Article

ATOSE: Audio Tagging with One-Sided Joint Embedding

Jaehwan Lee 1, Daekyeong Moon 2, Jik-So Kim 2 and Minkyoung Cho 2,*

1 Com2Us Corporation, Seoul 08506, Republic of Korea; jaehwlee@com2us.com
2 Department of Computer Engineering, Myongji University, Yongin 17058, Republic of Korea; dkmoon@mju.ac.kr (D.M.); jiksoo@mju.ac.kr (J.-S.K.)
* Correspondence: mkcho@mju.ac.kr; Tel.: +82-31-330-6435

Abstract: Audio auto-tagging is the process of assigning labels to audio clips for better categorization and management of audio file databases. With the advent of advanced artificial intelligence technologies, there has been increasing interest in directly using raw audio data as input for deep learning models in order to perform tagging and eliminate the need for preprocessing. Unfortunately, most current studies of audio auto-tagging cannot effectively reflect the semantic relationships between tags—for instance, the connection between “classical music” and “cello”. In this paper, we propose a novel method that can enhance audio auto-tagging performance via joint embedding. Our model has been carefully designed and architected to recognize the semantic information within the tag domains. In our experiments using the MagnaTagATune (MTAT) dataset, which has high inter-tag correlations, and the Speech Commands dataset, which has no inter-tag correlations, we showed that our approach improves the performance of existing models when there are strong inter-tag correlations.

Keywords: deep learning; music auto-tagging; joint embedding

1. Introduction

Audio auto-tagging automatically assigns metadata, or tags, to audio clips. Basically, these tags denote attributes, such as the sound type, the music genre, or specific instruments, that are designed to provide a substantial summary of the audio content. This process can help in organizing extensive audio file databases and formulating content-driven recommendations.

In the early stages of audio auto-tagging, traditional machine learning techniques, such as decision trees [1], the K-nearest neighbor algorithm [2], and support vector machines [3], were commonly employed after extracting a wide range of simple features from the audio [4–6]. Later on, using preprocessing techniques like spectrograms and mel-spectrograms, raw audio data could be transformed into image data and then fed into machine learning models. With the advance of artificial intelligence techniques, most recent work on audio auto-tagging can directly deal with the raw data without preprocessing by leveraging image-based deep learning models, such as convolutional neural networks (CNNs), to extract acoustic features and correlate them with the respective tags.

However, these recent deep learning models still lack utilization of the semantic information linking tags. For instance, the association between “classical music” and “cello” is intuitively stronger than that between “techno” and “quiet”. To address this issue, we introduce a joint embedding model that utilizes the semantic information from tags in order to enhance the performance of prior auto-tagging models. Our approach involves two distinct modules: one for extracting features from audio domain data, and another for tag domain data. Then, they are subsequently embedded into a shared (tag) feature space. Given the generalizability of our method, we could incorporate the audio feature extraction components from previously established audio auto-tagging models.

To summarize, the contributions of our paper are as follows:

---

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).
• We extract features from diverse audio data through asymmetric multi-modal embedding and project these features onto a relatively simple tag domain. Therefore, we are able to maintain various feature vectors of the audio data continuously and also encapsulate the correlation between the tag data and the audio data within the features. In contrast to traditional methods that directly map different modalities of data into one common embedding space, we are able to sufficiently maintain the distinct characteristics of each data modality through our one-sided joint embedding technique;

• Our experiments showed that the proposed model outperforms the previous methods with datasets with higher tag correlations. We drew this result from comparative experiments using two datasets with distinct characteristics: one with a high degree of tag correlation and another with quite independent tags. The experiments showed that relevant internal tag information can improve the final results, whereas independent tags degrade the performance;

• Our method is a general framework so it can be applied to existing auto-tagging models to utilize the tag semantic information without additional training data.

