ReconVAT: A Semi-Supervised Automatic Music Transcription Framework for Low-Resource Real-World Data

Kin Wai Cheuk*
Dorien Herremans
Information Systems Technology and Design
Singapore University of Technology and Design
Singapore
{kinwai_cheuk,dorien_herremans}@mymail.sutd.edu.sg

Li Su
Institute of Information Science
Academia Sinica
Taiwan
lisu@iis.sinica.edu.tw

ABSTRACT

Most of the current supervised automatic music transcription (AMT) models lack the ability to generalize. This means that they have trouble transcribing real-world music recordings from diverse musical genres that are not presented in the labelled training data. In this paper, we propose a semi-supervised framework, ReconVAT, which solves this issue by leveraging the huge amount of available unlabelled music recordings. The proposed ReconVAT uses reconstruction loss and virtual adversarial training. When combined with existing U-net models for AMT, ReconVAT achieves competitive results on common benchmark datasets such as MAPS and MusicNet. For example, in the few-shot setting for the string part version of MusicNet, ReconVAT achieves F1-scores of 61.0% and 41.6% for the note-wise and note-with-offset-wise metrics respectively, which translates into an improvement of 22.2% and 62.5% compared to the supervised baseline model. Our proposed framework also demonstrates the potential of continual learning on new data, which could be useful in real-world applications whereby new data is constantly available.

CCS CONCEPTS

• **Applied computing** → **Sound and music computing**; • **Computing methodologies** → **Semi-supervised learning settings**; **Neural networks**.

KEYWORDS

semi-supervised training, virtual adversarial training, audio processing, automatic music transcription, music information retrieval

ACM Reference Format:
Kin Wai Cheuk, Dorien Herremans, and Li Su. 2021. ReconVAT: A Semi-Supervised Automatic Music Transcription Framework for Low-Resource Real-World Data. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3475405

*Also with Institute of High Performance Computing (IHPC), Agency for Science, Technology and Research (A*STAR).

MM '21, October 20–24, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475405

1 INTRODUCTION

Automatic Music Transcription (AMT), is a fundamental problem in the field of Music Information Retrieval (MIR). According to the definition from the field of Music Information Retrieval (MIR) [3], AMT aims at transcribing music audio files into symbolic representations such as piano rolls [7] or music scores [6, 38, 39], which is very similar to Automatic Speech Recognition (ASR) [1, 8]. These symbolic representations have a wide range of applications including music indexing [15, 42], music generation [20], music recommendation system (MRS) [9, 32, 48], music analysis [22, 26, 30], and automatic music accompaniment [31].

Recent advances in fully supervised deep learning have enabled AMT models [18, 19, 23, 24] to achieve state-of-the-art performance for solo piano pieces, given sufficient labelled training data. While acoustic audio recordings as well as the aligned midi labels for piano music can be easily obtained by using a hybrid acoustic/midi piano such as the Yamaha Disklavier [16], this is not the case for other musical instruments such as violin and clarinet. At the time of writing, hybrid versions of these musical instruments are still not available. They are either midi controllers that lack the capability to produce realistic acoustic sound, or fully acoustic instruments without the capability to record the midi annotation in real-time. Therefore, the paired acoustic recordings and midi annotations for these instruments are very expensive to obtain, and hence, very limited. Supervised models fail to function well for these instruments.

Self-supervised or semi-supervised learning is an underexplored area in AMT. Existing unsupervised models have only been applied to specific musical instruments. For example, Berg-Kirkpatrick et al. [5] proposed an unsupervised graphical model using prerecorded key-wise piano samples to reconstruct the original signal. Upon successful reconstruction, the model could infer the transcription result via both the predicted onset locations and the selected piano samples for reconstructing the spectrogram. Choi and Cho [14] also employed a similar approach in their unsupervised drum transcription model. This approach, however, only works when the musical instrument is a percussion or plucked instrument type with a clear transient, immediately followed by a natural decay (piano is considered as a percussion instrument due to the hammering mechanism). Musical instruments which produce an increasing or fluctuating amplitude after the transient are unable to be represented by a fixed audio sample, and hence cannot be properly transcribed using the above-mentioned approach. Examples of such instruments include string instruments that are capable of starting a long note softly

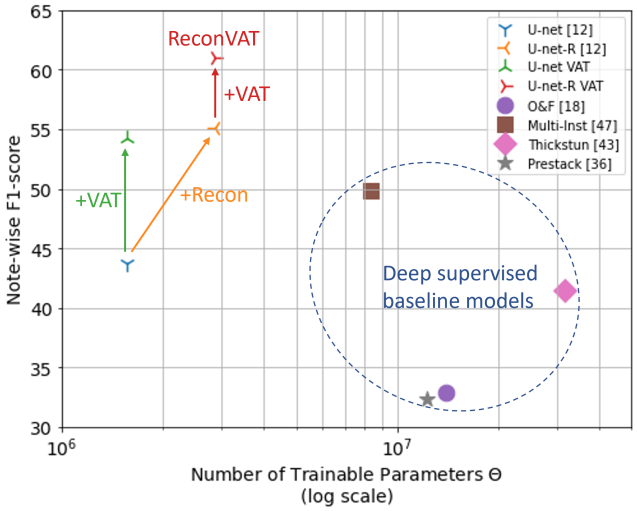

Figure 1: A scatter plot showing the note-wise F1-score and the number of model parameters for different models trained on the string version of MusicNet (see Section 4.2 for details).

followed by a crescendo through gradually increasing the bow pressure; and woodwind instruments that can sustain a note as long as the player's lung capacity can handle.

In this paper, we propose a semi-supervised AMT framework, which we refer to as **ReconVAT**, based on the idea of spectrogram reconstruction [12] and virtual adversarial training (VAT) [34]. ReconVAT leverages unlabelled data to improve the transcription accuracy with only a limited amount of labelled data, and it works well on various musical instruments such as piano, string instruments, as well as woodwind instruments. We also show that our framework has important applications such as continual learning with new unlabelled recordings, and being able to transcribe music genres that are outside of the labelled training set. More importantly, all of these can be achieved with only a small number of model parameters compared to existing deep learning models for AMT as shown in Figure 1. This makes our framework attractive and practically usable for real-world applications deployed on mobile devices. To the best of our knowledge, this is the first semi-supervised deep learning framework for instrument-agnostic AMT at the time of writing.

The contributions of this paper can be summarized as follows:

- We propose a semi-supervised framework for AMT that generalizes well across *different kinds of musical instruments*.
- We leverage existing models by integrating them into the proposed semi-supervised framework to achieve state-of-the-art transcription accuracy for low-resource scenario.
- We demonstrate possible applications in continual learning on music genres that are not present in the train set.

2 METHOD

In this section, we will formulate automatic music transcription (AMT) mathematically. Then we will introduce related work, and describe how to combine both spectrogram reconstruction [12] and VAT [34] to be our proposed semi-supervised framework ReconVAT.

2.1 Problem Definition

The goal of AMT is to convert audio data into symbolic music data [3, 4]. In this paper, we consider the case of converting spectrograms into piano rolls. Given a normalized input spectrogram $X_{\text{spec}} \in [0,1]^{T\times F}$, where T is the number of timesteps and F is the number of frequency bins, we want to have a model $p(Y_{\text{post}}|X_{\text{spec}},\theta)$, with a set of trainable parameters θ, that infers the posteriorgram $Y_{\text{post}} \in [0,1]^{T\times N}$. Here N is the note range for the musical instrument, for example, $N = 88$ for piano transcription since there are 88 keys on the keyboard. The ground truth piano roll $Y_{\text{roll}} \in \{0,1\}^{T\times N}$ is the symbolic notation we want to predict. This is done by simply applying a threshold (e.g. 0.5) to the Y_{post}.

