JEPOO: Highly Accurate Joint Estimation of Pitch, Onset and Offset for Music Information Retrieval

Haojie Wei[1*], **Jun Yuan**[2], **Rui Zhang**[3], **Yueguo Chen**[1†], **Gang Wang**[2]
[1]School of Information, Renmin University of China, Beijing, China
[2]Huawei Noah's Ark Lab, Shenzhen, China
[3]Tsinghua University [‡]
{weihaojie, chenyueguo}@ruc.edu.cn, {yuanjun25, wanggang110}@huawei.com, rayteam@yeah.net

Abstract

Melody extraction is a core task in music information retrieval, and the estimation of pitch, onset and offset are key sub-tasks in melody extraction. Existing methods have limited accuracy, and work for only one type of data, either single-pitch or multi-pitch. In this paper, we propose a highly accurate method for joint estimation of pitch, onset and offset, named JEPOO. We address the challenges of joint learning optimization and handling both single-pitch and multi-pitch data through novel model design and a new optimization technique named Pareto modulated loss with loss weight regularization. This is the first method that can accurately handle both single-pitch and multi-pitch music data, and even a mix of them. A comprehensive experimental study on a wide range of real datasets shows that JEPOO outperforms state-of-the-art methods by up to 10.6%, 8.3% and 10.3% for the prediction of Pitch, Onset and Offset, respectively, and JEPOO is robust for various types of data and instruments. The ablation study validates the effectiveness of each component of JEPOO.

1 Introduction

Music information retrieval (MIR) is an essential infrastructure supporting the daily use of the large music platforms. Melody is critical to music retrieval such as query/search by singing and content based music recommendation. Melody is also the core of music understanding as agreed by many existing studies [Hawthorne *et al.*, 2018a; Kim *et al.*, 2018; Gfeller *et al.*, 2020; Gardner *et al.*, 2021; Hawthorne *et al.*, 2021]. However, the vast majority of music data are in their audio form, typically `.wav` or `.mp3` files, which do not directly reflect the melody of the music data. To obtain the melody of music, we have to perform the so-called *melody extraction*, which converts an audio file into a sequence of notes, consisting of (i) the fundamental frequency f0 (termed *pitch*) of the note, (ii) the start of the note (termed *onset*) and (iii) the end of the note (termed *offset*). Specifically, melody

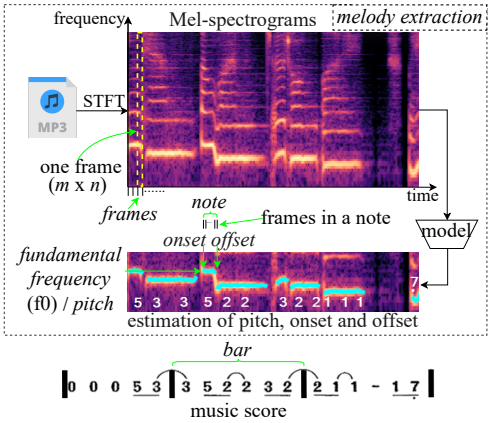

Figure 1: Melody extraction.

extraction aims to predict the value of f0, the onset, and the offset of each note based on the mel-spectrogram of an audio file as shown in Figure 1, which is a 2-dimensional matrix denoting the strength of every frequency during every time segment (termed *frame*). It is challenging because the boundaries of a note (i.e., its onset and offset) are usually blurry and noisy (c.f. Figure 1), where the yellow signals tend to have sharp jumps and blurry edges around the beginning and ending of a note. There has been continued research on melody extraction, but the effectiveness of existing work still needs substantial improvements to be really useful in practice. There are two major challenges described below.

Challenge 1: Joint learning optimization. One line of studies (e.g., CREPE [Kim *et al.*, 2018] and SPICE [Gfeller *et al.*, 2020]) focus on predicting only pitches but not onsets/offsets. The accuracy of these methods is limited and sensitive to noise (c.f. Figure 7), because they do not utilize the timing of the notes (onsets/offsets) and do not consider long-term context. It is reasonable to believe that the prediction of pitch and the prediction of onset/offset should benefit each other. Therefore, another line of studies (e.g., OAF [Hawthorne *et al.*, 2018a] and its follow-ups [Hawthorne *et al.*, 2018b; Kim and Bello, 2019; Kelz *et al.*, 2019]) perform joint learning of the two tasks *pitches* and *onsets/offsets*. However, they perform joint learning by

[*]Work done when being an intern at Huawei Noah's Ark Lab.
[†]Corresponding Author
[‡]www.ruizhang.info

simply summing up the different tasks' losses as the objective function, which do not address the following problems: (i) *Model design*. What is the suitable model structure for the joint prediction of pitch and onset/offset. (ii) *Data imbalance*. There is a huge imbalance in the labeled data; Specifically, there are much more negative labels (frames with no pitch) than positive labels (a frame with at least one pitch), and there are even fewer frames with onset/offset labels than those with pitches because a note has only one onset and one offset frame but many frames with pitches (c.f. Figure 1). (iii) *Imbalance between the importance of pitch and onset/offset prediction*. Our ultimate goal is to achieve a balanced accuracy of three tasks, pitch prediction, onset prediction and offset prediction. Weighting one task too high may result in high accuracy in one task but low accuracy in the others. The optimal weights between the losses of the three tasks may not be the same, which is what existing studies assume when they simply summing up the objective functions of different tasks. We need to learn the appropriate weights between the three tasks, which is challenging.

Challenge 2: Handling both single-pitch and multi-pitch data. Another deficiency of both lines of studies, which seems to be coincidental with the application each line of studies have targeted, is that they have been designed for either single-pitch prediction or multi-pitch prediction. This makes them work well for only one case, but not the other case or a mix of both cases. Specifically, CREPE [Kim *et al.*, 2018], SPICE [Gfeller *et al.*, 2020] and their follow-ups have mainly used the single-pitch (**SP**) dataset such as `MDB-stem-synth` [Salamon *et al.*, 2017]. We call them *single-pitch prediction* (**SPP**) algorithms because they were designed to predict only one pitch at any timestamp. In comparison, OAF [Hawthorne *et al.*, 2018a] and its follow-ups [Hawthorne *et al.*, 2018b; Kim and Bello, 2019; Kelz *et al.*, 2019] focus on multi-pitch (**MP**) datasets such as `MAESTRO-V1.0.0` [Hawthorne *et al.*, 2018b] and `MAPS` [Emiya *et al.*, 2009]. We call them *multi-pitch prediction* (**MPP**) algorithms because they were designed to predict multiple pitches at any timestamp.

Existing SPP algorithms perform poorly on MP datasets since they are not able to predict multiple pitches at the same timestamp. Existing MPP algorithms perform poorly on SP datasets because they were trained on MP data and tend to predict multiple pitches in a frame. Since SPP is a special case of MPP, we may improve MPP algorithms by retraining them on SP data. However, their performance on a mix of SP and MP data is still poor because their decision boundaries are different caused by the positive/negative label imbalance. Specifically, SP data has a much lower positive/negative label ratio than that of MP data, because in SP data, there is usually one pitch (positive label) in a frame while in MP data, there are usually multiple pitches in a frame. In real settings, we do not know whether the music data is SP or MP in advance, so neither SPP or MPP algorithms do well in generic settings.

