

# SEMI-SUPERVISED MUSIC TAGGING TRANSFORMER

Minz Won<sup>1,2</sup>    Keunwoo Choi<sup>2</sup>    Xavier Serra<sup>1</sup>

<sup>1</sup> Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain

<sup>2</sup> ByteDance, Mountain View, California, United States

minz.won@upf.edu

## ABSTRACT

We present Music Tagging Transformer that is trained with a semi-supervised approach. The proposed model captures local acoustic characteristics in shallow convolutional layers, then temporally summarizes the sequence of the extracted features using stacked self-attention layers. Through a careful model assessment, we first show that the proposed architecture outperforms the previous state-of-the-art music tagging models that are based on convolutional neural networks under a supervised scheme.

The Music Tagging Transformer is further improved by noisy student training, a semi-supervised approach that leverages both labeled and unlabeled data combined with data augmentation. To our best knowledge, this is the first attempt to utilize the entire audio of the million song dataset.

## 1. INTRODUCTION

Automatic music tagging is a classification task whose objective is computational understanding of music semantics. From a given audio excerpt, a trained music tagging model predicts relevant tags (e.g., genre, mood, instrument, decade, region) based on its acoustic characteristics. The task has attracted music information retrieval (MIR) researchers due to its wide pragmatic usages in many applications. Especially, there is a strong demand from industries that have music recommendation services from large-scale music libraries. Thanks to the recent advances in deep learning, mostly convolutional neural networks (CNNs) [1], the performances of music tagging models have been significantly enhanced by leveraging large-scale data with various deep architectures [2–5]. However, there still are two limitations in the current music tagging research: i) chunk-based prediction and ii) a limited amount of labeled data for supervised learning.

Music signals are in the form of sequential data. In this sequence, regarding typical tags, some acoustic characteristics may appear locally (e.g., instruments) while some others may span over the sequence (e.g., mood, genre). This means a successful music tagging model needs to be

able to extract both local and global features. Fully convolutional network [2], one of the very early deep learning models for music tagging, was designed to capture both local and global features by increasing the size of the overall receptive fields with max-pooling. More recently, however, it is shown that training with a smaller hop size with shorter audio chunks is beneficial for music tagging [6]. This approach has been adopted in many CNN-based models [3–5], where the models are trained with short audio chunks (3 to 5 second long), densely striding max-pooling, and a global pooling layer. To predict music tags of a 3-minute song, for example, the audio is split into multiple short audio chunks, and the model makes predictions on each chunk. Then, the predictions are aggregated through majority vote or global average-/max-pooling. This means that on a track level, the current music tagging models are performing like a bag-of-features model [7] instead of modeling music representation as a sequence.

Another limitation of the current music tagging research is a limited amount of labeled data. The Modern deep learning models are data-hungry. However, manually labeling music tags is time-consuming and requires domain expertise. In pursuit of large-scale research, the million song dataset (MSD) [8], which literally includes a million songs in it, became popular in music tagging research. Among the million songs, however, only about 24% are labeled with at least one of the top-50 music tags. Most of the previous music tagging research has only utilized the labeled data while discarding 76% of the songs in the dataset. This type of setup (i.e., a small labeled dataset along with a large unlabeled dataset) is not limited to the MSD but can be found often in the real world regardless of the domain. To leverage the unlabeled data, self-supervised [9–13] and semi-supervised [14–16] learning have been actively explored in computer vision and natural language processing.

In this paper, we present Music Tagging Transformer that is trained with a semi-supervised approach. Our main contribution is three-fold: (i) through a careful model assessment, we show that our Music Tagging Transformer outperforms the previous works, (ii) we show that we can use unlabeled data to improve music tagging performances via semi-supervised learning, (iii) we provide a new split of the MSD to solve known issues of the previous one. Reproducible code is available online. <sup>1</sup>

<sup>1</sup> <https://github.com/minzwon/semi-supervised-music-tagging-transformer>



## 2. RELATED WORKS

### 2.1 Music Tagging

#### 2.1.1 CNN-based models

The research using CNNs accelerated the improvement of music tagging models. Choi et al. proposed to adopt CNNs for music tagging [2]. Pons et al. intended to inject domain knowledge in their architecture by designing vertically and horizontally long filters so that the network can capture timbral and temporal features, respectively [3]. Lee et al. formalized music tagging as a fully end-to-end process by using a 1D CNN and raw audio inputs [4]. Won et al. proposed to use a learnable harmonic front end for a 2D CNN [5]. More recently, the authors of [6] evaluated all these CNN-based music tagging models under the same experimental environment. Two of the main conclusions of [6] are to recommend (i) using mel spectrogram inputs, and (ii) using the most granular 2D filters (i.e.,  $3 \times 3$  convolution) instead of manual design choices. That means, a simple 2D CNN with mel spectrogram inputs, which is prevalent and sometimes referred to as *vgg-ish* model, is still outperforming the other music tagging models.