2.2 Spectrogram Reconstruction

Cheuk et al. [12] proposed a model consisting of a transcriber $p(Y_{\text{post}}|X_{\text{spec}},\theta)$ and a reconstructor $q(X_{\text{recon}}|Y_{\text{post}},\phi)$. The reconstructor uses the posteriorgram generated from the transcriber as input to reconstruct the spectrograms X_{recon}. Therefore, in addition to the transcription loss $L_{\text{trans}}(Y_{\text{post}}, Y_{\text{label}})$, there is also a reconstruction $L_{\text{recon}}(X_{\text{recon}}, X_{\text{spec}})$ to be minimized.

The reconstructed spectrograms X_{recon} are then used to train the same transcriber $p(Y'_{\text{post}}|X_{\text{recon}},\theta)$ again, resulting in one extra transcription loss $L_{\text{trans}}(Y'_{\text{post}}, Y_{\text{label}})$. Cheuk et al. [12] has shown that training the model in this manner results in a consistently better model. Their reported results, however, are unable to beat the state-of-the-art AMT models. We will show in Table 1 that their model can be modified to compete with the state-of-the-art AMT models. Although it is not demonstrated in their paper, they also claim that their model has the potential to be trained in an unsupervised manner. We will therefore also show in Section 2.4 that when combined with virtual adversarial training [34], we can modify the spectrogram reconstruction framework [12] to be a semi-supervised model for AMT.

2.3 Virtual Adversarial Training

Virtual adversarial training (VAT), as presented by Miyato et al. [34] is an extended version of adversarial training (AT) proposed by Goodfellow et al. [17]. In AT, labels are required to calculate the the adversarial vectors. In the case where we do not have access to the labels, Miyato et al. [34] proved that the adversarial vector can be obtained via Equation (1):

$$r_{\text{adv}}^{\text{VAT}} = \epsilon \left(\nabla_r D \left[p(Y_{\text{pred}}|X,\theta), p(Y_{\text{adv}}|X+r,\hat{\theta}) \right] \right), \qquad (1)$$

where r is a randomly initialized noise vector, and Y_{adv} is the output obtained using the adversarial input $X_{\text{adv}} = X + r$.

By doing so, it is possible to perform adversarial training using unlabelled datasets. Most existing literature applies VAT to static classifications [1, 8, 27, 28, 33, 35]. While SeqVAT [10] is designed for sequential labelling, it is a one-hot prediction system. To the best of our knowledge, ReconVAT is the first framework capable of multi-hot sequential labelling for polyphonic AMT. In the next section,

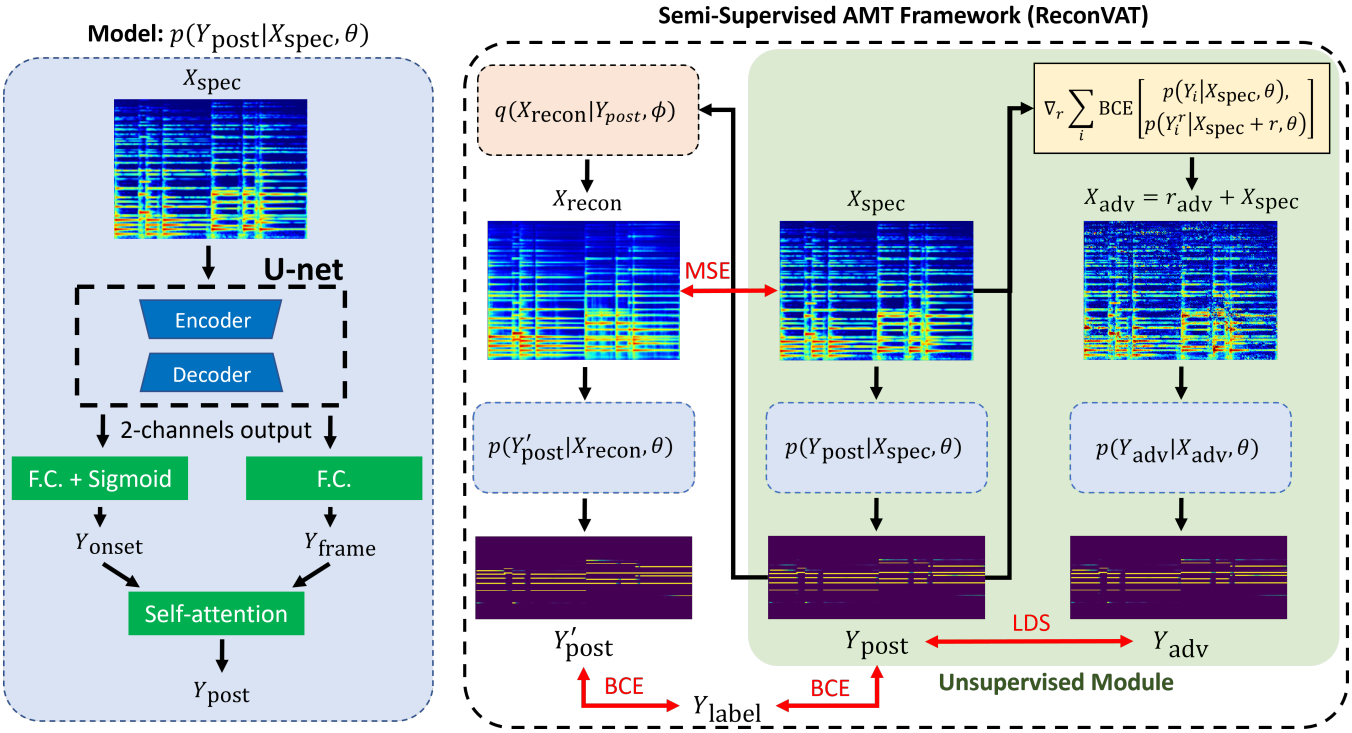

Figure 2: The left-hand side of the figure shows the modified version of the AMT model proposed in [12] such that it supports onset prediction. The right-hand side of the figure shows the proposed framework for semi-supervised AMT. For simplicity, we omit the onset prediction in the figure by showing the case when $i = \{\text{post}\}$. The region highlighted in green is the unsupervised module which supports training not only on labelled samples, but also on unlabelled samples.

we will describe how to combine both spectrogram reconstruction and VAT to obtain a semi-supervised framework for AMT.

2.4 Proposed Framework – ReconVAT

The left-hand side of Figure 2 shows our modified version of the model proposed by Cheuk et al. [12]. We improve it by introducing a two-channel output, one channel for the onset prediction Y_{onset}, and another channel for the frame feature extraction Y_{frame}. The posteriorgram is obtained from a self-attention layer which takes the concatenation of Y_{onset} and Y_{frame} as the input. This modification increases the model's flexibility. For example, if we want to also include the offset prediction Y_{offset}, we can have a three-channel output instead. For simplicity, we will only explore the case of one-channel (Y_{frame}) and two-channel (Y_{frame} and Y_{onset}) prediction in this paper. The implementation details will be discussed in Section 3.4.

The right-hand side of Figure 2 shows our proposed ReconVAT. It consists of three branches. The framework starts with the middle branch where it takes X_{spec} as the input and outputs a posteriorgram Y_{post}. The branch on the left then takes the Y_{post} as its input and generates a reconstructed spectrogram X_{recon} using the reconstructor $q(X_{\text{recon}}|Y_{\text{post}},\phi)$ mentioned in 2.2. The reconstructed spectrogram X_{recon} is passed to the same model again to obtain another posteriorgram Y'_{post}. The two posteriorgrams Y_{post} and Y'_{post} should be as close to the label Y_{label} as possible.