The rest of this paper is structured as follows: in Section 2, we describe the background and related research work. Section 3 describes the components, loss functions, and training algorithms for the proposed joint embedding model. Section 4 presents the experimental environments and the results. Finally, in Section 5, we conclude this paper and discuss the future research directions.

2. Related Work

Previously, audio auto-tagging models employed preprocessing techniques, such as spectrograms and mel-spectrograms [7], to process raw audio data before using machine learning or deep learning techniques for tagging. However, recent auto-tagging deep learning models directly handle raw data without preprocessing.

Dieleman and Schrauwen [8] initially used raw waveforms as input for CNN models directly and achieved comparable results to the previous models using mel-spectrograms. Lee et al. [9] proposed a CNN-based deep learning model called SampleCNN for audio auto-tagging. They sampled a raw audio frame with small size filters (e.g., two or three) through multiple CNN layers to obtain its frame-level representation. Later, they extended their results with squeeze and excitation [10] techniques and residual connection [11–13].

Won et al. [14] proposed HarmonicCNN using an audio representation module with a set of harmonic filters as a feature extraction module. Note that a harmonic is an integer multiple of the fundamental frequency. By combining harmonic knowledge and data-based learning methods, they built an audio representation module to reduce the dimensions of data waveforms and achieve fast convergence.

Raavanelli and Bengio [15] attempted a novel approach by using frequency filters in the frequency domain instead of the small-sized CNN filters traditionally employed for audio feature extraction. They proposed SincNet, which can learn using bandpass filters from raw audio data waveforms instead of using fixed frequency filters. Note that the sinc function is a function represented by the ratio of the sine function and its variables and has a rectangular shape in the frequency domain. The bandpass filter implemented as a sinc function learns only the cut-off frequency that can pass through the band, regardless of the length. As a result, the authors were able to dramatically decrease the number of parameters that need to be learned.

All of the above research work mainly used techniques to extract features from audio and predict tags but did not utilize the inherent relationships between the tags. Joint embedding is an advanced technique that can consolidate data from multiple domains or modalities into a coherent representation and is useful in tasks such as cross-modal retrieval and multi-modal learning.

Elizalde [16] proposed a framework for learning joint embeddings in tags and acoustic space, which was applied to a cross-modal audio search engine. Instead of using separate
embedding networks to extract features from audio and tag data, they trained a common embedding network, known as a Siamese network, with both audio and tag data. Using this network, they improved both text and audio queries for search and retrieval.

Favory [17,18] demonstrated a joint embedding scheme using two distinct autoencoders to extract features for sound data and tag data, respectively. In order to ensure that the latent vectors from each autoencoder were as similar as possible, they added a constraint to maximize the similarity between the latent vectors as a regularization term in their loss function.

Akbari et al. [19] proposed the Video-Audio-Text Transformer (VATT) model, which learns multi-modal representations using large-scale unlabeled video data. VATT linearly projects each modality (video, audio, text) into a feature vector and then feeds it into a shared transformer encoder. The authors achieved high performance in video action recognition, acoustic event detection, and image classification as downstream tasks after pretraining the model via self-supervised learning with large-scale video data.

However, these models may encounter an asymmetry issue when mapping the diverse range of audio data and the relatively limited range of tag data into a common embedding space. To address this problem, we propose an audio auto-tagging model with joint embedding that can learn not only acoustic features from the audio domain but also semantic information from the tag domain.

3. Architecture: Audio Auto-Tagging with One-Sided Joint Embedding

In this section, we present our audio auto-tagging model with a one-sided joint embedding technique. Our model consists of four modules: a tag autoencoder that extracts tag domain features, a feature extractor that extracts acoustic features of the audio domain from raw waveforms, a projector that maps audio domain features to tag domain features, and a classifier for predicting whether tags should be included. The overall architecture of our model is shown in Figure 1. The remainder of this section is devoted to the detail of the modules and how to train the model with two-phase steps.

