Acoustic-to-Articulatory Mapping With Joint Optimization of Deep Speech Enhancement and Articulatory Inversion Models

Abdolreza Sabzi Shahrebabaki, Giampiero Salvi, Torbjørn Svendsen, *Senior Member, IEEE*, and Sabato Marco Siniscalchi, *Senior Member, IEEE*

***Abstract*—We investigate the problem of speaker independent acoustic-to-articulatory inversion (AAI) in noisy conditions within the deep neural network (DNN) framework. In contrast with recent results in the literature, we argue that a DNN vector-to-vector regression front-end for speech enhancement (DNN-SE) can play a key role in AAI when used to enhance spectral features prior to AAI back-end processing. We experimented with single- and multi-task training strategies for the DNN-SE block finding the latter to be beneficial to AAI. Furthermore, we show that coupling DNN-SE producing enhanced speech features with an AAI trained on clean speech outperforms a multi-condition AAI (AAI-MC) when tested on noisy speech. We observe a 15% relative improvement in the Pearson's correlation coefficient (PCC) between our system and AAI-MC at 0 dB signal-to-noise ratio on the Haskins corpus. Our approach also compares favourably against using a conventional DSP approach to speech enhancement (MMSE with IMCRA) in the front-end. Finally, we demonstrate the utility of articulatory inversion in a downstream speech application. We report significant WER improvements on an automatic speech recognition task in mismatched conditions based on the Wall Street Journal corpus (WSJ) when leveraging articulatory information estimated by AAI-MC system over spectral-alone speech features.**

***Index Terms*—Deep neural network, acoustic-to-articulatory inversion, speech enhancement, multi-task training, speaker independent models.**

I. Introduction

THE human speech production system contains several organs, namely, lungs; trachea; larynx; throat; oral and nasal cavities. The oral cavity comprises several anatomical elements, such as velum, tongue, teeth, jaw and lips. Those elements are considered as the articulators. Articulator movements result in the production of various speech sounds. The problem of estimating the articulators' movements from the acoustic speech signal is referred to as acoustic-to-articulatory inversion (AAI). In recent years, AAI has attracted increasing attention because of its potential applications in speech processing. Examples include low bit rate coding [1], automatic speech recognition (ASR) [2]–[4], speech synthesis [5], [6], computer aided pronunciation training (CAPT) [7], [8], depression detection from speech [9], [10], and speech therapy [11], [12]. The articulators' movements can be measured and parameterized through various techniques, for instance real-time magnetic resonance imaging (rt-MRI) [13], X-ray microbeam [14], electromagnetic articulography (EMA) [15], and ultrasound [16]. Nevertheless, obtaining articulatory measurements is not practical in real world applications since it requires instrumentation not available outside laboratories, and imposes heavy burdens on the subjects. As a consequence, estimation of these parameters from the available source of information, which is the speech signal, must be achieved through an AAI system. Unfortunately, this inversion problem is highly non-linear and non-unique [3], [17], which means that different articulator configurations can produce the same sound. Moreover, coarticulation [18], i.e., the impact of adjacent phonemes on the articulators' movement, makes the AAI problem even harder. In addition, articulatory measurements are only available for a limited number of speakers. This limitation introduces an additional complexity to the AAI problem and urges building up speaker independent AAI systems (SI-AAI) that can be utilized for speech databases with no articulatory recordings.

The majority of available AAI works mainly focused on two different aspects: (i) the acoustic feature representation, and (ii) the solution to the AAI regression problem with different techniques. Different acoustic representations, such as Line Spectral Frequencies (LSFs) [19], Perceptual Linear Predictive coding (PLP) [20] and Mel-Frequency Cepstral Coefficients (MFCCs) [21] have been widely used for the AAI task. Filter-Bank Energies (FBEs) from STRAIGHT spectra [22] have also been employed as the input of the AAI system [23]. Among these features, MFCCs are reported to perform better compared to other features for SI-AAI [24], [25].

In the literature, various techniques are applied to the AAI problem, e.g. search-based algorithms in the joint codebook

Manuscript received March 21, 2021; revised August 8, 2021 and November 8, 2021; accepted November 23, 2021. Date of publication December 7, 2021; date of current version December 24, 2021. This work was supported in part by the Research Council of Norway through the Project AULUS, in part by the NTNU through the Project ArtiFutt, and in part through the PRIN2017 Project under Grant JNKCYZ_002. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Tom Backstrom *(Corresponding author: Abdolreza Sabzi Shahrebabaki.)*

Abdolreza Sabzi Shahrebabaki, Torbjørn Svendsen, and Sabato Marco Siniscalchi are with Signal processing, Norwegian University of Science and Technology, 7491 Trondheim, Norway (e-mail: abdolreza.sabzi@ntnu.no; torbjorn.svendsen@ntnu.no; marco.siniscalchi@unikore.it).

Giampiero Salvi is with Signal processing, Norwegian University of Science and Technology, 7491 Trondheim, Norway, and also with KTH Royal Institute of Technology, 114 28 Stockholm, Sweden (e-mail: giampiero.salvi@ntnu.no).

Digital Object Identifier 10.1109/TASLP.2021.3133218

of the acoustic-articulatory space [26], [27], non-parametric and parametric statistical methods, such as support vector regression (SVR) [28], local regression approach based on K-nearest neighbour [29], joint acoustic-articulatory distribution by utilizing Gaussian mixture models (GMMs) [30], hidden Markov models (HMMs) [7], mixture density networks (MDNs) [31], deep neural networks (DNNs) [4], [32], and recurrent neural networks (RNNs) [23], [33]–[39]. Among those methods, the neural network based models outperform the rest by having the ability of dealing well with large context size and better modelling of acoustic and articulatory spaces.

It is also important to remark that most of the available AAI research is accomplished using clean data, with the goal of improving the AAI accuracy either for speaker dependent, or speaker independent cases. Real world speech applications, however, suffer from the presence of environmental noise in the recordings, which in turn leads to a performance degradation of the AAI system. There are few works available for AAI in noisy conditions. Most of these works are in the field of robust ASR [4], [40], and use synthetically generated speech data obtained with an articulatory synthesizer and the Task Dynamics and Applications (TADA) system [41]. To the best of the authors' knowledge, there is only one work dealing with real articulatory measurements in noisy conditions [42], where the authors compared the accuracy of two AAI systems. One system was trained on clean data (AAI-C); the other system was built using multi-condition speech data (AAI-MC), including clean data. For the AAI-C system the noisy test data were optionally enhanced by minimum mean square error (MMSE) based speech enhancement (SE) [43]. The outcome of the study was twofold. First, AAI-MC seemed to be the best solution for dealing with noisy data. Second, MMSE-based SE on the noisy data led to a drop in the AAI-C performance on noisy data compared to both AAI-MC and to AAI-C with unprocessed speech. Such an outcome contrasts with the naïve expectation that enhanced speech should yield an improved performance. A possible explanation of the unexpected outcome could be that distortions and artifacts introduced by the MMSE-based method may have reduced the quality of the enhanced speech with respect to the AAI task.

Although SE based on the MMSE approach did not seem useful in AAI applications in noisy conditions, we observe that deep neural network (DNN) based approaches to SE have recently been shown to better overcome musical noise issues and introduce less distortion than traditional digital signal processing (DSP) methods [44]–[46]. Therefore, we argue that DNN-SE can play a key role in AAI too if used at a pre-processing stage before the downstream speech applications, as demonstrated for other tasks in [47]–[49], for instance. Our goal is therefore to clean up the input signal before sending it to the off-the-shelf AAI-C, avoiding the need to build an AAI-MC system leveraging multi-condition data. In addition, for speech recognition in noisy conditions which is more applicable in the daily usage, it would be helpful to apply enhancement as a pre-processor, and estimate the articulatory trajectories and subsequently utilize them in the recognition task.

We design our SE system using deep neural networks vector-to-vector regression with the goal to enhance the speech features. The deep model used for the AAI task is stacked on top of the SE model, allowing for joint optimization of the full model for further improved performance. In this way the overall neural model learns to enhance the noisy speech in a helpful way for the AAI goal. To better appreciate our experimental evidence, we compare and contrast our proposed approach with the MMSE-based speech enhancement with an improved noise estimation method, namely minima controlled recursive averaging (IMCRA) [50]. The IMCRA algorithm produces less distortion in the enhanced speech compared to the original MMSE based approaches, e.g. [43]. We will refer to IMCRA based system as DSP-SE. Moreover, we assessed the role that articulatory information, extracted with the proposed solution, could play in a downstream speech application using an ASR task in noisy condition, namely a hand-crafted noisy version of the Wall Street Journal (WSJ) task [51]. Experimental evidence clearly demonstrates the beneficial effect of combining articulatory information with standard spectral-based speech features when decoding noisy speech data using a character-based encoder-decoder end-to-end ASR system leveraging both a hybrid connectionist temporal classification (CTC) loss function, and the attention mechanism.

The rest of the paper is organized as follows. In section II different neural architectures are described. Section III introduce the corpora which are utilized in this research work, and in Section IV, different experiments are conducted and the results are discussed. Section V concludes our work and suggests future work.

II. AAI Systems

In this section, three different systems are described. The first system performs acoustic-to-articulatory inversion (AAI) directly on (noisy) speech using a deep model; the second systems consists of a feed-forward deep neural network based speech enhancement module (DNN-SE) and an AAI module based on a deep architecture; finally, the third system combines the DNN-SE and DNN-AAI module into a single deep architecture and joint training is used to fine-tune the overall AAI system. In the following, those three systems are discussed in detail.

