# Harmonic Preserving Neural Networks for Efficient and Robust Multipitch Estimation

Chin-Yun Yu, Jing-Hua Lin, and Li Su
Institute of Information Science, Academia Sinica, Taiwan
E-mail: lisu@iis.sinica.edu.tw Tel: +886-2788-3799

***Abstract*—Multi-pitch estimation (MPE) is a fundamental yet challenging task in audio processing. Recent MPE techniques based on deep learning have shown improved performance but are computation-hungry and relatively sensitive to the variation of data such as noise contamination, cross-domain data, etc. In this paper, we present the harmonic preserving neural network (HPNN), a model that incorporates deep learning and domain knowledge in signal processing to improve the efficiency and robustness of MPE. The proposed method starts from the multi-layered cepstrum (MLC), a feature representation that utilizes repeated Fourier transform and nonlinear scaling to suppress the non-periodic components in signals. Following the combined frequency and periodicity (CFP) principle, the last two layers of the MLC are integrated to suppress the harmonics of pitches in the spectrum and enhance the components of true fundamental frequencies. A convolutional neural network (CNN) is then placed to further optimize the pitch activation. The whole system is constructed as an end-to-end learning scheme. Improved time efficiency and performance robustness to noise and cross-domain data are demonstrated with experiments on polyphonic music in various noise levels and multi-talker speech.**

## I. Introduction

Multi-pitch estimation (MPE), the task to detect the concurrent and time-varying pitch values in audio signals, is one of the most fundamental task in automatic music transcription (AMT) [1] and speech recognition [2]. The MPE task has been considered as a challenging task mainly because the features of individual pitches in a sound mixture are highly overlapped with each other [3]. Such a phenomenon usually confronts the hope to set a generalized rule in designing the audio features for MPE. Over the past years, we have witnessed a paradigm shift in the MPE technology development, as data-driven MPE with modern deep learning methods have taken over from most of the traditional rule-based and feature-based pitch detection methods, with strongly improved performance on various genres of music or speech data. In the deep learning approach, neural networks with a huge number of parameters are trained on the highly-augmented, large-scale labeled data to fit all the possible characteristics of signals [4], [5], [6]. Computing cost in both the training and inference stages then appears as a central concern for the applications which need portable, speed-up, and robust MPE solutions, for example, a music practicing tool which requires MPE on mobile device and under noisy environments [7], [8].

It has been noticed that the traditional rule- and feature-based pitch detection methods play an important role in designing an efficient and robust neural network for MPE. Among these approaches, domain knowledge of music theory and signal processing are integrated into the design process to simplify the problem. For example, the FifthNet chroma extractor utilizes known structures of music intervals in the spectral features to compress a neural work for chord recognition [9], and the harmonic CQT (HCQT) adopts multi-channel data representations to enhance the saliency of fundamental frequencies against harmonics [10]. Also related to the exploration of harmonic structure in multi-pitch signals, a recently proposed method called *multi-layered cepstrum* (MLC) shows that, inspired from the multi-layer operation of deep neural networks, multi-pitch saliency can also be computed by repeated operations of generalized cepstrum, which is a discrete Fourier transform (DFT) followed by a power-scaled nonlinear activation function [11]. MLC also follows the *combined frequency and periodicity* (CFP) principle [12], [13], [14], which states that pitch saliency is the consensus of spectral and cepstral features, as the cepstral feature can well suppress the harmonics in the spectral feature; this principle is by far one of the most general and representative rules considered in the traditional rule-based and feature-based MPE methods.

In this paper, we integrate the ideas of CFP and MLC with deep learning, and propose the harmonic preserving neural network (HPNN), which allows the parameters of the power-scaled activation functions in MLC to be trainable and appends a small-size neural network at the output of MLC in order to better predict the pitch values at each frame. Since the DFT operation is fixed in the network, the HPNN is guided to learn pitch information from the harmonic/ periodic structures underlying in audio features, and is therefore robust to noise interference and cross-dataset inferences. Also, the number of trainable parameters in HPNN is much smaller than other MPE networks, and this can reduce the time in the training stage. Due to the smaller model size and the fast operation of discrete Fourier transform (DFT), the testing time can also be reduced. These characteristics of HPNN are verified through experiments on a wide variety of signals, including polyphonic music signal in clean and noisy conditions and multi-talker speech signals, by comparing to state-of-the-art multi-pitch estimation methods of both music and speech.