To address above challenges, we propose a highly accurate method for joint estimation of pitch, onset and offset (JEPOO). Challenge 1 arises from three problems, (i) model design, (ii) data imbalance and (iii) multi-task weight allocation. To address problem (i), we design a model structure which has parameter sharing and feature fusion. Focal loss [Lin *et al.*, 2017] is a popular approach to problem (ii), and Pareto optimization [Lin *et al.*, 2019] is a popular approach to problem (iii). However, we find that a direct application of focal loss or Pareto optimization separately yields very limited performance gain. Further, the gain of applying focal loss and Pareto optimization together is less than the sum of the gains of applying each technique separately. We believe this is because the weights obtained by one technique may conflict with those obtained by the other technique to some extent. Therefore, we propose a novel way to combine the two as follows. In focal loss, the $(1-\hat{y})^{\gamma}$ value is used to set the weight for each sample. Since Pareto optimization produces the weights of the different tasks, we replace the $(1-\hat{y})^{\gamma}$ value in focal loss by the task weight resulted from Pareto optimization. The intuition is that the higher the weight of a task, the higher the weight of the samples in that task. This way, we achieve much higher accuracy when using Pareto optimization together with focal loss, and we call the resulted loss as *Pareto modulated loss* (PML). Moreover, to avoid imbalance between the losses of different tasks, we impose a regularization on the weights of the losses of the three tasks, which we call *loss weight regularization* (LWR). The ultimate optimization method for JEPOO is PML with LWR.

Challenge 2 is caused by the different decision boundaries of SPP and MPP algorithms resulted from the positive/negative label imbalance. Our proposed PML has the effect of focal loss, which addresses data imbalance and enlarges the difference of the prediction values of positive and negative samples for both SP and MP data. Therefore, our method is robust w. r. t. different decision boundaries (Figure 6) and addresses Challenge 2.

Our contributions are summarized as follows: i) We propose JEPOO, a highly accurate method for joint estimation of pitch, onset and offset. We address the challenges of joint learning optimization and handling different types of data by novel model design and a new optimization technique named Pareto modulated loss with loss weight regularization. ii) This is the first work that can accurately handle both SP and MP music data or a mix of them. iii) A comprehensive experimental study on a wide range of real datasets shows that JEPOO significantly outperforms state-of-the-art methods by up to 10.6%, 8.3% and 10.3% for the prediction of Pitch, Onset and Offset, respectively. Moreover, JEPOO's performance is robust for different types of datasets and instruments. The code of JEPOO is available at https://gitee.com/mindspore/models/tree/master/research/recommend/JEPOO.

2 Related Work

2.1 Pitch Prediction

Pitch prediction, also termed pitch estimation, has been studied extensively [Noll, 1967]. Once we have extracted the pitches, then we can use them as features for MIR by recent recommendation algorithms such as [Su *et al.*, 2021; Wang *et al.*, 2021]. Existing work on pitch estimation largely falls into two categories, SPP and MPP.

For SPP, traditional heuristic methods, such as ACF [Dubnowski *et al.*, 1976], YIN [De Cheveigné and Kawahara,

2002] and pYIN [Mauch and Dixon, 2014], employ a certain candidate-generating function to produce the pitch curve. Recently, some neural network based models have been proposed, such as CREPE [Kim *et al.*, 2018], SPICE [Gfeller *et al.*, 2020]. The accuracy of these methods is limited because they do not utilize the timing of the notes and hence cannot learn long-term sequential patterns.

For MPP, there are mainly two types of methods, including frame-level transcription methods and note-level transcription methods[Benetos *et al.*, 2018]. The frame-level transcription methods, such as OAF [Hawthorne *et al.*, 2018a], ADSRNet[Kelz *et al.*, 2019], Non-Saturating GAN [Kim and Bello, 2019], using CNN and LSTM to predict pitch results in each frame. While note-level transcription models, such as sequence-to-sequence [Hawthorne *et al.*, 2021] and MT3 [Gardner *et al.*, 2021] formulate the note as event to get the predictions using Transformer. But there is no research try to unify SPP and MPP algorithms as far as we know.

2.2 Joint Learning in Melody Extraction

Joint learning has been applied to MIR. For example, in [Choi *et al.*, 2017], a transfer learning approach is used to solve classification and regression tasks in MIR simultaneously. In [Bittner *et al.*, 2018], a multi-task deep learning model is used for melody, vocal and bass line estimation tasks. Besides, Hawthorne *et al.* proposed OAF [Hawthorne *et al.*, 2018a] to jointly learn pitch, onset/offset together, by using onset/offset predictions to rectify pitch predictions. There are several follow-ups [Hawthorne *et al.*, 2018b; Kim and Bello, 2019; Kelz *et al.*, 2019]. Unlike OAF, ADSRNet [Kelz *et al.*, 2019] shares the bottom parameters and uses a strong temporal prior in the form of a handcrafted HMM to rectify pitch predictions. However, these joint models do not consider the balance between different tasks at all.

In order to balance different tasks in joint learning, there are some optimization methods for multi-task learning, such as GradNorm [Chen *et al.*, 2018], DWA [Liu *et al.*, 2019], DTP [Guo *et al.*, 2018] and Pareto [Lin *et al.*, 2019]. GradNorm and DWA make each task learn at a similar rate. DTP allows the model to give difficult task a bigger weight so as to dynamically prioritize difficult tasks during training. And Pareto determines the weight of each task through Pareto optimal solutions. However, none of them has been used in MIR as far as we know.

3 Method

The problem of melody extraction is formulated in Appendix A.1.

3.1 Model Structure Design

The overall structure is illustrated in Figure 2, which has three key mechanisms designed for the joint learning of pitch and onset/offset prediction: (i) Shared bottom layers, (ii) task-specific multi-label sequential predictors, and (iii) fusion of high level features, which are detailed below.

Shared bottom layers. To capture common features of all sub-tasks, we stack several ReConv blocks as the shared bottom as shown in Figure 3. A ReConv block contains two base convolutional layers and a skip-connection layer. The skip-connection layer is a convolutional layer with 1×1 kernel, while other kernels of base convolutional layers are 3×3. After element-wisely summing the output of skip-connection layer and last base convolutional layer, the result goes through a Relu function and becomes the output. By using ReConv block, the model can utilize multi-level features and become much deeper than conventional melody extraction models.

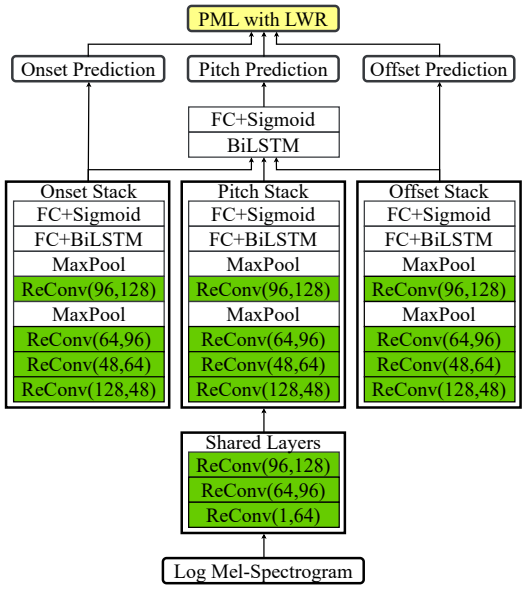

Figure 2: The overall structure of JEPOO.