A. DNN for Acoustic-to-Articulatory Inversion (DNN-AAI)

A speaker-independent (SI) design is used to deploy the DNN-AAI system, so that test speakers are removed from the developing material during the training phase. The input features are the standard MFCCs. These features have been shown to attain better performance than other speech features for the SI task [24] when higher order cepstral coefficients are removed. The smooth nature of articulatory trajectories and co-articulation effect suggest that the input temporal context should be long enough to capture the needed information with respect to to the output trajectories [3]. We select every other frame in a $2 \times M_{\text{aai}}$ window preceding and succeeding the current frame to construct the following extended input vector:

$$X_{\text{aai}}[n] = \big[X[n - 2 \times M_{\text{aai}}]^T, \ldots, X[n-2]^T, X[n]^T,$$

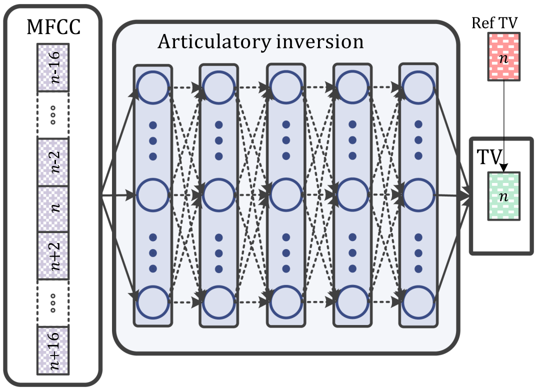

Fig. 1. DNN based AAI system with 340 ms input context of MFCCs, and tract variables (TV) as the output.

$$X[n+2]^T, \dots, X[n+2 \times M_{\text{aai}}]^T\big]^T, \qquad (1)$$

where X_{aai} is the contextualized MFCC vector for the AAI system and $[.]^T$ indicates the transpose operator. Employing every other frame gives us the benefit of longer temporal context with less parameters in the AAI model with no performance degradation.[1] Fig. 1 shows the structure of input data for a DNN-AAI system, where the output features are tract variables (TVs), which are described later in Section IV. The input features and output targets of the DNN-AAI system are mean and variance normalized at an utterance level. DNN-AAI systems trained on clean and multi-condition noisy speech data will in the following be denoted AAI-C and AAI-MC, respectively.

B. DNN-SE for AAI

In this solution, a DNN is first built to map noisy speech features into estimated clean features using a regression framework [46]. The AAI-C system is then used to estimate the articulatory trajectories. The DNN-SE system is based on a feed-forward layered structure of non-linear hidden layers and a linear output layer. The non-linear blocks allow the network to better handle the complex interactions between degraded noisy signal and its clean counterpart, as argued in [46]. The input features for the DNN-SE are globally mean and variance normalized Log Power Spectra (LPSs). LPSs have been obtained by taking the log of the squared magnitude of the signal's short-time Fourier transform (STFT). The DNN-SE enhances only the magnitude spectrum; therefore, the noisy phase is used in the reconstruction step (synthesis). In this work, we synthesise the enhanced speech waveform from enhanced magnitude and noisy phase spectrum using the the overlap-add method [52], which was also used in [46], to be able to assess the quality of the enhanced speech. To take into account context information, M_{se} previous and future frames around the current frame are used at the DNN-SE input:

$$S_{\text{se}}[n] = \left[S[n-M_{\text{se}}]^T, \dots, S[n]^T, \dots, S[n+M_{\text{se}}]^T\right], \quad (2)$$

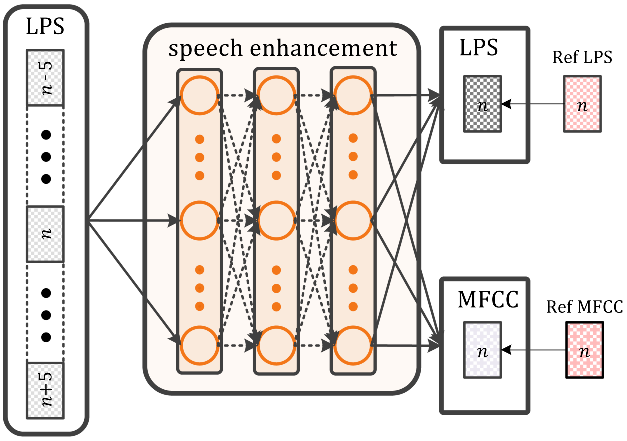

Fig. 2. DNN based SE system with 120 ms context of noisy LPSs, and clean LPSs and MFCCs as the output.

where the S_{se} is the contextualized LPS of the noisy signal as the input vector.

It should be noted that M_{se} is shorter than M_{aai}, that is, less context is taken into account in the SE step. That is coherent with the non-stationary property of noises, which enables the network to have a better estimation of short-time noise spectrum to be suppressed. At a target level, there are several possible choices, namely, only clean LPS can be used in a single-task learning procedure, or both MFCC and LPS can be employed in a multi-task scenario. In the multi-task case, the back propagated loss from the MFCC output layer acts as a regularizer and would prevent the model to over-fit to the training data. Moreover, the MFCC-related output layer can be directly used as an input of the AAI-C system. Fig. 2 shows a sketch of DNN-SE system with multiple output tasks.

Although MFCCs can be derived from LPS through a transformation, joint estimation of enhanced LPS and MFCCs may impose additional constraints unavailable in the direct prediction of clean LPS. As discussed in [53], Mel-filtering is applied to make the acoustic features consistent with human auditory perception. However there is so far no prior auditory knowledge adopted in the LPS domain except for the log-compression, and clean LPS features could therefore be better predicted with a MFCC constraint imposed at the output layer. Furthermore, the correlation information among different channels can be incorporated in each MFCC coefficient due to the discrete cosine transformation (DCT) [54] operation. Therefore, we expect that correlated and consistent distortion across different frequency bins can be learned when predicting the clean LPS. Differently from [53] the DCT block in our pipeline also performs dimensionality reduction, since we use MFCCs for the AAI block.

C. Joint DNN-SE and DNN-AAI

In Sections II-A and II-B, we described the two independent DNN-based systems for AAI and SE task respectively, where the DNN-SE module could be employed in a pre-processing step before the target AAI task to be accomplished with the DNN-AAI system. Since the two independent systems are built

[1] Experiments with different decimation factors, D, showed no PCC degradation for $D = 2$ and a moderate degradation for $D = 3$ and 4.

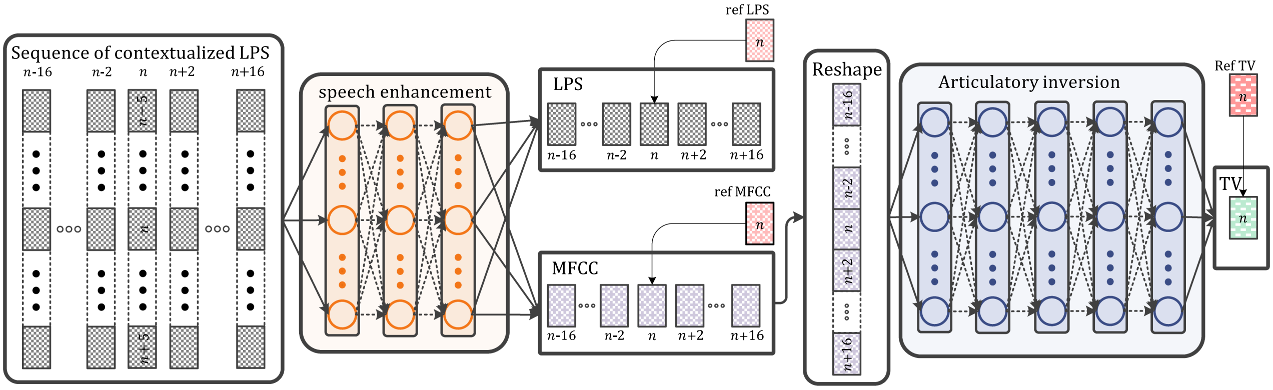

Fig. 3. Network structure of joint training of SE and AAI systems ($M_{se} = 5, M_{aai} = 8$).

within the connectionist framework, we can stack them back-to-front and obtain a single overall AAI system. The overall system can be further fine-tuned using the same loss employed to build the DNN-AAI system. However, the fusion of those two systems into a single one is challenging, because of the different temporal contexts used to build the two systems independently. As mentioned in Section II-B, the DNN-SE input context size (M_{se}) is smaller than AAI-C one (M_{aai}). The required frames for building the AAI input need to be provided at the input layer of the DNN-SE module. To this end, a sequence S_{joint} of contextualized speech vectors is presented at the input of the joint system, where the sequence is built as follows:

$$S_{\text{joint}}[n] = \left[S_{\text{se}}[n - 2 \times M_{\text{aai}}]^T, \ldots, S_{\text{se}}[n-2]^T, S_{\text{se}}[n]^T,\right.$$
$$\left. S_{\text{se}}[n+2]^T, \ldots, S_{\text{se}}[n + 2 \times M_{\text{aai}}]^T\right], \quad (3)$$

The DNN-SE module thus generates all needed input frames for the AAI module, $X_{\text{aai}}[n]$. In the training stage, back propagated error for the enhancement part is limited to that referring to the middle contextualized vector in the input sequence. The proposed architecture is illustrated in Fig. 3 where the LPS and MFCC tasks are considered for the current time n. In this way the network parameters are trained on the current time n, while being able to deal with the different time-varying nature of the events to be handled in the two modules. For the AAI module, the output concerned with the MFCC task for each input sequence of contextualized LPSs is reshaped to build the AAI input vector. The overall system loss function based on mean squared error (MSE) is formulated as follows:

$$L_{\text{joint}} = \frac{1}{N}\sum_{i=1}^{N} ||\boldsymbol{y}_i^{\text{LPS}} - \hat{\boldsymbol{y}}_i^{\text{LPS}}||^2 + ||\boldsymbol{y}_i^{\text{MFCC}} - \hat{\boldsymbol{y}}_i^{\text{MFCC}}||^2$$
$$+ ||\boldsymbol{y}_i^{\text{TV}} - \hat{\boldsymbol{y}}_i^{\text{TV}}||^2, \quad (4)$$

where, $\boldsymbol{y_i}^{(\ldots)}$s are the reference output vectors, $\hat{\boldsymbol{y}}_{\boldsymbol{i}}^{(\ldots)}$s are the estimated vectors for each output and N is the number of training samples.

