Speaker-dependent multipitch tracking using deep neural networks

Yuzhou Liu[a] and DeLiang Wang[b]
Department of Computer Science and Engineering, The Ohio State University, Columbus, Ohio 43210, USA

(Received 14 March 2016; revised 15 December 2016; accepted 22 December 2016; published online 1 February 2017)

Multipitch tracking is important for speech and signal processing. However, it is challenging to design an algorithm that achieves accurate pitch estimation and correct speaker assignment at the same time. In this paper, deep neural networks (DNNs) are used to model the probabilistic pitch states of two simultaneous speakers. To capture speaker-dependent information, two types of DNN with different training strategies are proposed. The first is trained for each speaker enrolled in the system (speaker-dependent DNN), and the second is trained for each speaker pair (speaker-pair-dependent DNN). Several extensions, including gender-pair-dependent DNNs, speaker adaptation of gender-pair-dependent DNNs and training with multiple energy ratios, are introduced later to relax constraints. A factorial hidden Markov model (FHMM) then integrates pitch probabilities and generates the most likely pitch tracks with a junction tree algorithm. Experiments show that the proposed methods substantially outperform other speaker-independent and speaker-dependent multipitch trackers on two-speaker mixtures. With multi-ratio training, the proposed methods achieve consistent performance at various energies ratios of the two speakers in a mixture.
 [http://dx.doi.org/10.1121/1.4973687]

[JFL] Pages: 710–721

I. INTRODUCTION

There is a long-standing interest in estimating the pitch, or the fundamental frequency ($F0$) of speech. A reliable estimation of pitch is critical for many speech processing applications, including automatic speech recognition for tonal languages (Chen *et al.*, 1997), speaker identification (Zhao *et al.*, 2012) and speech separation (Wang and Brown, 2006). Over the last few decades, various algorithms have been designed for tracking the pitch of a single speaker (Talkin, 1995; Boersma, 2001; Cheveigné and Kawahara, 2002), and they achieve good performance under clean or modestly noisy conditions. However, pitch tracking when speech is severely corrupted by interfering speakers is still a challenging problem.

This paper is concerned with multipitch tracking when two speakers are talking simultaneously in a monaural (single-microphone) recording. Although microphone-array approaches are widely used for multi-source tracking, monaural solutions are easier to apply and may complement array-based techniques. A number of studies have investigated the problem of monaural multipitch tracking. Wu *et al.* (2003) propose a probabilistic representation of pitch and tracked continuous pitch contours with a hidden Markov model (HMM). Sha and Saul (2005) model the instantaneous frequency spectrogram with nonnegative matrix factorization (NMF) and use the inferred weight coefficients to determine pitch candidates. Bach and Jordan (2005) propose direct probabilistic modeling of the spectrogram and track several pitches with a factorial HMM (FHMM). Christensen and Jakobsson (2009) describe statistical, filtering and subspace methods for both single- and multi-pitch estimation. Hu and Wang (2010) propose a tandem algorithm that performs pitch estimation and voiced speech segregation jointly, producing a set of pitch contours and their associated binary masks. Jin and Wang (2011) improve the system by Wu *et al.* (2003) by designing new techniques for channel selection and pitch score estimation in the context of reverberant and noisy signals. The abovementioned studies build a general system without modeling the characteristics of any specific speaker, and can thus be denoted as speaker-independent models. Although most speaker-independent models perform well for estimating pitch periods, they can not assign pitch estimates to the underlying speakers for multipitch tracking. To alleviate this problem, Hu and Wang (2013) build their system on the tandem algorithm (Hu and Wang, 2010) and group simultaneous pitch contours into two speakers using a constrained clustering algorithm. Similarly, Duan *et al.* (2014) take the pitch estimates of speaker-independent multipitch trackers as input and stream pitch points by clustering. However, both approaches achieve limited improvement as individual pitch contours and points are usually too short to contain enough speaker information for clustering. On the other hand, speaker-dependent models have been investigated recently. Wohlmayr *et al.* (2011) model the probability of pitch periods using speaker-dependent Gaussian mixture models (GMMs), and then use a speaker-dependent FHMM to track pitches of two simultaneous speakers. They have shown significant improvement over a speaker-independent approach (Wu *et al.*, 2003).

In this paper, we propose a speaker-dependent and discriminative technique to model the pitch probability at each

[a]Electronic mail: liuyuz@cse.ohio-state.edu
[b]Also at Center for Cognitive and Brain Sciences, The Ohio State University, Columbus, Ohio 43210, USA.

time frame. Specifically, we use deep neural networks (DNNs) to model the posterior probability that a pair of frequency bins (pitch states) is pitched given frame-level observations. A DNN is a feedforward neural network that contains more than one hidden layer (Hinton *et al.*, 2006). Recently, Han and Wang (2014) use DNNs to model the posterior probability of pitch states for single-pitch tracking in noisy conditions, which motivates the use of DNNs for multipitch tracking in this study. To leverage individual speaker characteristics, we train a DNN for each speaker enrolled in the system, denoted as speaker-dependent DNNs or SD-DNNs. We also train DNNs for different pairs of speakers, denoted as speaker-pair-dependent DNNs or SPD-DNNs. We then extend the DNN based models to relax practical constraints. To deal with unseen speakers, we train three gender-pair-dependent DNNs (male-male, male-female, and female-female, denoted as GPD-DNNs) as a generalization of SPD-DNNs. GPD-DNNs only require gender information during testing. With insufficient training data, direct training of SD-DNNs or SPD-DNNs may result in overfitting. To examine this issue, we conduct a fast adaptation of GPD-DNNs for each speaker pair with limited training data. Also, the utterances of the two speakers in a mixture usually have different energy ratios, leading to a ratio mismatch between training and test. We address this problem by including various speaker energy ratios in training, denoted as the multi-ratio training.

After estimating the posterior probability of pitch states, we use an FHMM for pitch tracking. Under the framework of the FHMM, the pitch state of each speaker evolves within its own Markov chain, while the emission probability is derived by the posterior probability estimated by DNNs. We then use the junction tree algorithm (Jordan *et al.*, 1999) to infer the most likely pitch tracks.

The rest of the paper is organized as follows. Section II gives an overview of the system architecture. Feature extraction is discussed in Sec. III. The details of DNN based posterior probability estimation are introduced in Sec. IV. Section V describes the FHMM for multipitch tracking. Experimental results and comparisons are presented in Sec. VI. Finally, we conclude the paper and discuss related issues in Sec. VII. A preliminary version of this paper (Liu and Wang, 2015a) was presented at Interspeech 2015. This paper extends Liu and Wang (2015a) in major ways, including gender-dependent DNNs, model adaptation, and multi-ratio training.

II. SYSTEM OVERVIEW

A diagram of our proposed multipitch tracker is illustrated in Fig. 1. The input to the system is a speech mixture v_t sampled at 16 KHz,

$$v_t = u_t^1 + u_t^2, \tag{1}$$

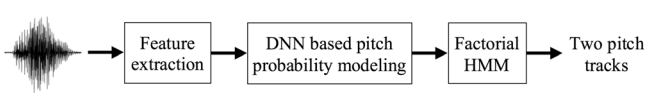

FIG. 1. Diagram of the proposed multipitch tracker.

where u_t^1 and u_t^2 are utterances of two speakers. Given the mixture, our system first extracts frame-level features $\mathbf{y}_m$ with a frame shift of 10 ms, which corresponds to the first module in the diagram.

In the second stage, features are fed into DNNs to derive the posterior probability of pitches at frame m, i.e., $p(x_m^1, x_m^2|\mathbf{y}_m)$, where x_m^1 and x_m^2 denote pitch states of two speakers at frame m. Both x_m^1 and x_m^2 have 68 states (s^1, s^2, s^3,…, s^{68}), where s^1 refers to an unvoiced or silent state, and s^2 to s^{68} encode different pitch frequencies ranging from 60 to 404 Hz (Han and Wang, 2014). Specifically, we quantize the pitch frequency range 60 to 404 Hz using 24 bins per octave on a logarithmic scale, resulting in a total of 67 bins. This frequency resolution provides two bins per semitone, and gives less than 3% of relative frequency difference between adjacent pitch states, adequate for continuous pitch tracking. $p(x_m^1 = s^i, x_m^2 = s^j|\mathbf{y}_m)$ equals one if groundtruth pitches fall into the ith and jth frequency bins, respectively. We propose two types of DNN to estimate the posterior probability, which are the speaker-dependent DNNs and the speaker-pair-dependent DNNs. We also explore several extensions. The detailed settings of DNNs can be found in Sec. IV.

The final module converts the posterior probability $p(x_m^1, x_m^2|\mathbf{y}_m)$ to the emission probability of an FHMM $p(\mathbf{y}_m|x_m^1, x_m^2)$. The junction tree algorithm is then applied to infer the most likely pitch tracks. Note that in the following sections, a pitch contour refers to a continuous pitch trajectory from the same speaker, and a pitch track refers to a set of pitch contours from the same speaker.