#### 2.1.2 Multiple instance learning

Music tagging can be seen as a multiple instance learning (MIL) [17] problem. A given music signal (a bag of multiple instances) will be labeled with a tag if a part (an instance) of the signal has a certain, relevant acoustic characteristic. The acoustic characteristic may span across the entire sequence (e.g., mood) or appear partially (e.g., instruments). There are two approaches of handling this aspect in music tagging. One is to train an instance-level model then aggregate the instance-level predictions; the other is to handle multiple instances in a single model.

The first approach has been used in many systems [3–5]. This approach is justified by our intuition – humans can predict music tags within just a few seconds. For example, people would not spend 3 minutes to determine whether a track is *rock*. In this approach, during the evaluation phase, the instance-level predictions are aggregated with a method such as majority vote, global max pooling, global average pooling, or adaptive pooling [18].

The second approach, the single model one, tackles the MIL problem in an end-to-end fashion. Fully convolutional network [2], an early music tagging model using CNN, models the entire 30-second mel spectrogram inputs. However, this model has to stick to a fixed input size and shows a relatively lower performance compared to other models that are trained with short audio chunks. To take the global structure into account while not losing local features, sequential modeling was added to the deep learning architectures. Convolutional recurrent neural network (CRNN) [19] is designed to capture local acoustic characteristics in a CNN front end and to summarize the sequence of the extracted features using an RNN back end. Similarly, another previous work [20] adopted a sequence model from the natural language processing field, the Transformer [21].

### 2.2 Transformers

#### 2.2.1 Overview

Transformer [21, 22] has shown its ability in sequence modeling, establishing itself *de facto* state-of-the-art in natural language processing. The structure of Transformer is a deep stack of self-attention layers. In each layer, by self-attention mechanism, the pairwise attention scores between every time step are calculated to output another sequence which includes a better context in each time step. In detail, the self-attention score is calculated as:

$$\text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V, \quad (1)$$

where  $d_k$  is the dimension of the *key* and  $Q$ ,  $K$ , and  $V$  are *query*, *key*, and *value*, respectively. Based on the relevance of *key* and *query* (i.e., relationship between two items), how much of *value* to be passed to the next layer is decided. Since Transformer has direct paths between each time step, it does not suffer from *vanishing gradient*, which is a critical problem of RNN families when modeling long sequences. However, Transformer may suffer from its memory complexity which is quadratic to the length of the sequence.

Due to its receptive field, which is unlimited within the input sequence, Transformer has become extremely popular in sequential data modeling such as text [21], symbolic music [23], and video [24]. It even demonstrated its representation power in non-sequential data such as image [25].

#### 2.2.2 Self-attention in music tagging

Hereinafter, we refer to the previous work [20] as CNNSA (acronym of convolutional neural network with self-attention) so that we can distinguish it from the proposed Music Tagging Transformer. The CNNSA model is the first music tagging model that uses Transformer. The model consists of a CNN front end and a Transformer back end. The CNN front end captures local (0.03 to 2.6 seconds) acoustic characteristics and the Transformer back end temporally summarizes the sequence of the features. Based on previous works, the CNNSA investigated two types of CNN front end which have two very different motivations. One is based on hand-crafted vertically and horizontally long filters [3] to capture timbral and temporal characteristics, respectively. The other is a more flexible, data-driven approach by using 1D CNN [4] with raw audio inputs. The CNNSA showed comparable results in music tagging but did not outperform other previous CNN approaches. Nevertheless, it demonstrated the Transformer’s sequence modeling ability and better temporal interpretability.

### 2.3 Semi-supervised Learning

With the advances of scalable hardware and training algorithms, the demand for labeled data has outpaced the progress of the size of datasets in many fields. As a solution, researchers started to develop methods that can take advantage of unlabeled data. Self-supervised [9–12] and

semi-supervised learning [14–16] aim at leveraging the abundant unlabeled data and have shown strong performances in various domains including computer vision and natural language processing.