III. Corpora and Data Representation

There are three tasks in this work, the main one is the AAI, the second one is speech enhancement, and the third task is automatic speech recognition. For the first task, two corpora are employed, the "Haskins Production Rate Comparison"database (HPRC) [55], which contains both acoustic and articulatory measurements, and the AURORA2 database [56] which contains eight noise types. For the second task, we additionally employ two datasets: TIMIT [57] with spoken American English; and Nonspeech [58] which contains 100 various noise types. For the third task, we use the WSJ dataset. In the following, the mentioned corpora are described in details. Furthermore, the representation of the acoustic and articulatory data is described.

A. Corpora

1) HPRC: The Haskins Production Rate Comparison (HPRC) database is selected as the main database for the AAI experiments. It contains recordings of eight native American English speakers, four female (F01-F04) and four male (M01-M04) speakers. There are 720 spoken utterances available in the dataset with both normal and fast speaking rate. For some of the normal speaking rate utterances, there are a few repetitions available. Speech waveforms are sampled at the rate of 44.1 kHz, and synchronous EMA recordings are available at a sampling rate of 100 Hz. EMA recordings are obtained from eight sensors, which record tongue rear or dorsum (TR), tongue blade (TB), tongue tip (TT), upper and lower lip (UL and LL), mouth left (ML), jaw or lower incisors (JAW) and jaw left (JAWL). The articulatory measurements are aligned to the occlusal plane in X, Y and Z directions, corresponding to movements from posterior to anterior, right to left and inferior to superior, respectively. The movements along the Y axis carry limited information. In this work, we employed only the X and Z directions of TR, TB, TT, UL, LL and JAW. Furthermore, we used 80% of data for training, 10% for validation, and the remaining 10% for test.

2) TIMIT: TIMIT [57], [59] is a speech corpus consisting of 6300 sentences spoken by 630 speakers, covering 8 major dialect regions of the United States. The dataset includes two dialect sentences (SA), 1890 phonetically diverse sentences (SI), and 450 phonetically compact sentences (SX). The training set is predefined and consists of all the SX and SI sentences from 462 speakers with a total of 3696 sentences. The sentences from the remaining 168 speakers constitute the full test set. We use the core test set [59], covering speech material from 24 speakers, for testing purposes. A validation set spoken by 50 speakers is used to prevent over-fitting and performance tuning with respect to the validation data. The core test set consists of 192 utterances, and the development set consists of 400 utterances.

3) Wall Street Journal - WSJ: The WSJ [51] corpus is in two distinct parts: WSJ0 and WSJ1. The SI-84 training material from the WSJ0 covers 7,193 utterances (15 hours). The SI-284 (80 hours) data is formed by combining training data from both the WSJ0 and WSJ1 (26,515 utterances). For development and evaluation, 503 utterances (1.1 h), and 333 utterances (0.7 h) are used, respectively. Clean waveforms, sampled at 16 kHz, and corresponding transcripts are provided for both WSJ0 and WSJ1. Waveforms were down-sampled to 8 kHz to carry out our downstream ASR experiments. Moreover, testing waveforms were corrupted with noise in order to create mismatched conditions between training (clean) and testing (noisy) and better assess the effect of introducing articulatory information into an end-to-end ASR system. More details are given in Section IV-G.

4) AURORA 2: AURORA 2 [60] is a corpus of noisy speech created by adding noise of various types and levels to clean speech recordings. In this work we only employ the noise recordings consisting of eight different noise types that are recorded in different places, namely, airport, crowd of people (babble), car, exhibition hall, restaurant, street, subway, and train station. The recordings contain stationary and non-stationary noise segments, and are sampled at a rate of 8 kHz.

5) Nonspeech: The Nonspeech dataset [58], which contains 100 different environmental noises, is recorded with a 20 kHz sampling rate and was downsampled to 8 kHz for our experiments. The noise types available in the dataset are as follows, N1-N17: Crowd noise, N18-N29: Machine noise, N30-N43: Alarm and siren, N44-N46: Traffic and car noise, N47-N55: Animal sound, N56-N69: Water sound, N70-N78: Wind, N79-N82: Bell, N83-N85: Cough, N86: Clap, N87: Snore, N88: Click, N88-N90: Laugh, N91-N92: Yawn, N93: Cry, N94: Shower, N95: Tooth brushing, N96-N97: Footsteps, N98: Door moving, N99-N100: Phone dialing.

6) Simulated Multi-Condition Dataset: Multi-condition waveforms are synthetically generated by randomly adding noise from AURORA2 and Nonspeech to the HPRC and TIMIT speech samples at different signal-to-noise ratios (SNR). The multi-condition data set also includes clean data. To match the 8 kHz sampling rate of the AURORA2 database the audio material from the other datasets is downsampled to 8 kHz. Another constraint is imposed by the 100 Hz sampling rate of the articulatory measurements, which leads to a frame shift of 10 ms to match the 100 Hz sampling rate.

B. Articulatory Data Representation

As reported in [61], geometrical transformations can be applied to the EMA measurements in order to transform those measurements into tract variables (TVs). TVs have the property of being more speaker independent than the original measurements, because they are relative measures and suffer less from non-uniqueness [62]. We use nine TVs, including Lip Aperture (LA), Lip Protrusion (LP), Jaw Angle (JA), Tongue Rear Constriction Degree (TRCD), Tongue Rear Constriction Location (TRCL). For TB and TT, we also calculate TBCD, TBCL, TTCD and TTCL, as explained below. The aforementioned geometrical transformations are defined as follows:

$$\mathrm{LA}[n] = \sqrt{(\mathrm{LL}_x[n] - \mathrm{UL}_x[n])^2 + (\mathrm{LL}_z[n] - \mathrm{UL}_z[n])^2}, \tag{5}$$

$$\mathrm{LP}[n] = \mathrm{LL}_x[n] - \underset{m \in \text{all utterances}}{\mathrm{median}} \mathrm{LL}_x[m]. \tag{6}$$

LA represents the Euclidean distance between LL and UL sensors. LP is defined as the movement of LL from its median position in the X direction,

$$\mathrm{JA}[n] = \sqrt{(\mathrm{JAW}_x[n] - \mathrm{UL}_x[n])^2 + (\mathrm{JAW}_z[n] - \mathrm{UL}_z[n])^2}, \tag{7}$$

is defined as the Euclidean distance between the JAW and UL sensors.

For each of the tongue sensors TR, TB and TT, two TVs are defined. Those TV features represent constriction locations, which are the deviations from median of the corresponding sensor along the X axis, and the constriction degree, which is the minimum distance between the corresponding tongue sensors position and the palate trace. TRCL and TRCD are defined as follows

$$\mathrm{TRCL}[n] = \underset{m \in \text{all utterances}}{\mathrm{median}} \mathrm{TR}_x[m] - \mathrm{TR}_x[n], \tag{8}$$

$$\mathrm{TRCD}[n] = \min\left\{\sqrt{(\mathrm{TR}_x[n] - x_{pal})^2 + (\mathrm{TR}_z[n] - z_{pal})^2}\right\}, \tag{9}$$

where x_{pal} and z_{pal} are the palate coordinates on the occlusal plane.

The remaining four variables TBCL, TBCD, TTCL and TTCD can be obtained in a similar way:

$$\mathrm{TBCL}[n] = \underset{m \in \text{all utterances}}{\mathrm{median}} \mathrm{TB}_x[m] - \mathrm{TB}_x[n], \tag{10}$$

$$\mathrm{TBCD}[n] = \min\left\{\sqrt{(\mathrm{TB}_x[n] - x_{pal})^2 + (\mathrm{TB}_z[n] - z_{pal})^2}\right\}, \tag{11}$$

$$\mathrm{TTCL}[n] = \underset{m \in allutterances}{\mathrm{median}} \mathrm{TT}_x[m] - \mathrm{TT}_x[n], \tag{12}$$

$$\mathrm{TTCD}[n] = \min\left\{\sqrt{(\mathrm{TT}_x[n] - x_{pal})^2 + (\mathrm{TT}_z[n] - z_{pal})^2}\right\}. \tag{13}$$

C. Acoustic Feature Representations

As discussed in the previous sections, we study three tasks. The first task is the AAI, which is the main task in the present work; the SE is the second task. Both tasks are addressed under the DNN framework. AAI models are trained over MFCC feature vectors, which are extracted using a 20 ms windowed signal with a frame shift of 10 ms. 13-dimensional MFCC feature vectors are extracted from 23 Mel-scaled filter banks. For the AAI system we set $M_{\text{aai}} = 8$. This moderately long temporal span covers 340 ms of the input acoustic data. As already mentioned, the temporal context improves the AAI performance due to the smooth varying nature of the articulator trajectories. For the SE system, the log power spectra (LPS) (256 coefficients) are calculated for 20 ms windowed signal with 10 ms frame shift. The temporal context with $M_{\text{se}} = 5$ spans past and future frames around the target frame at time n, that is equivalent to 120 ms of speech.

IV. Experiments and Results

The key experiments reported in this section are concerned with AAI, and the effect of speech enhancement on AAI. Speech enhancement quality is also reported for all of the DSP- and DNN-based systems investigated in this work. Finally, the role of AAI system in a downstream speech application is assessed using an ASR task in noisy condition.