The branch on the right-hand side is the unsupervised module which uses VAT. To obtain the adversarial spectrogram X_{adv}, we apply a modified version of VAT that works better for AMT (Section 3.4). Using this adversarial spectrogram, we obtain another posteriorgram Y_{post} via the same model.

For labelled spectrograms, all three branches are used. For unlabelled spectrograms, we only use the middle and the right branches (highlighted in green in Figure 2). This framework is trained by minimizing both the supervised loss L_{l} (Equation 5) and the unsupervised loss L_{ul} which will be discussed in detail in Section 3.5.

3 EXPERIMENTS

In this section, we describe the datasets and the experiments for demonstrating the power of our proposed semi-supervised framework ReconVAT.

3.1 MAPS dataset

The MAPS dataset [16] consists of nine folders, each folder contains 30 full-length midi recordings. In seven of these folders, the audio recordings are synthesized from the midi annotations using different virtual piano software such as Steinberg, Native Instruments, and Sampletekk. Only in the folders `ENSTDkAm` and `ENSTDkCl`, the audio recordings are recorded simultaneously with the midi recordings using a Yamaha Disklavier. We follow the existing consensus [12, 18, 23, 36, 41] that the seven folders containing artificially

generated audio recordings should be used as the training set, and the other two folders, `ENSTDkAm` and `ENSTDkCl`, as the test set.

Since some music pieces appear in both the training and the test set, we follow the existing literature [18, 41] to remove overlapping songs from the training set that are also present in the test set, thus reducing the size of the training set from 210 music pieces down to 139 pieces. Following existing conventions [12, 18, 19, 41], all audio recordings are downsampled from 44.1 kHz to 16 kHz.

To demonstrate the effectiveness of our VAT model, we train our model using the following three versions of the MAPS dataset:

3.1.1 Full version. This version uses all 139 available pieces from MAPS as the labelled training set. To demonstrate the ability of leveraging unlabelled data using our VAT model, we use the training set from MAESTRO [19] as the unlabelled dataset (967 music recordings). The labelled training batch size N_{l} and the unlabelled training batch size N_{ul} are both 8.

3.1.2 Small version. In this version, only one folder (`AkPnBcht`, containing 23 non-overlapping songs) from MAPS is used as the labelled training set. We keep using the same 967 music recordings from MAESTRO as our unlabelled set for our VAT model. Again, N_{l} and N_{ul} are both 8 in this version.

3.1.3 One-shot version. Only one music recording (`chp_op31` from the `AkPnBcht` folder) is used as the labelled training set. The unlabelled set consists of the same 967 music recordings from MAESTRO as the above two versions. Due to the fact that there is only one labelled training sample, N_{l} is 1 and the unlabelled training batch size N_{ul} remains 8.

3.2 MusicNet dataset

MusicNet [44] contains both audio recordings and annotations of various types of musical instruments such as those from the string family and the woodwind family. To prove that our model also works for different types of musical instruments, we perform our experiments on the following variations of MusicNet:

3.2.1 String version. In the official training set provided by MusicNet, there are 8 genres of music that contain string instruments. We select only one piece from each genre from the official training set, forming our own labelled training set. The remaining pieces of each genre are used as the unlabelled training set for our VAT framework. By doing so, there are eight labelled samples and 104 unlabelled samples in our training set. We pick four string pieces from the official test set provided by MusicNet as our test set. The IDs of the four pieces are 2191, 2628, 2106, and 2298 which are solo violin, accompanied violin, string quartet, and solo cello respectively. More details about the data splitting can be found in the supplementary material[2].

The labelled training batch size N_{l} and the unlabelled training batch size N_{ul} are both 8.

3.2.2 Woodwind version. Similar to the string version, we pick only one piece from six different woodwind genres from MusicNet as the labelled training set and use the remaining pieces in each genre as the unlabelled training set. This results in six labelled training samples and 21 unlabelled training samples. The official test set provided by MusicNet contains only two pieces (1819, 2416) from the woodwind family, which belong to the Pairs Clarinet-Horn-Bassoon genre. We use these two pieces as our test set. Again, more details can be found supplementary material[2]. N_{l} is 1 and N_{ul} is 8 in this version.

3.3 Data Processing

We extract Mel spectrograms on-the-fly from the audio clips using a GPU-based audio processing library nnAudio [11]. Following Hawthorne et al. [18], we use a Hann window size of 2,048, a hop size of 512, and 229 Mel bins as the parameters of our Mel spectrograms X_{spec}. To extract a fixed length spectrogram, we crop the audio clips into segments of 327,680 sample points using random sampling during each iteration, which results in Mel spectrogram with 640 timesteps, and 229 Mel frequency bins. We compress the magnitude of the spectrograms by taking the natural logarithm and then normalizing the magnitude for each spectrogram into the range $[0, 1]$. i.e. $X_{\text{spec}} \in [0, 1]^{640\times 229}$.

As for our ground truth labels, we extract the onset, duration, and pitch information from the midi annotations to produce tsv files for the ground truth. These tsv files are read and converted into piano rolls in the form of a binary matrix $Y_{\text{label}} \in \{0, 1\}^{640,F}$. Since most musical instruments in the dataset are within the 88 notes range (note A0 to note C8), we use $F = 88$ in all our experiments.

3.4 Implementation Details

All models and experiments, including the baseline models, are implemented in PyTorch. To ensure transparency and fairness, we train all our models without tricks such as label smoothing [47], weighted cross entropy [18], and focal loss [29, 46]. We believe that these tricks would in general improve the the transcription accuracy, and it is beyond the scope of this paper to explore this.

We adopt U-net models specifically designed for pitch detection [12, 21] and integrate them into the VAT framework [34]. While we follow mostly the same design as in [12], we modify the final layer of the decoder so that it has the flexibility to output two channels as shown in Figure 2. One of the channels is fed to a fully connected layer with sigmoid activation to predict the onsets $Y_{\text{onset}} \in [0, 1]^{T\times F}$, and the other channel is fed to a linear fully connected layer to obtain the features $Y_{\text{frame}} \in \mathbb{R}^{T\times F}$. The concatenated output $Y_{\text{onset}} \bigoplus Y_{\text{frame}}$ is fed to a relative local 1D self-attention layer [37, 40] to obtain the posteriorgram $Y_{\text{post}} \in [0, 1]^{T\times 88}$. We binarize the posteriorgram with a threshold of 0.5 to obtain the predicted piano roll $Y_{\text{roll}} \in \{0, 1\}^{T\times 88}$. If the two-channel output is used, we follow the inference method from the Onsets and Frames model [18] to obtain a refined piano roll by using both Y_{onset} and Y_{frame} to filter out notes that do not have a onset. Otherwise, we directly use the posteriorgram to obtain the piano roll. In addition, we also replace all of the LSTM layers in [12] with local relative self-attention layers, since it has been shown that self-attention layers perform as good as LSTM layers while providing the extra benefit of being able to train in parallel [45, 47].

We also modify the original VAT method [34] so that it works better for AMT. Firstly, since polyphonic AMT is a timestep-wise multiclass classification problem (multiple pitches can occur at the same time), we replace the Kullback–Leibler divergence (KL-div)

with binary cross entropy (BCE) when calculating the local distributional smoothness (LDS). Secondly, we normalise the adversarial vector r_{adv} along the timestep dimension as shown in Equation (2):

$$r_{\text{adv}} = \epsilon \left[\frac{g_1}{\|g_1\|_2}, \frac{g_2}{\|g_2\|_2}, \cdots, \frac{g_T}{\|g_T\|_2} \right] \tag{2}$$

where ϵ is a parameter that controls the magnitude of the adversarial vector r_{adv}, and g_t for $1 \leq t \leq T$ is the timestep-wise gradient obtained from Equation (3)

$$g = \nabla_r \sum_i \text{BCE}\left[p(Y_{\text{i}}|X_{\text{spec}}, \hat{\theta}), p(Y_{\text{adv}}|X_{\text{spec}} + r, \hat{\theta}) \right] . \tag{3}$$

If the onsets prediction module is included, then $i = \{\text{onset}, \text{post}\}$. Otherwise, there is only one term in Equation (3), i.e. $i = \{\text{post}\}$. As in [34], the weight of the model is considered as a constant $\hat{\theta}$ when calculating the gradient g.