In many self- and semi-supervised learning approaches, the models are trained to return noise-invariant predictions [11, 12, 15]. When there is an apple on the table, for example, it is always an apple even if we take a look at it from a different angle or a different distance, under different lights, or through glass. This ‘noise’ is usually realized in a form of data augmentation. In detail, with a self-supervised learning scheme, models are trained to optimize the agreement between different views of the same input [11, 12].

On the contrary, semi-supervised learning takes advantage of *both* existing labeled data and unlabeled data. One effective way of handling the data is to formalize the problem as teacher-student learning [15, 26, 27]. In teacher-student learning, a teacher model is first trained with labeled data in a supervised scheme and then, a student model is trained to mimic the teacher’s behavior by predicting pseudo-labels [28] that are generated by the teacher model. The teacher-student training has been actively explored with the purpose of domain adaptation [26], knowledge distillation [29], and knowledge expansion [15]. Especially, noisy student training [15] successfully takes advantage of the teacher-student training with the aforementioned noise invariance.

### 3. MODELS

#### 3.1 Short-chunk ResNet

As a simple 2D CNN with mel spectrogram inputs outperforms other music tagging models [6], we use a short-chunk ResNet model as our baseline. The model is trained with 3.69-second short audio chunks (instance-level), then the predictions are later aggregated by averaging them in the evaluation phase. It is a seven-layer CNN and each layer comprises of  $3 \times 3$  convolution with residual connection [30], batch normalization [31], rectified linear unit (ReLU) non linearity, and  $2 \times 2$  max pooling. This is a variant of the prevalent *vgg-ish* model, but due to its specific characteristics (i.e., short-instance-level training and densely striding max pooling), it is referred to as short-chunk ResNet [6]. We used a publicly available implementation <sup>2</sup>.

#### 3.2 Music Tagging Transformer

The proposed Music Tagging Transformer consists of two parts: a CNN front end and a Transformer back end as in Figure 1. On a high level, our structure is similar to the previous CNNSA [20] where CNN captures local spectrotemporal features and Transformer globally summarizes the sequence of the extracted features. However, there are two main differences in the front end: the CNN architecture and the reshaping layer.

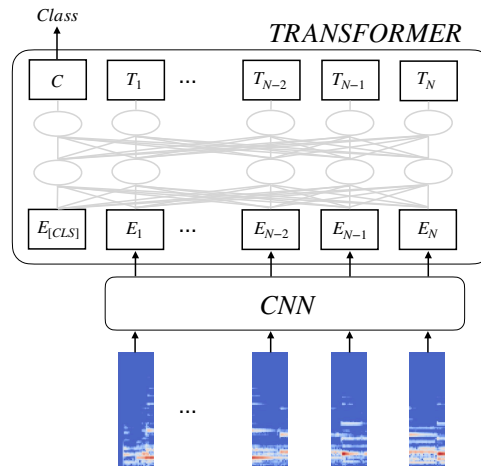


Figure 1: Proposed Music Tagging Transformer.

| layer                    | output shape                        |
|--------------------------|-------------------------------------|
| Input                    | $B \times 1 \times F \times T$      |
| Conv ( $3 \times 3$ )    | $B \times C \times F \times T$      |
| MaxPool ( $2 \times 2$ ) | $B \times C \times F/2 \times T/2$  |
| Conv ( $3 \times 3$ )    | $B \times C \times F/2 \times T/2$  |
| MaxPool ( $2 \times 2$ ) | $B \times C \times F/4 \times T/4$  |
| Conv ( $3 \times 3$ )    | $B \times C \times F/4 \times T/4$  |
| MaxPool ( $2 \times 1$ ) | $B \times C \times F/8 \times T/4$  |
| Reshape                  | $B \times (C \cdot F/8) \times T/4$ |
| Fully-connected          | $B \times C' \times T/4$            |

Table 1: Front end CNN of Music Tagging Transformer.