Moreover, several experiments have been carried out to validate the proposed approach and fine tune all models. With respect to the optimization of network parameters, the AAI systems investigated in the present work have been built leveraging clean and multi-condition data, resulting in AAI-C and AAI-MC systems, respectively. The AAI-MC system is considered in the present study, because it was recently reported as the best AAI solution in noisy conditions [42]. For the evaluation of the optimized networks, clean, multi-condition and enhanced multi-condition data are used. Moreover, all of the AAI experiments are carried out in mismatched speaker conditions using the leave-one-speaker-out (LOSO) cross-validation scheme during the training phase. For speech enhancement, we compare and contrast the IMCRA-based DSP approach (DSP-SE), and the feature-based vector-to-vector regression with deep models for speech enhancement approach (DNN-SE) discussed in [46]. The DNN-SE is deployed using a deep feed-forward neural network with three hidden non-linear layers, each having 1024 nodes. ReLU activation functions [63] were employed in both AAI and SE neural modules; whereas, a linear activation function was used at the output layer. The PCC criterion was used to select the best performing network on the validation data. Moreover, early stopping prevents over-fitting to the training data, and training is halted either when the PCC on validation data does not improve for 10 consecutive epochs, or a total number of epochs equal to 100 has been reached. The ADAM optimizer [64] was employed to minimize the MSE between the ground-truth and estimated tract variables. All neural models implemented in our work were built using the Tensorflow library [65] with Keras API [66]. Drop-out [67] was used to contrast over-fitting, and a drop-out rate of 10% was used in each hidden layer. Different DNN-SE systems have been built using a different experimental setups, namely:

1) matched speakers, noise types, and SNRs between training and testing phases;
2) mismatched speakers but matched noise types and SNRs between training and testing phases;
3) mismatched speakers, noise types and SNRs between training and testing phases.

The purpose of latter experimental setup is to verify the applicability of DNN-SE in real-world conditions, where having similar speakers, noise types and SNRs is highly unlikely. In our experiments, we consider SNR levels in the range between -5 dB to 20 dB in incremental steps of 5 dB. In the following, experiments and results are presented and discussed in more detail, yet we first introduce the metrics used in this work to assess all systems.

A. Test Data

Because we employ several corpora in this work, the data split needs to be clarified. In all simulations, the test set is from the HPRC database. In the case of multi-condition data, the test set is distorted by additive noises from AURORA2.

B. Performance Metrics

The Pearson's correlation coefficient (PCC) was used as a measure of accuracy between the estimated and the reference TVs in the AAI systems. The reason for choosing PCC is that the PCC is a normalized measure and varies between -1 to 1, and it is independent from the difference in articulatory measurement's ranges which is related to speakers' anatomies. A higher value of the PCC shows better performance of inversion system.

Perceptual evaluation of speech quality (PESQ) was used to evaluate the quality of the enhanced speech [68]. For computing the PESQ, enhanced speech waveforms were synthesized from the enhanced LPS and the noisy phase spectra. The PESQ score ranges from -0.5 to 4.5, and the higher the PESQ score, the closer the enhanced speech is to the original clean speech. Indeed, PESQ has been proven to provide a high correlation to the quality scores rated by humans [69].

C. DNN-AAI Results

Using LOSO cross-validation during training, each of the eight speakers, in turn, becomes a test speaker while the remaining seven speakers are used in the training phase. Reported results are thus averaged across all test speakers. Several experiments varying the number of hidden layers and nodes in the DNN were carried out. PCC is used to select the best AAI system using the validation data. In particular, the following configurations were investigated: $[100, 300, 500, 1000]$ nodes, and [2, 3, 4, 5] hidden layers. The PCC value is reported in the upper panel in Figure 4 when clean data are used; PCC curves show that the best performing AAI system has 5 hidden layers with 100 nodes per layer. As the amount of available data is limited, it is reasonable that increasing the number of parameters would not lead to a performance improvement. The same set of experiments was executed using multi-condition data, and results are reported in the lower panel of Figure 4. We can see that either 4 or 5 hidden layers with 300 nodes can lead to the best PCC score. For our following experiments we have chosen the configuration with 4 hidden layers to save computational resources.

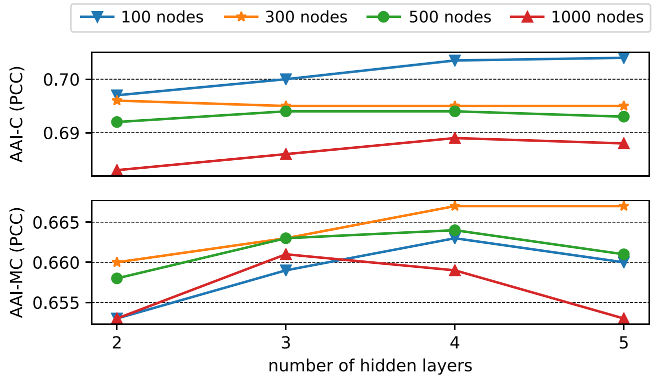

Fig. 4. Average PCC performance vs AAI DNN parameters with matched training and test data: clean data (top panel) and multi-condition data (bottom panel).

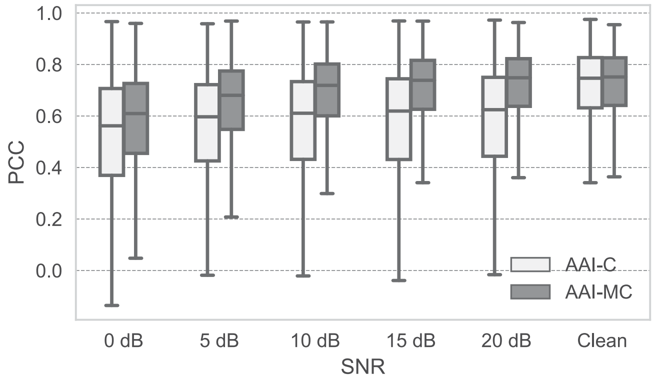

Fig. 5. Average PCC for multi-condition data with respect to different SNR levels. The box plots represent the minimum, first quartile, median, third quartile, and the maximum of average PCC values.

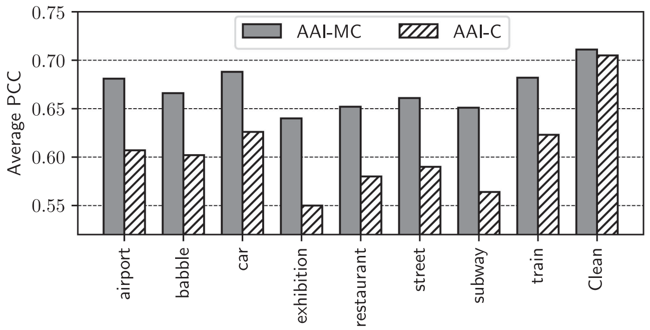

Fig. 6. Average PCC for multi-condition data on AAI-C and AAI-MC models, with respect to different noise types.

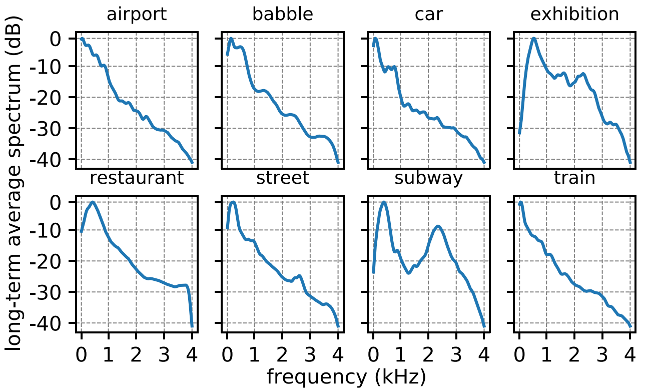

Fig. 7. Long-term average spectrum of different noise types in Aurora 2 database.

After tuning the neural parameters, the average PCC on the test set is calculated and reported with respect to two different aspects, namely:

1) SNR level, and
2) noise type.

Experimental results for different SNRs are shown in Fig. 5 for both AAI-C and AAI-MC systems. It can be observed that AAI-MC attains almost similar PCC on clean data and noisy data at SNRs $\geq$ 15 dB. It can be concluded that the required speech information for the inversion are obtainable at these SNRs. The high standard deviation in the PCC distribution in Fig. 5 is due to several factors, e.g., different test speakers performance, different variation range for each of the TVs and the effect of various noise types.

The effect of different noise types on the performance of AAI-C and AAI-MC systems is shown in Fig. 6 that shows the average PCC over different speakers and SNRs. It can be observed that 'exhibition' and 'subway' noises have the greatest negative effects on AAI accuracy and cause a significant performance drop; in contrast, 'car' and 'train' noises have a minor negative effects on the final AAI accuracy. Inspecting the long term averaged power spectrum of different noise types in Fig. 7, we can observe that a common feature of the noise types that cause the most severe degradation of the AAI performance is that they have considerable energy in frequency bands between 1 kHz and 3 kHz. For clean data, AAI-MC performs slightly better than the AAI-C, which can be explained thinking of the larger amount of training data used to build the system, i.e., a consequence of the data-augmentation effect, especially data at an SNR equal to 20 dB.

D. DNN-SE Results

The DNN-SE system has been trained in three different scenarios. We briefly describe each scenario along with the corresponding training procedure in the following.

DNN-SE1 - Matched Speakers, Noise Types, and SNRs: The HPRC dataset is used for the speech material, and the AURORA 2 noises are added to it in order to synthetically simulate noisy speech. All of the eight possible noises are added to the speech waveforms at different SNR levels. The same speakers, noise types and SNR levels are employed for creating training, validation, and test data. Furthermore, these settings are used in both the single and multi-task approaches (see Section II-B). SNR levels are [0, 5, 10, 15, and 20] dB.

