A Dual-Path Speaker-Independent Acoustic-to-Articulatory Inversion Model Based on Content and Speaker Information Disentanglement

Qiang Fang
Institute of Linguistics, Chinese Academy of Social Sciences
E-mail: fangqiang@cass.org.cn Tel: +86-10-85195375

***Abstract*—Speaker-independent acoustic-to-articulatory inversion (AAI) is a technique for estimating articulatory trajectories from input acoustic speech signals for unseen speakers. This is of great importance for several theoretical and practical issues and has attracted increasing attention from researchers in recent years. In this study, we propose a dual-path speaker-independent AAI model to address this issue. In the proposed model, the input speech is decomposed into speaker and content embeddings. These two types of embeddings are independently inverted to the articulatory counter parts and then fused through a pretrained decoder of an articulatory variational autoencoder. The results indicate that the proposed approach significantly improves the performance of speaker-independent AAI in terms of the root mean square error (RMSE), while the Pearson correlation coefficient (PCC) is comparable to that of the state-of-art approach with MFCC input only.**

I. Introduction

Estimating the configuration of the vocal tract from acoustic input is known as Acoustic-to-Articulatory Inversion (AAI) [1]. Several studies have been conducted on synchronous acoustic-articulatory databases in which AAI is formulated as a regression problem. Researchers have attempted to seek better input features [2, 3] or to devise effective models to enhance AAI performance [4-12]. Most of the aforementioned studies focused on AAI in speaker-dependent scenarios. Recently, an increasing number of studies attempted to extend their concern to speaker-independent scenario by utilizing a few publicly available acoustic-articulatory databases through various approaches.

These approaches can be classified into the following categories.

i.) prototype-based approach, which assumes that the similarities between articulatory representations are highly correlated with those between acoustic features. It typically defines a set of prototype acoustic-articulatory pairs [13], and the articulatory trajectories are approximated by a weighted sum of the prototype trajectories. Here, the weights are estimated based on the similarity between the input and the prototype acoustic features. The strength of this approach is that it does not require adaptation data from new speakers. However, the inference speed is constrained because it must calculate the similarity between the input and prototypes for each frame.

ii.) feature adaptation approach, which adapts the acoustic features of a target speaker to those of the reference speaker and then utilizes the AAI model trained on the reference speaker's data to estimate the articulatory trajectories [14].

iii.) model adaptation approach, which estimates the articulatory trajectories by adapting the articulatory-to-acoustic mapping of an HMM-based speech production model [15].

iv.) model interpolation approach, which creates a model for a new speaker by weighting the reference models. For instance, Ji et al. [16] first trained several HMM-based speaker-dependent AAI models as reference models. The weights of these reference acoustic HMM models were estimated for a new speaker by maximizing the likelihood of the interpolated model on the new speaker's acoustic data. Under the assumption that the similarity between speakers in the acoustic domain is highly correlated with the similarity between speakers in the articulatory domain, these estimated weights were used to interpolate the reference articulatory generation models for the AAI of a new speaker.

v.) speaker-independent training approach, which directly trains the model on pooled acoustic-articulatory data. Various methods have been explored for AAI. Wu et al. [17] conducted AAI from acoustic feature to a transformed articulatory space that suppresses the speaker differences. Sivaraman et al. [18] further implemented a vocal tract length normalization procedure to reduce the differences in acoustic domain. Wang et al. [19] applied speech decomposing network to distill the information in acoustic feature. They further incorporated a phoneme stream to enhance AAI performance [20]. Chung et al. [21] utilized a generative adversarial network (GAN) to improve the performance of speaker-independent AAI, where a MDPD module was devised to discriminate real and generated articulatory trajectories.

Because the speaker feature is relatively stable and time-invariant, whereas the content feature varies significantly along the temporal dimension, the inversion functions for the speaker

979-8-3315-7206-8/25/$31.00 ©2025 IEEE

and content information may differ in complexity. This has not been addressed in previous studies. In this paper, we propose a dual-path AAI framework to independently deal with content and speaker inversion.