The CNNSA [20] investigated two different CNN front ends that are from two opposite motivations. One is using hand-crafted filter design on mel spectrogram inputs to leverage domain knowledge [3], the other is, with a fully data-driven spirit, one that learns from raw audio in an end-to-end fashion [4]. However, we follow a suggestion from a more recent work [6] – we use  $3 \times 3$  convolution filters with residual connections [30] on mel spectrogram inputs. Table 1 outlines our 3-layered CNN front end where  $B$  is the batch size,  $C$  is the number of convolution channels,  $F$  is the number of mel bins,  $T$  is the number of frames, and  $C'$  is the number of attention channels of Transformer. This CNN front end i) helps the model to capture local representations and ii) reduces the time resolution of the input so that it is feasible to train the following back end.

At the end of the CNN, the second and the third dimensions are reshaped into a single dimension. This flattening is motivated by Vision Transformer (ViT) [25] which reshapes a 2D image patch into a one-dimensional array. As a result, the output of the CNN is a sequence of short-chunk audio features where a chunk corresponds to approximately 0.1 second. It is input to the back end Transformer. This is in contrast to the CNNSA [20] that used the frequency-axis max-pooling at the end of its front end. In other words, in Music Tagging Transformer, the attention layers are given more detailed spectral information.

Our back end Transformer architecture is nearly iden-

<sup>2</sup> <https://github.com/minzwon/sota-music-tagging-models>

tical to the previous works [20, 21] except for the number of parameters and input lengths. After a hyperparameter search, we chose 4 layers, 256 attention dimensions, and 8 attention heads. At the input stage, positional embedding [21] is applied and a special token embedding  $E_{[CLS]}$  is inserted so that the Transformer can perform sequence classification (Figure 1) as a downstream task.

### 3.3 Noisy Student Training

To leverage unlabeled data, we investigate noisy student training [15], a successful semi-supervised learning approach. Table 2 provides an overview of noisy student training with pseudocode. First, we train a teacher model  $\mathcal{T}$  with a conventional supervised learning approach (line 1–6). Then, we train a student model  $\mathcal{S}$  with two types of losses. The first loss,  $l_1$ , is coming from the typical supervised approach with labeled data (line 10–11) as done for the teacher model. The other loss,  $l_2$ , is from unlabeled inputs and the corresponding pseudo-labels provided by the teacher model (line 12–15). In order to make the student model perform beyond mimicking the teacher model, data augmentation is applied (line 13). Both hard and soft labels can be used for the pseudo-labels [15] and we use soft labels in our work. If the training is successful, the trained student model would outperform the teacher model. Furthermore, the whole training process can be done iteratively by using the student as a new teacher model and training another student model to obtain an even better performing model. For a stronger teacher model, we used data augmentation in our supervised learning pipeline as well (line 1–6 and 10–11). As a result, the only pipeline without data augmentation is pseudo-label generation (line 12).

The size of the student model can be identical or larger than the teacher model. In this case, one can interpret the training process as *knowledge expansion* [15], meaning the knowledge in the teacher model is upgraded in the student model. One can also design the student model to be smaller than the teacher model, making the process *knowledge distillation* [32]. Knowledge expansion and knowledge distillation are complementary; depending on the use-case, one could pursue either performance or efficiency. We investigate both directions in this paper.

### 3.4 Data Augmentation

Data augmentation is a key to success in noisy student training. In our experiments, we take advantage of *Audio Augmentations* library [33] which is easily integrated to PyTorch data pipeline. The applied data augmentation methods are as follows:

- **Polarity inversion.**
- **Additive noise** by  $k_{snr} \in \{0.3, 0.5\}$ .
- **Random gain** by  $\mathcal{A} \in \{-20, -1\}$  dB.
- **High-pass filter** by  $f_H \in \{2200, 4000\}$  Hz.
- **Low-pass filter** by  $f_L \in \{200, 1200\}$  Hz.