![Figure 1. ATOSE Architecture.](image)

3.1. Model Architecture

First, we describe our tag autoencoder, which extracts tag features. Autoencoders are well-known models that train input data to be the same as output data and are composed of an encoder and a decoder. The encoder receives data and transforms them into a latent vector form that has the characteristics of the data, and this latent vector is then passed to the decoder to approximately restore the original input data. The latent space of this vector is considered as the inherent feature space of the tags. Our **tag autoencoder (TAE)** uses
an overcomplete autoencoder to effectively extract additional meaning information that includes the relationships between the tags that are embedded.

An overcomplete autoencoder is a type of autoencoder that uses a vector of larger dimensions than the dimensions of the input to be learned, even though most autoencoders generally have smaller latent dimensions than the input dimensions. The reason we chose larger latent dimensions is that it is advantageous if the tag information and the information for the relations between tags are stored independently in separate dimensions. In general, it is difficult to train overcomplete autoencoders since the model may tend to learn identity mapping given the data. To overcome the challenge, we train the autoencoder in conjunction with an audio feature extractor and a projection module, subject to constraints, aiming to ensure that the resultant latent feature vector from the autoencoder aligns closely with the projected tag feature vector derived from the audio data. This can be considered as regularization that makes the training possible while avoiding identical mapping. Additionally, during training, we randomly select positions from a single audio file to generate multiple fixed-length samples. This data augmentation technique effectively trains the autoencoder to learn from different samples with the same tag label. It helps improve the model's generalization and performance. Note that we set the latent vector dimensions of the overcomplete autoencoders as approximately two times larger than the nearest power of two of the input tag dimensions. Also, we apply layer normalization [20] to each layer to avoid extracting unnecessary features.

Now, we describe how to extract the features from audio and project them to the tag feature space. The feature extractor (FE) is a module responsible for extracting the acoustic features of an audio domain. As the proposed joint embedding method was designed to be applicable to existing models, we utilized previous models when implementing the FE. Specifically, we employed SampleCNN [9,12,13], HarmonicCNN [14], and SincNet [15] to predict audio tags by extracting features from the waveforms of the audio. Additional classifier layers were then used to classify the resulting tags. In order to ensure consistency across all models, we standardized the dimensionality of the acoustic features, transforming them into vectors of 1,024 dimensions.

The projector (PR) module is responsible for mapping diverse audio features provided by the FE to a relatively simpler tag latent domain. Unlike traditional joint embedding methods that map significantly different characteristic domains into a common domain, we employ a one-sided joint embedding approach. The approach enables us to map the complex audio domain to the simple tag domain while preserving the distinct characteristics of both domains and capturing additional relational information. The PR consists of two dense layers and a unit normalization layer. The input dimensions are the audio feature dimensions, and the output dimensions are set to match the latent vector dimensions of the TAE. During optimization, the resulting vector of the projection model is adjusted to closely resemble the latent vector of the TAE. A more detailed explanation of this matter is provided in the following section.

The classifier (CL) module classifies acoustic features extracted from the trained feature extractor into tags. The classifier connects two fully connected layers of 256 dimensions and one fully connected layer. It predicts the inclusion of tags represented by the multi-hot vectors.

Note that the implementations and experimental results for all models are available at https://github.com/jaehwlee/jetatag (accessed on 31 July 2023).

3.2. Training Procedure

Our joint embedding training algorithm consists of two stages that are specifically designed to enable our feature extractor to learn the semantic information of the tags. In order to ensure stable convergence, we employ scheduled Adam and SGD optimizers [21] with a batch size of 16 in both stages.

Stage one: We begin by training three components: the tag autoencoder (TAE), the feature extractor (FE), and the projector (PR). The TAE takes tag data in the form of
a multi-hot vector as input and extracts a tag embedding vector while simultaneously reconstructing the tags. On the other hand, the FE extracts acoustic features from audio input data and the PR projects these acoustic features into the tag domains, ensuring that they resemble the tag embedding vector obtained from the TAE.