Once we obtain the adversarial vector r_{adv}, we can calculate the LDS. By the same logic as above, the LDS can contain either one or two terms depending on the model output:

$$\text{LDS}_* = \frac{\sum_i \text{BCE}\left[p(Y_{\text{i}}|X^*_{\text{spec}}, \hat{\theta}), p(Y_{\text{adv}}|X^*_{\text{spec}} + r_{\text{adv}}, \theta) \right]}{N_*} . \tag{4}$$

From Equation 4, we can see that the label Y_i^{label} is not required to calculate the LDS. Therefore, LDS is an unsupervised loss that can be calculated using both labelled spectrograms $X^{\text{l}}_{\text{spec}}$ and unlabelled spectrograms $X^{\text{ul}}_{\text{spec}}$. We will denote the LDS calculated using $X^{\text{l}}_{\text{spec}}$ as LDS_{l} and the LDS calculated using $X^{\text{ul}}_{\text{spec}}$ as LDS_{ul}. Unlike the original VAT [34], we normalise LDS_{l} and LDS_{ul} by its respective batch size N_{l} and N_{ul}, rather than summing both LDS_{l} and LDS_{ul} together and normalize with $N_{\text{l}} + N_{\text{ul}}$. By doing so, we prevent N_{ul} from interfering with LDS_{l} and N_{l} from interfering with LDS_{ul}.

3.5 Training Objective and Optimization

As mentioned in Section 3.4, we have the supervised objective L_{l} that requires labels, and the unsupervised objective L_{ul} that does not require any label. The final objective L being minimized during training contains three terms as shown in Equation (7):

$$L_{\text{l}} = \sum_i \text{BCE}\left[Y_i, Y_i^{\text{label}} \right] + \sum_i \text{BCE}\left[Y_i^{\text{recon}}, Y_i^{\text{label}} \right] \tag{5}$$

$$L_{\text{ul}} = \frac{\text{LDS}_{\text{l}} + \text{LDS}_{\text{ul}}}{2} \tag{6}$$

$$L = L_{\text{l}} + \alpha L_{\text{ul}} + L_{\text{recon}} \tag{7}$$

where α is the weighting for L_{ul}, which is set to 1 throughout all our experiments; L_{recon} is the reconstruction loss mentioned in Section 2.2. We observe the same model behaviour as reported in [34], that is, controlling the ϵ in Equation (2) alone is sufficient to control the model performance without the need to change α.

To minimize the objective L, we use Adam [25] optimizer with a learning rate of 0.001 and a learning rate decay of 2% every 1,000 iterations. When training, our framework includes three forward passes during each iteration. One forward pass for L_{l}, one forward pass for LDS_{l}, and one forward pass for LDS_{ul}. We define one epoch as 10 iterations. During the parameter search, we split our training set into 80% for training and 20% for validating. The optimal value for ϵ in Equation (2 is mostly within the range between 1 and 2, and depends on the model architecture and the dataset. This value can be easily obtained after a few trials.

3.6 Evaluation Metrics

Following existing literature [12, 13, 18, 19, 23, 24], we report the frame-wise, note-wise, and note-with-offset-wise metrics to evaluate our model performance comprehensively. For note-wise metric, we use a onset tolerance of 50ms; for note-with-offset-wise metric, we use an offset tolerance of 50ms or 20% of the note duration, whichever is larger [2]. Readers are referred to Cheuk et al. [13] which explains the differences between these metrics in detail in their Section IV-C. In our experiments, we use the implementations from `mir_eval`[1] to calculate and report the above-mentioned metrics.

4 RESULTS

4.1 Effectiveness of VAT

We compare our proposed models to the Onsets and Frames model [18] and the Multi-Instrument AMT model [47] as they show good performance on the MAPS and MusicNet datasets respectively. We exclude the models proposed by Pedersoli et al. [36] and Thickstun et al. [43] in our results below since their performance is worse than the Multi-Instrument AMT model [47]. In Table 1-2, we use R to represent the reconstruction module, and O to represent the onset module. Therefore U-net-RO means that the U-net model contains both a reconstruction and onset module. The columns of the tables represent the precision (P), recall (R), and F1-score for each of the metrics mentioned in Section 3.6. Our proposed models and the baseline models are trained on the same labelled data, and only the proposed semi-supervised models are able to leverage the unlabelled data mentioned in Section 3.

4.1.1 Full MAPS. We can see that when using the VAT (row A3-A4, A7, A8), all three metrics generally improve compared to their respective counterparts without the VAT (row A1-A2, A5, A6). When using onset inference (A5-A8), the note-wise and note-with-offset-wise metrics are improved by at least 7 percentage points. The model using both the onset inference as well as our proposed framework (row A8) performs as good as the state-of-the-art Onsets and Frames model [18] (row 9) for this dataset.

4.1.2 Small MAPS. The middle part of Table 1 shows that when the number of labelled training samples is reduced by over 80% from 139 to 23 audio clips, the advantage of the VAT module becomes more obvious. Similar to the full MAPS dataset, the models with VAT module outperform their counterparts that do not use VAT. Moreover, our proposed framework (row B8) outperforms the Onsets and Frames model (B9) by 6, 5.1, 4.4 percentage points in terms of frame-wise, note-wise, and note-with-offset-wise F1-scores, which can be translated into improvements in performance of 11.5%, 8.1%, and 14.1% respectively.

4.1.3 One-shot MAPS. The bottom part of Table 1 shows that when we reduce the number of labelled training audio clip even further to only one, our proposed framework (C8) outperforms the Onsets and frames model (C9) by 23.7, 17, and 12.9 percentage points.

[1] `https://github.com/craffel/mir_eval`

Between the models that use and do not use onset inference, we can see that onset inference has the tendency of decreasing the frame-wise F1-score while improving the note-wise F1-scores. This is due to the unreliability of the frame-wise metric [13, 18]. Cheuk et al. [13] has provided a few examples and shown that a high frame-wise score does not guarantee a good transcription. Nonetheless, these three experiments have shown that VAT is a very effective semi-supervised method, that allows the use of unlabelled training samples to greatly improve the model performance in cases where the number of labelled samples is scarce.

4.1.4 String MusicNet. The top section of Table 2 shows the performance of our proposed framework on the string subset of MusicNet (3.2.1). Although most models with the VAT outperform their counterparts without the VAT, U-net-RO VAT on row D4 performs worse than its counterpart without the VAT. Moreover, using the onset inference (row D1-4 and D9) does not improve the transcription accuracy in this setting, on the contrary, it worsens the model performance. There are two possible reasons for this. First, we believe that the onset inference only works well for piano only, and it cannot generalize well to other musical instruments such as those from the string and the woodwind family. Second, we believe that the onset labels for MusicNet are not completely accurate, since the labels are generated using dynamic time warping (DTW) [43]. Therefore, inaccurate onset labels might confuse the VAT. Using no labels might be better than using inaccurate labels, which is one of the advantages of using VAT.

Now, let us consider models (row D5-D8) that do not use onset inference. We will use the Multi-Instrument AMT model (row D10) [47], which is the state-of-the-art model for the MusicNet dataset at the time of writing, as the baseline model. Since the baseline model [47] is much deeper than the U-net model (row D5), it outperforms the U-net model. By applying the reconstruction module to the U-net model (row D6), the U-net model begins to outperform the baseline model. When we further apply VAT to the U-net models (row D7-D8), the transcription accuracy becomes even better. The best model, U-net-R VAT (row D8), outperforms the baseline model by 3.9, 11.1, and 11.3 percentage points in terms of frame-wise, note-wise, and note-with-offset-wise metrics.