## II. Methods

The proposed HPNN model incorporates the MLC-CFP-CNN process altogether thanks to the flexibility of deep learning neural networks. Fig. 1 illustrates its architecture. In

a nutshell, it performs MLC with the trainable power-scale nonlinear function, and next it employs CFP to transform the last two MLC time-domain and frequency-domain outputs into unified time-frequency representations. The stacked audio features are then fed into a convolutional neural network (CNN) for further identifying the fundamental frequencies.

The MLC repeatedly operates DFT, filtering, and nonliner activation by $N$ times. Since the input is the raw audio, the DFT operation is essentially the short-time Fourier transform (STFT): given a frame of input signal $\mathbf{x} \in \mathbb{R}^M$, the $M$-point DFT matrix $\mathbf{F} \in \mathbb{C}^{M\times M}$, the high-pass filter $\mathbf{W} \in \mathbb{R}^{M\times M}$, and the nonlinear activation function $\sigma$, $1 < n \leq N$, the $n$-th-layer output of MLC is

$$\mathbf{z}^{(1)} = \sigma^{(1)}\left(\mathbf{W}^{(1)}|\mathbf{F}\mathbf{x}|\right),$$

$$\mathbf{z}^{(n)} = \sigma^{(n)}\left(\mathbf{W}^{(n)}\mathbf{F}\mathbf{z}^{(n-1)}\right).$$

It is noted that $\mathbf{z}^{(n)}$ is in the frequency domain when $n$ is odd and cepstral (i.e. time) domain when $n$ is even. The high-pass filter $\mathbf{W}^{(n)}$ aims to remove the slow-varying portion, that is, the low-frequency or low-quefrency components, on the assumption that they are irrelevant to the fundamental frequency. Then the nonlinear function:

$$\sigma(x) := \mathrm{ReLU}(x)^{\gamma}, \tag{1}$$

is an element-wise root-power operation to fit humans' perception scale [15], and ReLU represents the rectified linear unit function. Empirically, $n \geq 2$ and $0 < \gamma < 2$ encompasses a majority of signal representations for pitch estimation [16]. As an extension, [11] has demonstrated that MLC is able to refine the salience layer by layer, and it resembles the multi-layered perceptron as DFT acts like the fully-connected layer and $\sigma^{(n)}$ is analogue to the activation function. In this work, to model Equation (1), we further design a *gamma layer* (see Fig. 1) to learn this power-scaled mapping.

The CFP approach states that both time-domain and frequency-domain information is equally important and thus describes a pitch object as a composite of frequency, periodicity, and harmonicity [13], [14]. To leverage the strength of CFP, we pass the last two outputs from MLC as input for later deep learning networks. More specifically, the last two outputs, one spectral feature and one cepstral feature, are concatenated into a 2-channel feature map for the next stage. The deep learning network is a CNN with three convolution layers. Different from the previous layers in MLC and CFP, all the convolutional kernels are learnable. The first convolution layer is designed to extract the harmonic information along the frequency-axis, and the second convolution layer is performing temporal smoothing along the time-axis, while the last one is set to filter again and adjust the output shape. The output of the CNN is a multi-hot vector representing the pitch activation for each frame. The pitch range of the output is from `A0` to `C8`, totaling 88 semitones wide. The length of the output vector is $88r$, where $r$ is a factor to control the resolution: for example, in the fine-resolution mode of our system, we have

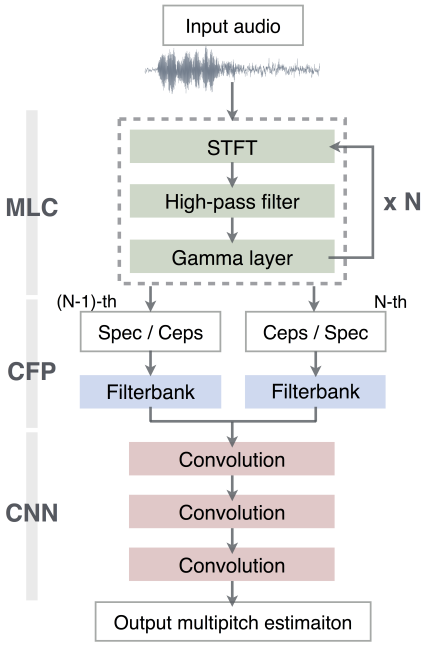


Fig. 1. A model architecture of the proposed harmonic preserving neural network (HPNN) for multipitch estimation.

pitch resolution set to a quarter of semitone, i.e., $r = 4$. See Section III-B for more detailed discussions.

## III. Experiment Settings

### A. Data

We use the MusicNet [6] and the MAPS [17] datasets to develop the proposed method. For cross-domain evaluation on speech data, we use the PTDB-TUG [18] to test the performance of models that are purely trained on music data.