We then define the objective functions of the modules to optimize. The loss for the TAE can be computed with the mean squared error to compare the correct and reconstructed tags as follows:

\[ L_{ae} = \sqrt{(\hat{y} - y)^2} \]

where \( y \) represents the ground-truth tags and \( \hat{y} \) the tags predicted by the TAE.

The loss for the joint embedding defines a cosine similarity function for two latent (tag) vectors from the tag data and the audio data as follows:

\[ L_{je} = \cos_{\text{loss}}(z_t, z_a) + \cos_{\text{loss}}(m - z_t, m - z_a) \]

where \( m \) is a constant (e.g., 0.4), \( z_t \) is the latent tag vector extracted from the TAE, and \( z_a \) is the latent tag vector obtained by projecting the acoustic features through the FE from the raw waveform using the PR. We add cosine embedding loss computation terms for \( m - z_a \) and \( m - z_t \) to prevent the latent vectors from converging to zero when their values are too small while narrowing the latent vector distance of the two domains in the joint embedding loss computation process. Note that \( \cos_{\text{loss}} \) is the pairwise loss, calculated as follows:

\[ \cos_{\text{loss}}(x, y) = \frac{x \cdot y}{||x|| \cdot ||y||} \]

Finally, the loss function \( L \) at stage one uses the value obtained by adding the tag autoencoder loss \( L_{ae} \) and the joint embedding loss \( L_{je} \).

\[ L = L_{ae} + L_{je} \]

**Stage two:** Recall that the feature extractor (FE) in the first stage has been trained to extract acoustic features from the input audio data. These acoustic features are then used as input for the classifier (CL), which predicts tags. Since the tags are in the form of multi-hot vectors, to measure the quality of the predictions, we use the binary cross-entropy between the ground-truth tags and the predicted tags.

\[ L_{c} = -\frac{1}{N} \sum_{n=1}^{N} [y_n \log \hat{y}_n - (1 - y_n) \log (1 - \hat{y}_n)] \]

where \( N \) is the total number of tags in the training dataset, \( y \) represents the ground-truth tags, and \( \hat{y} \) represents the tags predicted by the classifier.

4. Experiments

We conducted experiments using two distinct audio tagging datasets: the MagnaTagATune (MTAT) dataset and the Speech Commands dataset. The MagnaTagATune (MTAT) dataset exhibits high inter-tag correlations, whereas the Speech Commands dataset consists of independent tags with no inter-tag correlations. Our objective was to demonstrate that our one-sided joint embedding approach enhances the performance of existing models for the MagnaTagATune dataset specifically, which is characterized by strong inter-tag correlations.

In order to evaluate the performance, we conducted performance comparisons of the models without joint embedding (referred to as “original”) and the models with joint embedding (referred to as “with JE”) for each dataset.

Note that, before training with the datasets, we performed re-sampling to reduce data complexity and regularization to encourage scales of different amplitudes for each data characteristic as follows. The data at 44.1 kHz were re-sampled to 16 kHz, and scaling was
performed by dividing all values by the amplitude with the largest absolute value among the amplitudes of the waveform for one song. After that, the normalized data were divided into segments to be used as an input to the deep learning model. At this time, the length of the segments varied according to the model to be used as the feature extractor: with HarmonicCNN, the length was set to 80,000 (5 s), and both SampleCNN and Tag-SincNet used 59,049 (approximately 2.86 s). While training, only one random segment per audio file was employed, and while verifying, the hop length was set to half the segment length to undertake performance evaluation with all samples included in one audio file in order not to miss any audio segments. Note that we unified the final feature dimensions to 1024 for the performance comparison.