III. FEATURE EXTRACTION

Features should encode the information of pitch and speaker identity at the same time. We compare three features: cochleagram, log spectrogram, and mel-frequency cepstral coefficients in our study. Cochleagram and log spectrogram are signal representations shown to be effective for speech separation, automatic speech recognition and speaker recognition. Unique harmonic structure of each speaker is reflected in both cochleagram and log spectrogram. Mel-frequency cepstral coefficients (MFCCs) are widely used in speech processing, and they are investigated here as a representative cepstral feature found to be useful for pitch estimation long ago (Noll, 1967).

A. Cochleagram

To get the cochleagram feature, we first decompose the input signal in time-frequency domain by using a bank of 64 gammatone filters whose center frequencies range from 50 to 8000 Hz. Gammatone filters model the impulse responses of auditory filters and are widely used (Holdsworth *et al.*, 1988). We divide each subband signal into 20 ms frames with a 10 ms frame shift. The cochleagram is derived by computing the energy of each subband signal at each frame. We then loudness compress the cochleagram with a cubic root operation to get the final 64-dimensional cochleagram feature (for a MATLAB implementation see OSU Perception and Neurodynamics Lab, 2008).

B. Log spectrogram

To get the spectrogram feature, the signal is first divided into 32 ms frames with a 10 ms frame shift. The frame length of log spectrogram is longer than that of the other two features in order to produce a finer resolution of the frequency axis. We then apply a Hamming window to each frame and derive the spectrogram using 1024-point FFT. Last, we compute the logarithm of the amplitude spectrum, and pick bins 2-65 (corresponding to a frequency range up to 1000 Hz) as our frame-level feature vector. The dimensionality of this feature is 64, and it is proposed by Wohlmayr *et al.* (2011) in their GMM-FHMM based multipitch tracker.

C. Mel-frequency cepstral coefficients

To compute MFCCs, we divide the input signal into 20 ms frames with a 10 ms frame shift. The power spectrogram is derived using short-time Fourier transform filtered by a Hamming window. Next we use a bank of 64 mel scale filters to convert the power spectrogram into mel scale. Last, logarithm compression and discrete cosine transform are applied to compute 31-dimensional MFCCs (Brookes, 2011).

D. Incorporating temporal context

To make use of the temporal context, we concatenate neighboring frames into one feature vector. Denoting the feature vector extracted within frame m as $\hat{\mathbf{y}}_m$, we have

$$\mathbf{y}_m = [\hat{\mathbf{y}}_{m-d}, \ldots, \hat{\mathbf{y}}_m, \ldots, \hat{\mathbf{y}}_{m+d}], \tag{2}$$

where d is chosen to be 5 (see Sec. IV B).

IV. DNN BASED PITCH PROBABILITY MODELING

DNNs have been successfully applied in various speech processing applications. In this section, we first introduce two types of DNN for posterior probability estimation. Next we extend the models to relax practical constraints.

A. Speaker-dependent DNNs

The goal of DNNs is to model the posterior probability that a pair of pitch states occurs at frame m, i.e., $p(x_m^1, x_m^2|\mathbf{y}_m)$. However, this would be difficult without the prior knowledge of the underlying speakers. We first focus on training speaker-dependent DNNs to model the posterior probability.

According to the chain rule in probability theory,

$$p(x_m^1, x_m^2|\mathbf{y}_m) = p(x_m^1|\mathbf{y}_m)p(x_m^2|x_m^1, \mathbf{y}_m), \tag{3}$$

we can estimate $p(x_m^1|\mathbf{y}_m)$ and $p(x_m^2|x_m^1, \mathbf{y}_m)$ in turn to get $p(x_m^1, x_m^2|\mathbf{y}_m)$. In this study, we estimate the pitch-state probability of speaker one $p(x_m^1|\mathbf{y}_m)$ by training a DNN. The input layer of the DNN corresponds to the frame-level feature vector of the mixture. There are four hidden layers in the DNN, and each hidden layer has 1024 rectified linear units (ReLU) (Glorot *et al.*, 2011). The reason we choose ReLU instead of sigmoid is that it alleviates the overfitting problem, leading to faster and more effective training/adaptation. The output layer has 68 softmax output units, denoted as $(O_1^1, O_1^2, \ldots, O_1^{68})$, where O_1^j estimates $p(x_m^1 = s^j|\mathbf{y}_m)$. Hence there are sixty-seven 0s and a 1 in the desired output. The value of 1 corresponds to the frequency bin of the groundtruth pitch. We use cross-entropy as the cost function. The standard backpropagation algorithm and dropout regularization (Hinton *et al.*, 2012) are used to train the network, with no pretraining. We adopt mini-batch stochastic gradient descent along with a momentum term (0.9) for the optimization. The choice of DNN parameters is justified in Sec. VI B. The training data contain mixtures of speaker one and a set of interfering speakers.

Figure 2 compares the groundtruth and estimated pitch-state probabilities of speaker one in a female-female test mixture. As shown in the figure, the DNN rather accurately models the conditional probability of x_m^1, even without knowing x_m^2. Therefore the same type of DNN can be used to model $p(x_m^1|x_m^2, \mathbf{y}_m)$ or $p(x_m^2|x_m^1, \mathbf{y}_m)$.

In the next step, we train another DNN to model $p(x_m^2|x_m^1, \mathbf{y}_m)$ using exactly the same structure and training methodology as for the first DNN. The output of the DNN is denoted as $(O_2^1, \ldots, O_2^{68})$. The original posterior probability $p(x_m^1, x_m^2|\mathbf{y}_m)$ can then be obtained by

$$p(x_m^1 = s^i, x_m^2 = s^j|\mathbf{y}_m) = O_1^i O_2^j. \tag{4}$$

Because we train a DNN for each enrolled speaker, we denote this model as the speaker-dependent DNN (SD-DNN).

B. Speaker-pair-dependent DNNs

A speaker-pair-dependent DNN (SPD-DNN) is a DNN trained on a specific pair of speakers. The structure of an SPD-DNN is quite similar to that of an SD-DNN. The input layer corresponds to the frame-level feature vector. There are four hidden layers with 1024 ReLU units. Instead of estimating the probability for only one speaker, we concatenate

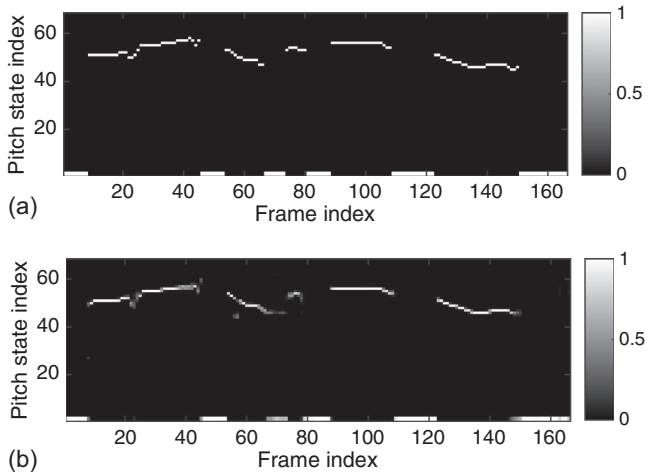

FIG. 2. Pitch probability modeling of the first speaker in a female-female mixture at 0 dB. (a) Groundtruth probabilities of pitch states. (b) Probabilities of pitch states estimated by a DNN.

the pitch-state probabilities of the other speaker into the DNN output. The resulting output layer has 136 units, denoted as $(O_1^1, \ldots, O_1^{68}, O_2^1, \ldots, O_2^{68})$. To correctly model the probability distribution, the activation function of the output layer is a softmax function. Assuming that output units before applying the activation function have values $(v_1^1, \ldots, v_1^{68}, v_2^1, \ldots, v_2^{68})$, we have

$$O_i^j = \frac{\exp\left(v_i^j\right)}{\sum_{k=1}^{68} \exp\left(v_i^k\right)} \quad \text{for} \quad i = 1 \text{ or } 2, \quad 1 \le j \le 68. \tag{5}$$

Other training details exactly follow SD-DNNs. The posterior probability of pitch states is estimated by

$$p(x_m^1 = s^i, x_m^2 = s^j | \mathbf{y}_m) = O_1^i O_2^j. \tag{6}$$

Because SPD-DNNs are trained on speaker pairs, they should accurately capture the underlying speaker information. On the other hand, for a system with N speakers enrolled, we need to train N SD-DNNs, but $[N(N-1)]/2$ SPD-DNNs.

C. Extensions

SD-DNNs and SPD-DNNs utilize detailed speaker information to estimate the posterior probability of pitch states. In this section, we introduce extensions to relax their practical constraints.

1. Gender-pair-dependent DNN

SD-DNNs and SPD-DNNs are not applicable to unseen speakers. To deal with this constraint, we extend our speaker-dependent models to gender-dependent ones. In this way, only the genders of the two underlying speakers are needed during testing.