DNN-SE2 - Mismatched Speakers, Matched Noise Types and SNRs: The speech material and noises are the same as those employed in the first experimental scenario. Mismatch between training and testing condition was inserted at a speaker level.

TABLE I
PESQ PERFORMANCE COMPARISON OF SINGLE-TASK (ST) AND MULTI-TASK (MT) SPEECH ENHANCEMENT SYSTEMS BASED ON DNN FOR THREE DIFFERENT SCENARIOS

SNR	Noisy	DSP-SE	DNN-SE1		DNN-SE2		DNN-SE3
			ST	MT	ST	MT	MT
-5 dB	—	—	—	—	—	—	2.359
0 dB	1.51	1.700	2.554	2.653	2.365	2.528	2.580
5 dB	1.75	2.077	2.767	2.873	2.544	2.729	2.770
10 dB	2.06	2.533	2.955	3.069	2.702	2.907	2.937
15 dB	2.47	2.950	3.104	3.224	2.828	3.048	3.074
20 dB	2.97	3.316	3.205	3.333	2.919	3.148	3.180

TABLE II
PERFORMANCE OF SI-AAI SYSTEMS TRAINED ON CLEAN AND MULTI-CONDITION DATA AND TESTED ON CLEAN, MULTI-CONDITION AND ENHANCED DATA

Test data	Enhancement	AAI-C	AAI-MC
Clean	None	0.705	0.710
Multi-Cond	None	0.595	0.665
Multi-Cond	DSP-SE	0.568	0.620
Multi-Cond	DNN-SE1-MT	0.699	0.711
Multi-Cond	DNN-SE1-ST	0.689	0.702
Multi-Cond	DNN-SE2-MT	0.670	0.701
Multi-Cond	DNN-SE2-ST	0.662	0.693
Multi-Cond	DNN-SE3-MT	0.678	0.697

For each speaker, a stand-alone network is built using the other seven speakers in the training phase, which is applied to speech from the given speaker in the testing phase. In doing so, the deep model is more realistic and better simulates real world applications compared to the previous scenario, which may be useful for a feasibility assessment. SNR levels are again [0, 5, 10, 15, and 20] dB.

DNN-SE3 - Mismatched speakers, noise types, and SNRs: In this third experimental scenario, the 8 kHz version of the TIMIT corpus is used for the speech material, and the challenging Nonspeech database is used for the noises. The validation set also comes from TIMIT and Nonspeech. The test set consists of material taken from the HPRC speakers and degraded by AURORA 2 noises. The SNR levels in training are [0, 5, 10, and 20] dB. Different SNRs, namely [-5, 0, 5, 10, 15, and 20] dB, are used in the test phase. These experimental conditions are closer to what one can expect in real production; moreover, our DNN-SE module is trained on independent data and noises with respect to the testing conditions, so it functions a general purpose SE tool.

Table I shows the average PESQ for models trained and tested as discussed above. A visual inspection of Table I shows that DSP-SE improves the average PESQ by 0.1 for 0 dB, 0.2 for 5 dB, 0.4 for 10 dB, 15 dB, and 20 dB. A main issue with the DSP-SE method is its poor performance at low SNRs, yet its strength is the inherent nature of the DSP solution that does not require training data and makes it a general SE tool for real-world applications. The best results are expected for DNN-SE1, which is trained in matched conditions for speakers, noise types, and SNR levels. Both single-task (DNN-SE1-ST) and multi-task (DNN-SE1-MT) configurations are evaluated. DNN-SE1-MT achieves a better performance than DNN-SE1-ST, as it can be observed comparing columns four and five in Table I. This confirms our intuition about the regularization effect of the multi-task configuration. Indeed, DNN-SE1-MT attains better PESQ compared to DSP-SE and DNN-SE1-ST in all tested SNRs.

Experiments in matched condition demonstrated the feasibility of our idea, and the positive effect of a multi-task configuration for a SE task. DNN-SE2 is built using a different training configuration, which takes into account a mild level of mismatch between training and testing phases. Therefore, a small drop in the SE performance is expected, and results reported in the sixth and seventh column in Table I confirm our expectation. Moreover, DNN-SE2-MT attains a performance comparable to DNN-SE1-ST in spite of the more challenging SE scenario. Given that multi-task is a viable way to boost SE performance in mismatched conditions, only DNN-SE3-MT is built in the third experimental scenario, the most realistic and challenging one. Since DNN-SE3-MT is trained as general purpose SE module, it is not a surprise that it shows PESQ values superior to those attained with DNN-SE2-MT. The key strength of DNN-SE3-MT compared to the DNN-SE2-MT is the larger number of speakers, and thereby speech material, used in the training phase along with the more challenging noises that the model had to deal with. In mismatched SNRs, very promising results are obtained; for example, at an SNR of 15 dB, DNN-SE3-MT slightly outperforms DNN-SE2-MT in terms of PESQ. For -5 dB the PESQ value is 2.359 which it is ≈0.2 less than 0 dB, and the PESQ value is 2.580 at 0 dB is ≈0.2 less than the PESQ value at 5 dB.

In general DNN-SE models have higher performance than DSP-SE at low SNR levels.

E. AAI on SE Data

We investigated the effect of enhancing the speech data prior to AAI. Because it is unlikely that clean data are available in real production, SE modules are employed prior to the AAI-C system, as a pre-processing step. In doing so, we can use an off-the-shelf AAI-C model without exploiting MC training. We compare and contrast the effect of both DSP-SE and DNN-SE on AAI, and the AAI-MC performance is reported to ease the comparison.

First, we notice from Table II that AAI-C tested on data enhanced by DNN-SE performs better than AAI-MC tested on multi-condition data without enhancement. On the one hand, it can be argued that the improvement comes from an increase of the neural parameters obtained by coupling two deep models. On the other hand, it should be noted that the DNN-SE and the AAI-C deep model were independently trained on different data, and our solution allows us to use an off-the-shelf AAI-C system avoiding training a new system from scratch. This aspect should not be underestimated in a production pipeline of a real complex system. It should also be recalled that [42] reported DSP-SE to cause a drop in the AAI performance. We therefore further compare DSP-SE and DNN-SE effects on AAI-C. In Table II, we see that DSP-SE coupled with AAI-C indeed causes 0.11 drop in the PCC compared with our DNN-SE-MT3 coupled with AAI-C. Most importantly, for AAI-C, applying DSP-SE to

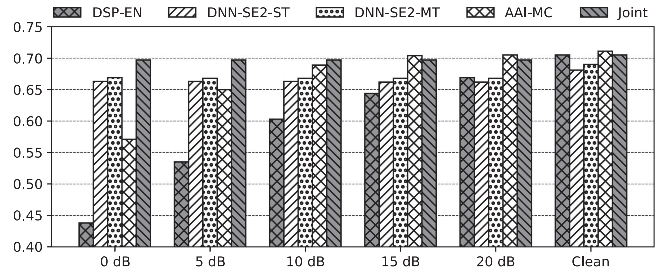

Fig. 8. PCC of the AAI-C on enhanced speech data, AAI-MC and joint systems on multi-condition data, at different SNRs.

the noisy data reduces the PCC by 4.5% relative compared to using the noisy data directly. That result is in line with [42], and it could be explained by the signal distortions usually introduced by DSP-SE, such as musical noise [70]. In contrast, our DNN-SE method does not cause any drop in the AAI performance, and our findings open up a new path for DNN-based front-end approaches in speech applications. In Table II, we see that the DNN-SE system has a significant improvement over the DSP-SE, an increase of 0.11 in terms of average PCC, and a relative improvement of 19.36% is achieved using DNN-SE3-MT system over DSP-SE. For the sake of completeness, we report experimental results with multi-task (MT) and single-task (ST) training strategies in Table II, both in matched and mismatched speaker scenarios. The multi-task DNN-SE methods outperform single-task counterparts; whereas, a drop in PCC is observed when moving from matched to mismatched speakers. However, speech enhancement is performed to avoid building an AAI-MC system, we provided results using AAI-MC on clean, noisy and enhanced data for completeness. Interestingly, enhancing the noisy speech with the DNN-SE based systems improves the AAI-MC model's performance, in contrast with what is observed for DSP-SE. It should be noted the better performance of separate DNN-SE1-MT model with AAI-MC model in comparison with the joint model performance is due to the matched speaker condition of SE module. A detailed comparison in terms of SNR values of the AAI-C on DSP-SE and DNN-SE systems, is shown in Fig. 8. Enhancement with DNN-SE2-MT always gives a better PCC in low SNR conditions. Moreover, DNN-SE2 and DSP-SE lead to similar PCC only in very high SNR. At 0 db, from Figs. 6 and 8, we see that AAI-MC attains a PCC of 0.579, and AAI-C on DNN-SE enhanced data attains a PCC of 0.67, which accounts for a 15% relative improvement in favor of the proposed DNN-SE based AAI-C approach.

The DNN-SE methods cause degradation for the clean data performance compared to the performance of clean data on the AAI-C. The performance degradation of AAI-C, for enhanced clean data by DNN-SE system, can be explained by over-smoothing of enhanced speech compared to the natural ones or enhanced by DSP-SE method.

F. Joint AAI and SE Based on DNN

So far we have investigated the AAI system either using stand alone AAI systems or decoupled SE and AAI system. We now address both the SE and AAI tasks under a unified DNN framework, by coupling the two deep architectures into a single network and leveraging the availability of the MFCC output in the SE module. Then, the overall network can be jointly fine-tuned with the goal of accomplishing AAI. Training the joint model is challenging because back-propagation of different tasks affect each other and make the convergence slower compared to learning different models designed to accomplish different tasks. To improve convergence, there are two alternative procedures available:

1) First, the speech enhancement module is trained while keeping AAI parameters frozen, so that the gradient flows back through the network layers until the enhancement module converges. Next, the speech enhancement module weights are kept frozen, and the AAI parameters are updated till convergence. In this way, the training scheme will be similar to AAI training with enhanced multi-condition data.
2) Initializing each the connectionist parameters with the pre-trained DNN-SE3 and AAI-C weights, and then fine-tune the whole system with the goal of accomplishing AAI. In this way both modules start from a better initialization starting point.