II. Proposed Method

A. Overall framework

The basic idea of the proposed approach is to decompose speech features into content and speaker embeddings using an acoustic Variational Autoencoder (ac-VAE). Here, we adopt the idea from Chou et al.'s [22] study to disentangle the content and speaker information of both acoustic and articulatory features using an ac-VAE and an ar-VAE, respectively. Subsequently, the content and speaker embeddings in articulatory domain are estimated from those in acoustic domain. Finally, articulatory trajectories are generated based on the inverted content and speaker embeddings using the decoder of an articulatory Variational Autoencoder (ar-VAE).

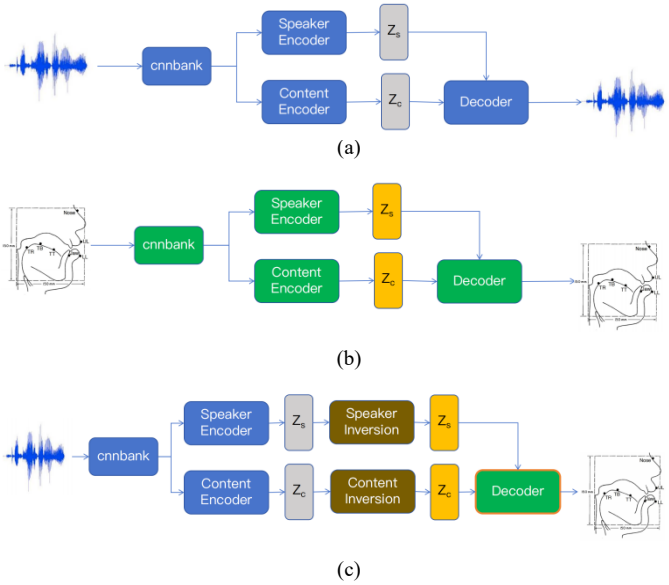

Fig. 1 Framework of the proposed dual-path AAI model. a) Structure of the ac-VAE. b) Structure of the ar-VAE. c) Structure of the AAI model.

To this end, we pretrain two VAEs, one for acoustic features, referred to as ac-VAE, and the other for articulatory features, denoted ar-VAE. Each VAE can extract the corresponding content and speaker embeddings via its encoder and synthesize either acoustic or articulatory features through its decoder. Consequently, the conventional AAI is formulated as a process of estimating articulatory content and speaker embeddings from their acoustic counterparts, followed by synthesizing articulatory trajectories by feeding the decoder of the ar-VAE with the inverted content and speaker embeddings.

Fig. 1c illustrates the framework of the proposed speaker-independent AAI model. It consists of several components: a CNN bank (cnnbank), a speaker encoder, and a content encoder in acoustic domain; a speaker inversion network, a content inversion network, and a decoder in the articulatory domain. The blue blocks in Fig. 1c represent the pretrained feature extraction and processing modules, content encoder, and speaker encoder. These modules are frozen while training the entire AAI model. The brown blocks denote the inversion networks that transform the content and speaker embeddings in acoustic domain to the corresponding embeddings in articulatory domain. The parameters of these modules are adjusted during the training phase. The green module outlined in orange is the decoder that synthesizes articulatory trajectories based on the input content and speaker embeddings. The parameters of this module are copied from the decoder of the ar-VAE and allowed to be tuned during the training phase.

B. Details of each module

The detailed structures of each component of the ac-VAE and ar-VAE are presented in this section and illustrated in Fig. 2.