A straightforward way to design a gender-dependent model is to follow the structure of SD-DNNs and train two DNNs for male and female speakers, respectively. This idea works well for male-female mixtures, but can not distinguish the two speakers of the same gender. Therefore we build our gender-dependent model by extending SPD-DNNs to gender-pair-dependent DNNs or GPD-DNNs. We train three GPD-DNNs for different gender pairs: male-female, male-male, and female-female. The structure of a GPD-DNN is chosen to be the same as an SPD-DNN for simplicity. For the male-female GPD-DNN, the pitch-state probabilities of the male speaker correspond to the first 68 output units, and the female speaker the remaining output units. For same-gender GPD-DNNs, the first 68 output units correspond to the speaker with lower average pitch, and the other output units correspond to the speaker with higher average pitch. Although this layout may lead to incorrect speaker assignment at some frames, it provides a reasonable way to distinguish two speakers with the little information available. Other training aspects exactly follow SPD-DNNs.

2. Adaptation of GPD-DNNs with limited training data

SD-DNNs and SPD-DNNs would overfit if we could not collect enough training data. One way to address this problem is to perform speaker adaptation of GPD-DNNs with limited data. Speaker adaptation of DNNs has been studied in automatic speech recognition. Two typical approaches include incorporating speaker-dependent information into DNN's input (Abdel-Hamid and Jiang, 2013; Saon *et al.*, 2013) and regularized retraining (Liao, 2013; Yu *et al.*, 2013). In the first approach, speaker-dependent information, like i-vectors and speaker codes, is incorporated into the input of DNNs and the original features are projected into a speaker-normalized space. In regularized retraining, the weights of DNNs are modified using the adaptation data. To ensure that the adapted model does not deviate too much from the original model, a regularization term is added to the cost function. Both approaches substantially improve the performance of unadapted DNNs.

We use regularized retraining to perform speaker adaptation. For each new speaker pair, we retrain all the weights of the corresponding GPD-DNN on limited adaptation data with a relatively small learning rate (0.001) and a weight decay (L_2 regularization) of 0.0001. Other training aspects follow those for training SPD-DNNs.

3. Multi-ratio training

Utterances of the two speakers in a mixture usually have different energy ratios. A ratio mismatch between training and test may result in performance degradation for supervised algorithms. Under the framework of GMM-FHMM, Wohlmayr and Pernkopf (2011) alleviate this problem by adding a gain parameter to the mean vectors of each GMM. An expectation-maximization based algorithm is then performed to estimate the gains for each test mixture. They further extend the EM-like framework to adapt model parameters to unseen acoustic environments and speakers (Wohlmayr and Pernkopf, 2013). However, it is unclear how to apply these techniques to DNNs.

Generally speaking, the performance of supervised learning is sensitive to the information contained in the training set. Therefore, a simple and effective way for improving generalization is to enlarge the training set by including various acoustic conditions (Chen *et al.*, 2016). In this study, we perform multi-condition training by creating mixtures at different speaker energy ratios, denoted as multi-ratio training. The resulting DNNs are denoted as ratio-adapted DNNs. The details of multi-ratio training are given in Sec. VI.

V. FACTORIAL HMM INFERENCE

Once all posterior probabilities are estimated by DNNs, we use a factorial HMM to infer the most likely pitch tracks. A factorial HMM is a graphical model that contains several Markov chains (Ghahramani and Jordan, 1997). In this study, we only discuss the case of two Markov chains, as shown in Fig. 3.

The hidden variables (x_m^1, x_m^2) are the pitch states of two speakers, and the observation variable is the feature vector

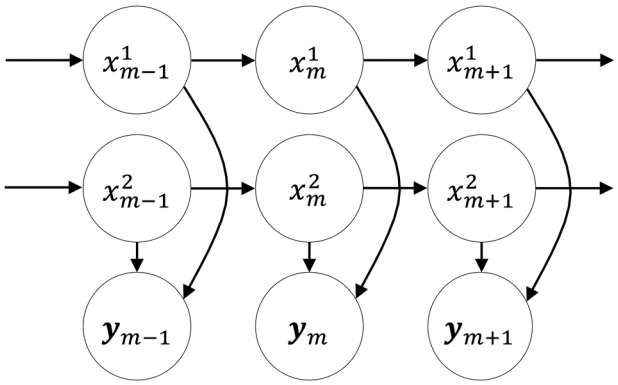

FIG. 3. A Factorial HMM with two Markov chains.

$\mathbf{y}_m$. The Markov assumption implies that $\mathbf{y}_m$ is independent of all variables given x^1_m and x^2_m. Assuming the total number of frames is M, we denote the sequence of variables in boldface: $\mathbf{X} = \cup_{m=1}^{M}\{x^1_m, x^2_m\}$, $\mathbf{Y} = \cup_{m=1}^{M}\{\mathbf{y}_m\}$. The overall joint probability of the model is given by

$$p(\mathbf{X},\mathbf{Y}) = p(x^1_1)p(x^2_1)p(\mathbf{y}_1|x^1_1,x^2_1)\prod_{m=2}^{M} p(x^1_m|x^1_{m-1}) \times p(x^2_m|x^2_{m-1})p(\mathbf{y}_m|x^1_m,x^2_m). \quad (7)$$

Prior probabilities and transition matrices of the hidden variables are computed from single-speaker recordings in the training set either speaker-dependently (for SD-DNNs and SPD-DNNs) or gender-dependently (for GPD-DNNs). To avoid a probability of zero, Laplace smoothing is applied during the computation, where we add one to each possible observation. The emission probability can be computed using the estimated posterior probability and Bayes rule,

$$p(\mathbf{y}_m|x^1_m,x^2_m) = \frac{p(x^1_m,x^2_m|\mathbf{y}_m)p(\mathbf{y}_m)}{p(x^1_m)p(x^2_m)}, \quad (8)$$

where $p(\mathbf{y}_m)$ is a constant for all feature vectors.

Once all probabilities are derived, we apply the junction tree algorithm to infer the most likely sequence of pitch states. The first step of this algorithm is to convert the directed graphical model to an undirected graphical model. In the next step, the nodes in the undirected graph are arranged to form a junction tree, where belief propagation is performed. For more details on the junction tree algorithm, we refer the interested reader to Jordan *et al.* (1999) and Wohlmayr *et al.* (2011). The time complexity of the junction tree algorithm is $O(2 \times 68^3 \times M)$ in our study. We then convert derived pitch states to the mean frequencies of the corresponding frequency bins. Because the resulting frequencies correspond to a rough sampling of possible pitch frequencies, we use a moving average window of length three to smooth frequencies and get final pitch estimates.

VI. EVALUATIONS AND COMPARISONS

A. Corpus and error measurement

For evaluations, we first use the GRID database (Cooke *et al.*, 2006), which is also used in Wohlmayr *et al.* (2011) hence facilitating our comparisons. The corpus consists of 1000 sentences spoken by each of 34 speakers (18 male, 16 female). Two male and two female speakers [No. 1, 2, 18, 20, same as Wohlmayr *et al.* (2011)], denoted as MA1, MA2, FE1, and FE2, are selected to train and test the proposed methods, except for GPD-DNNs which are tested on the same four speakers but trained on another set of speakers. We denote these four speakers as Set One. For each speaker in Set One, 950 sentences are selected for training, 40 sentences are used for choosing the best DNN weights during training, and the remaining ten sentences are used for testing. Note that all test sentences used in Wohlmayr *et al.* (2011) are also included in our test set. Another ten male and nine female speakers (numbers 3, 5, 6, 9, 10, 12, 13, 14, 17, 19; 4, 7, 11, 15, 16, 21, 22, 23, 24)[1] are used in the training of SD-DNNs and GPD-DNNs, where again for each speaker we select 950 sentences for training, and 40 sentences for selecting the best DNN weights. We denote these twenty speakers as Set Two. Reference pitches are extracted from single-speaker sentences using RAPT (Talkin, 1995), which outperforms other pitch trackers on clean speech signals (Drugman and Alwan, 2011). Although RAPT makes minor mistakes like pitch halving and doubling, these errors are not severe. Since the main challenge in multipitch tracking is the interference of another pitched sound, we treat thus derived pitch as groundtruth.