---

### Noisy Student Training

---

**Input** labeled data  $X$ , labels  $Y$ , unlabeled data  $Z$

**Models** teacher model  $\mathcal{T}$ , student model  $\mathcal{S}$

**Functions** loss function  $\mathcal{L}$ , data augmentation  $\mathcal{A}$ ,  
back propagation  $\mathcal{B}$

**Train**

```

1  for  $x \in X, y \in Y$ 
2  do
3   $p \leftarrow \mathcal{T}(x)$  // predict
4   $l \leftarrow \mathcal{L}(p, y)$  // get loss
5   $\mathcal{T} \leftarrow \mathcal{B}(\mathcal{T}, l)$  // update teacher model
6  end do
7  end for
8  for  $x \in X, y \in Y, z \in Z$ 
9  do
10  $p_1 \leftarrow \mathcal{S}(x)$  // predict
11  $l_1 \leftarrow \mathcal{L}(p_1, y)$  // get supervised loss
12  $\psi \leftarrow \mathcal{T}(z)$  // generate pseudo-label
13  $\hat{z} \leftarrow \mathcal{A}(z)$  // data augmentation
14  $p_2 \leftarrow \mathcal{S}(\hat{z})$  // predict
15  $l_2 \leftarrow \mathcal{L}(p_2, \psi)$  // get semi-supervised loss
16  $\mathcal{S} \leftarrow \mathcal{B}(\mathcal{S}, l_1 + l_2)$  // update student model
17 end do
18 end for

```

---

**Table 2:** Pseudocode of noisy student training.

- **Delay** by  $t \in \{200, 500\}$  ms.
- **Pitch shift** by  $n \in \{-7, 7\}$  semitones.
- **Reverb** by room size  $s \in \{0, 100\}$ .

Each augmentation method is activated independently with a probability  $p \in \{0.3, 0.7\}$ .

## 4. DATASET

We use the million song dataset (MSD) [8] which consists of one million songs with audio features and metadata. Most of the previous works [2–6] relied on the Last.fm tags, a set of crowdsourced music tags, as ground truth. A popular approach is to take the most frequent 50 tags and select tracks that have at least one of the tags. This results in 242k songs, which are split into train, validation, and test sets.<sup>3</sup> Hereinafter, we refer to this as a conventional (MSD) split. Since this split has been widely used, we use it to benchmark the proposed Music Tagging Transformer against the previous works.

We also suggest a new split to alleviate some known problems of the conventional split. There are two problems – First, since the MSD music tags are collected from users, some of them are very noisy, and that may lead to noisy (and incorrect) evaluation [34]. Second, a strict split of music items requires taking the artist information into consideration since often, songs and labels from the same

<sup>3</sup> [https://github.com/keunwoochoi/MSD\\_split\\_for\\_tagging](https://github.com/keunwoochoi/MSD_split_for_tagging)

artist heavily resemble each other. However, the conventional split was done without such consideration, having caused unintended information leakage between the training and evaluation sets. Ultimately, this would cause an overly optimistic evaluation. As a solution, we use manually cleaned data from a previous work [35] and take the top 50 tags. We also propose a new split of MSD that does not share any artist among training/validation/test sets and is extended to more tracks. We name this ‘CALS split’ (cleaned and artist-level stratified split). CALS split consists of 233k labeled tracks and 516k unlabeled tracks. To the best of our knowledge, this is the first attempt to utilize the entire MSD audio, although we have to discard the rest 250k tracks to avoid information leakage by shared artists.

When noisy student training is used, it is common to have an unlabeled set that is significantly bigger than the labeled set. For example, in computer vision, 81 million unlabeled items were used along with 1.2 million labeled items [15], making the ratio of the semi-supervised set to be 67.5 (81/1.2). However, with 233k labeled items and 516k unlabeled items, our ratio is only around 2.3. This might be a factor that limits us from fully exploring the potential advantage of semi-supervised learning as we will discuss in Section 5.

## 5. EXPERIMENT

We use the MSD that was introduced in detail in Section 4. All the audio signals are pre-processed to 22,050 Hz sample rate and converted to short-time Fourier transform representations with a 1024-point FFT and 50%-overlapping Hann window. Finally, we convert them to log mel spectrograms with 128 mel bins.

All models that are used or introduced in this paper are optimized using Adam [36] with a learning rate of 0.0001. The best model is selected based on the binary cross entropy loss of the validation set and early stopping is applied when the validation loss does not improve for 20 epochs.

Music tagging models are typically evaluated with Area Under Receiver Operating Characteristic Curve (ROC-AUC). However, it is known that ROC-AUC may report overly optimistic results with highly skewed data [37]. Therefore, as our main evaluation metrics, we use not only ROC-AUC but also Area Under Precision-Recall Curve (PR-AUC).

### 5.1 Performance with the conventional split

Table 3 summarizes the performance of previous systems and the proposed model using the conventional split. The ROC-AUC and PR-AUC of the many previous models have been under 0.89 and 0.33, respectively. Our model, Music Tagging Transformer, outperforms the previous state-of-the-art models, harmonic CNN and short-chunk ResNet. The improvement, especially on PR-AUC, is non-trivial and even larger with data augmentation.