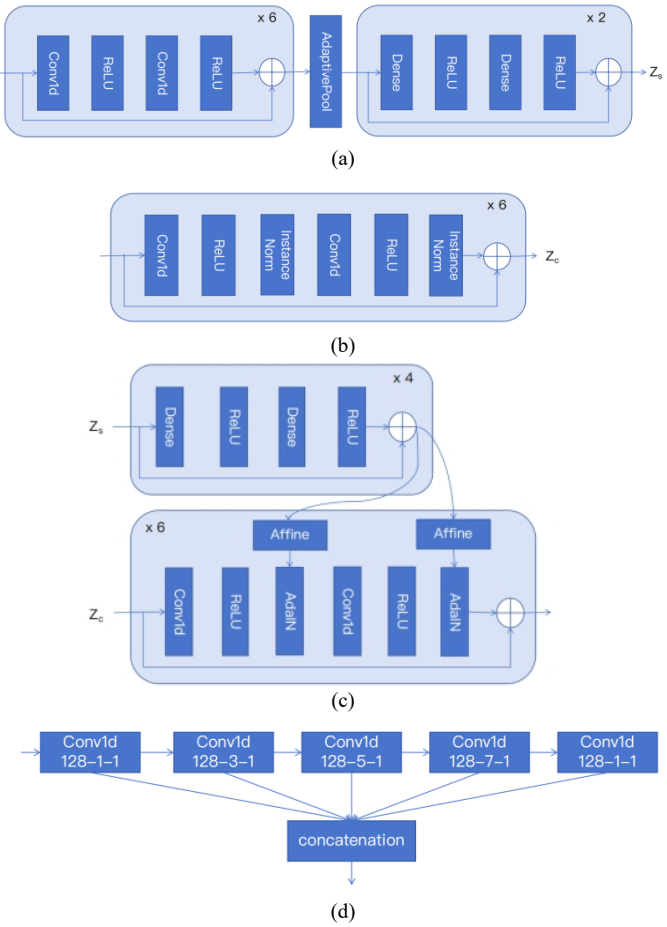

Fig. 2 The detailed structure of (a) speaker encoder, (b) content encoder, (c) decoder, and (d) cnnbank.

i.) The Speaker encoder

The speaker encoder consists of six conv-residual blocks, followed by an adaptive pooling layer and two dense-residual blocks. In each conv-residual block, there are two 1-D convolutional layers equipped with the 'ReLU' activation function, followed by a residual connection. The input and output channels, kernel size, and stride are 128, 128, 5, and 1,

respectively. In each dense-residual block, there are two dense layers equipped with the 'ReLU' activation function, followed by a residual connection. The input and output dimensions of the dense layer are both 128.

ii.) The content encoder

The content encoder consists of six conv-InstNorm-residual blocks. In each conv-InstNorm-residual block, there are two 1-D convolutional layers equipped with the 'ReLU' activation function, each of which is followed by an instance normalization layer. A residual connection is added to each conv-InstNorm-residual block. The parameters of the convolutional layers are the same as those of the convolutional layers in the speaker encoder.

Instance normalization without an affine transformation is employed as the instance normalization layer. It is supposed to eliminate speaker-specific information while preserving content-specific information [22]. The formula for instance normalization without an affine transformation is given by Eq. 1.

$$\boldsymbol{y} = \frac{\boldsymbol{x}-\boldsymbol{\mu}}{\boldsymbol{\sigma}} \tag{1}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are the mean and variance of embedding $\boldsymbol{x}$ of the utterance, respectively; and $\boldsymbol{y}$ is the normalized version of $\boldsymbol{x}$.

iii.) The decoder

The decoder consists of two types of modules: a speaker information transformation block and a speaker-content information fusion block.

The speaker information transformation module converts the speaker embedding $\boldsymbol{z}_s$ into mean and variance vectors required for the speaker-content information fusion block. It consists of four dense-residual blocks, each of which has two dense layers equipped with the 'ReLU' activation function, followed by a residual connection. The input and output dimensions of the dense layer are both 128.

The speaker-content information fusion module fuses the content embedding $\boldsymbol{z}_c$ and speaker information to synthesize articulatory trajectories. It consists of six conv-AdaIN blocks, each of which has two 1-D convolutional layers equipped with the 'ReLU' activation function, followed by an adaptive instance normalization layer (AdaIN). The parameters of the convolutional layers are the same as those of the convolutional layers in the speaker encoder. In each AdaIN layer, the features are normalized using standard instance normalization, with the scaling and translation coefficients from the speaker transformation module, as demonstrated in Eq.2.

$$\boldsymbol{y} = \boldsymbol{\gamma} \cdot \frac{\boldsymbol{x}-\boldsymbol{\mu}}{\boldsymbol{\sigma}} + \boldsymbol{\beta} \tag{2}$$

where $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are the scaling and translation coefficients estimated by the speaker encoder, respectively; and $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are the mean and variance of embedding $\boldsymbol{x}$ of the utterance, respectively.

iv.) The cnnbank

The architecture of the cnnbank is identical to that described in [12]. It consists of five 1-dimensional convolutional layers in the cnnbank, with kernel sizes of 1, 3, 5, 7, and 9, respectively. The strides of all the kernels are 1, and no pooling layers are included in the cnnbank.