To mix two sentences u^1_t and u^2_t, we first select a speaker ratio R in dB, and amplify one of the speakers by R dB. A mixture with a speaker ratio of R dB is created by combining the resulting sentences using: $v_t = 10^{R/20}u^1_t + u^2_t$ or $v_t = u^1_t + 10^{R/20}u^2_t$. Note that if we choose a speaker ratio of 0 dB, the two equations to derive v_t are the same. For comparison reasons, we use a matched speaker ratio of 0 dB in the training and test of SD-DNNs, SPD-DNNs, GPD-DNNs and adaptation of GPD-DNNs. Unmatched speaker ratios are used to test multi-ratio training. The details of the training and test set are as follows:

- SD-DNNs: training mixtures are created by mixing each sentence of the target speaker in Set One with 60 random sentences in Set Two at 0 dB. Thus there are 57 000 training mixtures created for every target speaker. The test is conducted within Set One. We exhaustively mix test sentences for each speaker pair in Set One at 0 dB, resulting in a total of $10 \times 10 \times 6 = 600$ test mixtures.
- SPD-DNNs: for each speaker pair in Set One, we build the training set by mixing sentences of the two speakers at 0 dB. We make sure that each sentence of one speaker is randomly mixed with 60 sentences of the other speaker. Therefore 57 000 mixtures are created to train each speaker pair. We use the same test set as for SD-DNNs.
- GPD-DNNs: the training is conducted within Set Two. For the male-female case, we randomly create 57 000 mixtures at 0 dB. For the same-gender case, we divide the training speakers into two groups, with the first group having higher average pitch. We then create 57 000 training mixtures by randomly mixing the utterances in the first group and those in the second group at 0 dB. The same test set is used as for SD-DNNs.

- Adaptation of GPD-DNNs: For each speaker pair in Set One, we randomly select 100 mixtures from the SPD-DNN's training set as the adaptation data. The same test set is used as for SD-DNNs.
- Multi-ratio training: the training is conducted for both SD-DNNs and SPD-DNNs. Mixtures are no longer created at 0 dB in this experiment. Instead, we randomly amplify one of the two speakers with a random ratio out of $R = \{-12, 6, 0, 6, 12\}$ dB for each training mixture. As for the test set, we alternately amplified one of the two sentences with a ratio out of $R = \{-15, -12, -9, -6, -3, 0, 3, 6, 9, 12, 15\}$ dB, which gives $10 \times 10 \times 6 \times 2 = 1200$ mixtures at each speaker energy ratio, and 13 200 mixtures in total; note that each mixture at 0 dB appears twice in test.

In addition, we test our proposed methods using the FDA database (Bagshaw *et al.*, 1993) where the groundtruth pitches are derived from laryngograph data.

We evaluate pitch tracking results using the error measure proposed in Wohlmayr *et al.* (2011), which jointly evaluates the performance in terms of pitch accuracy and speaker assignment. Assuming that the ground truth pitch tracks are F_m^1 and F_m^2, we globally assign each estimated pitch track to a groundtruth pitch track based on the minimum mean square error and denote the assigned estimated pitch tracks as f_m^1 and f_m^2. The pitch frequency deviation of speaker i, $i \in \{1, 2\}$, is

$$\Delta f_m^i = \frac{|f_m^i - F_m^i|}{F_m^i}. \tag{9}$$

The voicing decision error E_{ij}, $i \neq j$, denotes the percentage of time frames where i pitch points are wrongly detected as j pitch points. For each speaker i, the permutation error E_{Perm}^i is set to one at time frames where the voicing decision for both estimates is correct, but Δf_m^i exceeds 20%, and f_m^i is within the 20% error bound of the other reference pitch, i.e., the error is due to incorrect speaker assignment. The overall permutation error E_{Perm} is the percentage of time frames where either E_{Perm}^1 or E_{Perm}^2 is one. Next, for each speaker i, the gross error E_{Gross}^i is set to one at time frames where the voicing decision for both estimates is correct, but Δf_m^i exceeds 20% with no permutation error. The overall gross error E_{Gross} is the percentage of time frames where either E_{Gross}^1 or E_{Gross}^2 is one. The fine detection error E_{Fine}^i is defined as the average of Δf_m^i in percent at time frames where Δf_m^i is smaller than 20%. $E_{\text{Fine}} = E_{\text{Fine}}^1 + E_{\text{Fine}}^2$. The total error is used as the overall performance measure:

$$E_{\text{Total}} = E_{01} + E_{02} + E_{10} + E_{12} + E_{20} + E_{21} + E_{\text{Perm}} + E_{\text{Gross}} + E_{\text{Fine}}. \tag{10}$$

B. Parameter selection

Because all proposed DNNs have similar structure, we conduct parameter selection for SPD-DNNs only. The best performing parameters are used in other models. We use a new pair of male speakers (numbers 26 and 28 in the GRID corpus) as the development set. For each speaker, 950 sentences are used for training, 40 sentences are used for choosing the best DNN weights during training and 10 sentences are used for test. Besides the matched 0 dB training and test condition, we also train the SPD-DNN with multi-ratio training. The details of the training and test set follow Sec. VI A. The results of multi-ratio training are averaged across all speaker ratios.

The size of the training set has strong impact on DNN's performance. We create five training sets by randomly mixing each sentence of one speaker with 5, 20, 40, 60 and 80 sentences of the other speaker, resulting in 4750, 19 000, 38 000, 57 000, 76 000 mixtures. An SPD-DNN is trained for each training set. The results are given in Fig. 4(a). In general, the total error decreases with the increase of the training size, and the improvement becomes small when the training size reaches 57 000. Taking the computational cost into consideration, we choose 57 000 training mixtures in the subsequent experiments.

Features are important to the system. As shown in Fig. 4(b), we compare three features: cochleagram, log spectrogram and MFCCs. We adopt the cochleagram feature in the subsequent experiments as it outperforms other two features.

To incorporate temporal dynamics, a context window is applied to the input feature. We have explored three values of the window size d (see Sec. III D). In Fig. 4(c), the total error substantially decreases when d is increased from 3 to 5, and remains the same when d reaches 7. Therefore we choose $d = 5$ for the cochleagram feature.

Next, we investigate the number of hidden units used in SPD-DNNs. Three numbers are compared: 512, 1024, and 1536. As shown in Fig. 4(d), the total error is reduced by more than 1.1% when the number is increased from 512 to 1024. However, further increasing the number of hidden units does not significantly boost the performance.

As described in Sec. II, we follow Han and Wang (2014) to use 68 pitch states to quantize the frequency range from 60 to 404 Hz. Another speaker-dependent multipitch tracking algorithm (Wohlmayr *et al.*, 2011) quantizes the frequency range from 80 to 500 Hz into 170 pitch states. We compare the two pitch quantizations using SPD-DNNs. The results are given in Fig. 4(e). Basically, more pitch states do not lead to better performance, probably because the frequency resolution with 170 pitch states is too fine for DNNs to make accurate probability estimates.

Other parameters, including the type of activation functions, the number of hidden layers, learning rate and mini-batch, are also chosen from the same development set.

C. Results and comparisons

We present our results, and compare with two state-of-the-art multipitch trackers: Jin and Wang (2011) and Wohlmayr *et al.* (2011). Jin and Wang's approach is designed for noisy and reverberant signals. They use correlogram to select reliable channels and track continuous pitch contours with an HMM. Wohlmayr *et al.* model speakers with GMMs, and use a mixture maximization model to obtain a probabilistic representation of pitch states. An FHMM is then applied to track pitch over time. The GMM-FHMM structure could also