The front end of our Music Tagging Transformer takes a sequence of chunks, where each of which represents a very short duration of the signal ( $\approx 0.1$  second) [6]. Be-

| Models                                   | ROC-AUC       | PR-AUC        |
|------------------------------------------|---------------|---------------|
| FCN [2]                                  | 0.8742        | 0.2963        |
| Musicnn [3]                              | 0.8788        | 0.3036        |
| Sample-level [4]                         | 0.8789        | 0.2959        |
| Sample-level+SE [38]                     | 0.8838        | 0.3109        |
| CRNN [19]                                | 0.8460        | 0.2330        |
| CNNSA [20]                               | 0.8810        | 0.3103        |
| Harmonic CNN [5]                         | 0.8898        | 0.3298        |
| Short-chunk CNN [6]                      | 0.8883        | 0.3251        |
| Short-chunk ResNet [6]                   | 0.8898        | 0.3280        |
| Transformer (proposed) <sup>§</sup>      | <b>0.8916</b> | <b>0.3358</b> |
| Transformer (proposed) + DA <sup>†</sup> | <b>0.8972</b> | <b>0.3479</b> |

**Table 3:** Performance comparison using the conventional MSD split for top-50 music tagging. The § and † marks mean they are based on the identical model architecture and training strategy; compared to the same, marked models in Table 4, only the dataset split is different.

| Models                      | #param | ROC-AUC       | PR-AUC        |
|-----------------------------|--------|---------------|---------------|
| ResNet [6]                  | 13.5m  | 0.9098        | 0.3525        |
| ResNet+DA                   | 13.5m  | 0.9141        | 0.3705        |
| ResNet+DA+KE                | 13.5m  | 0.9165        | 0.3728        |
| ResNet+DA+KD                | 3.4m   | <b>0.9171</b> | <b>0.3742</b> |
| Transformer <sup>§</sup>    | 4.6m   | 0.9188        | 0.3775        |
| Transformer+DA <sup>†</sup> | 4.6m   | 0.9191        | 0.3845        |
| Transformer+DA+KE           | 4.6m   | 0.9204        | 0.3839        |
| Transformer+DA+KD           | 0.5m   | <b>0.9217</b> | <b>0.3889</b> |

**Table 4:** Performance comparison using the CALS MSD split for music tagging.

cause 0.1 second would be too short to represent musical characteristics alone, we interpret that the experimental results would mean our Transformer back end plays a role of sequential feature extractor beyond simple bag-of-feature aggregation. This may be an important aspect of the proposed model since sequential modeling is what the self-attention mechanism is the best suit.

The data augmentation we adopted contributes to improvements of 0.0056 ROC-AUC and 0.0119 PR-AUC. These are bigger than many of the improvements we have seen between different architecture choices. This emphasizes that data augmentation should be considered when developing a music tagging model.

### 5.2 Performance with the CALS split

For a deeper and more accurate analysis of the proposed models and methods, we use the proposed CALS split and run experiments with various configurations. Table 4 presents the experimental results of short-chunk ResNets (baseline architecture) and Music Tagging Transformers. For both of the models, it also summarizes the results of supervised models (the baseline among training methods), models with data augmentation (DA), models with

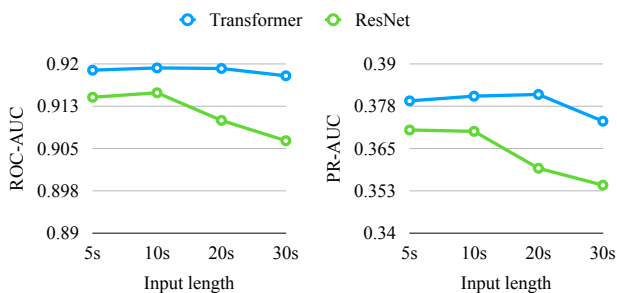


Figure 2: Performance with different input lengths.

| width | depth | ROC-AUC       | PR-AUC        |
|-------|-------|---------------|---------------|
| 32    | 4     | 0.9118        | 0.3528        |
| 64    | 4     | 0.9178        | 0.3754        |
| 128   | 4     | <b>0.9194</b> | 0.3776        |
| 256   | 4     | 0.9191        | <b>0.3845</b> |
| 512   | 4     | 0.9177        | 0.3788        |
| 768   | 4     | 0.9162        | 0.3707        |
| 1024  | 4     | 0.9174        | 0.3736        |
| 256   | 1     | 0.9180        | 0.3736        |
| 256   | 2     | 0.9193        | 0.3805        |
| 256   | 4     | <b>0.9199</b> | 0.3814        |
| 256   | 8     | 0.9181        | <b>0.3826</b> |
| 256   | 12    | 0.9165        | 0.3780        |
| 256   | 16    | 0.9169        | 0.3785        |

Table 5: The performance of Music Tagging Transformer with varying width and depth of the attention layers.