III. DATABASE

In this study, we utilize the Haskins Production Rate Comparison (HPRC) database [23], which comprises speech and EMA signals of approximately 7.9 hours. The recordings were based on 720 phonetically balanced Harvard sentences (IEEE 1969) spoken by eight American speakers (4 male and 4 female) at two production rates (normal and fast). Errors and mispronunciations were monitored and corrected by immediate repetition. The sampling rates were 44.1kHz and 100Hz for the speech and EMA signals, respectively.

Eight coils were glued to the speech organs of interest while recording the EMA data. These included the Tongue Rear (TR), Tongue Blade (TB), Tongue Tip (TT), Upper Lip (UL), Lower Lip (LL), Mouth Left (ML), lower medial incisor (Jaw), and Jaw Left (JawL). Each coil provided a vector of six real-valued measurements: the posterior-anterior (x), left-right (y), and superior-inferior (z) coordinates and three rotation angles of the coil. The kinematic trajectories were filtered using a forward-backward Butterworth low-pass filter with a cutoff frequency of 20Hz, corrected for head movement, and aligned to the occlusal plane. In the AAI experiments, only the x and z coordinates of the articulatory trajectories are of interest.

IV. EXPERIMENTS

A. Materials

To conduct the AAI experiments, the speech signals are converted into 40-dimensional MFCCs using the TorchAudio toolkit. For this purpose, the speech signals are chopped into sequences of segments with a Hanning window of a length of 30ms. The stride between two consecutive windows is 10ms. An FFT with a length of 1024 is used to convert the wave signals into their frequency-domain representations. The frequency components ranging from 50Hz to 8000Hz are analyzed using 128 Mel-filters, adhering to the 'HTK-Mel' specification. Finally, a 40-dimensional MFCC vector is generated for each frame.

For the articulatory representation, we adopt the x and z coordinates of coil TR, TB, TT, UL, LL, and JAW. The trajectories of these coils are mean-normalized, and the origin is shifted to the mean position of the jaw. Portions corresponding to silence at the beginning and end of each utterance are eliminated.

To train speaker-independent AAI models, we use an 8-fold cross-validation paradigm. In each fold, data from 7 subjects are used for training, and data from the remaining subject are used for testing.

B. Training losses

i.) Loss for VAE

Let $\boldsymbol{x}$ represent the acoustic or articulatory feature vector, $\boldsymbol{E_s}$, $\boldsymbol{E_c}$, and $\boldsymbol{D}$ denote the speaker encoder, content encoder, and decoder, respectively. $\boldsymbol{E_s}$ and $\boldsymbol{E_c}$ are trained to generate speaker embedding $\boldsymbol{z_s}$ and content embedding $\boldsymbol{z_c}$, respectively. We assume that $p(\boldsymbol{z_c}|\boldsymbol{x})$ follows a conditionally independent Gaussian distribution $N(\boldsymbol{\mu}, \Lambda)$, where Λ is a diagonal matrix. Therefore, the training loss can be formulated as Eq.3-5.

$$L = L_{rec} + L_{KL} \tag{3}$$

$$L_{rec} = \frac{1}{N}\sum_{i=1}^{N}\left\|D(\boldsymbol{z}_c^i, \boldsymbol{z}_s^i) - \boldsymbol{x}^i\right\| \tag{4}$$

$$L_{KL} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j}^{d}(\Lambda_{ij} + \mu_{ij}^2 - 1 - log\Lambda_{ij}) \tag{5}$$

ii.) Loss for AAI

As illustrated in Fig. 1, there are three modules whose parameters must be adjusted during the training phase. To optimize these parameters, the loss function for the proposed AAI model comprises three components, as formulated in Eq.6-9.