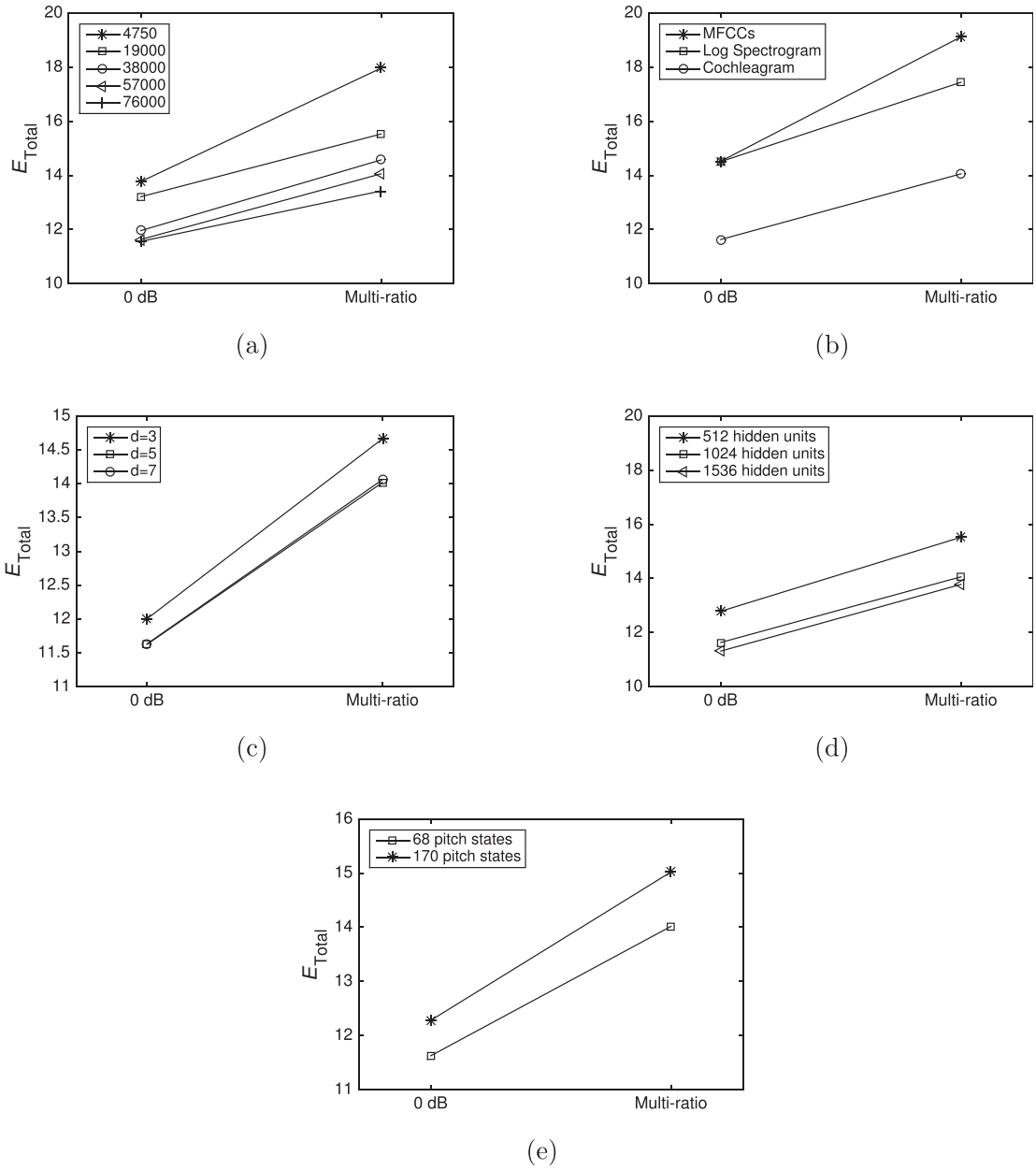

FIG. 4. Average E_{Total} of SPD-DNNs with different (a) sizes of training set, (b) features, (c) sizes of context window, (d) numbers of hidden units, (e) numbers of pitch states.

be extended to be gender-dependent. We denote the speaker-dependent and gender-dependent models from Wohlmayr *et al*. as Wohlmayr *et al*. SD and Wohlmayr *et al*. GD, respectively. Wohlmayr *et al*. train their models on the GRID database with groudtruth pitches obtained also by RAPT. The test mixtures used in their study are included in our test set, and we directly adopt the trained GMM/FHMM models posted on their website for comparison.

TABLE I. E_{Total} for different multipitch trackers on 600 test mixtures of the GRID Corpus.

		E_{01}	E_{02}	E_{10}	E_{12}	E_{20}	E_{21}	E_{Gross}	E_{Fine}	E_{Perm}	E_{Total}
Jin and Wang	Mean	4.54	1.25	6.97	5.51	1.94	**12.81**	4.80	6.93	6.47	**51.21**
	Std[a]	2.34	1.38	3.55	3.33	2.16	5.54	4.65	3.17	5.34	11.71
Wohlmayr *et al*. SD	Mean	1.81	0.06	5.89	2.68	1.39	**10.81**	0.93	2.79	0.37	**26.73**
	Std	1.64	0.26	3.42	2.18	2.06	5.26	1.14	0.73	0.79	9.49
SD-DNN	Mean	1.98	0.13	2.01	5.70	0.07	**2.74**	0.72	2.32	1.01	**16.69**
	Std	1.61	0.40	2.02	5.57	0.27	2.06	1.25	0.84	2.23	7.90
SPD-DNN	Mean	1.69	0.07	1.59	3.19	0.05	**2.55**	0.52	1.95	0.15	**11.77**
	Std	1.42	0.26	1.54	2.09	0.24	1.94	0.91	0.33	0.54	3.29

[a]Standard deviation (Std).

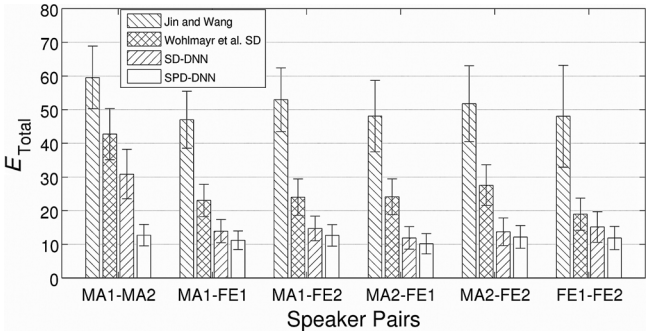

FIG. 5. E_{Total} of different approaches tested on six pairs of speakers. Error bars depict the mean and standard deviation of a method on the test mixtures of a given speaker pair.

We first evaluate the SD-DNN and SPD-DNN based methods. Table I compares the SD-DNN and SPD-DNN based methods with the other multipitch trackers on 600 test mixtures. Speaker-dependent approaches perform substantially better than the speaker-independent approach, and our SD-DNN and SPD-DNN based methods cut E_{Total} by more than 10% compared to Wohlmayer *et al.* SD. The major improvement in E_{Total} comes from E_{21}, which implies that our methods estimate pitch more accurately when the two speakers are both voiced. The SPD-DNN method performs

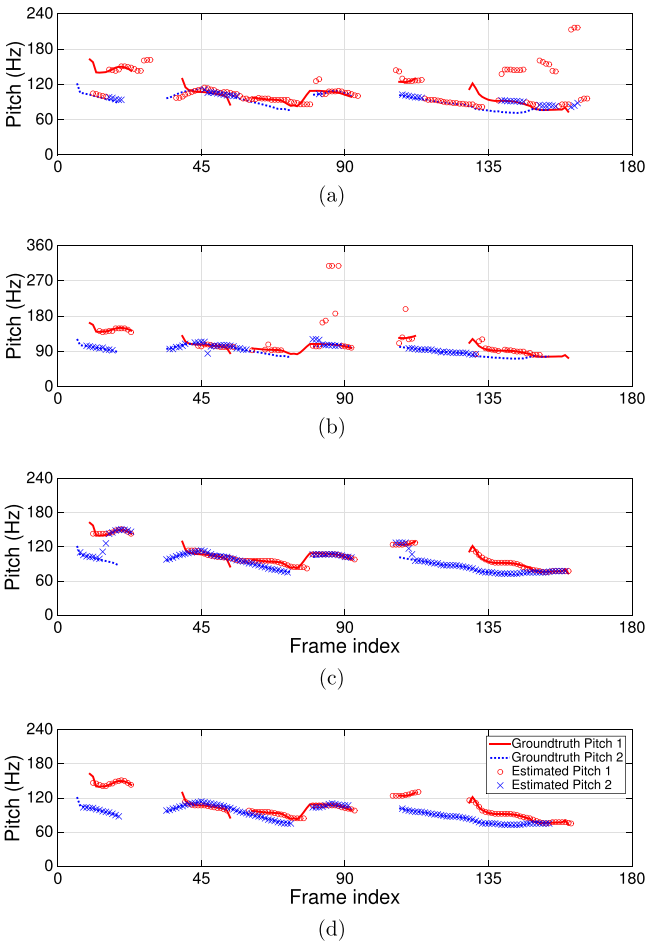

FIG. 6. (Color online) Multipitch tracking results on a test mixture (pbbv6n and priv3n) of the MA1-MA2 speaker pair. (a) Groundtruth pitch (lines and dotted lines) and estimated pitch (circles and crosses) by Jin and Wang. (b) By Wohlmayr *et al.* SD. (c) By SD-DNN. (d) By SPD-DNN.

TABLE II. Average E_{Total} for SD-DNN and Wohlmayr *et al.* SD on 600 test mixtures of the GRID Corpus.

Training utterances per speaker	497		950
Feature type	Log spectrogram	Cochleagram	Cochleagram
SD-DNN	19.02	17.22	**16.69**
Wohlmayr *et al.* SD	26.73	—	—

better than the SD-DNN method, which is not surprising as SPD-DNNs are trained on individual speaker pairs. We further illustrate E_{Total} for each of the six speaker pairs in Fig. 5. As shown in the figure, our methods have lower errors across all pairs. SD-DNNs and SPD-DNNs perform comparably on five speaker pairs, and the latter achieve significantly lower E_{Total} on the most difficult pair of MA1-MA2. Figure 6 illustrates pitch tracking results on a test mixture of MA1-MA2. Jin and Wang's approach fails to assign pitches to the underlying speakers. The approach by Wohlmayr *et al.* works better in terms of speaker assignment, but performs poorly when two pitch tracks are close to each other. Moreover, their resulting pitch contours lack continuity. The SD-DNN produces much smoother pitch contours. However, it still has incorrect speaker assignment at a few frames. The SPD-DNN generates very good pitch tracks in both pitch accuracy and speaker assignment.