The MusicNet dataset consists of 330 pieces of solo or multi-instrumental chamber music recordings, covering 11 kinds of instruments. The MAPS dataset contains 270 pieces of piano solo in nine different recording setups; seven kinds of audio tracks are obtained from piano synthesizers, and the remaining two kinds are real-world recordings.

For MusicNet, following the setting in [19], we use 314 pieces for training, 6 pieces for validation, and 3 pieces (id 2303, 1819, and 2382) for testing. As to MAPS, we adopt the *Configuration II* [20] to split the dataset. The model is trained on 210 tracks that are created with synthesized pianos, 180 pieces for training and 30 pieces for validation. For evaluation, the model is tested on the 60 tracks generated from authentic piano recordings, the ENSTDkAm and ENSTDkCl subsets.

The PTDB-TUG dataset collects recordings from ten male and ten female English native speakers from different home countries, reading 2,342 phonetically rich sentences taken from the TIMIT corpus [21]. For the ground-truth, we adopt the pitch trajectories extracted with Praat [22] provided in [23], rather than the reference provided along with the speech recordings.

TABLE I
DETAILS OF THE PROPOSED HPNN.

| Layer name | | Layer size | Output size | Trainable |
|---|---|---|---|---|
| MLC x N | STFT | 1, (8192, 8192) | 1, (17, 8192) | False |
| | High-pass filter | 1, (8192, 8192) | 1, (17, 8192) | False |
| | Gamma layer | – | 1, (17, 8192) | True |
| CFP | Filterbank | 1, (8192, 454) | 1, (17, 454) | False |
| | Concatenate | – | 2, (17, 454) | False |
| CNN | Conv2D | 24, (1, 101) | 24, (17, 354) | True |
| | Conv2D | 48, (17, 1) | 48, (1, 354) | True |
| | Conv2D | 1, (1, 4) stride=(1, 4) | 1, (1, 88) | True |

### B. Implementation

In this work, the DFT layer is implemented with Hann window as follows: we choose a frame size of 16,384 samples, and a hopsize of 512 inspired by [19]. Along with a window size of $M$ = 8,192, it results in $17 = 1+(16384-8192)/512$ number of regions per frame. That means, each training instance is a single-channel time-frequency image with size (17, 8192). For the high-pass filters, the cutoff frequency for spectrum is set to 27.5 Hz, the frequency of `A0`, and the cutoff quefrency for cepstrum is set to 0.24 ms, the period of `C8`. For CFP we employ the filterbank with a scalable pitch resolution and a range corresponding to the 88 piano keys plus 25 more semitones to cover the first four harmonics and sub-harmonics for each pitch. To divide a semitone by a factor of $r$, the number of triangular filters is $(88 + 25)r + 2$. The outputs of MLC-CFP are then successively convolved with kernels of shape $(1, (25r + 1))$, $(17, 1)$, and $(1, r)$. In the following experiments, the filterbank for CFP is implemented in two levels of resolutions: coarse and fine. The coarse one has $r = 1$, delivering a pitch resolution equal to one semitone, while the fine one has a $r = 4$, rendering an interval of 48 pitch bands per octave. And if hoping to down-sample the resolution back to one semitone, stride $(1, r)$ is applied to the last convolution layer.

For the CNN, the three types of kernels in the three convolution layers are designed to match the time-frequency patterns efficiently: the first is to learn the harmonic patterns in the frequency domain, the second is for temporal smoothing, and the third one directly applies stride convolution to merge the feature map to piano roll. The output vectors of the CNN is designed to have 88 and 352 pitch classes for inference on music and speech data, respectively. Model details in the fine-resolution mode are provided in Table I.

We also augment the training data by modulation and jitter following [19]. By random modulation, a pitch-shifting within $\pm$5 semitones, we enlarge the dataset by an order of magnitude while the modified audio is still perceptually natural. In addition, a continuous pitch-jitter is applied in a much smaller range of $\pm$0.1 semitone, which enables the model to be more robust to tuning variations.

These neural network models are implemented in PyTorch 1.0.1. The operation of DFT is directly implemented with the built-in function `torch.stft` in PyTorch. For the gamma layer, the power value $\gamma$ needs to be set as a trainable parameter, which is implemented by setting $\gamma$ as `Parameter` and its attribute `requires_grad` as `trainable`. For the convolution layers, the activation function is ReLU, and the model is optimized by minimizing the binary cross entropy. We train the CNN with the Adam optimizer with learning rate 0.001. All the models are trained on two NVIDIA GTX 1080Ti GPUs. The OS is Ubuntu 16.04 LTS with Xeon E5-2620 2.1GHz CPU, and the RAM is 64 GB in total. The source codes[1] are provided for reproducibility.