TABLE III
JOINT SPEECH ENHANCEMENT AND ARTICULATORY INVERSION PERFORMANCE IN TERMS OF PESQ AND PCC

	0 dB	5 dB	10 dB	15 dB	20 dB
PESQ	2.655	2.864	3.050	3.197	3.301
PCC	0.697	0.697	0.697	0.697	0.697

We decided to use the second approach to carry out joint training of the SE and AAI blocks. The LOSO cross-validation approach is utilized for training of the joint model. The multi-condition data is kept the same as in the previous experiments, to have comparable results. Table III reports results with joint training. It is interesting to see that we can improve both SE and AAI tasks in terms of PESQ and PCC, respectively. It should be recalled that the DNN-SE has a primary task which corresponds to enhancing the LPS speech vector. By comparing PESQ values in Tables I and III, we can observe that the SE module in the joint model attains results close to the DNN-SE1-MT model which is the best performing enhancement model presented in this work. The AAI performance for different SNR levels are the same to the third decimal place. The AAI performance of the joint model on multi-condition data is PCC=0.697, and the AAI-C model performance on clean data is PCC=0.705. The joint model performance is closer to the AAI-C system on clean data than the performance of either the AAI-MC system on multi-condition data (PCC=0.665), or the AAI-C system on DNN-SE3-MT data (PCC=0.678). This performance is expected considering that the AAI part is tuned for the enhanced data in the joint training of the enhancement and inversion systems. In addition, from Fig. 8, it can be observed AAI-MC system performs better than the AAI-C system with enhanced data by DNN-SE modules at SNR$\geq$10 dB. The joint model decrease this under performing.

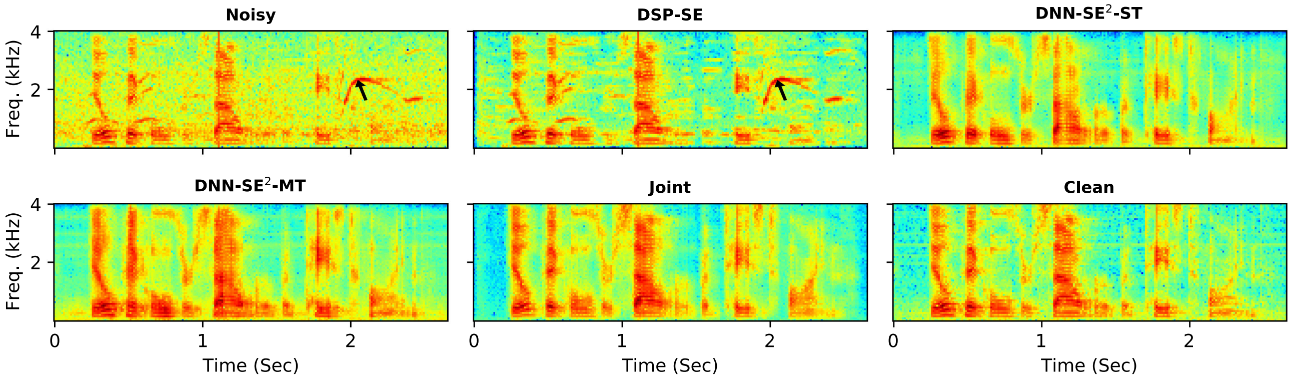

Fig. 9. Spectrogram of the utterance "Dill pickles are sour but taste fine," corrupted by Exhibition noise at SNR=5 dB. (a) noisy speech with (PESQ=1.768), (b) enhanced by DSP-SE (PESQ=1.815), (c) single-task DNN based model (PESQ=2.204), (d) multi-task DNN based model (PESQ=2.55), multi-task DNN based model jointly with the articulatory inversion (PESQ=2.89), and (f) the clean speech signal. Black arrows indicate the high energy whistle sound.

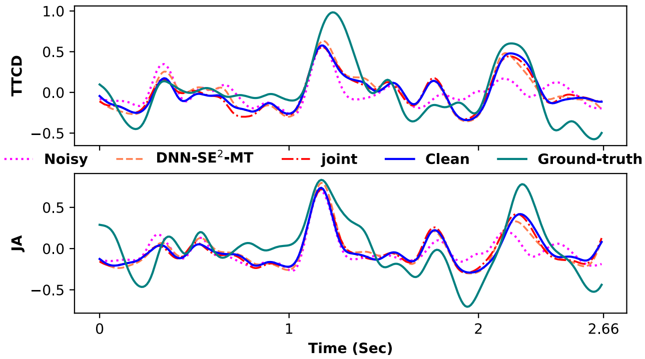

Fig. 10. TTCD and JA trajectories for utterance "Dill pickles are sour but taste fine," distorted by exhibition noise at SNR=5 dB.

Fig. 9 shows spectrograms for a testing utterance corrupted by exhibition noise at an SNR equal to 5 dB, clean, and enhanced with different SE methods. The DSP-SE method clearly introduces some distortions in the form of musical noise. Moreover, it could not remove high energy whistle sound (indicated by black arrows) starting at ∼1.93s. The DNN based methods are instead able to suppress different noise characteristics in the noisy signal but the over-smoothing affects the higher frequency components. However, the unwanted whistle sound is completely suppressed by all of DNN-SE methods.

TTCD and JA trajectories for the same selected utterance are depicted in Fig. 10. The "Noisy‘' one is estimated using AAI-MC model, and the other trajectories are enhanced and predicted by AAI-C model. From those trajectories in Fig. 10, we can argue that the enhanced speech by DNN-SE methods allows to obtain AAI accuracy like those obtained on the clean speech signal. The estimated trajectories by the AAI-MC with the noisy data as the input are very different with the estimated trajectories by the AAI-C model, e.g. the estimated JA at ∼ 2s which is due to the whistle distortion.

G. AAI for ASR

We now turn our attention on assessing the role of articulatory information on downstream speech tasks. To this end, a continuous word recognition task is considered, namely the WSJ0 [51], and several end-to-end automatic speech recognition (ASR) systems are built and contrasted to demonstrate the effect of TV information on the ASR performance in both clean, and noisy conditions. The word error rate (WER) is selected as the metric to compare the accuracy of all systems deployed in this section.

Clean data is already available with the WSJ0 corpus, and noisy data are synthetically generated by adding two noise types, namely exhibition, and subway. In the previous sections, the most adverse effects on AAI accuracy were caused by these noise types. Two SNR levels are used for training and testing, namely 0 dB and 10 dB. WSJ waveforms are downsampled from 16 kHz to 8 kHz. 60-dimensional log Mel filter bank energy (FBE) features were extracted using a 512-point short-time Fourier transform to compute the spectra of each overlapping windowed frame. A 32-ms Hamming window and a 16-ms window shift were adopted. The end-to-end ASR systems are all based on the end-to-end ESPnet recognizer [71], which is a character-based encoder-decoder model leveraging both a hybrid connectionist temporal classification (CTC) loss function, and an attention mechanism [72]. The encoder part contains 12 layers of BLSTM with 2048 cells, six layers of LSTM for the decoder with 2048 cells, and a location-aware attention mechanism with 10 convolution filters of length 100. The CTC loss and the attention loss were weighted by 0.2 and 0.8 respectively. Words are obtained from characters using an RNN language model, utilizing one LSTM layer with 1000 cells, which is trained on 65000 words from the WSJ1 corpus. In our experiments, the "dev93" part of WSJ0 corpus is used for parameter tuning. The actual evaluation is carried out on the for the "eval92" part.

We built two ASR systems using different data conditions, namely clean or noisy (0 dB and 10 dB), and different input speech features, namely FBEs, or FBEs and TVs. The first system is trained on clean data and used FBE features; we refer to this system as **System 1**, and it sets a WER lower-bound when testing on clean data, and an upper-bound in noisy conditions. The second system, **System 2** is trained on clean data and leverages both FBE and TV features. System 2 allows us to assess the effect of articulatory information on the downstream ASR task. Table IV shows all results gathered in our experiments. System

TABLE IV
WER for the "Eval92" Part of WSJ Database for the two Mentioned ASR Systems

Test Condition	System 1	System 2
Clean FBEs	5.3	—
Clean FBEs + TV (AAI-MC)	—	5.5
Clean FBEs + TV (DNN-SE+AAI-C)	—	5.4
Clean FBEs + TV (Joint)	—	5.4
Enh Clean FBEs	6.1	—
10 dB FBEs	49.4	—
10 dB FBEs + TV (AAI-MC)	—	22.6
10 dB FBEs + TV (DNN-SE+AAI-C)	—	19.8
10 dB FBEs + TV (Joint)	—	19.1
Enh 10 dB FBEs	42.3	—
0 dB FBEs	78.2	—
0 dB FBEs + TV (AAI-MC)	—	57.8
0 dB FBEs + TV (DNN-SE+AAI-C)	—	51.4
0 dB FBEs + TV (Joint)	—	49.8
Enh 0 dB FBEs	68.4	—

1 is evaluated on three different conditions, namely clean, noisy, and enhanced FBE features obtained with DNN-SE3-MT. System 2 leverages TV features, which are obtained with the AAI-C model described in Section IV-C in the training phase. In the testing phase, however, TV features are obtained either using the AAI-MC model in Section IV-C, the DNN-SE3-MT+AAI-C model discussed in Section IV-E, or the joint model discussed in Section IV-F. A visual inspection of Table IV reveals that System 1 attains the best results on clean FBE features with a WER equal to 5.3%, and attains the worst WER (6.13%) when tested in clean condition on enhanced data, as expected. The use of TV features along with clean FBE does not cause a significant increase of the WER. In noisy conditions, namely testing on FBE extracted on waveforms at 10 dB and 0 dB SNRs, we can see that System 1 attains the worst WERs as expected.Interestingly, the injection of TV features in System 2 boosts the ASR recognition performance significantly. Given that System 2 is also trained on clean FBE features as System 1, the latter results allow us to argue that articulatory information plays a key role in the selected downstream speech tasks. Moreover, the estimated TVs from the joint model have the most effect on the System 2 performance in terms of WER.