Task-specific multi-label sequential predictors. A note may contain hundreds of frames, so we design sequential predictors to utilize long-term features to achieve better performance. Specifically, we design a task-specific sequential block, which consists of 4 ReConv blocks, a max pooling layer, a BiLSTM and a full connection layer with sigmoid. The input of task-specific multi-label sequential predictors is the output of Shared bottom layer. We call the parameters of different tasks as onset stack, pitch stack and offset stack. The output of these stacks is 88-elements vectors, whose elements are the predicted probability of corresponding labels.

Fusion of high level features. We use the predictions of onset and offset to help the pitch prediction in our model. Specifically, we concatenate the output of three stacks as the input of a BiLSTM layer along with full connection layer and sigmoid function to get pitch predictions. We do not use the pitch prediction to help onset/offset prediction due to the fol-

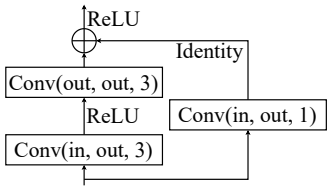

Figure 3: The details of residual convolution (ReConv) block.

lowing reason. Onset/offset data only has positive labels in the starting/ending frames of a note, while pitch data has positive labels in almost all the frames of a note. The different distribution of pitch labels and onset/offset labels cause two impact. On the one hand, the positive labels of pitch between the starting and ending frames can be noise to Onset/Offset prediction. On the other hand, pitch prediction are inaccurate at boundaries of notes, and the onset/offset prediction can rectify the boundaries of note. Our experimental study validates the effectiveness of this design (see Appendix A.7).

3.2 Pareto Modulated Loss With Loss Weight Regularization (PML with LWR)

To address the problems of data imbalance and multi-task weight allocation, we propose to use a combination of focal loss and Pareto optimization. Next, we firstly describe a naive combination of Pareto optimization and focal loss below, but it only yields limited performance improvement. We then present Pareto Modulated Loss (PML) with Loss Weight Regularization (LWR), which works much better.

Naive Combination of Pareto Optimization and Focal Loss (Naive Optimization). Focal loss is popular to balance the category within a sub-task by setting hyper-parameter α and shift the decision boundaries. Pareto optimization is popular to allocate the weight of sub-tasks. A naive way to combine these two techniques is to multiple the Focal loss with the task weight produced by Pareto optimization as follows, and we call it the naive optimization.

$$\mathcal{L}_{task} = -\sum_{i}^{n} \omega^i \alpha^i y^i (1-\hat{y}^i)^{\gamma^i} \log \hat{y}^i \tag{1}$$

where ω^i is a Pareto optimal solution for *i*th sub-task and y^i is the label of *i*th sub-task, n is the total number of sub-tasks.

Our empirical evaluation in section 5.2 shows that the naive optimization addresses data imbalance and weight allocation to some extent, but the improvement is limited. The reason may be as follows. Firstly, focal loss needs to search two hyper-parameters for each sub-task, resulting in high cost of grid search. As we mentioned above, data of different tasks varies hugely, so it is hard to determine the scale and precision of data related hyper-parameter γ for each sub-task and results in sub-optimal hyper-parameters. What's worse, as the number of tasks grows, grid search space grows exponentially. Secondly, Pareto optimal solution may be imbalance between tasks, resulting in some tasks hard to optimize. The model may need to be trained many times before making all the tasks are optimized.

PML with LWR. We propose a novel Pareto Modulated Loss (PML) by integrating Pareto optimization and focal loss to reduce the high training cost of focal loss. In addition, we design a loss weight regularization method to avoid the imbalance between the loss weights of Pareto optimal solution.

PML is based on our observation that the difficulty of a task reflects the difficulty of its own data. Thus, we try to replace the item $(1-\hat{y})^{\gamma}$ of focal loss by the task weight produced by Pareto optimization. This way, data in different batches have different weights, and the grid search space becomes at 50% less than the focal loss.

After getting the Pareto optimal weights $[\omega^1, ..., \omega^n]$ of different tasks' loss, we use a MLP layer to get the final weights. i.e. $[\omega^1_{PML}, .., \omega^n_{PML}] = softmax(\mathbf{W}[\omega^1, ..., \omega^n] + \mathbf{b})$. The formal form of PML is written as follows:

$$\mathcal{L}_{task} = -\sum_{i}^{n} \omega^i_{PML} \alpha^i y^i \log \hat{y}^i \tag{2}$$

PML obtains more discriminative ability than Pareto optimization without introducing any hyper-parameter. Moreover, PML has the effect of focal loss, which enlarges the difference between positive and negative sample predictions.

To avoid imbalance of loss weights, we design a simple yet efficient Loss Weight Regularization (LWR) as follows:

$$\mathcal{L}_{re} = \sum_{i}^{n} ||n\omega^i_{PML} - 1||_p \tag{3}$$

ω_{PML} represents the loss weight of different sub-tasks. Because $\sum_i^n \omega^i_{PML} = 1$, the LWR gives penalty to the weight that is far away from the average.

Then final PML with LWR loss is defined as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{task} + \lambda \mathcal{L}_{re} \tag{4}$$

Compared with PML, PML with LWR only introduces one more hyper-parameter than Pareto optimization, the weight of LWR item λ. The grid search cost is much less than the native optimization.

4 Experimental Setup

Datasets. To compare with previous pitch prediction methods on SP and MP data, we use three real datasets MDB-stem-synth [Salamon *et al.*, 2017], MAPS [Emiya *et al.*, 2009] and MAESTRO-V1.0.0 [Hawthorne *et al.*, 2018b]. *MDB-stem-synth* is a SP dataset. It contains 230 resynthesized monophonic music files spanning 25 musical instruments corresponding perfect f0 annotation. *MAPS* has a very small percentage of SP data and a majority of MP data. It contains 270 raw audio recordings of piano music and corresponding MIDI-annotated piano recordings. *MAESTRO-V1.0.0* is a MP dataset larger than MAPS. It contains 172.3 hours of paired audio and MIDI recordings from ten years of International Piano-e-Competition.

Evaluation Metrics. Following MT3 [Hawthorne *et al.*, 2018a] and CREPE [Kim *et al.*, 2018], we use 4 metrics to evaluate our model. These metrics are computed by mir_eval [Raffel *et al.*, 2014]. The details are described as follows: *The F1 score of pitch prediction (Pitch)* uses a binary measure of whether the prediction of a frame and the ground truth matches. Each second will be divided into a fixed number of frames, and the sequence of notes is represented as a binary matrix of size [frames × 88], which indicates the presence or absence of an active note at a given pitch and time. *Note with onset (Onset)* considers a prediction to be correct if it has the same pitch and is within ±50ms of a reference onset. *Note with onset and offset (Onset&Offset)*. In addition to matching onsets and pitches as above, this metric requires the note to also end in this frame (offset). *Voicing false alarm rate (VFA)* computes the proportion of non-melody frames in the ground truth but are mistakenly predicted as melody frames.