### C. Baseline Methods

Two MPE algorithms, one for music and the other for speech, are considered as the baseline methods. For music MPE, the state-of-the-art music transcription method based on an end-to-end translation-invariant network is included [19]. The network is designed to learn patterns that are invariant to translations in the frequency domain from raw audio signals. For speech MPE, there are still few studies using deep learning techniques, so we consider the spectral modeling method based on maximal likelihood (ML) and constrained clustering, proposed by Duan *et al.* [24]. The method is open-sourced and has the highest average accuracy among the three state-of-the-art methods for MPE in the two-talker scenario in [23].

### D. Evaluation Metrics

We report the precision (P), recall (R), and F1-score (F) with a common pitch tolerance of 0.5 semitone: a detection is true if it is deviated within $\pm$0.5 semitones from the ground truth. The P, R, and F are defined as $TP/(TP + FP)$, $TP/(TP + FN)$, and $2PR/(P + R)$, where TP is true positive, FP is false positive, and FN is false negative. When evaluating on music datasets (i.e., MAPS and MusicNet), the inferred pitch estimation results across the entire test data are evaluated with *mir_eval* [25], while for speech data (i.e., PTDB-TUG), these frame-level scores are computed clip by clip with the *mpa_eval* toolbox[2] and get averaged.

## IV. RESULTS

Four sets of experiments and results are presented in this section. First, the proposed model out of different training settings are evaluated and discussed on the polyphonic music data, MusicNet and MAPS. Second, to verify the robustness of the model to noise interference, we further evaluate the performance of MusicNet test set under noise contamination with various levels of signal-to-noise ratio (SNR). Third, to understand more about the behaviors of the HPNN, a parametric study is performed on how the number of the DFT layers (i.e. $N$) affects the resulting MPE performance. Lastly, to verify the robustness to cross-domain evaluation, we perform a cross-domain test that uses our model trained on a polyphonic music data (i.e. MusicNet train set) to test on the MPE of multi-talker speech data.

[1] https://github.com/brontosaurusJH/HPNN-multipitch-estimation
[2] http://www2.ece.rochester.edu/projects/air/resource.html

TABLE II
MULTI-PITCH ESTIMATION PERFORMANCE ON MUSIC DATASETS (IN %).

| Methods | | | Testing data | | | | | |
|---|---|---|---|---|---|---|---|---|
| Model | Training data | Data augmentation | MusicNet | | | MAPS | | |
| | | | P | R | F | P | R | F |
| HPNN-coarse | MusicNet | n/a | 55.52 | 74.65 | 63.68 | 69.27 | 75.59 | 72.29 |
| HPNN-fine | MusicNet | n/a | 59.33 | 72.98 | 65.45 | 72.37 | 71.31 | 71.84 |
| | MAPS | n/a | 47.94 | 62.78 | 54.36 | 69.95 | 74.40 | 72.11 |
| | MusicNet | Pitch-shift and jitter | 60.61 | 72.47 | 66.01 | 72.48 | 72.32 | 72.40 |
| Trans-inv [19] | MusicNet | Pitch-shift and jitter | 68.06 | 76.25 | 71.92 | 79.94 | 74.09 | 76.91 |

### A. MPE on Polyphonic Music

We evaluate four variations of the proposed model and the translation-invariant neural network (denoted as Trans-inv) [19] on the MusicNet and MAPS test sets. For the proposed models, the four variants of setup differ in pitch resolutions–coarse or fine, and training data–MAPS, MusicNet, or MusicNet with data augmentation. The cases of cross-dataset evaluation are presented in Table II, that is, the HPNN trained on MusicNet is evaluated on MAPS, and vice versa.

In this cross-dataset evaluation, each model achieves better results in MAPS no matter it is trained on MAPS or MusicNet. It results from that MusicNet is composed of multi-instrument recordings therefore more complicated, which also leads to that the model trained on MAPS has the lowest evaluation results on MusicNet. For models trained on MusicNet, higher pitch resolution raises the performances when evaluated on MusicNet, but not on MAPS. The MAPS dataset is composed of piano solos so to split an octave into 48 semitones is superfluous and hence increases the odds of making mistakes. In general, the HPNN trained on MusicNet has more robust performances when evaluated on both music datasets, and when evaluated on MAPS, it even outperforms the model trained on MAPS if the data augmentation is applied during the training process. On top of fine resolution, data augmentation further improves the performance.