DA and knowledge expansion (KE), and models with DA and knowledge distillation (KD). Note that the two bottom rows of Table 3 correspond to the 5th and 6th rows of Table 4 as marked with § and †.

For both short-chunk ResNet [6] and the Music Tagging Transformer, we observe constant improvements when data augmentation and noisy student training (knowledge expansion) are applied accumulatively. This shows that for both of the architectures, the size of the dataset is a factor that limits the performance of the models.

In Section 4, we mentioned that our ratio of the semi-supervised set is relatively small. There are two observations that may be related to it. First, unlike a previous work in computer vision [15], we could not observe any performance gain by iterating the noisy student training (i.e., repeating to use a student model as the next teacher model). Second, interestingly, the student model with smaller parameters (models with DA and KD) showed better performance than larger models (models with DA and KE). This would be explained more clearly if the models are trained with a significantly richer dataset, one that is bigger and/or has more diverse data. Unfortunately, we could not run such an experiment due to the lack of a suitable dataset.

### 5.3 More Hyperparameter Search

In this section, we present the experiment results with various model configurations. First, we trained our Music Tag-

ging Transformer and short-chunk ResNet with varying input lengths to assess our proposed model’s ability to handle long sequences. As shown in Figure 2, on both of the metrics, short-chunk ResNet shows a noticeable performance degradation as the audio input gets longer. This shows that the global max pooling in the short chunk ResNet is not perfectly suitable for a long signal. Meanwhile, the Music Tagging Transformer shows consistent performances in general. An exception is when the input is 30-second long. We suspect the performance drop of the Music Tagging Transformer happens because the model cannot take advantage of random cropping data augmentation effect since the 30-second is the full length of the MSD previews.

Second, we investigate different Transformer parameters to figure out the best performing setup. As summarized in Table 5, Transformer achieved the best performance with attention channels (width) at 128 and 256, and their depth of 4 and 8 layers. However, these optimal parameters are dataset-dependent; as generally observed, a larger network structure would perform better if a larger amount of training data is provided.

## 6. CONCLUSION

In this paper, we proposed a new architecture, Music Tagging Transformer, and improved its tagging performance with a semi-supervised scheme: noisy student training. Experimental results showed that the proposed architecture outperforms the previous state-of-the-art models in supervised music tagging using the MSD [8]. The results also indicate that the tagging models can be further enhanced using noisy student training – with either knowledge expansion and knowledge distillation. We also provided an analysis result that shows Music Tagging Transformer can handle long audio inputs better than the previous CNN architectures do.

In future work, our Transformer can be further utilized in various MIR tasks. Since Transformer can perform both sequence-level and token-level classification, it can be used in not only music tagging but also tasks such as beat detection and melody extraction. Finally, by combining the multiple MIR tasks in a multi-task learning scheme, Transformer can be trained as a general purpose music representation learning model.

Another important direction to explore is self-supervised learning. Our Music Tagging Transformer can be pre-trained in a self-supervised scheme such as masked embedding prediction [21] or contrastive learning [11–13]. Finally, the pre-trained model can be optimized together with semi-supervised learning to further improve the performance [16].

## 7. ACKNOWLEDGEMENT

This work was funded by the predoctoral grant MDM-2015-0502-17-2 from the Spanish Ministry of Economy and Competitiveness linked to the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502).