Methods	F1(%) on MDB-stem-synth(SP)			F1(%) on MAPS(SP&MP)			F1(%) on MAESTRO(MP)		
	Pitch	Onset	Onset&Offset	Pitch	Onset	Onset&Offset	Pitch	Onset	Onset&Offset
PYIN* [Mauch and Dixon, 2014]	79.6	56.5	56.4	12.5	28.4	28.1	10.5	25.0	24.4
CREPE* [Kim *et al.*, 2018]	90.6	78.5	78.5	26.0	41.3	40.8	21.5	41.3	40.2
OAF-retrain [Hawthorne *et al.*, 2018a]	95.3	90.9	89.8	79.7	81.9	61.7	89.7	94.1	79.6
OAF* [Hawthorne *et al.*, 2018a]	65.5	38.2	26.5	71.7	80.8	40.8	90.2	95.3	80.5
ADSRNet† [Kelz *et al.*, 2019]	——	——	——	77.2	81.4	56.1	——	——	——
Non-Saturating GAN† [Kim and Bello, 2019]	——	——	——	——	——	——	91.4	95.6	81.3
KJN† [Kwon *et al.*, 2020]	——	——	——	——	——	——	83.8	94.7	79.4
sequence-to-sequence* [Hawthorne *et al.*, 2021]	20.0	29.8	22.7	47.1	75.7	35.4	66.0	96.0	83.5
MT3* [Gardner *et al.*, 2021]	12.0	4.8	2.3	74.4	80.7	51.6	86.0	95.0	80.0
JEPOO	**97.1**	**96.0**	**95.6**	**81.6**	**84.2**	**65.6**	**93.0**	**96.5**	**84.0**

Table 1: Performance comparison on both SP and MP datasets in terms of F1 score. * means we reproduce the results using authors' open source checkpoints. † represents we copy the results from original papers. OAF-retrain represents retraining OAF on three open datasets.

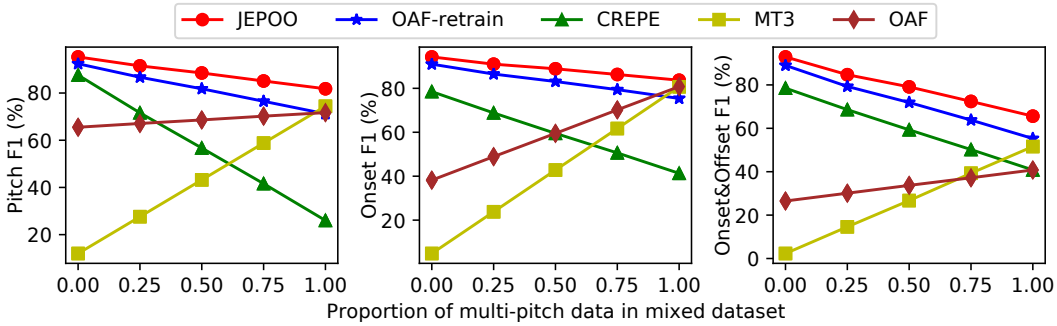

Figure 4: Performance on synthetic test datasets with different proportion of multi-pitch data. The results of CREPE, MT3 and OAF are reproduced by using authors' open source checkpoints. OAF-retrain represents retraining OAF on synthetic train dataset.

5 Experimental Results

Firstly, we compare JEPOO with SOTA methods on both real datasets and synthetic datasets of mixed SP and MP data. We then perform an ablation study to understand the effectiveness of the techniques we propose. Finally, we investigate the robustness of JEPOO on different types of datasets and instruments. The implementation details and comparison systems are provided in Appendix A.2 and A.3 respectively.

5.1 Main Results

Experiment on real datasets. Table 1 shows the results of comparing JEPOO with SOTA methods on the three aforementioned real datasets. We observe that JEPOO outperforms all the other methods on the datasets and all the tasks (Pitch, Onset, Onset&Offset) consistently. JEPOO outperforms the naive joint learning method OAF at all metrics by up to 30%, 58%, 69% in pitch, onset and offset prediction, respectively. This confirms that naive joint learning is far from optimal and our methods are necessary. JEPOO also outperforms SPP methods (CREPE, PYIN) on SP data, and MPP methods (all the rest) on MP data, respectively. Even we retrain OFA on SP data, and get the improved OAF-retrain method, JEPOO still outperforms OAF-retrain significantly on all tasks. These results show the effectiveness of our model design and optimization techniques.

Moreover, it should be noted that existing methods do not perform well on SP and MP data, simultaneously. As shown in Table 1, SPP models (CREPE and PYIN) perform poorly on MP datasets. MPP models (OAF, MT3 and sequence-to-sequence) perform poorly on SP datasets.

Experiment on synthetic dataset mixing SP and MP data. In real application scenarios, melody extraction models need to handle both SP and MP music data, because we are unable to know the data type in advance. Unfortunately, the current datasets are not suitable to evaluate such ability. Though MAPS is created to evaluate both SPP and MPP, the amount of different data are unbalance. Multi-pitch frames are 6 times as many as single-pitch frames. Worse still, the proportion of single-pitch frames in real audios is unknown.

To create a dataset that can evaluate the ability of JEPOO on handling both SP and MP data, we take two steps: *Firstly*, we mix the train dataset of MAPS and MDB-stem-synth to create synthetic train dataset. *Secondly*, to simulate various situations of real audio, we create multiple test datasets with different proportions of multi-pitch data by adding different amount of MDB-stem-synth test data into MAPS test dataset. The proportion ranges from 0 to 1 with the step of 0.25.

We retrain JEPOO and OAF on the synthetic train dataset. Figure 4 shows the evaluation results on different synthetic test datasets. In this figure, JEPOO outperforms all the other methods at any proportion of MP data significantly. JEPOO outperforms SOTA method by up to 10.6%, 8.3% and 10.3% at pitch, onset and offset prediction, respectively. By contrast, current models only work well on one type of data. For example, the SPP model CREPE decreases fastest as the raising of the proportion of MP data. MT3 performs worst when SP data is in majority. OAF-retrain gets second best results, but the gap of JEPOO and OAF-retrain increases with increasing of the proportion of MP data. These results indicate the ability of JEPOO to handle SP and MP data simultaneously.

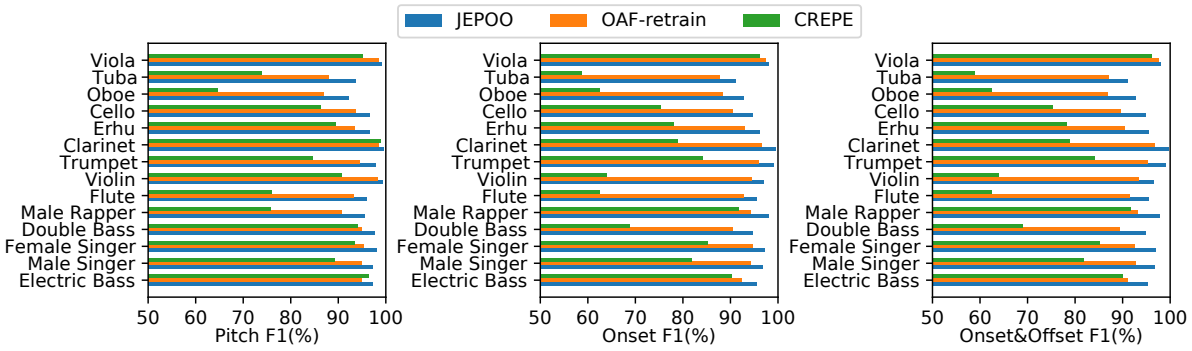

Figure 5: The performance of different models with different instruments. OAF-retrain represents retraining OAF.