$$L = L_c + L_s + L_{AAI} \tag{6}$$

$$L_c = \frac{1}{N}\sum_{i=1}^{N}\left\|\boldsymbol{z}_c^i - \hat{\boldsymbol{z}}_c^i\right\| \tag{7}$$

$$L_s = \frac{1}{M}\sum_{i=1}^{M}\left\|\boldsymbol{z}_s^i - \hat{\boldsymbol{z}}_s^i\right\| \tag{8}$$

$$L_{AAI} = \frac{1}{N}\sum_{i=1}^{N}\left\|\boldsymbol{x}^i - \hat{\boldsymbol{x}}^i\right\| \tag{9}$$

where N and M denote the number of feature frames and utterances, respectively; $\boldsymbol{z}_c^i$ and $\hat{\boldsymbol{z}}_c^i$ are the ground truth and estimated content embeddings, respectively; $\boldsymbol{z}_s^i$ and $\hat{\boldsymbol{z}}_s^i$ represent the ground truth and estimated speaker embeddings, respectively; $\boldsymbol{x}^i$ and $\hat{\boldsymbol{x}}^i$ stand for the ground truth and estimated articulatory vectors, respectively.

C. Detailed setting

To train the ac-VAE and ar-VAE, we utilize the Adam optimizer with a learning rate of 1.0e-4, β_1 of 0.9, β_2 of 0.999, and a batch size of 128. To mitigate the risk of overfitting, we implement a dropout rate of 0.3 for of each layer, along with a weight decay of 1.0e-4.

The 'ReduceLROnPlateau' scheduler is employed to adjust the learning rate, and the model is trained for 3000 iterations using an early stopping strategy.

To train the AAI model, the same optimizer and parameter settings are used, except that the learning rate is set to 1.0e-3. The AAI model is trained for 300 iterations using an early stopping strategy.

D. Comparison with SOTA

Compared with the SAFN model [19], the proposed dual-path AAI model discards the Auxiliary Feature Network designed to estimate auxiliary lip features and creates separate inversion pathways for content and speaker embeddings. Therefore, it is more appropriate to compare the results of the proposed dual-path model with those of the SAFN model, although the performance of the MCAD [21] and SPN models [20] is better than that of the SAFN model. Consequently, the SAFN model [19] is used as the baseline for evaluating the proposed AAI model.

In addition, we compare the results with those of the SPN model, which uses phoneme information streams as extra input features, and those of the MCAD model, which uses either MFCC or WavLM as input features.

V. Results

Table 1 presents the average RMSEs and PCCs across the 8-fold cross-validation experiments of recent studies and our study. Here, MCAD[1] [21] refers to the MCAD model that utilizes WavLM features, while MCAD[2] [21] is the MCAD model that employs MFCC features; SPN [20] denotes the model with the inputs of both MFCC and phoneme information; SANF [19] denotes the model depicted in Wang et al.'s paper; DP-VAE_FR denotes the model in which the decoder is copied from the pretrained ar-VAE and frozen during the training phase of the AAI model; DP-VAE denotes the model in which the initial parameters of the decoder are copied from the pretrained ar-VAE, while fine-tuned during the training phase of the AAI model.

Table 1 Results of the average RMSEs and PCCs across the 8-fold cross-validation experiments

	RMSE	PCC
MCAD[1] [21]	2.49	0.85
MCAD[2] [21]	3.36	0.64
SPN [20]	2.54	0.81
SAFN [19]	2.72	0.75
DP-VAE_FR	2.63	0.70
DP-VAE	2.57	0.72

A. Mean RMSE

As shown in Table 1, the mean RMSEs of DP-VAE_FR and DP-VAE are 2.63mm and 2.57mm, respectively. Both are less than those of the MACD[2] and SAFN models. This indicates that the proposed dual-path inversion models outperform the one-path models in terms of RMSE. Moreover, the DP-VAE model achieves the lowest RMSE among the models with only MFCC input. In addition, the performance of the DP-VAE model is better than that of the DP-VAE_FR model. This suggests that the fine-tuned decoder compensates for the

mismatch between the ground truth and estimated articulatory content and speaker embeddings to a certain extent.