As for the comparison to the baseline models, Table II shows that the F-score of the Trans-inv is 4.5–6% higher than our best model. However, the Trans-inv model is less efficient than the proposed one, as shown in Table III. More specifically, Table III summarizes the number of trainable parameters, the time consumed to finish the training process on the MusicNet training set, and the testing time. The training time is measured in seconds, while the testing time is measured with the proportion to the length of the input signals (e.g. 0.25x real time means the algorithm takes 15 seconds to estimate the pitch values for an input of one minute long). The number of trainable parameters of our model is at the order of $10^5$ while the Trans-inv is $10^9$ and therefore the clear contrast to computing power demands. For the training time, the Trans-inv model takes more than eight times longer to finish the training process. Though the two models might take different number of epochs to converge to the same criterion, a more detailed look at the training process shows that we train our model with 16k steps and it takes less than one hour. With the same computation power and learning rate, Trans-inv takes one more hour to finish the same amount of steps of training, and in fact it requires around 64k steps to complete the training process so it needs about 7.5 hours. For the testing time, we find that when inferring on the same audio clip, our model is by 1.75 times faster than Trans-inv on average.

TABLE III
COMPARISON OF MODEL EFFICIENCY.

| | HPNN-Fine | Trans-inv [19] | Ratio |
|---|---|---|---|
| Number of trainable parameters | 14,743 | 114,311,168 | 7,754 |
| Training time (seconds) | 3,251 | 26,782 | 8.24 |
| Testing time (prop. real time) | 0.134x | 0.235x | 1.75 |

TABLE IV
RESULTS OF NOISE ROBUSTNESS TESTS IN TERMS OF F-SCORE (IN %).

| Methods | Clean | SNR | | | |
|---|---|---|---|---|---|
| | | 30dB | 20dB | 10dB | 0dB |
| HPNN-fine | 66.01 | 64.38 | 64.1 | 62.68 | 57.24 |
| Trans-inv [19] | 71.92 | 65.43 | 53.1 | 38.28 | 24.59 |

### B. Noise-robustness Test

To evaluate the robustness of the proposed model to noise interference, we conduct a series of tests for the HPNN-fine model[3] and the Trans-inv model on the MusicNet test set under noise contamination with different levels of SNR. We consider different levels of additive pink noise, adapting the test data under five conditions, where the SNR values are $\infty$ dB (clean), 30 dB, 20 dB, 10 dB, and 0 dB. The simulated pink noise is provided by the `python-acoustics` package[4] and is frame-wise mixed with the test signals.

The results presented in Table IV show that Trans-inv outperforms HPNN-fine by 5.8% under clean condition, but such a lead drops to only 1.05% (64.38% for HPNN-fine and 65.43% for Trans-inv) when SNR = 30 dB, which is a condition still better than many real-world environments. As the SNR value keeps declining, the performances of Trans-inv drop even more rapidly. From clean testing data to SNR being 0 dB, the F-score decreases by less than 9% for the proposed model, while Trans-inv suffers from a loss more than 45%. Such robustness to noise is mainly contributed by the DFT layers and the nonlinear activation function; the former guides the model to focus on modeling the harmonic/ periodic structures which are truly relevant to pitch in the feature

[3]Hereafter the HPNN-fine narrows to the proposed architecture trained on the augmented MusicNet with fine resolution.

[4]https://github.com/python-acoustics/python-acoustics

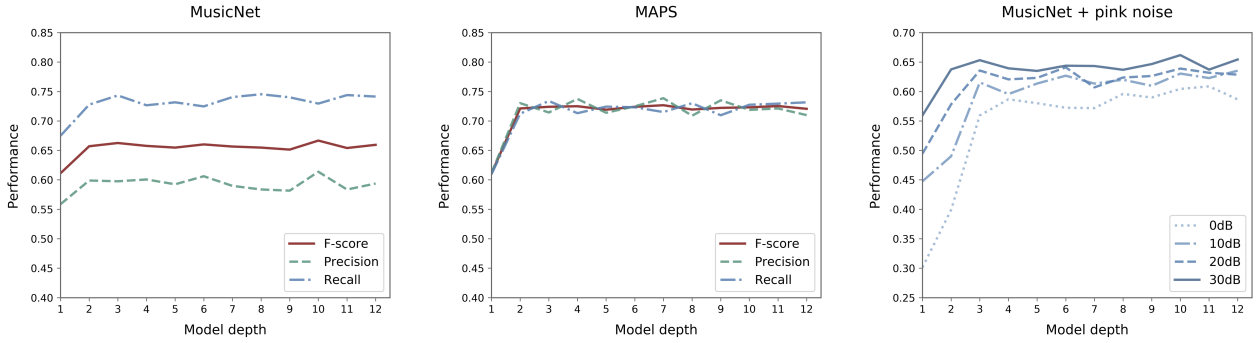


Fig. 2. Performances as a function of model depth. We recommend a depth of six layers of MLC as it finds a balance between performances and computing resources.