4.1.5 Woodwind MusicNet. The bottom section of Table 2 shows the results for the woodwind subset of MusicNet (Section 3.2.2). Since the Onsets and Frames model does not work well for this dataset either, we did not spend time experimenting with it. Just like all of the results reported above, the VAT module is very effective in improving the transcription accuracy. The best model being the one with both the reconstruction and the VAT module (row E4), and it outperforms the baseline model by 10.3, 6.9 percentage points in terms of note-wise and note-with-offset-wise metrics. The improvement for the frame-wise metrics is not obvious, however, we must keep in mind that this is not a reliable metric to evaluate the transcription accuracy as pointed out previously in Section 3.6 as well as existing literature [13, 18].

4.2 Model Compactness

A comparison of number of trainable model parameters for the baseline models and the proposed models is shown in Figure 1. It can be seen from the figure that a deep model does not necessarily yield a high transcription accuracy when the labelled training data is limited. The Onsets and Frames model [18] and the Prestack-Unet [36] have a high number of parameters, yet they do not perform well when the labelled data is scarce. While Thickstun's model [43] performs better than the two baseline models, its number of parameters is 10 times more than our proposed framework (U-net-R VAT). We use the Resnet-18 version of Prestack-Unet since the Resnet-32 version is too huge to run on our GPU. Another baseline model, the Multi-Instrument AMT model, performs better than the plain U-net model. With VAT, however, the U-net models already outperform the baseline model while keeping the number of trainable parameters low. We can also see that the VAT improves the model performance without adding extra parameters to the model. Therefore, VAT is a very effective method to improve the transcription accuracy by leveraging unlabelled training data when the labelled training data is limited.

5 APPLICATIONS

Our proposed semi-supervised framework allows for two important applications: continual learning and knowledge transfer to unseen music genres. We will discuss these two properties and their potential applications.

5.1 Continual Learning

The loss function of our proposed semi-supervised AMT framework contains a supervised term L_{l} and an unsupervised term L_{ul}. Even when we encounter new unseen, unlabelled data, we can still use this new data to minimize the unsupervised part of the model L_{ul}. That means, the proposed model can be *retrained* with any new data that was not collected before. Therefore, our proposed framework is capable of improving itself via new unlabelled data.

To confirm our framework's ability of continual learning, we take the string and woodwind subset of MusicNet as an example. We first train our models for 4,000 epochs (row 1 and 4 of Table 3, denoted as "4k"), and save the weights. These weights are then used as starting weights when we train the model for another 4,000 epochs with two different conditions: (1) without new data (row 2 and 5, denoted as "8k"); (2) using the test data as the unlabelled data as well as the existing data (row 3 and 6, denoted as "4k + 4k"). For the string subset of MusicNet, the model has already converged at 4,000 epochs, additional supervised training does not change the performance much. When we include the test data as the unlabelled data, it further pushes the accuracy around 1 percentage point higher. The same goes for the woodwind dataset. Although the improvement is relatively subtle at the moment, we plan to investigate ways to further improve this in future research. This property leads to the next application.

5.2 Case Study: Transcribing Unseen Genres

In some cases, we have some labels in one data domain, while the target domain we are interested in might not contain any labels at all. A model that can be trained on one domain and its knowledge then transferred to the target domain will be very useful. For example, we have some labelled data for classical woodwind music, but we want our model to be able to transcribe clarinet covers of Japanese pop

Table 1: Transcription accuracy (mean + STD) in terms of different metrics and their respective precision (P), recall (R), and F1-score (F1) when training on different variations of the MAPS dataset. The accuracy values are averaged over the test clips in the dataset.