The mean RMSEs of MCAD[1] and MCAD[2] are 2.47mm and 3.36mm, respectively, which are significantly different. As is known, MCAD[1] and MCAD[2] utilize the inversion model with the same structure, and the only difference between them is the input feature. This suggests that the performance in terms of RMSE can be further improved if the MCAD model is trained with appropriate features. The RMSE of MCAD[2] is greater than those of DP-VAE_FR and DP-VAE. Nevertheless, the RMSE of MCAD[1] is less than those of DP-VAE_FR and DP-VAE. Based on the above analysis, we argue that the MCAD model is more appropriate for WavLM feature than MFCC feature for developing a speaker-independent AAI. Consequently, different models should be designed for different input features.

The RMSE of SPN is 2.54mm, which is slightly less than that of DP-VAE (2.57mm). Compared with SAFN, SPN and DP-VAE improve AAI performance in different ways. SPN improves performance by introducing an extra information stream, whereas DP-VAE enhances performance by inverting the speaker and content embeddings independently.

B. Mean PCC

As shown in Table 1, the mean PCC of DP-VAE_FR and DP-VAE are 0.70 and 0.72, respectively. Both are higher than that of MCAD[2] but lower than those of MCAD[1], SPN, and SAFN. The PCC of DP-VAE is close to that of SAFN (0.72 vs. 0.75), both of which have only MFCC input.

C. Ablation study on embeddings

Table 2 Average RMSEs and PCCs of the ablation studies on latent embeddings.

	RMSE	PCC
DP-VAE	2.57	0.72
DP-VAE_cnt	2.27	0.78
DP-VAE_spk	2.26	0.71
DP-VAE_cnt_spk	1.95	0.77

These results indicate that the proposed dual-path model outperforms the state-of-the-art model with only MFCC input (SAFN) in terms of RMSE but is inferior to SAFN in terms of PCC. To clarify the cause, we perform several ablation experiments in which the inputs to the decoder are replaced with the corresponding embeddings extracted from articulatory data. DP-VAE_cnt denotes the DP-VAE model in which content embeddings are replaced with those extracted from articulatory data. DP-VAE_spk denotes the DP-VAE model in which speaker embeddings are replaced with those extracted from articulatory data. DP-VAE_cnt_spk denotes the DP-VAE model in which both content and speaker embeddings are replaced with those extracted from articulatory data.

The results of the average RMSEs and PCCs of the ablation studies on latent embeddings are shown in Table 2. The RMSE of DP-VAE_cnt is 2.27mm, which is 0.3mm lower than that of DP-VAE. The PCC of DP-VAE_cn is 0.78, which is 0.06 higher than that of DP-VAE. This suggests that improving the quality of content embedding in articulatory domain can enhance AAI performance in terms of RMSE and PCC.

The RMSE of DP-VAE_spk is 2.26mm, which is 0.31mm lower than that of DP-VAE. The PCC of DP-VAE_spk is 0.7, which is slightly lower than that of DP-VAE. This suggests that improving the quality of speaker embedding in articulatory domain can improve AAI performance in terms of RMSE but not PCC.

The RMSE of DP-VAE_cn_spk is 1.95mm, which is 0.32mm lower than that of DP-VAE_cnt. This indicates that the performance of DP-VAE_cnt in terms of RMSE can be further improved by using speaker embeddings of higher quality. However, the performance of DP-VAE_cnt in terms of PCC is slightly degraded, from 0.78 to 0.77. This supports the conjecture that improving the quality of content embedding enhances the AAI performance in terms of both RMSE and PCC, whereas improving the quality of speaker embedding merely enhances the performance of RMSE.

