WWW '22)*, Contrastive learning with positive-negative frame mask for music representation (New York, 2022), pp. 2906–2915
22. K. Koutini, J. Schlüter, H. Eghbal-zadeh, G. Widmer, Efficient training of audio transformers with patchout. arXiv preprint arXiv:2110.05069 (2021)
23. J. Revaud, J. Almazán, R.S. Rezende, C.R.d. Souza, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Learning with average precision: training image retrieval with a listwise loss (Seoul, 2019), pp. 5107–5116
24. A. Brown, W. Xie, V. Kalogeiton, A. Zisserman, in *European Conference on Computer Vision*, Smooth-ap: Smoothing the path towards large-scale image retrieval (Springer, 2020), pp. 677–694
25. T. Li, Z. Zhang, L. Pei, Y. Gan, Hashformer: Vision transformer based deep hashing for image retrieval. IEEE Sig. Process. Lett. **29**, 827–831 (2022)
26. J.C. Brown, Calculation of a constant q spectral transform. J. Acoust. Soc. Am. **89**(1), 425–434 (1991)
27. G. Doras, G. Peeters, in *Proceedings of ISMIR (International Society of Music Information Retrieval)*, Cover detection using dominant melody embeddings (Delft, 2019)
28. D.P. Ellis, MIREX 2006, Identifying "cover songs" with beat-synchronous chroma features (MIREX, 2006), pp. 1– 4
29. J.H. Jensen, M.G. Christensen, D.P. Ellis, S.H. Jensen, in *2008 IEEE international conference on acoustics, speech and signal processing*, A tempo-insensitive distance measure for cover song identification based on chroma features (IEEE, 2008), pp. 2209–2212
30. L. Maršík, M. Rusek, K. Slaninová, J. Martinovič, J. Pokorný, in *IFIP International Conference on Computer Information Systems and Industrial Management*, Evaluation of chord and chroma features and dynamic time warping scores on cover song identification task (Springer, 2017), pp. 205–217
31. F. Takuya, in *Proceedings of the International Computer Music Conference 1999*, Realtime chord recognition of musical sound: Asystem using common lisp music (Beijing, 1999)
32. E. Gómez, P. Herrera, in *Proceedings of ISMIR (International Society of Music Information Retrieval)*, The song remains the same: identifying versions of the same piece using tonal descriptors. (Victoria, 2006), pp. 180–185
33. J. Serra, X. Serra, R.G. Andrzejak, Cross recurrence quantification for cover song identification. New J. Phys. **11**(9), 093017 (2009)
34. J. Salamon, J. Serrà, E. Gómez, in *Proceedings of the 21st International Conference on World Wide Web*, Melody, bass line, and harmony representations for music version identification (Lyon, 2012), pp. 887–894
35. B. Logan, in *In International Symposium on Music Information Retrieval*, Mel frequency cepstral coefficients for music modeling (Citeseer, 2000)
36. B. McFee, J.P. Bello, *in Proceedings of ISMIR (International Society of Music Information Retrieval)*, Structured training for large-vocabulary chord recognition. (Suzhou, 2017), pp. 188–194
37. F. Yesiler, C. Tralie, A.A. Correya, D.F. Silva, P. Tovstogan, E. Gómez Gutiérrez, X. Serra, in *Proceedings of the 20th Conference of the International Society for Music Information Retrieval (ISMIR 2019): 2019 Nov 4-8; Delft, The Netherlands.[Canada]: ISMIR; 2019.*, Da-tacos: A dataset for cover song identification and understanding (International Society for Music Information Retrieval (ISMIR), 2019)
38. F. Yesiler, J. Serrà, E. Gómez, in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Accurate and scalable version identification using musically-motivated embeddings (IEEE, 2020), pp. 21–25
39. F. Yesiler, J. Serrà, E. Gómez, Less is more: faster and better music version identification with embedding distillation. arXiv preprint arXiv:2010.03284 (2020)
40. Z. Yu, X. Xu, X. Chen, D. Yang, in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Learning a representation for cover song identification using convolutional neural network (IEEE, 2020), pp. 541–545
41. X. Xu, X. Chen, D. Yang, in *2018 IEEE International Conference on Multimedia and Expo (ICME)*, Key-invariant convolutional neural network toward efficient cover song identification (IEEE, 2018), pp. 1–6
42. R. Hadsell, S. Chopra, Y. LeCun, in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, Dimensionality reduction by learning an invariant mapping (IEEE, 2006), pp. 1735–1742
43. K.Q. Weinberger, J. Blitzer, L. Saul, in *Proceedings of the 19th International Conference on Neural Information Processing Systems (NIPS'05)*, Distance metric learning for large margin nearest neighbor classification (Vancouver, 2005)
44. C. Burges, R. Ragno, Q. Le, in *Proceedings of the 20th International Conference on Neural Information Processing Systems (NIPS'06)*, Learning to rank with nonsmooth cost functions (Vancouver, 2006)
45. K. He, Y. Lu, S. Sclaroff, in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR'18)*, Local descriptors optimized for average precision (Salt Lake City, 2018), pp. 596–605
46. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
47. Q. Wang, B. Li, T. Xiao, J. Zhu, C. Li, D.F. Wong, L.S. Chao, Learning deep transformer models for machine translation. arXiv preprint arXiv:1906.01787 (2019)
48. H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, H. Jégou, in *International conference on machine learning*, Training data-efficient

image transformers & distillation through attention (PMLR, 2021), pp. 10347–10357
49. H. Robbins, S. Monro, A stochastic approximation method. Ann. Math. Stat. **22**(3), 400-407 (1951)
50. D.P. Ellis, B.M. Thierry, in *Proceedings of ISMIR (International Society of Music Information Retrieval)*, Large-scale cover song recognition using the 2d fourier transform magnitude (Porto, 2012)
51. A. El-Nouby, N. Neverova, I. Laptev, H. Jégou, Training vision transformers for image retrieval. arXiv preprint arXiv:2102.05644 (2021)

Publisher's Note