		Frame			Note			Note w/ offset		
	Full	P	R	F1	P	R	F1	P	R	F1
A1	U-net [12]	84.6 ± 6.0	70.8 ± 8.9	76.7 ± 6.5	55.8 ± 12.6	62.0 ± 11.9	58.4 ± 11.7	34.5 ± 11.3	38.5 ± 12.0	36.2 ± 11.4
A2	U-net-R [12]	86.2 ± 6.2	72.7 ± 10.0	78.4 ± 7.0	68.5 ± 10.5	61.0 ± 13.1	64.2 ± 11.4	45.5 ± 11.1	40.8 ± 12.9	42.8 ± 11.9
A3	U-net VAT	86.6 ± 5.4	71.5 ± 9.4	77.9 ± 6.7	64.5 ± 13.2	64.2 ± 12.6	64.0 ± 12.3	40.8 ± 11.6	40.9 ± 12.1	40.6 ± 11.5
A4	U-net-R VAT	88.8 ± 6.0	**72.7 ± 9.0**	**79.5 ± 6.5**	74.0 ± 9.3	63.3 ± 13.3	67.9 ± 11.2	49.7 ± 10.1	42.9 ± 12.5	45.8 ± 11.3
A5	U-net-O	89.6 ± 6.0	58.8 ± 9.9	70.4 ± 7.5	85.8 ± 7.8	66.3 ± 11.0	74.5 ± 9.2	53.1 ± 9.5	41.5 ± 11.3	46.4 ± 10.6
A6	U-net-RO	89.9 ± 6.6	60.4 ± 10.8	71.6 ± 8.3	86.1 ± 7.9	67.3 ± 11.2	75.2 ± 9.2	52.8 ± 10.1	41.7 ± 11.6	46.4 ± 10.9
A7	U-net-O VAT	**90.9 ± 6.1**	60.5 ± 9.5	72.2 ± 7.5	**89.8 ± 8.3**	65.0 ± 11.1	75.1 ± 9.5	**58.7 ± 9.9**	42.9 ± 11.3	49.4 ± 10.8
A8	U-net-RO VAT	85.9 ± 7.2	72.0 ± 8.7	77.9 ± 6.5	80.9 ± 7.0	70.6 ± 11.2	75.1 ± 8.6	54.3 ± 9.8	**47.6 ± 11.8**	**50.5 ± 10.6**
A9	O&F [18]	89.3 ± 6.4	65.6 ± 9.7	75.2 ± 7.3	85.2 ± 7.8	**73.3 ± 11.4**	**78.6 ± 9.3**	53.8 ± 9.8	46.7 ± 12.0	49.8 ± 10.9
	Small	P	R	F1	P	R	F1	P	R	F1
B1	U-net [12]	75.4 ± 6.6	57.1 ± 9.6	64.5 ± 7.3	35.4 ± 8.3	57.5 ± 11.6	43.5 ± 8.9	17.0 ± 6.9	27.6 ± 10.6	20.9 ± 8.0
B2	U-net-R [12]	81.2 ± 6.1	**61.1 ± 11.1**	**69.1 ± 8.1**	51.0 ± 10.3	**61.3 ± 12.4**	55.3 ± 10.6	26.2 ± 9.8	**31.8 ± 12.4**	28.6 ± 10.7
B3	U-net VAT	79.1 ± 6.6	56.9 ± 11.7	65.4 ± 8.7	52.4 ± 11.9	60.1 ± 12.6	55.5 ± 11.3	26.2 ± 9.8	30.3 ± 11.6	27.8 ± 10.3
B4	U-net-R VAT	79.7 ± 6.1	59.9 ± 11.0	67.7 ± 7.8	57.2 ± 11.9	61.0 ± 12.0	58.6 ± 11.1	29.2 ± 10.3	31.3 ± 11.3	30.0 ± 10.5
B5	U-net-O	88.3 ± 6.3	38.4 ± 10.2	52.6 ± 9.8	81.9 ± 7.9	50.1 ± 11.3	61.6 ± 9.8	41.0 ± 10.1	25.6 ± 10.1	31.2 ± 10.2
B6	U-net-RO	88.0 ± 6.3	44.9 ± 11.2	58.5 ± 10.1	83.4 ± 8.7	55.8 ± 12.2	66.3 ± 10.3	42.8 ± 10.3	29.2 ± 10.7	34.4 ± 10.6
B7	U-net-O VAT	89.5 ± 6.6	41.0 ± 10.5	55.4 ± 10.0	**86.8 ± 8.8**	52.3 ± 12.1	64.7 ± 10.7	44.0 ± 10.8	27.0 ± 10.4	33.1 ± 10.7
B8	U-net-RO VAT	**90.0 ± 6.2**	43.9 ± 10.6	58.2 ± 9.7	86.2 ± 8.6	57.1 ± 11.5	**68.2 ± 10.0**	**44.6 ± 11.7**	30.0 ± 11.0	**35.6 ± 11.3**
B9	O&F [18]	89.7 ± 5.8	37.7 ± 10.4	52.2 ± 10.4	85.3 ± 8.6	51.2 ± 13.1	63.1 ± 11.4	41.8 ± 10.5	25.5 ± 10.2	31.2 ± 10.4
	One-shot	P	R	F1	P	R	F1	P	R	F1
C1	U-net [12]	61.9 ± 5.8	41.8 ± 9.1	49.2 ± 6.9	24.7 ± 7.6	49.6 ± 10.6	32.2 ± 7.9	8.8 ± 5.6	17.0 ± 8.5	11.3 ± 6.5
C2	U-net-R [12]	74.8 ± 5.9	41.2 ± 10.4	52.2 ± 8.5	33.6 ± 9.4	54.5 ± 11.5	40.7 ± 8.8	12.5 ± 7.2	19.8 ± 10.0	15.0 ± 8.1
C3	U-net VAT	75.6 ± 6.6	**46.1 ± 11.1**	**56.2 ± 9.0**	42.7 ± 11.0	55.2 ± 12.7	47.1 ± 9.6	17.1 ± 7.8	22.1 ± 9.7	18.9 ± 8.1
C4	U-net-R VAT	71.6 ± 6.0	43.6 ± 9.5	53.3 ± 7.3	36.4 ± 8.5	**61.5 ± 12.0**	45.2 ± 8.5	13.6 ± 6.6	**22.8 ± 10.4**	16.8 ± 7.7
C5	U-net-O	87.7 ± 7.2	17.4 ± 7.0	28.4 ± 9.3	**85.7 ± 6.6**	27.7 ± 10.0	40.9 ± 11.3	**34.5 ± 10.6**	11.4 ± 6.1	16.8 ± 7.7
C6	U-net-RO	87.7 ± 7.5	21.4 ± 9.0	33.4 ± 11.1	82.0 ± 7.1	33.4 ± 11.7	46.3 ± 11.9	34.4 ± 10.0	14.6 ± 7.9	20.0 ± 9.0
C7	U-net-O VAT	72.9 ± 9.4	22.9 ± 8.8	33.5 ± 9.1	57.9 ± 10.2	43.6 ± 12.4	47.9 ± 7.6	17.1 ± 7.8	13.0 ± 7.0	14.2 ± 6.8
C8	U-net-RO VAT	86.1 ± 6.8	31.4 ± 9.8	45.0 ± 10.3	77.2 ± 9.8	51.5 ± 12.4	**60.7 ± 9.6**	31.4 ± 10.4	21.1 ± 9.1	**24.8 ± 9.2**
C9	O&F [18]	**88.0 ± 6.8**	12.4 ± 4.8	21.3 ± 7.2	83.5 ± 8.2	30.5 ± 10.4	43.7 ± 11.2	23.1 ± 11.4	8.3 ± 4.7	11.9 ± 6.3

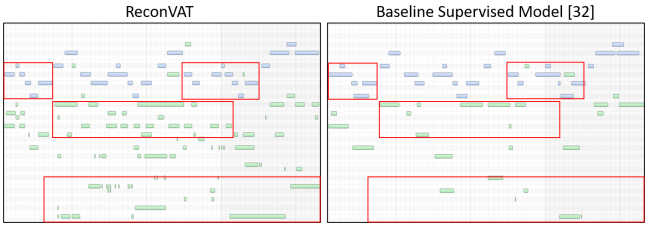

Figure 3: Transcribed piano rolls for J-pop. Blue notes indicate the main melody; green notes indicate the accompaniment (piano and drums). The red boxes indicate major differences in the piano rolls.

music. Our proposed framework, as shown in Figure 3, is capable of tackling this task.

We downloaded a few Japanese cover songs from YouTube, and we study the transcription results produced by both the "woodwind 4k + 4k" model reported in Table 3 and the best supervised baseline model [47] trained for 8,000 epochs using only labelled data. Due to page limitations, we only show one of the cover songs "Lemon" in Figure 3. More examples can be found in the demo page provided as part of the supplementary material[2]. Since Japanese Pop music is not included in the woodwind version of MusicNet data, the supervised baseline model trained on only MusicNet produces a piano roll with a lot of missing details such as the melody and the bass indicated by the red boxes. Training the supervised model for more epochs does not help as the loss already converged. Our proposed semi-supervised framework (ReconVAT) trained on both labelled and unlabelled data tries to capture more details than the fully supervised baseline model [47]. For example, in the upper part of the piano roll (Figure 3), the baseline model fails to capture the rhythmic patterns that are only found in pop music. Moreover, there are also rhythmic patterns for the piano accompaniment that are specific to pop music, and the baseline model failed to transcribe these unseen piano patterns (middle and lower part in Figure 3). One might argue that the transcription result for ReconVAT is noisier in the bass region (bottom part of the piano roll). This is due to the drum patterns in pop music. Since the labelled data in MusicNet does not contain any drum beat as the accompaniment, the baseline model simply ignores the drum sounds in the pop music. Our proposed model, however, is aware of the presence of the drum beats by training with the unlabelled pop music. It therefore

[2]https://kinwaicheuk.github.io/ReconVAT

Table 2: Transcription accuracy (mean ± STD) in terms of different metrics and their respective precision (P), recall (R), and F1-score (F1) when training on the different variations of the MusicNet dataset. The accuracy values are averaged over the test clips in the dataset.