D. Ablation studies on model structure

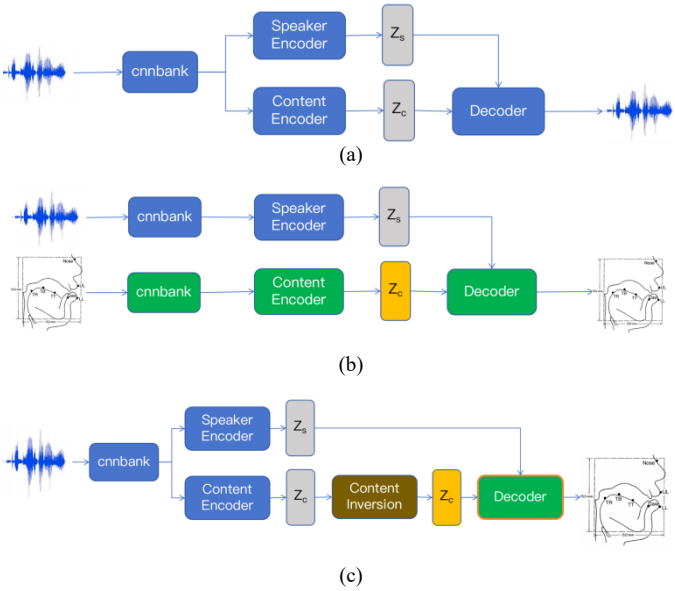

Fig. 3 Framework of the proposed AAI approach without the speaker inversion network. a) Structure of the ac-VAE. b) Structure of the ar-VAE. c) Structure of the DP-VAE_acspkEmb model.

Based on the above hypothesis, we propose a new model structure to deal with speaker embedding (shown in Fig. 3). This is referred to as DP-VAE_acspkEmb. The main difference between the model in Fig. 3 and the model in Fig. 1 is that the speaker embedding from acoustic domain, rather than from articulatory domain, is utilized when training the ar-VAE. Hence, the speaker inversion network is no longer necessary in the AAI model. In this manner, the mismatch between the estimate and ground truth can be suppressed, and the quality of the speaker embedding can be improved.

Table 3. Average RMSEs and PCCs of the ablation studies on latent embeddings.

	RMSE	PCC
SPN [20]	2.54	0.81
SAFN [19]	2.72	0.75
DP-VAE	2.57	0.72
DP-VAE_acspkEmb	2.45	0.73

Table 3 presents the average RMSEs and PCCs for the SPN, SAFN, DA-VAE, and DP-VAE_acspkEmb models. The RMSE and PCC are 2.45mm and 0.73 for DP-VAE_acspkEmb, respectively. As expected, the RMSE decreases by 0.12mm compared to that of DP-VAE, and the PCC is slightly higher than that of the DP-VAE (0.73 vs. 0.72). Furthermore, the RMSE of DP-VAE_acspkEmb is lower than that of SPN, which involves phoneme streams as extra information. There is an approximate 10% relative improvement in terms of RMSE compared with SAFN.

VI. Conclusion

In this study, we propose a dual-path speaker-independent AAI model that independently inverts the content and speaker embeddings from acoustic domain to articulatory domain. This process is followed by synthesizing articulatory trajectories using a pretrained decoder of an ar-VAE. The RMSE obtained from the proposed models (DP-VAE) is lower than that of the SOTA model with only MFCC input (SAFN). However, the PCC of the proposed model is lower than that of the SOTA model with only MFCC input (SAFN). The results of the ablation studies suggest that improving the quality of speaker embedding can enhance the performance in terms of RMSE, and improving the quality of content embeddings can enhance the performance in terms of both RMSE and PCC. The results of the DP-VAE_acspkEmb model support the former part of our argument. However, further experiments are necessary to test this hypothesis.

VII. Acknowledgement

This work is supported by the National Natural Science Foundation of China (No.61977049) and the Key Laboratory of Linguistics, Chinese Academy of Social Sciences (Project No.2024SYZH001).

References

[1] 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. Acoust. Soc. Am.*, vol. 63, no. 5, pp. 1535–1555, 1978.

[2] C. Qin and M. A. Carreira-Perpinan, "A Comparison of. Acoustic Features for Articulatory Inversion," presented at the Interspeech, 2007, pp. 2469–2472.

[3] S. Udupa, C. Siddarth, and P. K. Ghosh, "Improved Acoustic-to-Articulatory Inversion Using Representations from Pretrained Self-Supervised Learning Models," presented at the ICASSP, 2023.