4.1. MagnaTagATune (MTAT) Dataset

The MagnaTagATune (MTAT) dataset [22] is a well-known dataset used for experiments with music auto-tagging tasks. The dataset consists of about 25,000 files with tags, and each file has a length of 29 sec and indicates whether it contains each of the 188 tags. In this paper, we adopt the experimental settings employed in previous studies [9,12–14], and we focused our experiments on the top 50 most frequently used tags and utilized data segmentation techniques.

In the music datasets, notable correlations can often be observed, such as the correlation between genre and associated instrument tags. This observation suggests that the proposed one-sided joint embedding technique, which takes advantage of tag correlations, has the potential to enhance the performance of classification tasks. Figure 2 shows the correlations among the top 10 highly correlated tags in relation to the “classical” tag.

![Image](Image.png)

Figure 2. The correlation of tags in the MTAT dataset.
We utilized the ground-truth tags from the MTAT dataset and calculated the correlations between tags. Note that we observed strong correlations between certain tags. For instance, the tag “classical” exhibited a high correlation of 0.44 with “strings”, 0.41 with “violin”, and 0.41 with “harp-sichord”. Similarly, the tag “strings” demonstrated a significant correlation of 0.46 with “violin”, 0.44 with “classical”, and 0.28 with “cello”. Furthermore, the tag “violin” displayed a substantial correlation of 0.46 with “strings”, 0.41 with “classical”, and 0.41 with “cello”. These correlations are shown in the provided figure, which depicts numerous tags that exhibit strong inter-correlations.

We used the area under the receiver operating characteristic curve (ROC-AUC) and the area under the precision-recall curve (PR-AUC) as evaluation metrics to measure the model’s performance with the MTAT dataset. The experimental results presented in Table 1 demonstrate the performance improvements achieved by all models incorporating joint embedding with the MTAT dataset. Specifically, the models showed enhancements in terms of the ROC-AUC, with improvements ranging from 1.45% to 2.15% p. Additionally, in terms of the PR-AUC, the models exhibited improvements ranging from 2.85% p to 6.11% p. Notably, our HarmonicCNN model leveraging our joint embedding technique achieved the best performance, with a PR-AUC of 91.27% and ROC-AUC of 53.78%. These results show the effectiveness of our joint embedding approach in the music auto-tagging task.

Table 1. Performance comparison with MTAT dataset.

<table>
<thead>
<tr>
<th>Feature Extractor</th>
<th>ROC-AUC Original</th>
<th>With JE</th>
<th>PR-AUC Original</th>
<th>With JE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SampleCNN</td>
<td>0.8688</td>
<td>0.8782</td>
<td>0.4014</td>
<td>0.4311</td>
</tr>
<tr>
<td>SampleCNN (+se)</td>
<td>0.8782</td>
<td>0.8857</td>
<td>0.4311</td>
<td>0.4596</td>
</tr>
<tr>
<td>SampleCNN (+res)</td>
<td>0.8712</td>
<td>0.8864</td>
<td>0.4201</td>
<td>0.4556</td>
</tr>
<tr>
<td>SampleCNN (+rese)</td>
<td>0.8736</td>
<td>0.8902</td>
<td>0.4211</td>
<td>0.4694</td>
</tr>
<tr>
<td>HarmonicCNN</td>
<td>0.8982</td>
<td>0.9127</td>
<td>0.4939</td>
<td>0.5378</td>
</tr>
<tr>
<td>Tag-SincNet</td>
<td>0.8294</td>
<td>0.8534</td>
<td>0.3330</td>
<td>0.3622</td>
</tr>
</tbody>
</table>

4.2. Speech Commands Dataset

The Speech Commands dataset [23] is used for tasks of limited-vocabulary speech recognition, primarily focused on identifying voice commands. It consists of approximately 105,000 voice files containing 35 simple word commands, such as “go”, “left”, “right”, “backward”, and “follow”. Each voice file is associated with a single word command, resulting in one-hot encoded vectors as ground-truth tags. Accordingly, we evaluated the performance of the models with this dataset using accuracy as the metric, which indicates the number of correctly identified commands within a dataset. This indicates that the tags in this dataset are independent of each other. To ensure the absence of tag correlations, we performed a thorough correlation analysis of the tags in the dataset, confirming the lack of significant correlations (See Figure 3).
Figure 3. The correlation of tags in the Speech Commands dataset