To further analyze the above-mentioned improvement achieved by our proposed methods, we compare our SD-DNN based method with Wohlmayr *et al.* SD using the same feature, namely, the log spectrogram feature described in Sec. III B, and the same training data, i.e., 497 training utterances per speaker. Specifically, we train the SD-DNN based method using three settings: (1) 497 training utterances per speaker with log spectrogram feature, (2) 497 training utterances per speaker with cochleagram feature, and (3) 950 training utterances per speaker with cochleagram feature (the proposed training setting). Results on the 600 test mixtures are shown in Table II. When using exactly the same feature and training data, the SD-DNN based method significantly outperforms Wohlmayr *et al.* SD. If we replace SD-DNN's input feature with cochleagram, the total error further decreases. Last, increasing the training size slightly boosts SD-DNN's performance. In conclusion, although features

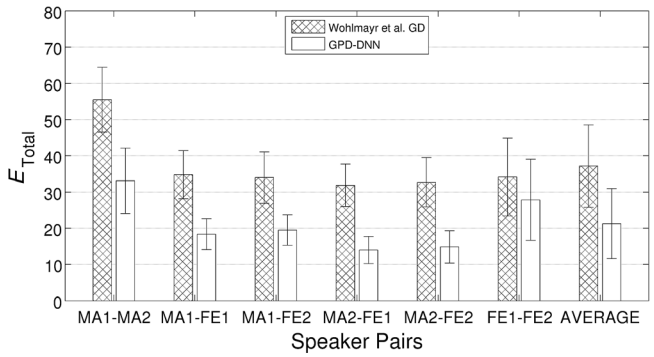

FIG. 7. E_{Total} of gender-dependent approaches. Error bars depict the mean and standard deviation of a method on the test mixtures of a speaker pair.

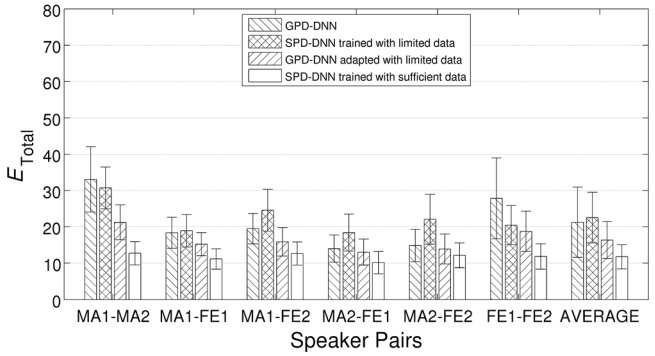

FIG. 8. Performance of GPD-DNN adaptation. Error bars depict the mean and standard deviation of a method on the test mixtures of a speaker pair.

and training sizes have an effect, the use of DNNs makes the most contribution to performance gains.

Next, we evaluate three extensions to the previous models. Figure 7 shows the performance of the GPD-DNN based method. It significantly outperforms the gender-dependent model by Wohlmayr *et al*. on all speaker pairs. The average E_{Total} of GPD-DNN is 15.89% lower than the gender-dependent model by Wohlmayr *et al*., and even 5.46% lower than the speaker-dependent model by Wohlmayr *et al*. However, the performance gap between GPD-DNNs and SD-DNNs/SPD-DNNs is larger than 4.5%. Therefore one should use SD-DNN/SPD-DNN based methods when speaker-dependent information is available.

Figure 8 shows the performance of GPD-DNN adaptation. Four models are compared across all speaker pairs: (1) GPD-DNNs, (2) SPD-DNNs directly trained with 100 mixtures per speaker pair, (3) GPD-DNNs adapted with 100 mixtures per speaker pair, (4) SPD-DNNs trained with 57 000 mixtures per speaker pair. As shown in the figure, SPD-DNNs trained with limited data perform better than GPD-DNNs on same-gender mixtures, but worse than GPD-DNNs on different-gender mixtures. GPD-DNN adaptation consistently outperforms the first two methods, resulting in 5% reduction in average E_{Total}. The results indicate the superiority of GPD-DNN adaptation for small training sizes.

Generalization to different speaker energy ratios is crucial to supervised multipitch trackers. Figure 9 shows the performance of SD-DNN, SPD-DNN, and the speaker-dependent models by Wohlmayr *et al*. at various speaker ratios. All models are trained at 0 dB, and results are averaged across all speaker pairs at each speaker ratio. As shown in the figure, the total error increases significantly when the speaker ratio deviates from 0 dB. Errors are not symmetric with respect to 0 dB, as we only scale the level of one speaker in order to create a specified ratio. For the speaker-dependent model by Wohlmayr *et al*., when the speaker ratio is positive, the mixture becomes dominated by the amplified speaker, misleading the GMM of the weak speaker. For the SD-DNN and SPD-DNN based methods, it is hard for DNNs to recognize the weak speaker when the speaker ratio is too low. We then apply multi-ratio training for SD-DNNs and SPD-DNNs, and compare them with Jin and Wang's unsupervised multipitch tracker as well as the gain-adapted version of the speaker-dependent models by Wohlmayr *et al*. (Wohlmayr and Pernkopf, 2011). Note that, unlike multi-ratio training, gain adaptation in Wohlmayr and Pernkopf (2011) uses an expectation-maximization based framework to estimate gains in test mixtures, thus no additional training is needed. The results are given in Fig. 10. The performance of multi-ratio trained DNNs remains high across all speaker ratios. At 0 dB, multi-ratio trained SD-DNNs and SPD-DNNs produce only 0.03% and 0.34% higher errors than SD-DNNs and SPD-DNNs trained in the matched 0 dB condition, indicating their strong generalization ability.

Noise robustness is also an important issue in multipitch estimation. We evaluate Jin and Wang's model, the speaker-dependent model by Wohlmayr *et al*., the SD-DNN based model and the SPD-DNN based model, when a speech shape noise (SSN) and a babble noise are mixed with two-speaker utterances. SSN is a stationary noise with no pitch, and babble noise is nonstationary with pitched portions. Specifically, we generate 100 test mixtures of MA1-MA2 at the speaker ratio of 0 dB. The test mixtures are then mixed with SSN and babble noise at the SNR of 5, 10, 20 and Inf dB. Here the SNR refers to the ratio of two-speaker-mixture power to the noise power, and Inf dB corresponds to the noise-free condition. Importantly, no retraining is performed for any system. The multipitch tracking results in background noise are given in Fig. 11. As shown in the figure, our methods remain robust to both kinds of noise, and outperform the comparison models.

In the above experiments, we use RAPT to extract the groundtruth pitch from single speaker recordings, which is not error-free as mentioned previously. We now evaluate our methods on the FDA database (Bagshaw *et al*., 1993), where

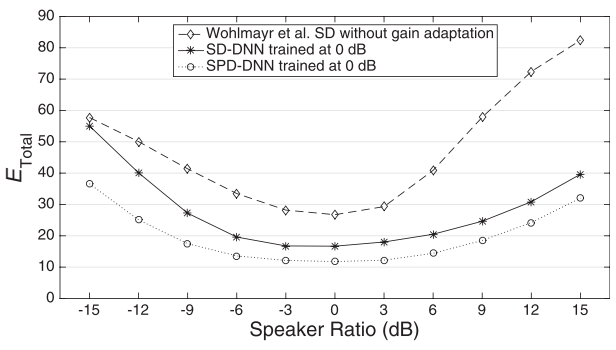

FIG. 9. Results of different approaches tested on 11 speaker ratios. Each data point represents E_{Total} averaged across 1200 test mixtures.

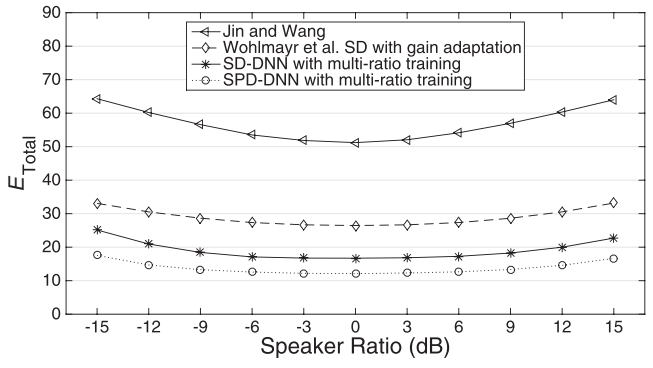

FIG. 10. Results of different approaches tested on 11 speaker ratios. Each data point represents E_{Total} averaged across 1200 test mixtures.