Through comparing with Table 1, we find that OAF trained on synthetic dataset get worse performance than the model trained on SP and MP dataset respectively, decreasing at pitch prediction by 2.9% on MDB-stem-synth test dataset and 8.5% on MAPS test dataset. While JEPOO trained on synthetic dataset improves 0.2% on MAPS test dataset. This result indicates that simply training model on synthetic data cannot improve model performance. Experiments on synthetic dataset of mixing SP and MP data show the ability of JEPOO to handle both SP and MP music data, which has great practical value in MIR systems.

5.2 Ablation Study

In this experiment, we not only evaluate different optimization techniques, but also compare JEPOO with naive joint learning, single task training on synthetic datasets. We conducted ablation studies of each component of JEPOO. More ablation studies about optimization techniques, sequential predictor, model structure and feature fusion are given in Appendix A.4, A.5, A.6 and A.7, respectively.

From Table 2, we observe that naive joint learning does

Methods	F1(%) on MAPS+MDB-stem-synth		
	Pitch	Onset	Onset&Offset
JEPOO	**87.6**	**88.3**	**77.4**
JEPOO with naive optimization	87.2	87.9	76.8
JEPOO with only FL	86.9	87.5	76.4
JEPOO with only Pareto	86.4	87.9	76.1
Naive joint learning of pitch, onset and offset	86.0	87.2	74.4
Naive joint learning of pitch and onset	85.5	87.4	74.3
Single model of pitch (SMP)	86.7	80.1	68.3

Table 2: Ablation study on different optimization techniques in terms of F1 score. The test dataset is 1:1 mixed SP and MP data.

not improve all sub-tasks, though pitch, onset and offset are highly related. For example, the naive joint learning of pitch, onset and offset decreases 0.7% at Pitch than SMP. While JEPOO with naive optimization and JEPOO both outperform naive joint model at all metrics. Although JEPOO with naive optimization performs better than JEPOO with only FL and JEPOO with only Pareto, but the improvement is only 1.2% at Pitch, which is less than the sum of separated improvement of focal loss (0.4%) and the Pareto optimization (0.9%). In addition, JEPOO with naive optimization achieves the same performance at Onset than JEPOO with only Pareto. JEPOO improves 1.6% at Pitch, 3.0% at Onset&Offset than naive joint model. Besides, the training cost of naive optimization is about four times as PML with LWR. Above results show that PML with LWR can better balance different sub-tasks and data in a low training cost than naive optimization.

5.3 Robustness of JEPOO

Experiment with different instruments. It is obviously that the performance of melody extraction varies on different instruments. To investigate the impact of instruments on our method, we deeper evaluate JEPOO on the multi-instrument dataset MDB-stem-synth along with OAF and CREPE. The reason that we only compare with these baselines is that OAF-retrain achieves second best results on MDB-stem-synth and CREPE is designed for this dataset. In this experiment, we use stratified sampling to split the train and test dataset, ensuring that the distribution of instruments is consistent between two datasets. By this way, the train and test datasets contain 25 instruments, such as bass, violin, flute and singing voice etc. We retrain JEPOO, OAF on the new multi-instrument dataset and evaluate these models on the new test dataset.

From the results on various instruments shown in Figre 5, there are following conclusions: **i)** JEPOO consistently outperforms OAF-retrain and CREPE at all metrics on all instruments. This result shows better discrimination ability of our method on different instruments than SOTA methods. **ii)** Our method has excellent generality on different instruments. From Figure 5, we can see that JEPOO achieves more stable performance than OAF-retrain and CREPE. The F1 score variances of JEPOO are only 4.0, 5.1 and 5.0 at Pitch, Onset and Onset&Offset, respectively. While the variances of OAF-retrain are 11.5, 8.0 and 9.6, and the variances of CREPE are 95.2, 135.1 and 135.1. The high variance of CREPE is because the checkpoint is trained on unbalanced instrumental datasets. The above conclusions demonstrate the high accuracy and robustness of JEPOO on multiple instruments.

Robustness on SP and MP datasets. To evaluate the robustness of JEPOO on handling SP and MP data, we train JEPOO and OAF on the synthetic train dataset and evaluate them with different positive thresholds on SP and MP dataset, respectively. There is a positive prediction when predicted probability is larger than the positive threshold. We test positive thresholds from 0.1 to 0.9 with step 0.1. The left of Fig-

ure 6 reports the result on SP dataset MDB-stem-synth and the right of Figure 6 reports the result on MP dataset MAPS. According to Figure 6, we can draw following conclusions:

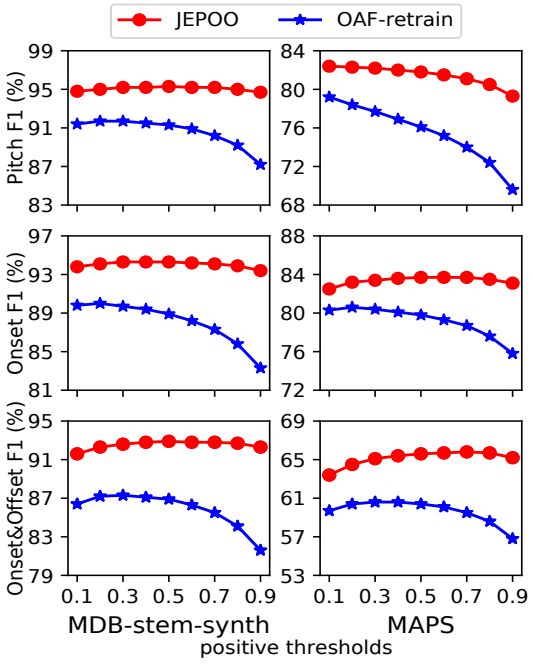

Figure 6: Comparison of different thresholds on MAPS (MP) and MDB-stem-synth(SP).

Firstly, JEPOO outperforms OAF-retrain on MDB-stem-synth and MAPS at all thresholds. Specifically, on MDB-stem-synth dataset, OAF-retrain achieves 91.7%, 90.0% and 87.2% at Pitch, Onset and Onset&Offset with the best threshold. JEPOO achieves 95.3%, 94.3% and 92.8% at Pitch, Onset and Onset&Offset when the best threshold is 0.5. There is a similar result on MAPS dataset. These results indicate that JEPOO can separate positive and negative samples clearly, making it robust against noise.

Secondly, JEPOO is more robust on both SP and MP data. The best threshold of JEPOO is near 0.5 on both SP and MP dataset, while the best threshold of OAF-retrain is just 0.2. When the threshold grows from 0.1 to 0.9, JEPOO decreases less than 0.7% at Pitch, 1.0% at Onset and 1.4% at Onset&Offset on SP dataset, and less than 3.1%, 1.2% and 2.4% on MP dataset. However, OAF-retrain decreases significantly when the threshold grows from 0.1 to 0.9, up to 9.6% at Pitch on MP dataset. Above results show the robustness of JEPOO on handing SP and MP datasets.