As can be seen from the experimental results in Table 2, most models with joint embedding exhibited a decline in performance with the dataset. Although one model showed a slight improvement, it was not significant. The underlying reason for this performance degradation was the lack of internal tag information in the dataset. Consequently, the additional tag information introduced by our joint embedding models may have introduced noise and potential misinterpretations.

Table 2. Performance comparison with Speech Command dataset.

<table>
<thead>
<tr>
<th>Feature Extractor</th>
<th>Accuracy</th>
<th>With JE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SampleCNN</td>
<td>0.9557</td>
<td>0.9483</td>
</tr>
<tr>
<td>SampleCNN (+se)</td>
<td>0.9595</td>
<td>0.9537</td>
</tr>
<tr>
<td>SampleCNN (+res)</td>
<td>0.9635</td>
<td>0.9476</td>
</tr>
<tr>
<td>SampleCNN (+rese)</td>
<td>0.9580</td>
<td>0.9312</td>
</tr>
<tr>
<td>HarmonicCNN</td>
<td>0.9635</td>
<td>0.9695</td>
</tr>
<tr>
<td>Tag-SincNet</td>
<td>0.9421</td>
<td>0.9038</td>
</tr>
</tbody>
</table>
### 4.3. Additional Mixed Experiment and Further Analysis

We conducted experiments with datasets with both high inter-tag correlations and zero correlations. Naturally, it would be expected that we would also investigate the case of intermediate tag correlations. However, due to the limited availability of public audio datasets, we chose to conduct our experiments using the DCASE 2017 dataset. Despite not being an ideal fit for the intermediate case, it is one of the more frequently used datasets in this field.

DCASE 2017 is a challenge dataset that was used for a sound event detection task in the Detection and Classification of Acoustic Scenes and Events workshop in 2017 [24]. The dataset was built using a part of Google’s AudioSet [25], which contains about 53,000 audio files each 10 s long classified into 17 categories of sound events, such as warning sounds and vehicle sounds. We used the F1-score as an evaluation metric to measure the model’s performance with the DCASE 2017 dataset.

Figure 4 shows the correlations among the top 10 highly correlated tags in relation to the “Fire truck (siren)” tag. We can observe correlations among the tags related to sirens. Note that the correlations are relatively weak, with values of less than 0.15.

![Correlation Matrix](image)

**Figure 4.** The correlation of tags in the DCASE 2017 dataset.
At the experimental results in Table 3, SampleCNN (+se) and HarmonicCNN showed performance improvements, with F1-scores rising from 0.5134 to 0.5312 and from 0.5052 to 0.5341, respectively. However, the results for the other models indicated degradation. We were unable to identify any specific trends when utilizing the semantic information of the tags with weak intermediate correlations.

Table 3. Performance comparison with DCASE 2017 dataset.

<table>
<thead>
<tr>
<th>Feature Extractor</th>
<th>F1-Score Original</th>
<th>F1-Score With JE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SampleCNN</td>
<td>0.4966</td>
<td>0.4583</td>
</tr>
<tr>
<td>SampleCNN (+se)</td>
<td>0.5134</td>
<td>0.5312</td>
</tr>
<tr>
<td>SampleCNN (+res)</td>
<td>0.5087</td>
<td>0.4448</td>
</tr>
<tr>
<td>SampleCNN (+rese)</td>
<td>0.5101</td>
<td>0.4534</td>
</tr>
<tr>
<td>HarmonicCNN</td>
<td>0.5052</td>
<td>0.5341</td>
</tr>
<tr>
<td>Tag-SincNet</td>
<td>0.4324</td>
<td>0.3623</td>
</tr>
</tbody>
</table>

In summary, through three different experiments, we confirmed that utilizing semantic tag information could enhance performance with high tag correlations. However, in cases with low tag correlations, we observed that performance could decrease. Additionally, we tried to understand why performance improved compared to the existing models when there were high tag correlations. For this purpose, for each given tag, we visualized the audio feature vectors with t-SNE and examined the differences compared to the original models. Although we discovered some meaningful changes in certain cases, we were unable to find consistent evidence across multiple trials.