maps, while the latter adjusts the relative scales in the feature maps such that the contribution of each concurrent pitch to the feature maps is balanced. In contrast, though [19] reports the convolution kernels in the first layer are observed to be DFT-like, such data-dependent kernels however are sensitive to noise. To summarize, in this audio degradation experiment, the proposed model better survives in low SNR scenarios than the other model that is more end-to-end. The proposed model is not only more economic, but also more robust for it applies Fourier basis as priors.

### C. Parametric Studies on Number of Layers

Fig. 2 shows the performances of the HPNN-fine models separately trained with a series of different numbers of MLC depths, $N$, from one ($N = 1$) to twelve ($N = 12$) layers, and then evaluated under the afore-mentioned three testing setups (i.e. MusicNet, MAPS, and MusicNet under different levels of pink noise contamination), as the title of each panel indicates. The left and middle panels show the F1-score, precision, and recall for MusicNet and MAPS, respectively. The right panel illustrates the F-score for MusicNet contaminated with various levels of additive pink noise as described in Section IV-B.

We find that the performances approach the optima for $N = 2$ in general, though improved performances can still be observed when the number of layers is increased. Taking the case of 0-dB SNR pink noise (the dotted line in the right panel at 2) as an example, the F1-score increases gradually with growing $N$ and achieves the maximum at $N = 11$. Deeper models might yield better performances, but that more resources lead to only marginal gains is not desirable. It can also be noted that the distances between precision and recall reach the smallest value at $N = 6$ for the MusicNet and MAPS datasets. Consequently, we recommend choosing a depth of six layers ($N = 6$) as a tradeoff between performances and computing resources. In our experiments, for $N = 6$, the learned $\gamma$ values from the the augmented HPNN-fine model trained on the MusicNet dataset are $(\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5, \gamma_6) = (0.0960, 0.6376, 0.4864, 0.6746, 0.6428, 0.3259)$. The learned values are different from what reported in previous works, for example, $(\gamma_1, \gamma_2, \gamma_3) = (0.24, 0.6, 1.0)$ in [16] and $(\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5, \gamma_6) = (0.1, 0.9, 0.9, 0.7, 0.8, 0.5)$ in [11]. However, these values are common in that 1) $\gamma_1$ is the smallest among all, and 2) all the $\gamma$ values are smaller than 1. This suggests that the power-scale parameters learned from data also follow the empirical rules mentioned in the literature of signal processing [11], [15], [16], while appear as an optimal solution that fits the training data.

TABLE V
RESULTS OF CROSS-DOMAIN EVALUATION (IN %).

| Methods | P | R | F |
|---|---|---|---|
| HPNN-fine | 49.38 | 46.89 | 47.67 |
| Duan *et al.* [24] | 67.41 | 53.98 | 59.72 |

Fig. 3 illustrates six examples, which are the output piano rolls of the HPNN-fine models trained with $N$ MLC layers, where $N = 1, 2, 3, \cdots, 6$. The frequency unit is in semitone. This example demonstrates the enhancement of pitch saliency grows with model depth $N$. We observe that when $N = 1$ (i.e. the upper-left sub-figure in Fig. 3), there are numbers of unwanted harmonics and also sub-harmonic components appearing on the feature map. These unwanted terms are greatly reduced for $N \geq 2$, while the salience of desired fundamental frequencies is preserved. The output piano rolls become more succinct as $N$ increases till $N = 6$.

### D. Domain-robustness Test

To further test the robustness of the proposed model on the variation of testing data, we consider a more challenging scenario: using the proposed MPE model trained on music to test on multi-talker speech signals. To find out how our proposed model that is trained on music performs on speech, we evaluate the HPNN-fine model on the PTDB-TUG and compare it with the model [24] proposed by Duan *et al.*, which is one of the MPE models that has been evaluated on multi-talker speech signals. To test the models in a multi-talker scenario, following [18], we choose two female (F01, F07) and two male (M04, M10) speakers from the PTDB-TUG. Among the rendered six speaker pairs, we randomly mix 100 utterances by equalizing the maximum of the magnitude and thus form a test set of 600 segments of two-talker speech