Non-Melody Frames Robustness. To evaluate robustness on non-melody frames, we use Voicing false alarm rate (VFA) to evaluate different models and report the results in Table 3. The smaller at VFA, the better performance of the model. In this experiment, we add white noise into clean audios to simulate real audios. We use Signal-to-noise ratio (SNR) to measure noise. The higher of SNR means the less of noise. Thus, the SNR INF in Table 3 represents raw clean audio.

JEPOO hardly predict pitch at non-melody frames on clean audios. In Table 3, JEPOO achieves only 0.2% at VFA, while CREPE is 48 times higher. Besides, when SNR is down to 50, the VFA of our model only increase to 3.6%, while OAF-retrain is up to 11.8% and CREPE is up to 28%. Although OAF-retrain only achieves 0.5% at VFA when no noise, but we find OAF-retrain predicts a pitch when the probability is higher than 0.03. This relative low threshold makes OAF is influenced by noise easily. As our model, the frame will be predicted positive only when the output probability is greater than 0.5. The above results show that JEPOO is more robust on non-melody frames and can predict correctly on almost all non-melody frames, since our model can separate positive and negative samples more clearly.

Methods	VFA(%) on MDB-stem-synth	
	SNR INF	SNR 50
CREPE [Kim *et al.*, 2018]	9.8	28.0
OAF-retrain [Hawthorne *et al.*, 2018a]	0.5	11.8
JEPOO	**0.2**	**3.6**

Table 3: Comparison of different models at non-melody frames in terms of VFA score.

Case study of Robustness. To intuitively understand the performance at non-melody frames, we visualize the result of a case to compare JEPOO and CREPE in Figure 7. We firstly extract clean vocals using ResUNetDecouple+ [Kong *et al.*, 2021] from the clip (12s-41s) of 403th song in the MIR_ST500 [Wang and Jang, 2021]. From this figure, we observe that our method predicts no pitch at non-melody frames, while CREPE predicts high pitch value at non-melody frames. This is because CREPE can be influenced by small noise (top left on mel-spectrogram) easily. In addition, the predictions of JEPOO coincide more closely with the ground truth, the red line in figure, than CREPE. Above results indicate the robustness of JEPOO in another aspect.

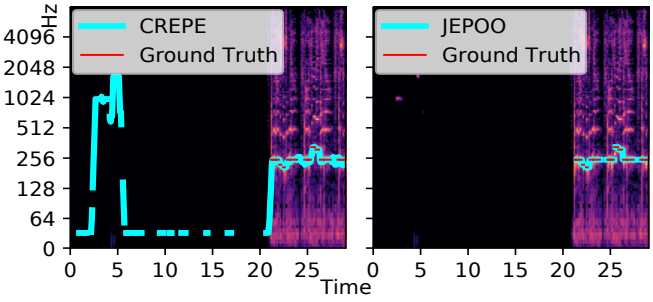

Figure 7: Case study of Robustness. The left is the predictions of CREPE, and the right is the predictions of JEPOO.

6 Conclusion

In this paper, we propose JEPOO, a highly accurate method for joint estimation of pitch, onset and offset by proposing novel model design and a new optimization technique named Pareto modulated loss with loss weight regularization. JEPOO significantly outperforms state-of-the-art methods by up to 10.6%, 8.3% and 10.3% for the prediction of Pitch, Onset and Offset, respectively, and JEPOO's performance is robust for different types of datasets and instruments.

A Appendix

A.1 Problem Formulation

In this section, we formulate the problem of *melody extraction*, for both SPP and MPP. In some studies, this is also called *music transcription*. The input is a Mel-spectrogram as shown in Figure 1, which is a 2-dimensional matrix $\mathrm{X}_{T\times F}$, where T is the count of audio frames and F is the number of frequency bins. Melody extraction coverts the Mel-spectrogram into a sequence of pitch values representing the f0 of the note; these are integer values typically in the range of 21 to 108. Thereby, melody extraction needs to predict the pitch value of each note, the starting frame of the note (onset) and the ending frame of the note (offset). In the literature, these are treated as multi-label classification problems for each frame. Specifically, each frame t of an audio is labeled by three 88-element one-hot arrays, $[\mathbf{y}^{p}, \mathbf{y}^{on}, \mathbf{y}^{off}]$. The elements of $\mathbf{y}^{p}$, $\mathbf{y}^{on}$ and $\mathbf{y}^{off}$ correspond to pitches, onsets and offsets, respectively. Thus, the melody extraction task can be formally written as $\mathcal{F} : X_{T\times F} \rightarrow Y_{T\times 3\times 88}$.

A.2 Implementation Details

The raw audio is sampled at 16 kHz and then transformed into Mel-spectrograms with log amplitude which has 229 mel bins. The hop length of Mel-spectrograms is 512 and the Hann window size is 2048. And we cut the frequency between 30Hz and 8000Hz to extract the Mel-spectrograms. Above audio processing uses librosa [McFee *et al.*, 2015], which is the same as OAF[Hawthorne *et al.*, 2018a].

The input and output channels of all ReConv blocks are marked in Figure 2, and all kernel sizes are introduced in Figure 3. Kernel size of MaxPool is (1, 2). The dimension of input and output of BiLSTM are both 768. We set the batch size as 16 and use the Adam optimizer [Kingma and Ba, 2014]. Each training audio is randomly selected from original audio with same duration, 12.8s. The learning rate is initialized as 0.0005 and reduced by 0.98 of the previous learning rate every 10000 iterations. We use the code released by [Lin *et al.*, 2019] to implement Pareto optimization. The initial weights of different sub-tasks are all 1, and we search new weights every 10 iterations. In loss weight regularization, we use L_2 distance, and the weight λ is 0.04.

We use the same way as CREPE[Kim *et al.*, 2018] to split MDB-stem-synth into train and test datasets. The split way of MAPS is the same as OAF [Hawthorne *et al.*, 2018a]. The split way of MAESTRO are introduced in [Hawthorne *et al.*, 2018b]. We process MIDI files of MAPS and MAESTRO using the same method as [Hawthorne *et al.*, 2018a]. The raw labels in time, frequency are transformed to note onset time, note offset time and midi numbers similar to MAPS and MAESTRO, when processing the MDB-stem-synth dataset.

A.3 Comparison System

We compare JEPOO with some recently proposed melody extraction methods. CREPE [Kim *et al.*, 2018] and pYIN [Mauch and Dixon, 2014] are the SPP methods and have open source checkpoints. We do not compare our model with SPICE [Gfeller *et al.*, 2020], because SPICE does not open code and use different metrics.

OAF [Hawthorne *et al.*, 2018a] is the first joint model and it has been designed for multi-pitch estimation. Its source code is publicly available and it has been trained on MP datasets. As discussed in Section 1, we may retrain it on SP data or mixed SP and MP data to get better performance, so we have retrained OAF on the various datasets respectively for each experiment and refer to this retrained version as *OAF-retrain*. The results of ADSRNet [Kelz *et al.*, 2019], Non-Saturating GAN [Kim and Bello, 2019] and KJN [Kwon *et al.*, 2020] are directly obtained from their original papers, because they do not make their code available.