## 8. REFERENCES

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems (NIPS)*, 2012.
- [2] K. Choi, G. Fazekas, and M. Sandler, “Automatic tagging using deep convolutional neural networks,” in *In Proc. of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, New York, USA, 2016.
- [3] J. Pons, O. Nieto, M. Prockup, E. Schmidt, A. Ehmann, and X. Serra, “End-to-end learning for music audio tagging at scale,” in *In Proc. of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018.
- [4] J. Lee, J. Park, K. L. Kim, and J. Nam, “Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms,” *In Proc. of the 14th Sound and music computing (SMC)*, 2017.
- [5] M. Won, S. Chun, , O. Nieto, and X. Serra, “Data-driven harmonic filters for audio representation learning,” *In Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.
- [6] M. Won, A. Ferraro, D. Bogdanov, and X. Serra, “Evaluation of cnn-based automatic music tagging models,” *In Proc. of Sound and Music Computing (SMC)*, 2020.
- [7] W. Brendel and M. Bethge, “Approximating cnns with bag-of-local-features models works surprisingly well on imagenet,” *International Conference on Learning Representations (ICLR)*, 2019.
- [8] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, “The million song dataset,” *In Proc. of the 12th International Conference on Music Information Retrieval (ISMIR)*.
- [9] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *arXiv preprint arXiv:1807.03748*, 2018.
- [10] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [11] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *International Conference on Machine Learning (ICML)*, 2020.
- [12] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar *et al.*, “Bootstrap your own latent: A new approach to self-supervised learning,” *arXiv preprint arXiv:2006.07733*, 2020.
- [13] J. Spijkervet and J. A. Burgoyne, “Contrastive learning of musical representations,” *In Proc. of International Society for Music Information Retrieval Conference (ISMIR)*, 2021.
- [14] O. Chapelle, B. Scholkopf, and A. Zien, “Semi-supervised learning,” *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 542–542, 2009.
- [15] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, “Self-training with noisy student improves imagenet classification,” in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [16] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton, “Big self-supervised models are strong semi-supervised learners,” in *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.
- [17] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, “Solving the multiple instance problem with axis-parallel rectangles,” *Artificial intelligence*, vol. 89, no. 1-2, pp. 31–71, 1997.
- [18] B. McFee, J. Salamon, and J. P. Bello, “Adaptive pooling operators for weakly labeled sound event detection,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 11, pp. 2180–2193, 2018.
- [19] K. Choi, G. Fazekas, M. Sandler, and K. Cho, “Convolutional recurrent neural networks for music classification,” in *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 2392–2396.
- [20] M. Won, S. Chun, and X. Serra, “Toward interpretable music tagging with self-attention,” *arXiv preprint arXiv:1906.04972*, 2019.
- [21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.
- [22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *Conference on Neural Information Processing Systems (NIPS)*, 2017.
- [23] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, “Music transformer,” *International Conference on Learning Representations (ICLR)*, 2019.
- [24] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, “Vivit: A video vision transformer,” *arXiv preprint arXiv:2103.15691*, 2021.

- [25] 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,” *International Conference on Learning Representations (ICLR)*, 2021.
- [26] J. Li, M. L. Seltzer, X. Wang, R. Zhao, and Y. Gong, “Large-scale domain adaptation via teacher-student learning,” *arXiv preprint arXiv:1708.05466*, 2017.
- [27] S. Kum, J.-H. Lin, L. Su, and J. Nam, “Semi-supervised learning using teacher-student models for vocal melody extraction,” *arXiv preprint arXiv:2008.06358*, 2020.
- [28] D.-H. Lee *et al.*, “Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks,” in *Workshop on challenges in representation learning, ICML*, 2013.
- [29] Y. Kim and A. M. Rush, “Sequence-level knowledge distillation,” *Proc. of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.
- [30] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016.
- [31] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International conference on machine learning (ICML)*, 2015.
- [32] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *Advances in neural information processing systems (NIPS), Deep learning workshop*, 2015.
- [33] J. Spijkervet, “Spijkervet/torchaudio-augmentations,” 2021. [Online]. Available: <https://doi.org/10.5281/zenodo.5042440>
- [34] K. Choi, G. Fazekas, K. Cho, and M. Sandler, “The effects of noisy labels on deep convolutional neural networks for music tagging,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2018.
- [35] M. Won, S. Oramas, O. Nieto, F. Gouyon, and X. Serra, “Multimodal metric learning for tag-based music retrieval,” in *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.
- [36] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *International Conference on Learning Representations (ICLR)*, 2015.
- [37] J. Davis and M. Goadrich, “The relationship between precision-recall and roc curves,” in *Proc. of international conference on Machine learning (ICML)*, 2006.
- [38] T. Kim, J. Lee, and J. Nam, “Sample-level cnn architectures for music auto-tagging using raw waveforms,” in *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.