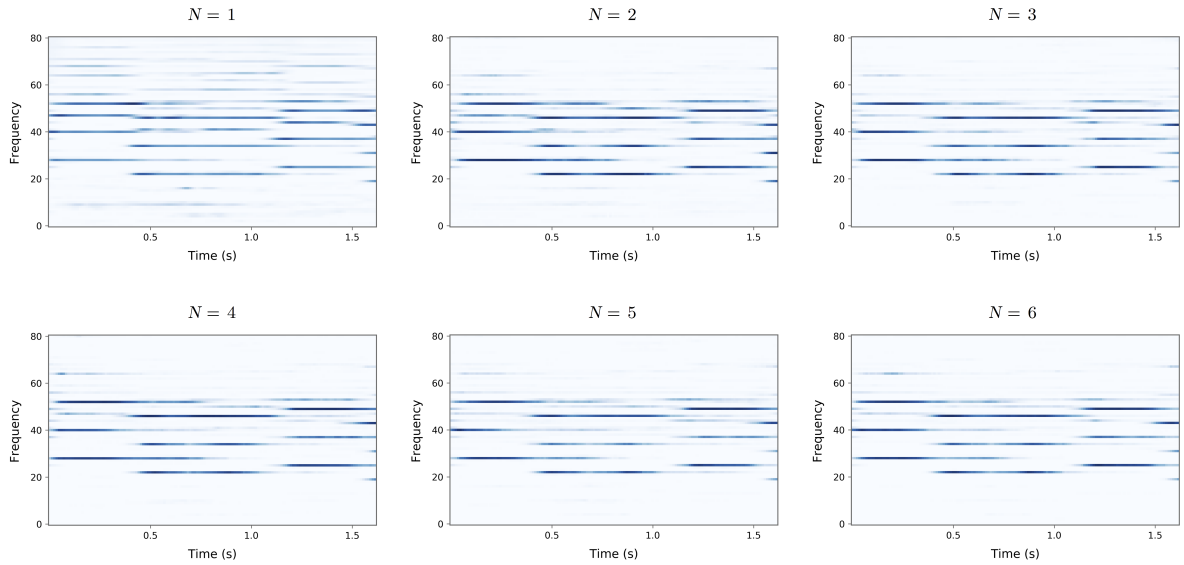


Fig. 3. An example of inference outputs of the HPNN-fine models with MLC depths from $N = 1$ to $N = 6$, on an excerpt of around 1.5 second from W.A. Mozart's *Serenade in E-flat Major*, K. 375, a wind quintet selected from MusicNet id 1819.

mixtures. Each utterance is cut at 70ms before the first and after the last occurrence of pitch values in the reference annotation. Both models are ignorant of the speaker number. To our knowledge, a cross-domain MPE task (i.e. music-to-speech MPE) has been rarely discussed in the literature except [23], though the work focuses more on the streaming of ideal MPE results rather than the MPE task itself.

The results presented in Table V quantifies the generalization ability of cross-domain MPE: it reaches an F1-score lower than [24] by around 12%, but can still correctly predict around half of the pitch values. The performance gap might stem from the consonants in speech, different behaviors of pitch contours between music and speech, and the irregular unvoiced intervals in speech, all of which exist in speech data but are more irrelevant to the harmonic structures of signals on which the HPNN model designed to focus.

## V. Conclusions

We have implemented the proposed HPNN method for multi-pitch estimation, and assessed the performance of the network under various scenarios. The HPNN method is efficient in time, robust to noise and cross-domain data, and still exhibits competitive F1-scores in comparison to state-of-the-art methods. The proposed method also represents a case that incorporates domain knowledge and data-driven deep learning techniques to tackle a problem. Since a performance gap between the HPNN and state-of-the-art methods still exists, the first task which needs further improvement will be to reduce the gap. Possible future directions therefore include the incorporation with a larger-scale CNN for optimization, using feature maps from more than the last two DFT layers, and considering other types of neural networks such as the U-Net [26], [27] for an output with a wider temporal context.

## References

[1] Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, and Anssi Klapuri, "Automatic music transcription: challenges and future directions," *J. Intelligent Information Systems*, vol. 41, no. 3, pp. 407–434, 2013.

[2] Mingyang Wu, DeLiang Wang, and Guy J Brown, "A multipitch tracking algorithm for noisy speech," *IEEE Trans. Audio, Speech, Lang. Proc. (TASLP)*, vol. 11, no. 3, pp. 229–241, 2003.

[3] Li Su and Yi-Hsuan Yang, "Escaping from the abyss of manual annotation: New methodology of building polyphonic datasets for automatic music transcription," in *International Symposium on Computer Music Multidisciplinary Research*. Springer, 2015, pp. 309–321.