MT3 [Gardner *et al.*, 2021] and sequence-to-sequence [Hawthorne *et al.*, 2021] are Transformer-based multi-pitch estimation models. The results of both models are reproduced by using authors' open source checkpoints. We do not compare our model with SpecTNT [Lu *et al.*, 2021], because it does not open code and is evaluated on different datasets.

A.4 Ablation Study of Optimization Techniques

Methods	F1(%) on MAPS		
	Pitch	Onset	Onset&Offset
JEPOO	**81.8**	**83.7**	**65.6**
JEPOO with naive optimization	81.5	83.1	64.9
JEPOO with only FL	80.5	82.6	64.1
JEPOO with only Pareto	80.8	82.2	63.9
Naive joint learning of pitch, onset and offset	79.8	82.1	63.5
Naive joint learning of pitch and onset	79.6	82.8	62.4
Single model of pitch (SMP)	80.2	75.1	58.6

Table 4: Ablation study on different optimization techniques in terms of F1 score. The test dataset is MAPS (MP) dataset.

In this section, we report more results of ablation studies for optimization techniques. Experimental settings are the same as those in Section 5.2. We evaluate different optimization techniques on MAPS and MDB-stem-synth datasets, and the results are shown in Table 4 and Table 5 respectively.

Methods	F1(%) on MDB-stem-synth		
	Pitch	Onset	Onset&Offset
JEPOO	**95.3**	**94.3**	**92.9**
JEPOO with naive optimization	94.9	94.1	92.5
JEPOO with only FL	94.3	92.5	92.1
JEPOO with only Pareto	94.1	91.4	92.0
Naive joint learning of pitch, onset and offset	93.8	90.4	91.6
Naive joint learning of pitch and onset	92.9	90.1	87.4
Single model of pitch (SMP)	94.2	82.5	80.8

Table 5: Ablation study on different optimization techniques in terms of F1 score. The test dataset is MDB-stem-synth (SP) dataset.

A.5 Comparison of BiLSTM and Transformer

In this section, we compare the effect of using BiLSTM or using Transformer in Table 6. We use Transformer with different layers to replace every BiLSTM in Figure 2. Besides, we also report the performance with different optimization techniques that we proposed. From this Table, we find that models with BiLSTM achieves better performance at two of the three metrics, Pitch and Onset&Offset. Based on this result, we adopt BiLSTM in our model design, rather than Transformer. In addition, 70% of models using PML with LWR have better

(BiLSTM or Transformer, #shared, #unshared)	F1(%) on MAPS+MDB-stem-synth					
	Base			PML+LWR		
	Pitch	Onset	Onset&Offset	Pitch	Onset	Onset&Offset
OAF-retrain	82.7	83.8	71.9	82.7	83.8	71.9
(BiLSTM, 6, 4)	86.8	87.6	75.4	87.2	88.0	76.6
(1 Transformer, 6, 4)	85.0	86.9	72.5	85.0	87.0	73.0
(2 Transformer, 6, 4)	86.5	87.9	75.8	86.4	88.2	75.6
(3 Transformer, 6, 4)	86.8	88.1	76.2	86.5	88.1	75.8
(4 Transformer, 6, 4)	86.5	87.7	75.7	86.5	87.5	75.8
(1 Transformer, 6, 8)	85.5	87.7	73.7	85.7	87.7	74.9
(2 Transformer, 6, 8)	86.1	88.0	75.5	86.3	88.5	75.9
(3 Transformer, 6, 8)	86.9	**88.1**	76.6	86.9	**88.6**	76.8
(4 Transformer, 6, 8)	86.6	87.7	76.1	86.9	88.2	76.4
JEPOO (BiLSTM, 6, 8)	**87.2**	87.9	**76.8**	**87.6**	88.3	**77.4**

Table 6: The comparison between using BiLSTM or using different layers of Transformer in terms of F1 score. #shared means the number of convolutional layers in shared layers, #unshared means the number of convolutional layers for specific sub-tasks. The test dataset is 1:1 mixed SP and MP data.

performance than the one using the naive optimization technique. This result indicates the generality of PML with LWR, and the effectiveness of PML with LWR on joint learning, which can be found in section 5.2.

A.6 Ablation Study of Convolutional Layers

In this section, we show the effect of convolutional layers and skip connection in Table 7. We adjust the number of convolutional layers in shared layers and sub-task's stacks. All models use PML with LWR. Compared to third line model (6, 4, True), the second line model (10, 4, True) adds four convolutional layers in shared layers, while the performance at Pitch, Onset and Onset&Offset metrics all decrease. This result indicates that oversharing decreases the discriminate ability of sub-tasks, and we need to limit the depth of shared layers. Compared to the model without skip connection, model with skip connection gets better performance with the same convolutional layers. This is because skip connection can utilize multi-level features and improve the performance. The fifth line model (6, 8, True) gets the best performance, and we adopt the configuration in other experiments.

(#shared, #unshared, skip-connection)	F1(%) on MAPS+MDB-stem-synth		
	Pitch	Onset	Onset&Offset
OAF-retrain	82.7	83.8	71.9
(10, 4, True)	80.4	81.0	62.0
(6, 4, True)	87.2	88.0	76.6
(6, 4, False)	87.1	87.9	76.6
(6, 8, True)	**87.6**	**88.3**	**77.4**
(6, 8, False)	87.2	88.1	77.0
(6, 12, True)	87.2	88.3	76.9
(6, 12, False)	87.2	87.7	76.4

Table 7: Comparison of different numbers of convolutional layers and whether to use skip connection in terms of F1 score. These models is trained on mixed train dataset of MDB-stem-synth and MAPS. The test dataset is 1:1 mixed SP and MP data. skip-connection means whether to use skip connection.

A.7 Ablation Study of Fusion Features

We evaluate the effect of different fusion methods in this section, and the results are shown in Table 8. The model in last line, whose pitch detector does not use the output of Onset stack and Offset stack. From third line to sixth line, these model's onset prediction and offset prediction fuse the output of pitch stack as input features with the corresponding weights, while pitch prediction uses onset and offset features as JEPOO. The model in the second line is JEPOO. Based on JEPOO, the model in the first line also fuses the onset prediction to offset, and the offset prediction to onset.

Fusion Methods	F1(%) on MAPS+MDB-stem-synth		
	Pitch	Onset	Onset&Offset
Fuse features for pitch, onset/offset	87.1	87.9	75.0
Fuse features for pitch	**87.6**	**88.3**	**77.4**
Fuse features for all sub-tasks(0.3)	87.0	87.8	75.2
Fuse features for all sub-tasks(0.5)	86.4	87.6	73.7
Fuse features for all sub-tasks(0.8)	83.8	84.8	63.9
Fuse features for all sub-tasks(1.0)	81.7	83.9	62.5
Without fusion	86.9	88.0	74.7

Table 8: Comparison of different fusion methods in terms of F1 score. These models is trained on mixed train dataset of MAPS and MDB-stem-synth. The test dataset is 1:1 mixed SP and MP data.