V. Conclusion

We have investigated into the speaker-independent AAI problem in noisy speech conditions. We have shown that DNN-based speech enhancement for input noisy signals can boost the performance of the AAI-C system trained on clean data. A good improvement was also observed for the AAI-MC system trained on multi-condition data. In the mismatched-speaker scenarios, enhancing multi-condition data with DNN-SE combined with the AAI-C model performed better than the straight AAI-MC system, which clearly demonstrates the effectiveness of the proposed speech enhancement pre-processing with deep models. Although the AAI-C system with speech enhanced by DNN-SE systems performs better than the AAI-MC system for noisy data, the performance at high SNR levels is degraded. To cope with this degradation, a joint model was proposed to perform both speech enhancement and articulatory inversion, which demonstrated its benefit over separate systems for each task. The joint system performance is close to the performance of clean data in AAI-C system. The key strength of applying DNN based enhancement methods prior to the AAI-C model, compared to the AAI-MC method are their better performance at low SNRs which is beneficial for ASR systems in presence of noise. Our experimental results also sheds new light on the AAI problem by contrasting what reported in the recent literature, namely speech enhancement does not bring any improvement when used in a pre-processing prior to AAI with noisy data [42]. Finally, we show that articulatory information can be useful in downstream speech applications, namely end-to-end ASR.

References

[1] J. Schroeter and M. M. Sondhi, "Speech coding based on physiological models of speech production," in *Proc. Adv. Speech Signal Process.*, 1992, pp. 231–267.

[2] J. Frankel and S. King, "ASR-articulatory speech recognition," in *Proc. 7th Eur. Conf. Speech Commun. Technol.*, 2001, pp. 599–602.

[3] V. Mitra, "Articulatory information for robust speech recognition," Ph.D. dissertation, Univ. Maryland, Maryland, 2010.

[4] V. Mitra, H. Nam, C. Y. Espy-Wilson, E. Saltzman, and L. Goldstein, "Articulatory information for noise robust speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 7, pp. 1913–1924, Sep. 2011.

[5] Z.-H. Ling, K. Richmond, and J. Yamagishi, "Articulatory control of HMM-based parametric speech synthesis using feature-space-switched multiple regression," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 1, pp. 207–219, Jan. 2013.

[6] K. Richmond and S. King, "Smooth talking: Articulatory join costs for unit selection," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 5150–5154.

[7] T. Hueber, A. Ben Youssef, G. Bailly, P. Badin, and F. Elisei, "Cross-speaker acoustic-to-articulatory inversion using phone-based trajectory HMM for pronunciation training," in *Proc. Interspeech*, 2012, pp. 783–786.

[8] O. Engwall, "Analysis of and feedback on phonetic features in pronunciation training with a virtual teacher," *Comput. Assist. Lang. Learn.*, vol. 25, no. 1, pp. 37–64, 2012.

[9] B. S. Helfer, T. F. Quatieri, J. R. Williamson, D. D. Mehta, R. Horwitz, and B. Yu, "Classification of depression state based on articulatory precision," in *Proc. Interspeech*, 2013, pp. 2172–2176.

[10] S. Sahu and C. Y. Espy-Wilson, "Speech features for depression detection," in *Proc. Interspeech*, 2016, pp. 1928–1932.

[11] D. W. Massaro, S. Bigler, T. Chen, M. Perlman, and S. Ouni, "Pronunciation training: The role of eye and ear," in *Proc. 9th Annu. Conf. Int. Speech Commun. Assoc.*, 2008, pp. 2623–2626.

[12] S. Fagel and K. Madany, "A 3-D virtual head as a tool for speech therapy for children," in *Proc. 9th Annu. Conf. Int. Speech Commun. Assoc.*, 2008.

[13] S. Narayanan, K. N. S. Lee, A. Sethy, and D. Byrd, "An approach to real-time magnetic resonance imaging for speech production," *J. Acoustical Soc. Amer.*, vol. 115, no. 4, pp. 1771–1776, 2004.

[14] J. R. Westbury, G. Turner, and J. Dembowski, *X-Ray Microbeam Speech Production Database User's Handbook*, 1st ed. Madison: Univ. Wisconsin, Waisman Center on Mental Retardation & Human Development, 1994.

[15] P. W. Schönle, K. Gräbe, P. Wenig, J. Höhne, J. Schrader, and B. Conrad, "Electromagnetic articulography: Use of alternating magnetic fields for tracking movements of multiple points inside and outside the vocal tract," *Brain Lang.*, vol. 31, no. 1, pp. 26–35, 1987.

[16] D. Porras, A. Sepúlveda-Sepúlveda, and T. G. Csapó, "DNN-based acoustic-to-articulatory inversion using ultrasound tongue imaging," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2019, pp. 1–8.

[17] K. Kirchhoff, "Robust speech recognition using articulatory information," Ph.D. dissertation, Univ. Bielefeld, 1999.

[18] S. Deena, S. Hou, and A. Galata, "Visual speech synthesis using a variable-order switching shared Gaussian process dynamical model," *IEEE Trans. Multimedia*, vol. 15, no. 8, pp. 1755–1768, Dec. 2013.

[19] R. Korin, P. Hoole, and S. King, "Announcing the electromagnetic articulography (day 1) subset of the MNGU0 articulatory corpus," in *Proc. Interspeech*, Florence, Italy, Aug. 2011, pp. 1505–1508.

[20] C. Qin and M. Á. Carreira-Perpiñán, "A comparison of acoustic features for articulatory inversion," in *Proc. 8th Annu. Conf. Int. Speech Commun. Assoc.*, 2007, pp. 2469–2472.

[21] P. K. Ghosh and S. Narayanan, "Automatic speech recognition using articulatory features from subject-independent acoustic-to-articulatory inversion," *J. Acoustical Soc. Amer.*, vol. 130, no. 4, pp. EL 251–EL 257, 2011.

[22] H. Kawahara, I. Masuda-Katsuse, and A. De Cheveigne, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech Commun.*, vol. 27, no. 3/4, pp. 187–207, 1999.

[23] A. S. Shahrebabaki, N. Olfati, A. S. Imran, S. M. Siniscalchi, and T. Svendsen, "A phonetic-level analysis of different input features for articulatory inversion," in *Proc. Interspeech*, 2019, pp. 3775–3779.

[24] P. K. Ghosh and S. Narayanan, "A generalized smoothness criterion for acoustic-to-articulatory inversion," *J. Acoustical Soc. Amer.*, vol. 128, no. 4, pp. 2162–2172, 2010.

[25] G. Sivaraman, V. Mitra, H. Nam, M. Tiede, and C. Espy-Wilson, "Unsupervised speaker adaptation for speaker independent acoustic to articulatory speech inversion," *J. Acoustical Soc. Amer.*, vol. 146, no. 1, pp. 316–329, 2019.

[26] B. S. Atal, J. J. Chang, M. V. Mathews, and J. W. Tukey, "Inversion of articulatory-to-acoustic transformation in the vocal tract by a computer-sorting technique," *J. Acoustical Soc. Amer.*, vol. 63, no. 5, pp. 1535–1555, 1978.

[27] S. Ouni and Y. Laprie, "Modeling the articulatory space using a hypercube codebook for acoustic-to-articulatory inversion," *J. Acoustical Soc. Amer.*, vol. 118, no. 1, pp. 444–460, 2005.

[28] A. Toutios and K. Margaritis, "A support vector approach to the acoustic-to-articulatory mapping," in *Proc. 9th Eur. Conf. Speech Commun. Technol.*, 2005, pp. 3221–3224.

[29] S. A. Moubayed and G. Ananthakrishnan, "Acoustic-to-articulatory inversion based on local regression," in *Proc. 11th Annu. Conf. Int. Speech Commun. Assoc., Spoken Lang. Process. All*, 2010, pp. 937–940.

[30] T. Toda, A. W. Black, and K. Tokuda, "Statistical mapping between articulatory movements and acoustic spectrum using a Gaussian mixture model," *Speech Commun.*, vol. 50, no. 3, pp. 215–227, 2008.

[31] K. Richmond, "A trajectory mixture density network for the acoustic-articulatory inversion mapping," in *Proc. 9th Int. Conf. Spoken Lang. Process.*, 2006, pp. 577–580.

[32] B. Uria, I. Murray, S. Renals, and K. Richmond, "Deep architectures for articulatory inversion," in *Proc. 13th Annu. Conf. Int. Speech Commun. Assoc.*, 2012, pp. 867–870.

[33] P. Liu, Q. Yu, Z. Wu, S. Kang, H. Meng, and L. Cai, "A deep recurrent approach for acoustic-to-articulatory inversion," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 4450–4454.

[34] X. Xie, X. Liu, and L. Wang, "Deep neural network based acoustic-to-articulatory inversion using phone sequence information," in *Proc. Interspeech*, 2016, pp. 1497–1501.

[35] A. Illa and P. K. Ghosh, "Representation learning using convolution neural network for acoustic-to-articulatory inversion," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 5931–5935.