[4] Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse H. Engel, and Douglas Eck, "Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset," in *7th International Conference on Learning Representations (ICLR)*, 2019.

[5] Curtis Hawthorne, Erich Elsen, Jialin Song, Adam Roberts, Ian Simon, Colin Raffel, Jesse Engel, Sageev Oore, and Douglas Eck, "Onsets and frames: Dual-objective piano transcription," in *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, 2018.

[6] John Thickstun, Zaid Harchaoui, and Sham Kakade, "Learning features of music from scratch," in *International Conference on Learning Representations (ICLR)*, 2017.

[7] Estefanía Cano, Christian Dittmar, Jakob Abeßer, Christian Kehling, and Sascha Grollmisch, "Music technology and education," in *Springer Handbook of Systematic Musicology*, pp. 855–871. Springer, 2018.

[8] Pablo A Alvarado and Dan Stowell, "Efficient learning of harmonic priors for pitch detection in polyphonic music," *arXiv preprint arXiv:1705.07104*, 2017.

[9] Ken O'Hanlon and Mark B Sandler, "The fifthnet chroma extractor," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3752–3756.

[10] Rachel M Bittner, Brian McFee, Justin Salamon, Peter Li, and Juan Pablo Bello, "Deep salience representations for f0 estimation in polyphonic music.," in *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017, pp. 63–70.

[11] Chin-Yun Yu and Li Su, "Multi-layered cepstrum for instantaneous frequency estimation," in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2018, pp. 276–280.

[12] G. Peeters, "Music pitch representation by periodicity measures based on combined temporal and spectral representations," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2006.
[13] L. Su and Y.-H. Yang, "Combining spectral and temporal representations for multipitch estimation of polyphonic music," *IEEE/ACM Trans. Audio, Speech, Lang. Proc. (TASLP)*, vol. 23, no. 10, pp. 1600–1612, 2015.
[14] L. Su, T.-Y. Chuang, and Y.-H. Yang, "Exploiting frequency, periodicity and harmonicity using advanced time-frequency concentration techniques for multipitch estimation of choir and symphony," in *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, 2016, pp. 393–399.
[15] Patrice Alexandre and Philip Lockwood, "Root cepstral analysis: A unified view. application to speech processing in car noise environments," *Speech Communication*, vol. 12, no. 3, pp. 277–288, 1993.
[16] Li Su, "Between homomorphic signal processing and deep neural networks: Constructing deep algorithms for polyphonic music transcription," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2017, pp. 884–891.
[17] Valentin Emiya, Roland Badeau, and Bertrand David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Trans. Audio, Speech, Lang. Proc. (TASLP)*, vol. 18, no. 6, pp. 1643–1654, 2010.
[18] Gregor Pirker, Michael Wohlmayr, Stefan Petrik, and Franz Pernkopf, "A pitch tracking corpus with evaluation on multipitch tracking scenario," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2011.
[19] John Thickstun, Zaid Harchaoui, Dean P Foster, and Sham M Kakade, "Invariances and data augmentation for supervised music transcription," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 2241–2245.
[20] Siddharth Sigtia, Emmanouil Benetos, and Simon Dixon, "An end-to-end neural network for polyphonic piano music transcription," *IEEE/ACM Trans. Audio, Speech, Lang. Proc. (TASLP)*, vol. 24, no. 5, pp. 927–939, 2016.
[21] John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett, "Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1," *NTIS. order number PB01–100354*, vol. 93, pp. 27403, 1993.
[22] Paul Boersma, "Praat, a system for doing phonetics by computer," *Glot. Int.*, vol. 5, no. 9, pp. 341–345, 2001.
[23] Zhiyao Duan, Jinyu Han, and Bryan Pardo, "Multi-pitch streaming of harmonic sound mixtures," *IEEE/ACM Trans. Audio, Speech, Lang. Proc. (TASLP)*, vol. 22, no. 1, pp. 138–150, 2014.
[24] Zhiyao Duan, Bryan Pardo, and Changshui Zhang, "Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions," *IEEE Trans. Audio, Speech, Lang. Proc. (TASLP)*, vol. 18, no. 8, pp. 2121–2133, 2010.
[25] 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 18th International Society for Music Information Retrieval Conference (ISMIR)*. Citeseer, 2014.
[26] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
[27] Andreas Jansson, Eric J. Humphrey, Nicola Montecchio, Rachel M. Bittner, Aparna Kumar, and Tillman Weyde, "Singing voice separation with deep u-net convolutional networks," in *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017, pp. 745–751.