From this table, we can conclude that the output of pitch stack may be the noise of onset and offset predictions. Because the performance of the fifth line model drops significantly than the last line model. Moreover, the performance increases as the weight of the pitch stack output decreases. Besides, our fusion method achieves the best performance. This may be because the majority of the pitch is not the onset/offset, and feeding such predictions to the predictor of onset/offset will actually harm the performance (making the model tend to always predict no onset/offset). On the other hand, the onset/offset frames indicate the beginning and the ending of pitches, so the onset/offset frames provide helpful information for the pitch prediction.

Ideally, the model should learn to ignore the noise of pitches in the onset and offset tasks if we fuse the feature of pitches, but this is only true when there is sufficiently labeled data. In reality, due to the nature of the data as described in Section 1, there are much more frames with pitch labels than the frames with onset/offset labels, and this makes it difficult for the model to learn the patterns.

Acknowledgments

This work is supported in part by the National Science Foundation of China under the grant 62272466, and Public Computing Cloud, Renmin University of China.

Contribution Statement

Haojie Wei, Jun Yuan and Rui Zhang are co-first authors, they contributed equally to this work. Haojie Wei designs and implements the detailed algorithms, performs experimental studies, and is crucially involved in every aspect of this work. Jun Yuan designs the naive Optimization method and Pareto modulated loss, and is involved in detailed model designs and experiments. Rui Zhang identifies the research problem, provides the main idea of the solution and some detailed designs, oversees the execution of the experiments and conducts fine editing and writing of the paper. Yueguo Chen coordinates the research activities and contributes to all the discussions. Gang Wang contributes to the design of the loss weight regularization method.

References

[Benetos *et al.*, 2018] Emmanouil Benetos, Simon Dixon, Zhiyao Duan, and Sebastian Ewert. Automatic music transcription: An overview. *IEEE Signal Processing Magazine*, pages 20–30, 2018.

[Bittner *et al.*, 2018] Rachel M Bittner, Brian McFee, and Juan P Bello. Multitask learning for fundamental frequency estimation in music. *arXiv preprint arXiv:1809.00381*, 2018.

[Chen *et al.*, 2018] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the International conference on machine learning (ICML)*, pages 794–803, 2018.

[Choi *et al.*, 2017] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. Transfer learning for music classification and regression tasks. *arXiv preprint arXiv:1703.09179*, 2017.

[De Cheveigné and Kawahara, 2002] Alain De Cheveigné and Hideki Kawahara. Yin, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America (JASA)*, pages 1917–1930, 2002.

[Dubnowski *et al.*, 1976] John Dubnowski, Ronald Schafer, and Lawrence Rabiner. Real-time digital hardware pitch detector. *IEEE Transactions on Acoustics, Speech, and Signal Processing (TASLP)*, pages 2–8, 1976.

[Emiya *et al.*, 2009] Valentin Emiya, Roland Badeau, and Bertrand David. Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle. *IEEE Transactions on Audio, Speech, and Language Processing (TASLP)*, pages 1643–1654, 2009.

[Gardner *et al.*, 2021] Josh Gardner, Ian Simon, Ethan Manilow, Curtis Hawthorne, and Jesse Engel. Mt3: Multi-task multitrack music transcription. *arXiv preprint arXiv:2111.03017*, 2021.

[Gfeller *et al.*, 2020] Beat Gfeller, Christian Frank, Dominik Roblek, Matt Sharifi, Marco Tagliasacchi, and Mihajlo Velimirović. Spice: Self-supervised pitch estimation. *IEEE Transactions on Audio, Speech, and Language Processing (TASLP)*, pages 1118–1128, 2020.

[Guo *et al.*, 2018] Michelle Guo, Albert Haque, De-An Huang, Serena Yeung, and Li Fei-Fei. Dynamic task prioritization for multitask learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 270–287, 2018.

[Hawthorne *et al.*, 2018a] Curtis Hawthorne, Erich Elsen, Jialin Song, Adam Roberts, and Ian Simon and. Onsets and frames: Dual-objective piano transcription. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2018.

[Hawthorne *et al.*, 2018b] Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. Enabling factorized piano music modeling and generation with the maestro dataset. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[Hawthorne *et al.*, 2021] Curtis Hawthorne, Ian Simon, Rigel Swavely, Ethan Manilow, and Jesse Engel. Sequence-to-sequence piano transcription with transformers. *arXiv preprint arXiv:2107.09142*, 2021.

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

[Kim and Bello, 2019] Jong Wook Kim and Juan Pablo Bello. Adversarial learning for improved onsets and frames music transcription. *arXiv preprint arXiv:1906.08512*, 2019.

[Kim *et al.*, 2018] Jong Wook Kim, Justin Salamon, Peter Li, and Juan Pablo Bello. Crepe: A convolutional representation for pitch estimation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 161–165, 2018.

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

[Kong *et al.*, 2021] Qiuqiang Kong, Yin Cao, Haohe Liu, Keunwoo Choi, and Yuxuan Wang. Decoupling magnitude and phase estimation with deep resunet for music source separation. *arXiv preprint arXiv:2109.05418*, 2021.

[Kwon *et al.*, 2020] Taegyun Kwon, Dasaem Jeong, and Juhan Nam. Polyphonic piano transcription using autoregressive multi-state note model. *arXiv preprint arXiv:2010.01104*, 2020.

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

[Lin *et al.*, 2019] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. Pareto multi-task learning. *Advances in neural information processing systems (NIPS)*, 2019.

[Liu *et al.*, 2019] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 1871–1880, 2019.

[Lu *et al.*, 2021] Wei-Tsung Lu, Ju-Chiang Wang, Minz Won, Keunwoo Choi, and Xuchen Song. Spectnt: a time-frequency transformer for music audio. *arXiv preprint arXiv:2110.09127*, 2021.

[Mauch and Dixon, 2014] Matthias Mauch and Simon Dixon. pyin: A fundamental frequency estimator using probabilistic threshold distributions. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 659–663, 2014.

[McFee *et al.*, 2015] Brian McFee, Colin Raffel, Dawen Liang, Daniel P Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, pages 18–25, 2015.

[Noll, 1967] A Michael Noll. Cepstrum pitch determination. *The journal of the Acoustical Society of America*, pages 293–309, 1967.

[Raffel *et al.*, 2014] Colin Raffel, Brian McFee, Eric J Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, Daniel PW Ellis, and C Colin Raffel. Mir_eval: A transparent implementation of common mir metrics. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 367–372, 2014.

[Salamon *et al.*, 2017] Justin Salamon, Rachel M Bittner, Jordi Bonada, Juan J Bosch, Emilia Gómez Gutiérrez, and Juan Pablo Bello. An analysis/synthesis framework for automatic f0 annotation of multitrack datasets. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2017.

[Su *et al.*, 2021] Yixin Su, Rui Zhang, Sarah Erfani, and Zhenghua Xu. Detecting beneficial feature interactions for recommender systems. In *Proceedings of the AAAI conference on artificial intelligence*, pages 4357–4365, 2021.

[Wang and Jang, 2021] Jun-You Wang and Jyh-Shing Roger Jang. On the preparation and validation of a large-scale dataset of singing transcription. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 276–280, 2021.

[Wang *et al.*, 2021] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. Combating selection biases in recommender systems with a few unbiased ratings. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 427–435, 2021.