[36] A. Illa and P. K. Ghosh, "Low resource acoustic-to-articulatory inversion using bi-directional long short term memory," in *Proc. Interspeech*, 2018, pp. 3122–3126. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1843

[37] T. Biasutto-Lervat and S. Ouni, "Phoneme-to-articulatory mapping using bidirectional gated RNN," in *Proc. Interspeech*, 2018, pp. 3112–3116. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1202

[38] A. S. Shahrebabaki, S. M. Siniscalchi, G. Salvi, and T. Svendsen, "Sequence-to-sequence articulatory inversion through time convolution of sub-band frequency signals," in *Proc. Interspeech*, 2020, pp. 2882–2886.

[39] A. S. Shahrebabaki, N. Olfati, S. M. Siniscalchi, G. Salvi, and T. Svendsen, "Transfer learning of articulatory information through phone information," in *Proc. Interspeech*, 2020, pp. 2877–2881.

[40] V. Mitra, G. Sivaraman, H. Nam, C. Espy-Wilson, E. Saltzman, and M. Tiede, "Hybrid convolutional neural networks for articulatory and acoustic information based speech recognition," *Speech Commun.*, vol. 89, pp. 103–112, 2017.

[41] H. Nam, L. Goldstein, E. Saltzman, and D. Byrd, "TADA: An enhanced, portable task dynamics model in MATLAB," *J. Acoustical Soc. Amer.*, vol. 115, no. 5, pp. 2430–2430, 2004.

[42] N. Seneviratne, G. Sivaraman, V. Mitra, and C. Espy-Wilson, "Noise robust acoustic to articulatory speech inversion," in *Proc. Interspeech*, 2018, pp. 3137–3141. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1509

[43] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, no. 2, pp. 443–445, Apr. 1985.

[44] Y. Wang and D. Wang, "Towards scaling up classification-based speech separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 7, pp. 1381–1390, Jul. 2013.

[45] D. S. Williamson, Y. Wang, and D. Wang, "Reconstruction techniques for improving the perceptual quality of binary masked speech," *J. Acoustical Soc. Amer.*, vol. 136, no. 2, pp. 892–902, 2014.

[46] Y. Xu, J. Du, L. Dai, and C. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 1, pp. 7–19, Jan. 2015.

[47] A. Triantafyllopoulos, G. Keren, J. Wagner, I. Steiner, and B. W. Schuller, "Towards robust speech emotion recognition using deep residual networks for speech enhancement," in *Proc. Interspeech*, 2019, pp. 1691–1695. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-1811

[48] C. Donahue, B. Li, and R. Prabhavalkar, "Exploring speech enhancement with generative adversarial networks for robust speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 5024–5028.

[49] B. Ghai, B. Ramanan, and K. Müller, "Does speech enhancement of publicly available data help build robust speech recognition systems?," 2019, *arXiv:1910.13488*.

[50] I. Cohen, "Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging," *IEEE Speech Audio Process.*, vol. 11, no. 5, pp. 466–475, Sep. 2003.

[51] John S. Garofolo *et al.*, *CSR-I (WSJ0) Complete LDC93S6A. Web Download*. Philadelphia, PA, USA: Linguistic Data Consortium, 1993.

[52] D. Griffin and J. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 2, pp. 236–243, Apr. 1984.

[53] Y. Xu, J. Du, Z. Huang, L.-R. Dai, and C.-H. Lee, "Multi-objective learning and mask-based post-processing for deep neural network based speech enhancement," 2017.

[54] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, no. 1, pp. 90–93, Jan. 1974.

[55] M. Tiede, C. Y. Espy-Wilson, D. G. V. Mitra, H. Nam, and G. Sivaraman, "Quantifying kinematic aspects of reduction in a contrasting rate production task," *J. Acoustical Soc. Amer.*, vol. 141, no. 5, pp. 3580–3580, 2017.

[56] E. Rothauser, "IEEE recommended practice for speech quality measurements," *IEEE Trans. Audio Electroacoust.*, vol. 17, no. 3, pp. 225–246, Sep. 1969.

[57] L. F. Lamel, R. H. Kassel, and S. Seneff, "Speech database development: Design and analysis of the acoustic-phonetic corpus," in *Proc. DARPA Speech Recognit. Workshop*, L. S. Baumann, Ed., Feb. 1986, pp. 100–109.

[58] G. Hu and D. Wang, "A tandem algorithm for pitch estimation and voiced speech segregation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 8, pp. 2067–2079, Nov. 2010.

[59] J. S. Garofolo, L. F. Lamel, W. M. Fischer, J. Fiscus, D. S. Pallett, and N. L. Dahlgren, "DARPA TIMIT acoustic-phonetic continuous speech corpus," *Tech. Rep. NISTIR* 4930, 1993.

[60] H.-G. Hirsch and D. Pearce, "The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proc. 6th Int. Conf. Spoken Lang. Process.*, 2000, pp. 29–32.

[61] A. Ji, "Speaker independent acoustic-to-articulatory inversion," Ph.D. dissertation, Univ. Maryland, Maryland, 2014.

[62] R. S. McGowan, "Recovering articulatory movement from formant frequency trajectories using task dynamics and a genetic algorithm: Preliminary model tests," *Speech Commun.*, vol. 14, no. 1, pp. 19–48, 1994.

[63] R. H. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit," *Nature*, vol. 405, no. 6789, pp. 947–951, 2000.

[64] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[65] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. 12th {USENIX} Symp. Operating Syst. Des. Implementation*, 2016, pp. 265–283.

[66] F. Chollet *et al.*, "Keras: The Python deep learning library," *Astrophys. Source Code Library*, 2018.

[67] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[68] I.-T. Recommendation, "Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," Rec. ITU-T P. 862, 2001.

[69] S. Fu, C. Liao, and Y. Tsao, "Learning with learned loss function: Speech enhancement with quality-net to improve perceptual evaluation of speech quality," *IEEE Signal Process. Lett.*, vol. 27, pp. 26–30, 2020.

[70] T. D. Tran, Q. C. Nguyen, and D. K. Nguyen, "Speech enhancement using modified IMCRA and OMLSA methods," in *Proc. Int. Conf. Commun. Electron.*, 2010, pp. 195–200.

[71] S. Watanabe *et al.*, "Espnet: End-to-end speech processing toolkit," in *Proc. Interspeech*, 2018, pp. 2207–2211. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1456

[72] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid CTC/attention architecture for end-to-end speech recognition," *IEEE J. Sel. Top. Signal Process.*, vol. 11, no. 8, pp. 1240–1253, Dec. 2017.

Abdolreza Sabzi Shahrebabaki received the B.Sc. degree in electrical engineering from Khajeh Nasir Toosi university of technology (KNTU), Tehran, Iran, in 2009, and the M.Sc. degree in electrical engineering from the Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran, in 2012. He is currently working toward the Ph.D. degree with the signal processing Group, Department of Electronic Systems, Faculty of Information Technology and Electrical Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway. His research interests include Articulatory inversion, voice conversion, analysis by synthesis of speech, speech enhancement, signal processing, and deep learning.

Giampiero Salvi received the M.Sc. degree in electronic engineering from Università la Sapienza, Rome, Italy, and the Ph.D. degree in computer science from KTH Royal Institute of Technology, Stockholm, Sweden. He is a Professor with the Department of Electronic Systems, the Norwegian University of Science and Technology's (NTNU), Trondheim, Norway, and an Associate Professor with KTH Royal Institute of Technology, Department of Electrical Engineering and Computer Science, Stockholm, Sweden. He was a Postdoctoral Fellow with the Institute of Systems and Robotics, Lisbon, Portugal. He was a Co-Founder of the company SynFace AB, active between 2006 and 2016. His main interests include machine learning, speech technology, and cognitive systems.

Torbjørn Svendsen (Senior Member, IEEE) received the Siv.Ing (M.Sc.) and Dr.Ing. degrees from the Norwegian Institute of Technology (NTH), in 1980, and 1985, respectively. He is a Professor with the Department of Electronics and Telecommunications, Norwegian University of Science and Technology's (NTNU). Dr. Svendsen has been a Research Scientist with SINTEF before joining NTH as an Associate Professor in 1988. Since 1995, he has been a Professor of speech processing with NTNU. He has had extended research stays at AT&T Bell Laboratories, Murray Hill, NJ, USA, AT&T Labs, Florham Park, NJ, USA, Griffith University, Brisbane, Australia, Queensland University of Technology, Brisbane, Australia and MIT. His research interests include automatic speech recognition, speech synthesis, speech coding and speech analysis and modeling. He has authored or coauthored more than 90 papers in these areas. Prof. Svendsen is a Member of IEEE Signal Processing Society (SPS) and the International Speech Communication Association (ISCA). He has been a Member of the IEEE SPS Speech Processing Technical Committee.

Sabato Marco Siniscalchi (Senior Member, IEEE) is a Professor with the University of Enna, Enna, Italy, an Adjunct Professor with the Norwegian University of Science and Technology's (NTNU), and an Affiliate Faculty with the Georgia Institute of Technology. He received doctorate degree in computer engineering from the University of Palermo, Palermo, Italy, in 2006. In 2006, he was a Post Doctoral Fellow with the Ga Tech. From 2007 to 2010, he joined NTNU, Norway, as a Research Scientist. From 2010 to 2015, he was an Assistant Professor, first, and an Associate Professor, after, at the Kore University. From 2017 to 2018, he was a Senior Speech Researcher with Siri Speech Group, Apple Inc., Cupertino CA, USA. He acted as an Associate Editor of the IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH AND LANGUAGE PROCESSING, from 2015 to 2019. Dr. Siniscalchi is an Elected Member of the IEEE SLT Committee from 2019 to 2021.