		Frame			Note			Note w/ offset		
	Strings	P	R	F1	P	R	F1	P	R	F1
D1	U-net-O	70.0 ± 8.4	25.3 ± 13.4	35.7 ± 13.8	59.9 ± 10.7	24.6 ± 13.0	33.9 ± 13.6	35.4 ± 11.3	15.0 ± 9.9	20.5 ± 11.0
D2	U-net-RO	79.1 ± 1.1	39.3 ± 18.7	50.1 ± 16.6	68.5 ± 12.3	39.4 ± 17.2	48.6 ± 16.1	49.0 ± 17.1	29.9 ± 17.8	36.3 ± 18.5
D3	U-net-O VAT	65.2 ± 18.9	27.6 ± 13.5	38.2 ± 15.6	52.7 ± 14.2	27.8 ± 14.5	35.7 ± 14.9	31.5 ± 13.4	17.4 ± 12.6	22.0 ± 13.5
D4	U-net-RO VAT	78.0 ± 4.0	36.9 ± 10.5	49.5 ± 10.1	**69.6 ± 13.7**	37.6 ± 11.4	48.4 ± 11.9	**50.1 ± 18.3**	27.8 ± 13.4	35.4 ± 15.4
D5	U-net [12]	67.1 ± 6.9	50.7 ± 12.9	57.4 ± 10.5	41.8 ± 12.5	46.1 ± 12.7	43.7 ± 12.5	24.6 ± 13.4	26.7 ± 14.0	25.6 ± 13.7
D6	U-net-R [12]	71.7 ± 2.9	**62.8 ± 10.6**	66.6 ± 6.9	53.1 ± 10.7	57.4 ± 11.9	55.1 ± 11.1	37.6 ± 14.1	**40.9 ± 16.4**	39.1 ± 15.2
D7	U-net VAT	76.1 ± 7.2	52.4 ± 11.9	61.7 ± 10.2	58.1 ± 13.9	51.2 ± 15.1	54.2 ± 14.3	39.6 ± 17.8	35.4 ± 18.4	37.2 ± 18.1
D8	U-net-R VAT	**78.9 ± 4.8**	60.7 ± 9.8	**68.4 ± 7.7**	63.6 ± 13.8	**58.8 ± 14.3**	**61.0 ± 13.8**	43.3 ± 18.8	40.2 ± 18.9	**41.6 ± 18.7**
D9	O&F [18]	75.3 ± 3.1	22.5 ± 12.0	33.1 ± 14.6	69.0 ± 15.8	22.6 ± 12.1	32.9 ± 14.8	41.7 ± 18.2	15.3 ± 11.2	21.7 ± 14.5
D10	Multi-Inst [47]	71.4 ± 6.2	59.5 ± 13.1	64.5 ± 9.6	56.5 ± 16.7	45.1 ± 18.4	49.9 ± 17.8	33.7 ± 19.3	27.8 ± 18.4	30.3 ± 18.9
	Woodwinds	P	R	F1	P	R	F1	P	R	F1
E1	U-net [12]	62.6 ± 9.6	65.7 ± 2.9	63.5 ± 3.7	33.6 ± 2.4	39.9 ± 1.1	36.4 ± 1.0	11.6 ± 1.4	13.7 ± 0.3	12.5 ± 0.9
E2	U-net-R [12]	65.4 ± 6.7	71.4 ± 4.3	**67.8 ± 1.7**	40.5 ± 5.8	**52.5 ± 0.6**	45.4 ± 3.5	16.9 ± 5.2	21.3 ± 3.5	18.7 ± 4.7
E3	U-net VAT	69.0 ± 9.6	62.4 ± 1.2	65.1 ± 3.7	44.4 ± 3.1	40.5 ± 2.5	42.2 ± 0.0	16.1 ± 1.6	14.6 ± 0.5	15.3 ± 0.5
E4	U-net-R VAT	**69.8 ± 8.3**	65.8 ± 2.0	67.4 ± 2.8	**48.6 ± 3.5**	47.9 ± 1.1	**48.2 ± 1.2**	**22.1 ± 3.1**	**21.7 ± 1.0**	**21.8 ± 2.0**
E5	Multi-Inst [47]	64.4 ± 9.4	**71.8 ± 2.7**	67.3 ± 4.1	43.5 ± 2.8	33.6 ± 0.6	37.9 ± 0.7	17.1 ± 2.5	13.2 ± 0.9	14.9 ± 1.5

Table 3: Continual learning of the AMT model on the String and Woodwind subsets of MusicNet. Our proposed framework (4k+4k) has the ability to adapt to unseen and unlabeled new data to improve itself.

	Frame	Note	Note w/ offset
String 4k	68.4 ± 7.7	61.0 ± 13.8	41.6 ± 18.7
String 8k	67.7 ± 8.0	61.1 ± 13.5	41.5 ± 18.8
String 4k+4k	**68.7 ± 8.0**	**62.7 ± 13.3**	**42.8 ± 18.9**
Woodwind 4k	67.4 ± 2.8	48.2 ± 1.2	21.8 ± 2.0
Woodwind 8k	**68.1 ± 2.8**	50.9 ± 1.3	23.3 ± 3.2
Woodwind 4k+4k	66.6 ± 0.4	**51.7 ± 2.2**	**23.9 ± 4.6**

attempts to transcribe the drum beats, making the transcription slightly noisier than the baseline model. Nonetheless, this example shows the success of our proposed model in transcribing unseen music genres.

6 DISCUSSION

Although VAT is found to be useful for our proposed ReconVAT, in our pilot study we observed instability of VAT in some cases. More specifically, we observed that VAT does not work well with some of the baseline models. Whenever VAT is used, the transcription accuracy for the baseline model will collapse to zero. Even when we pretrain the baseline model to first reach their best performance, the moment the VAT kicks in, it causes a sudden increase in the transcription loss after only one forward step and weight update. The transcription loss does not decrease when the VAT component is present. We discovered that the dropout layers are the culprit causing this problem. Removing the dropout layers from the baseline models can prevent the above-mentioned problem from happening, but at the same time, the transcription accuracy for the baseline models are severely comprised without the dropout layers. The dropout layers somehow cause instability of the gradient g (Equation (3)), making it change a lot during each iteration, and it eventually leads to a vanishing gradient issue, and hence the gradient explosion of the $\frac{g_t}{\|g_t\|}$ term. Although the models proposed by Thickstun et al. [43] and Pedersoli et al. [36] do not have any dropout layers, their models are too resources consuming and take too much time to train. Therefore, we do not find enough motivation to apply VAT on them.

The U-net model proposed by Cheuk et al. [12] and Hung et al. [21] does not use any dropout layers, they are compact in size, and work well with our proposed framework. However, we believe that $p(Y_{\text{post}}|X_{\text{spec}}, \theta)$ may be replaced by any type of model as long as it does not affect the stability of the gradient g. Based on these results, we believe that future research opportunities for AMT lie in the semi-supervised or even unsupervised techniques that work well in scenarios with insufficient labelled data instead of just exploring deep fully supervised models that only work well with abundant labelled data.

7 CONCLUSION

In this paper, we proposed a VAT based semi-supervised AMT framework, **ReconVAT**, that works well for different kinds of musical instruments such as strings and woodwinds. We demonstrated its power of leveraging unlabelled data to enhance the transcription accuracy when the availability of labelled data is limited. Our proposed framework also generalizes better to other genres that are not present in the training dataset such as music covers of Japaneses pop music. The compactness of our model also allows it to be easily deployed in real-world applications[2].

ACKNOWLEDGMENTS

This work is supported by A*STAR SING-2018-02-0204, MOE Tier 2 grant MOE2018-T2-2-161, and SRG ISTD 2017 129.

REFERENCES

[1] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. 2016. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*. PMLR, 173–182.

[2] Mert Bay, Andreas F Ehmann, and J Stephen Downie. 2009. Evaluation of Multiple-F0 Estimation and Tracking Systems.. In *ISMIR*. 315–320.

[3] Emmanouil Benetos, Simon Dixon, Zhiyao Duan, and Sebastian Ewert. 2019. Automatic Music Transcription: An Overview. *IEEE Signal Processing Magazine* 36 (2019), 20–30.

[4] Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, and Anssi Klapuri. 2013. Automatic music transcription: challenges and future directions. *Journal of Intelligent Information Systems* 41 (2013), 407–434.

[5] Taylor Berg-Kirkpatrick, Jacob Andreas, and Dan Klein. 2014. Unsupervised transcription of piano music. In *Advances in neural information processing systems*. 1538–1546.

[6] Ralf Gunter Correa Carvalho and Paris Smaragdis. 2017. Towards end-to-end polyphonic music transcription: Transforming music audio directly to a score. In *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 151–155.

[7] Ali Taylan Cemgil. 2004. *Bayesian music transcription*. Ph.D. thesis, Radbound University Nijmegen Netherlands.

[8] William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2016), 4960–4964.

[9] Hung-Chen Chen and Arbee LP Chen. 2001. A music recommendation system based on music data grouping and user interests. In *Proceedings of the tenth international conference on Information and knowledge management*. 231–238.