[4] K. Richmond, S. King, and P. Taylor, "Modelling the. uncertainty in recovering articulation from acoustics," *Computer Speech & Language*, vol. 17, no. 2–3, pp. 153–172, 2003.

[5] S. Hiroya, and M. Honda, "Estimation of Articulatory. Movements from Speech Acoustics Using an HMM-Based Speech Production Model," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 2, pp. 175–185, 2004.

[6] C. Qin and Carreira-Perpiñán M. A, "An empirical. investigation of the nonuniqueness in the acoustic-to-articulatory mapping," presented at the Interspeech, 2007.

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

[8] T. Toda, A. W. Black, and K. Tokuda, "Statistical mapping. between articulatory movements and acoustic spectrum using a Gaussian mixture model," *Speech Communication*, vol. 50, pp. 215–227, 2008.

[9] B. Uria, I. Murray, S. Renals, and K. Richmond, "Deep. Architectures for Articulatory Inversion," presented at the Interspeech, 2012.

[10] Z. Y. Wu, K. Zhao, X. X. Wu, X. Y. Lan, and H. Meng, "Acoustic to articulatory mapping with deep neural network," *Multimed Tools Appl*, vol. 74, no. 22, pp. 9889–9907, 2015, doi: 10.1007/s11042-014-2183-z.

[11] P. Zhu, L. Xie, and Y. Chen, "Articulatory Movement. Prediction Using Deep Bidirectional Long Short-Term Memory Based Recurrent Neural Networks and Word/Phone Embeddings," presented at the Interspeech, 2015, pp. 2192–2196.

[12] A. S. Shahrebabaki, S. M. Siniscalchi, G. Salvi1, and T. Svendsen, "Sequence-to-sequence articulatory inversion through time convolution of sub-band frequency signals," presented at the Interspeech, 2020, pp. 2882–2886.

[13] A. Afshan and P. K. Ghosh, "Improved subject-independent. acoustic-to-articulatory inversion," *Speech Communication*, vol. 66, pp. 1–16, 2015.

[14] T. Hueber, G. Bailly, P. Badin, and F. Elisei, "Speaker. adaptation of an acoustic-articulatory inversion model using cascaded Gaussian mixture regressions," presented at the Interspeech, 2013, pp. 2753–2757.

[15] S. Hiroya and M. Honda, "Speaker adaptation method for. acoustic-to articulatory inversion using HMM based speech production model," *IEICE Trans. Inf. & Syst.*, vol. 87, pp. 1071–1078, 2004.

[16] A. Ji, M. T. Johnson, and J. J. Berry, "Parallel Reference. Speaker Weighting for Kinematic-Independent Acoustic-to-Articulatory Inversion," *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, vol. 24, no. 10, pp. 1865–1875, 2016.

[17] P. Wu *et al.*, "Speaker-Independent Acoustic-to-Articulatory. Speech Inversion," presented at the ICASSP, 2023, pp. 5060–5064.

[18] G. Sivaraman, V. Mitra, H. Nam, M. Tiede, and C. Espy-Wilson, "Unsupervised speaker adaptation for speaker independent acoustic to articulatory speech inversion," *J. Acoust. Soc. Am.*, vol. 146, pp. 316–329, 2019.

[19] J. R. Wang, J. Liu, L. Zhao, S. Wang, R. Yu, and L. Liu, "Acoustic-to-articulatory inversion based on speech decomposition and auxiliary feature," presented at the ICASSP, 2022.

[20] J. R. Wang *et al.*, "Two-Stream Joint-Training for Speaker. Independent Acoustic-to-Articulatory Inversion," presented at the ICASSP, 2023.

[21] W. Chung and H. Kang, "Speaker-Independent Acoustic-to-Articulatory Inversion through Multi-Channel Attention Discriminator," presented at the Interspeech, 2024, pp. 1540–1544.

[22] J. Chou, C. Yeh, and H. Y. Lee, "One-shot Voice Conversion. by Separating Speaker and Content Representations with Instance Normalization," presented at the Interspeech, 2019, pp. 664–668.

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