5. Conclusions

In this paper, we proposed an audio auto-tagging model with joint embedding that can learn not only acoustic features from the audio domain but also semantic information from the tag domain. Our approach utilizes asymmetric multi-modal embedding to extract features from diverse audio data and project them onto a simplified tag domain. This allows for the continuous maintenance of various audio feature vectors while encapsulating the correlations between tags and audio data within the features.

Our experimental results demonstrated that our proposed model surpassed previous methods with the MTAT dataset, which has higher tag correlations. Specifically, the HarmonicCNN model with joint embedding achieved the highest performance, with a PR-AUC of 91.27% and an ROC-AUC of 53.78%. Our approach is a general method that can be applied to diverse datasets where there are inter-tag correlations for the ground-truth labels.

Recently, audio auto-tagging research has focused on pretrained models with large-scale datasets to enhance performance, similar to natural language processing research [26–28]. Our study stands apart by prioritizing performance enhancement without additional large-scale datasets. We tap into the intrinsic semantic information within tags to improve results for a given dataset. In future work, we will explore some applications, like movie genre prediction [29], and we plan to extend our research to various modalities, including images and texts.

Author Contributions: Conceptualization, J.L. and M.C.; methodology, J.L. and M.C.; software, J.L. and M.C.; validation, J.L., D.M., J.-S.K., and M.C.; formal analysis, J.L. and M.C.; investigation, J.L. and M.C.; resources, J.-S.K. and M.C.; data curation, J.-S.K. and M.C.; writing—original draft preparation, J.L. and M.C.; writing—review and editing, D.M., J.-S.K., and M.C.; visualization, J.L.; supervision, J.-S.K. and M.C.; project administration, J.-S.K. and M.C.; funding acquisition, D.M., J.-S.K., and M.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (no. NRF-2019R1A2C1005360) and the 2022 Research Fund of Myongji University.

Institutional Review Board Statement: Not applicable.
Informed Consent Statement: Not applicable.

Data Availability Statement: The data are available at https://github/jaehwlee/jetatag (accessed on 31 July 2023) or upon request to the corresponding author.

Acknowledgments: The authors sincerely thank the computer engineering department of Myongji University for generously providing the necessary computing resources for conducting the experiments.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adam</td>
<td>Adaptive Moment Estimation</td>
</tr>
<tr>
<td>CL</td>
<td>Classifier</td>
</tr>
<tr>
<td>CNN</td>
<td>Convolution Neural Network</td>
</tr>
<tr>
<td>FE</td>
<td>Feature Extractor</td>
</tr>
<tr>
<td>JE</td>
<td>Joint Embedding</td>
</tr>
<tr>
<td>MTAT</td>
<td>MagnaTagATune</td>
</tr>
<tr>
<td>PR</td>
<td>Projector</td>
</tr>
<tr>
<td>PR-AUC</td>
<td>Area Under the Precision-Recall Curve</td>
</tr>
<tr>
<td>ROC-AUC</td>
<td>Area Under the Receiver Operating Characteristic Curve</td>
</tr>
<tr>
<td>SGD</td>
<td>Stochastic Gradient Descent</td>
</tr>
<tr>
<td>TAE</td>
<td>Tag Autoencoder</td>
</tr>
<tr>
<td>VATT</td>
<td>Video-Audio-Text Transformer</td>
</tr>
</tbody>
</table>

References


Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.