[10] Luoxin Chen, Weitong Ruan, Xinyue Liu, and Jianhua Lu. 2020. SeqVAT: Virtual adversarial training for semi-supervised sequence labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 8801–8811.

[11] K. W. Cheuk, H. Anderson, K. Agres, and D. Herremans. 2020. nnAudio: An on-the-Fly GPU Audio to Spectrogram Conversion Toolbox Using 1D Convolutional Neural Networks. *IEEE Access* 8 (2020), 161981–162003.

[12] Kin Wai Cheuk, Yin-Jyun Luo, Emmanouil Benetos, and Dorien Herremans. 2020. The Effect of Spectrogram Reconstructions on Automatic Music Transcription:An Alternative Approach to Improve Transcription Accuracy. In *International Conference on Pattern Recognition (ICPR)*. IEEE, Washington, DC, USA, 9091–9098.

[13] Kin Wai Cheuk, Yin-Jyun Luo, Emmanouil Benetos, and Dorien Herremans. 2021. Revisiting the Onsets and Frames Model with Additive Attention. In *Proceedings of the International Joint Conference on Neural Networks*. IEEE, In press. https://doi.org/10.1109/SPW.2018.00014

[14] Keunwoo Choi and Kyunghyun Cho. 2019. Deep Unsupervised Drum Transcription. In *ISMIR*, Arthur Flexer, Geoffroy Peeters, Julián Urbano, and Anja Volk (Eds.). 183–191. http://archives.ismir.net/ismir2019/paper/000020.pdf

[15] Michael Scott Cuthbert and Christopher Ariza. 2010. music21: A toolkit for computer-aided musicology and symbolic music data. In *ISMIR*.

[16] Valentin Emiya, Nancy Bertin, Bertrand David, and Roland Badeau. 2010. MAPS-A piano database for multipitch estimation and automatic transcription of music. *Hal Inria* (2010).

[17] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations*.

[18] Curtis Hawthorne, Erich Elsen, Jialin Song, Adam Roberts, Ian Simon, Colin Raffel, Jesse Engel, Sageev Oore, and Douglas Eck. 2017. Onsets and Frames: Dual-Objective Piano Transcription. In *ISMIR*.

[19] Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. 2019. Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset. In *International Conference on Learning Representations*. https://openreview.net/forum?id=r1lYRjC9F7

[20] Yu-Siang Huang and Yi-Hsuan Yang. 2020. Pop Music Transformer: Generating Music with Rhythm and Harmony. *arXiv preprint arXiv:2002.00212* (2020).

[21] Yun-Ning Hung, I Ping Chiang, Yian Chen, and Yi-Hsuan Yang. 2019. Musical Composition Style Transfer via Disentangled Timbre Representations. In *IJCAI*. 4697–4703.

[22] Zheng Jiang and Roger B Dannenberg. 2019. Melody identification in standard midi files. In *Proceedings of the 16th sound & music computing conference*. 65–71.

[23] Rainer Kelz, Sebastian Böck, and Gerhard Widmer. 2019. Deep polyphonic adsr piano note transcription. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 246–250.

[24] Jong Wook Kim and Juan Pablo Bello. 2019. Adversarial Learning for Improved Onsets and Frames Music Transcription. *International Society forMusic Information Retrieval Conference* (2019), 670–677.

[25] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[26] Qiuqiang Kong, Keunwoo Choi, and Yuxuan Wang. 2020. Large-Scale MIDI-based Composer Classification. *arXiv preprint arXiv:2010.14805* (2020).

[27] Florian L Kreyssig and Philip C Woodland. 2020. Cosine-distance virtual adversarial training for semi-supervised speaker-discriminative acoustic embeddings. *Interspeech* (2020).

[28] Toyoaki Kuwahara, Yuichi Sei, Yasuyuki Tahara, Ryohei Orihara, and Akihiko Ohsuga. 2019. Model Smoothing Using Virtual Adversarial Training for Speech Emotion Estimation. In *2019 IEEE International Conference on Big Data, Cloud Computing, Data Science & Engineering (BCD)*. IEEE, 60–64.

[29] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*. 2980–2988.

[30] Zhang Liumei, Jiang Fanzhi, Li Jiao, Ma Gang, and Liu Tianshi. 2021. K-means clustering analysis of Chinese traditional folk music based on midi music textualization. In *2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP)*. IEEE, 1062–1066.

[31] José Pedro Magalhaes. 2015. CHORDIFY: THREE YEARS AFTER THE LAUNCH. In *ISMIR*.

[32] Albert Meroño-Peñuela, Rinke Hoekstra, Aldo Gangemi, Peter Bloem, Reinier de Valk, Bas Stringer, Berit Janssen, Victor de Boer, Alo Allik, Stefan Schlobach, et al. 2017. The MIDI linked data cloud. In *International Semantic Web Conference*. Springer, 156–164.

[33] Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. *ICLR* (2016).

[34] T. Miyato, S. Maeda, M. Koyama, and S. Ishii. 2019. Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 8 (2019), 1979–1993. https://doi.org/10.1109/TPAMI.2018.2858821

[35] Dae Hoon Park and Yi Chang. 2019. Adversarial sampling and training for semi-supervised information retrieval. In *The World Wide Web Conference*. 1443–1453.

[36] Fabrizio Pedersoli, George Tzanetakis, and Kwang Moo Yi. 2020. Improving Music Transcription by Pre-Stacking A U-Net. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 506–510.

[37] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. 2019. Stand-Alone Self-Attention in Vision Models. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2019/file/3416a75f4cea9109507cacd8e2f2aefc-Paper.pdf

[38] Miguel A. Román, Antonio Pertusa, and Jorge Calvo-Zaragoza. 2018. An End-to-end Framework for Audio-to-Score Music Transcription on Monophonic Excerpts.. In *ISMIR*. 34–41.

[39] Miguel A. Román, Antonio Pertusa, and Jorge Calvo-Zaragoza. 2019. A Holistic Approach to Polyphonic Music Transcription with Neural Networks. In *ISMIR*. 731–737.

[40] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-Attention with Relative Position Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 464–468. https://doi.org/10.18653/v1/N18-2074

[41] Siddharth Sigtia, Emmanouil Benetos, and Simon Dixon. 2015. An End-to-End Neural Network for Polyphonic Piano Music Transcription. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 24 (2015), 927–939.

[42] JQ Sun and Seok-Pil Lee. 2017. Query by singing/humming system based on deep learning. *Int. J. Appl. Eng. Res* 12, 13 (2017), 973–4562.

[43] John Thickstun, Zaïd Harchaoui, Dean P. Foster, and Sham M. Kakade. 2017. Invariances and Data Augmentation for Supervised Music Transcription. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2017), 2241–2245.

[44] John Thickstun, Zaïd Harchaoui, and Sham M. Kakade. 2016. Learning Features of Music From Scratch. In *ICLR*, Vol. abs/1611.09827.

[45] Minz Won, Sanghyuk Chun, and Xavier Serra. 2019. Toward Interpretable Music Tagging with Self-Attention. arXiv:1906.04972 [cs.SD]

[46] Yu-Te Wu, Berlin Chen, and Li Su. 2019. Polyphonic Music Transcription with Semantic Segmentation. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 166–170. https://doi.org/10.1109/ICASSP.2019.8682605

[47] Y. T. Wu, B. Chen, and L. Su. 2020. Multi-Instrument Automatic Music Transcription With Self-Attention-Based Instance Segmentation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28 (2020), 2796–2809. https://doi.org/10.1109/TASLP.2020.3030482

[48] Genki Yamaguchi and Makoto Fukumoto. 2019. A music recommendation system based on melody creation by interactive GA. In *2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*. IEEE, 286–290.