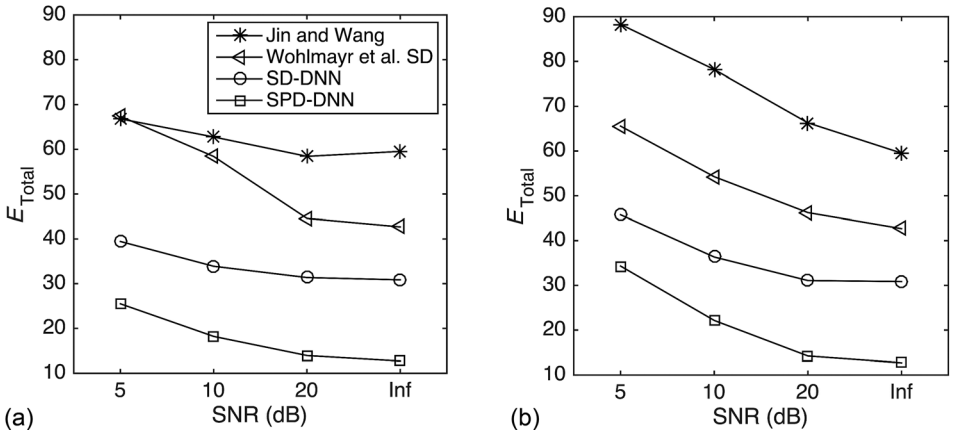

FIG. 11. E_{Total} of different approaches tested on the MA1-MA2 speaker pair mixed with (a) speech shape noise, (b) babble noise.

the groundtruth pitch is directly given by laryngograph data. The corpus consists of recordings of 50 sentences by each of two speakers (a male and a female). For each speaker, we choose 40 sentences for testing and 40 test mixtures are created by mixing the test sentences at 0 dB. Because the dataset is not large enough for training SD-DNNs and SPD-DNNs, we conduct experiments with GPD-DNN and GPD-DNN adaptation. A ratio-adapted GPD-DNN trained on the GRID database is used for pitch-state probability estimation. We also perform speaker adaptation of the GPD-DNN with 10 adaptation sentences per speaker, i.e., 10×10 adaptation mixtures. We compare the two methods with Jin and Wang's speaker-independent model as well as the gain-adapted version of the gender-dependent model by Wohlmayr *et al.* E_{Total} of different approaches is shown in Fig. 12. Results indicate that our GPD-DNN based method outperforms other approaches. The adaptation of the GPD-DNN further reduces the average total error by 8.69%.

In addition to E_{Total}, we use another metric to compare the performance in this experiment: overall multipitch accuracy used by Duan *et al.* (2014). To compute this accuracy, we first assign each estimated pitch track to a groundtruth pitch track. For each estimated pitch track, we call a pitch estimate at a frame correct if it deviates less than 10% from its corresponding groundtruth pitch. The overall multipitch accuracy is defined as

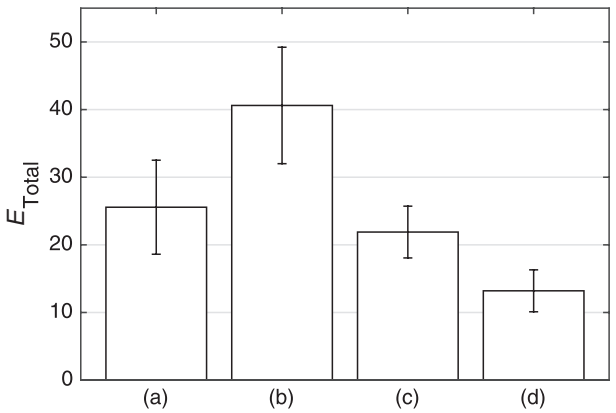

FIG. 12. Results of different approaches tested on the FDA corpus. (a) Jin and Wang. (b) Wohlmayr *et al.* GD with gain adaptation. (c) Ratio-adapted GPD-DNN. (d) Speaker adaptation of GPD-DNN. Error bars depict the mean and standard deviation of a method on the test mixtures.

$$\text{Accuracy} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}, \tag{11}$$

where TP (true positive) is the total number of correctly estimated pitches, FP (false positive) is the total number of pitches that appear in some estimated pitch track but do not belong to the corresponding groundtruth pitch track, and FN (false negative) denotes the total number of pitches that appear in some groundtruth pitch track but do not belong to the corresponding estimated pitch track. Different assignments of estimated pitch tracks give us different accuracies, and we choose the highest value to represent the overall accuracy. Similar to Fig. 12, the GPD-DNN and GPD-DNN adaptation achieve accuracies of 70.02% and 82.61%. The other two approaches have accuracies lower than 50%.

Last, we compare the computational complexity of different approaches. We directly use the program from the author websites of Jin and Wang (2011) and Wohlmayr *et al.* (2011). The program from Jin and Wang (2011) is implemented in Java, and the program by Wohlmayr *et al.* (2011) is implemented in MATLAB. Our program is a mixture of MATLAB and Caffe (a deep learning framework). One hundred mixtures with the total length of 179.7 s are created for this evaluation. The test is performed on a machine with an Intel i7-4770k CPU (3.5 GHz) and 32 GB memory. All computations are performed on the CPU within a single thread. Table III shows the average processing time per one second mixture. Results indicate that our methods are a lot more efficient. There are two main reasons why Wohlmayr *et al.* SD is slower. First, the number of pitch states used in SD of Wohlmayr *et al.* is 170, while in our study it is 68. Second, the mixmax interaction model in SD of Wohlmayr *et al.* occupies 85% of total running time. In our study the corresponding module is DNN, and it only takes less than 0.4 s for one second mixture.

In addition to the above comparisons, we have compared with Hu and Wang (2013), where a clustering algorithm is used to group short pitch contours into two speakers.

TABLE III. Running time comparison for different approaches.

	Jin-Wang	Wohlmayr *et al.* SD	SD-DNN	SPD-DNN
Time (s)	7.77	20.12	0.60	0.43

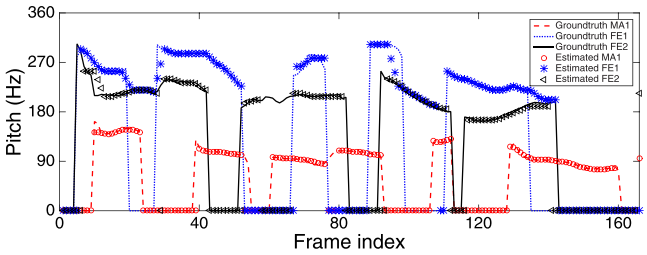

FIG. 13. (Color online) Multipitch tracking results by SD-DNNs on a three-speaker test sample by mixing MA1, FE1, and FE2 at equal sound levels.

We found that this method performs better than Jin and Wang's method, but worse than the speaker-dependent method by Wohlmayr *et al*. The details of this comparison can be found in Liu and Wang (2015b). NMF based approaches have been used in multipitch tracking (Sha and Saul, 2005; Peharz *et al*., 2011). Since a gain-adapted GMM-FHMM based approach has been shown to match the performance of an NMF-FHMM based approach at various speaker energy ratios (Wohlmayr and Pernkopf, 2011; Peharz *et al*., 2011), we do not directly compare our methods with NMF based algorithms.

VII. CONCLUDING REMARKS

We have proposed speaker-dependent and speaker-pair-dependent DNNs to estimate the posterior probabilities of pitch states for two simultaneous speakers. Taking advantage of discriminative modeling and speaker-dependent information, our approach produces good pitch estimation in terms of both accuracy and speaker assignment, and significantly outperforms other state-of-the-art multipitch trackers. The SPD-DNN based method performs especially well when the two speakers have close pitch tracks. In order to relax constraints, we have introduced three extensions to SD-DNNs and SPD-DNNs. Gender-pair-dependent DNNs are designed for unseen speakers during testing, and they perform substantially better than other speaker-independent and gender-dependent approaches on both GRID and FDA databases. Given limited speaker-dependent training data, speaker adaptation is effective for reducing tracking errors. Last, multi-ratio trained SD-DNNs and SPD-DNNs produce consistent results across various speaker ratios.

To apply our speaker-dependent models requires that the identities of the two speakers be known beforehand. Recently, Zhao *et al*. (2015) proposed a DNN-based cochannel speaker identification algorithm, which can reliably identify the speakers in two-speaker mixtures. Such an algorithm could be used to first identify the two speakers in an input mixture, thus helping select trained SD-DNNs or SPD-DNNs for pitch estimation. When the speakers in a mixture are not enrolled, we can use a similar cochannel gender pair detection algorithm as a front-end for gender-pair-dependent multipitch tracking. Our experiments show that the accuracy of such gender pair detector is perfect.

Although the proposed models are designed for two-speaker mixtures, they can be extended to mixtures with more than two speakers. To illustrate this extension, Fig. 13 shows an example when three speakers, i.e., MA1, FE1, and FE2 in the GRID database, are mixed in one test sample with equal energy ratio between every pair of speakers. We first use three SD-DNNs trained on the GRID database to estimate pitch-state probabilities for the three speakers. An FHMM with three Markov chains is then employed to connect all probabilities. No retraining is performed for this experiment. As shown in the figure, our algorithm does a decent job tracking three pitch tracks simultaneously. Extensions to more speakers can be achieved in a similar manner. It is worth noting that this relatively straightforward extension is an advantage of our speaker-dependent modeling and our use of FHMM that is not shared by the HMM based model in Jin and Wang (2011). Many multipitch trackers deal with interfering speakers and additive noise at the same time (Wu *et al*., 2003; Jin and Wang, 2011). We have illustrated the noise-robustness of our models without retraining. Better results are expected if we further include noise corrupted mixtures in the training data set.

To make use of the temporal context, we concatenate neighboring frames into a feature vector. Such a method can only capture temporal dynamics in a limited span. On the other hand, recurrent neural networks (RNNs) have self connections through time. Studies have shown that RNNs are good at modeling sequential data like handwriting (Graves *et al*., 2008) and speech (Vinyals *et al*., 2012). We plan to explore RNNs in future work to better capture the temporal context.

ACKNOWLEDGMENTS

We would like to thank M. Wohlmayr, M. Stark, and F. Pernkopf for providing their pitch tracking code to us. This research was supported by an AFOSR grant (FA9550-12-1-0130), an NIDCD grant (R01 DC012048), and the Ohio Supercomputer Center.

[1]This list follows gender-dependent training set by Wohlmayr *et al*. (2011). However, since speaker No. 8 is wrongly marked as a female speaker in their training set, we eliminate this speaker in our study. Results show that the elimination leads to little performance change.

Abdel-Hamid, O., and Jiang, H. (**2013**). "Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code," in *Proceedings of ICASSP*, pp. 7942–7946.

Bach, F., and Jordan, M. (**2005**). "Discriminative training of hidden Markov models for multiple pitch tracking," in *Proceedings of ICASSP*, pp. 489–492.

Bagshaw, P. C., Hiller, S. M., and Jack, M. A. (**1993**). "Enhanced pitch tracking and the processing of F0 contours for computer aided intonation teaching," in *Proceedings of Eurospeech*, pp. 1003–1006.

Boersma, P. (**2001**). "Praat, a system for doing phonetics by computer," Glot Int. **5**, 341–345.

Brookes B. (**2011**). "Voicebox: Speech processing toolbox for MATLAB," http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html (Last viewed July 28, 2016).

Chen, C., Gopinath, R., Monkowski, M., Picheny, M., and Shen, K. (**1997**). "New methods in continuous mandarin speech recognition," in *Proceedings of Eurospeech*, pp. 1543–1546.

Chen, J., Wang, Y., Yoho, S. E., Wang, D. L., and Healy, E. W. (**2016**). "Large-scale training to increase speech intelligibility for hearing-impaired listeners in novel noises," J. Acoust. Soc. Am. **139**, 2604–2612.

Cheveigné, A. D., and Kawahara, H. (**2002**). "YIN, a fundamental frequency estimator for speech and music," J. Acoust. Soc. Am. **111**, 1917–1930.

Christensen, M. G., and Jakobsson, A. (**2009**). "Multi-pitch estimation," Synth. Lectures Speech Audio Process. **5**, 1–160.

Cooke, M., Barker, J., Cunningham, S., and Shao, X. (**2006**). "An audio-visual corpus for speech perception and automatic speech recognition," J. Acoust. Soc. Am. **120**, 2421–2424.
Drugman, T., and Alwan, A. (**2011**). "Joint robust voicing detection and pitch estimation based on residual harmonics," in *Proceedings of Interspeech*, pp. 1973–1976.
Duan, Z., Han, J., and Pardo, B. (**2014**). "Multi-pitch streaming of harmonic sound mixtures," IEEE/ACM Trans. Audio, Speech, Lang. Process. **22**, 138–150.
Ghahramani, Z., and Jordan, M. (**1997**). "Factorial hidden Markov models," Mach. Learn. **29**, 245–273.
Glorot, X., Bordes, A., and Bengio, Y. (**2011**). "Deep sparse rectifier neural networks," in *Proceedings of AISTATS*, pp. 315–323.
Graves, A., Liwicki, M., Bunke, H., Schmidhuber, J., and Fernández, S. (**2008**). "Unconstrained on-line handwriting recognition with recurrent neural networks," in *Proceedings of NIPS*, pp. 577–584.
Han, K., and Wang, D. L. (**2014**). "Neural network based pitch tracking in very noisy speech," IEEE/ACM Trans. Audio, Speech, Lang. Process. **22**, 2158–2168.
Hinton, G., Osindero, S., and Teh, Y. W. (**2006**). "A fast learning algorithm for deep belief nets," Neural Comput. **18**, 1527–1554.
Hinton, G., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (**2012**). "Improving neural networks by preventing co-adaptation of feature detectors," arXiv:1207.0580, pp. 1–18.
Holdsworth, J., Nimmo-Smith, I., Patterson, R., and Rice, P. (**1988**). "Implementing a gammatone filter bank," Tech. Report, MRC Applied Psychology Unit, Cambridge, pp. 1–5.
Hu, G., and Wang, D. L. (**2010**). "A tandem algorithm for pitch estimation and voiced speech segregation," IEEE Trans. Audio, Speech, Lang. Process. **18**, 2067–2079.
Hu, K., and Wang, D. L. (**2013**). "An unsupervised approach to cochannel speech separation," IEEE Trans. Audio, Speech, Lang. Process. **21**, 122–131.
Jin, Z., and Wang, D. L. (**2011**). "HMM-based multipitch tracking for noisy and reverberant speech," IEEE Trans. Audio, Speech, Lang. Process. **19**, 1091–1102.
Jordan, M., Ghahramani, Z., Jaakkola, T., and Saul, L. K. (**1999**). "An introduction to variational methods for graphical models," Mach. Learn. **37**, 183–233.
Liao, H. (**2013**). "Speaker adaptation of context dependent deep neural networks," in *Proceedings of ICASSP*, pp. 7947–7951.
Liu, Y., and Wang, D. L. (**2015a**). "Speaker-dependent multipitch tracking using deep neural networks," in *Proceedings of Interspeech*, pp. 3279–3283.
Liu, Y., and Wang, D. L. (**2015b**). "Speaker-dependent multipitch tracking using deep neural networks," Tech. Report OSU-CISRC-8/15-TR12, Department of Computer Science and Engineering, The Ohio State University, Columbus, OH, pp. 1–22.
Noll, A. (**1967**). "Cepstrum pitch determination," J. Acoust. Soc. Am. **41**, 293–309.
OSU Perception and Neurodynamics Lab (**2008**). "Downloadable Code," available at http://web.cse.ohio-state.edu/pnl/shareware/cochleagram/ (Last viewed January 3, 2017).
Peharz, R., Wohlmayr, M., and Pernkopf, F. (**2011**). "Gain-robust multi-pitch tracking using sparse nonnegative matrix factorization," in *Proceedings of ICASSP*, pp. 5416–5419.
Saon, G., Soltau, H., Nahamoo, D., and Picheny, M. (**2013**). "Speaker adaptation of neural network acoustic models using i-vectors," in *Proceedings of ASRU*, pp. 55–59.
Sha, F., and Saul, L. K. (**2005**). "Real-time pitch determination of one or more voices by nonnegative matrix factorization," in *Proceedings of NIPS*, pp. 1233–1240.
Talkin, D. (**1995**). "A robust algorithm for pitch tracking (RAPT)," in *Speech Coding Synthesis* (Elsevier, New York), pp. 495–518.
Vinyals, O., Ravuri, S. V., and Povey, D. (**2012**). "Revisiting recurrent neural networks for robust ASR," in *Proceedings of ICASSP*, pp. 4085–4088.
Wang, D. L., and Brown, G., Eds. (**2006**). "Feature-based speech segregation," in *Computational Auditory Scene Analysis: Principles, Algorithms and Applications* (Wiley-IEEE Press, Hoboken, NJ), Chap. 3, pp. 81–111.
Wohlmayr, M., and Pernkopf, F. (**2011**). "EM-based gain adaptation for probabilistic multipitch tracking," in *Proceedings of Interspeech*, pp. 1969–1972.
Wohlmayr, M., and Pernkopf, F. (**2013**). "Model-based multiple pitch tracking using factorial HMMs: Model adaptation and inference," IEEE Trans. Audio, Speech, Lang. Process. **21**, 1742–1754.
Wohlmayr, M., Stark, M., and Pernkopf, F. (**2011**). "A probabilistic interaction model for multipitch tracking with factorial hidden Markov models," IEEE Trans. Audio, Speech, Lang. Process. **19**, 799–810.
Wu, M., Wang, D. L., and Brown, G. (**2003**). "A multipitch tracking algorithm for noisy speech," IEEE Trans. Speech Audio Process. **11**, 229–241.
Yu, D., Yao, K., Su, H., Li, G., and Seide, F. (**2013**). "KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *Proceedings of ICASSP*, pp. 7893–7897.
Zhao, X., Shao, Y., and Wang, D. L. (**2012**). "CASA-based robust speaker identification," IEEE Trans. Audio, Speech, Lang. Process. **20**, 1608–1616.
Zhao, X., Wang, Y., and Wang, D. L. (**2015**). "Cochannel speaker identification in anechoic and reverberant conditions," IEEE/ACM Trans. Audio, Speech, Lang. Process. **23**, 1727–1736.