



Enhancing Acoustic-to-Articulatory Inversion with Multi-Target Pretraining for Low-Resource Settings

Jesuraj Bandekar, Prasanta Kumar Ghosh

Department of Electrical Engineering, Indian Institute of Science, Bengaluru, India

jesurajbandekar.661@gmail.com, prasantag@gmail.com

Abstract

Acoustic-to-Articulatory Inversion (AAI) estimates vocal tract articulator movements from speech, benefiting tasks like ASR, speech synthesis, and speaker verification. While deep learning-based methods (CNNs, RNNs, Transformers) have advanced AAI, recent studies show that Self-Supervised Learning (SSL) features further enhance performance, particularly in low-resource settings. However, SSL feature extractors introduce inference latency and computational overhead. To address this, we propose a novel pretraining method leveraging three target representations—Phoneme Labels, Articulatory Feature Labels, and Critical-articulator Labels—eliminating the need for an SSL extractor during inference. We evaluate our approach against both baseline and SSL-based models across various data conditions. Results demonstrate that our method consistently improves AAI performance, particularly in low-resource scenarios, while significantly reducing inference costs without sacrificing accuracy.

Index Terms: Self-Supervised Learning, Acoustic to Articulatory Inversion, Multi-Target Pretraining

1. Introduction

The position and movement of vocal tract articulators play a crucial role in speech production. These rapid articulatory motions are closely linked to vocal tract morphology, pronunciation, and the speaker's language. Recent research has demonstrated the usefulness of Acoustic-to-Articulatory Inversion (AAI) in various speech-related applications, including Automatic Speech Recognition (ASR) [1, 2], Speech Synthesis [3], Speaker Verification [4], and pronunciation training [5].

Various techniques have been explored for Acoustic-to-Articulatory Inversion (AAI), including quantizing the articulatory space [6], Gaussian Mixture Models (GMMs) [7, 8], and Hidden Markov Models (HMMs) [9]. More recently, deep learning-based approaches have demonstrated significant improvements in AAI performance. Convolutional Neural Networks (CNNs) [10, 11] have proven effective, while Recurrent Neural Networks (RNNs) [12, 13] have shown strong results for sequence-to-sequence AAI tasks. Furthermore, the adoption of transformer-based models [14] has led to substantial performance gains, further advancing AAI research.

Various training strategies have been explored to enhance the performance of AAI models. [15] introduces a two-stream joint training approach for speaker-independent AAI, while [16] investigates multi-task learning to improve model generalization. Feature decomposition techniques have been employed in [17] to enhance speaker-independent AAI performance. Additionally, [18] explores classification-based loss on a quantized articulatory space to improve articulatory trajectory prediction.

These advancements demonstrate the effectiveness of different training methodologies in refining AAI models.

Another significant challenge in improving AAI performance is the limited availability of high-quality acoustic-articulatory data, especially for unseen speakers. Unlike ASR and speech synthesis, AAI datasets are scarce, affecting model generalization. Researchers have explored transfer learning techniques to develop low-resource AAI models to address this issue. [19] introduces a transfer learning-based approach using a generic AAI (GM AAI) model with a Long Short-Term Memory (LSTM) network to enhance performance in data-constrained scenarios. Similarly, [20] leverages phonetic information within a transfer learning framework to further improve AAI accuracy. These studies show that transfer learning helps mitigate data constraints and improves model robustness in low-resource settings.

In recent years, research has increasingly explored the use of features from Self-Supervised Learning (SSL) models as inputs for AAI instead of traditional Mel-Frequency Cepstral Coefficients (MFCC). [21] investigates various SSL features for AAI, while [22] applies SSL features to predict articulatory trajectories in dysarthric patients. [23] examines SSL-based inputs for AAI in a cross-lingual setting, and [24] analyzes outputs from different SSL model layers for patient-aware AAI. Additionally, [25] proposes leveraging a pre-trained model to learn linguistic invariants before integrating it with a standard inversion model. Collectively, these studies highlight that SSL-derived features provide significant improvements in AAI performance across various conditions, particularly in scenarios with unseen speakers.

While pre-trained SSL models have demonstrated significant improvements in AAI performance, especially in low-resource settings, their use as feature extractors comes with considerable drawbacks. These models are typically much larger than AAI models, leading to increased inference time and higher computational costs. Deploying such models for real-time applications can be impractical, particularly in scenarios with limited processing power or latency constraints. To address these challenges, we propose an alternative pretraining approach that directly optimizes the AAI model itself using multiple target representations. By eliminating the need for an external SSL feature extractor, our method reduces computational overhead while still leveraging the benefits of pretraining. We systematically compare our approach with both baseline models and SSL-based feature extractor models across different training data sizes. Additionally, we analyze the model's performance in low-resource scenarios, exploring whether this pretraining strategy can match or even surpass the performance of SSL-enhanced methods while maintaining efficiency during inference.

Hence the contributions of our work are as follows -

1. We propose a novel pretraining method for AAI models using three target representations—Phoneme Labels, Articulatory Feature Labels, and Critical-articulator Labels—to enhance AAI performance.
2. We compare the performance of the proposed methodology with the baseline model
3. We also compare the performance of the proposed methodology with the SSL feature extractor-based model
4. To analyse the efficacy for low-resource settings we use different amounts of training data and then compare the change in performance for all the models.
5. The proposed model achieves faster inference as it does not rely on an extra SSL model for feature extraction,

2. Dataset

We use the SpireEMA dataset [26], which consists of 460 sentences from the MOCHA-TIMIT dataset [27]. These utterances were spoken by 38 speakers aged between 20 and 28. All participants are fluent English speakers with no reported speech impairments.

For articulatory data, we use electromagnetic articulography (EMA) recordings that capture time-varying articulatory movements. The data was collected using the Electromagnetic Articulograph AG501 [28], with six sensors placed on different articulators: the Upper Lip (UL), Lower Lip (LL), Jaw, Tongue Tip (TT), Tongue Body (TB), and Tongue Dorsum (TD). We consider articulatory trajectories in the midsagittal plane, which provide both horizontal (x-axis) and vertical (y-axis) movement information. This results in a total of 12 articulatory trajectories across the six articulators.

Out of the 38 speakers, we use 32 for training, development, and evaluation. The development and test sets contain unseen utterances to ensure robust evaluation. Additionally, we evaluate performance on a separate test set comprising six unseen speakers, where both the utterances and speakers are entirely new to the model. The SpireEMA dataset is open-source and publicly available on Hugging Face ¹

For pretraining, we use the LibriSpeech ASR dataset [29], specifically the **train-100** subset, which contains 100 hours of speech data. To obtain frame-level phoneme alignments, we process the data using the Kaldi toolkit [30].

3. Methodology

In this work, we investigate different pretraining targets for AAI and subsequently fine-tune the model using varying amounts of EMA data. The trained models are then evaluated on the test set to assess their effectiveness. For pretraining, we explore three distinct target representations:

- **Phoneme Labels:** We use frame-aligned phoneme labels as ground truth for classification. The model is trained using Cross-Entropy loss to predict the phoneme label corresponding to each frame.
- **Articulatory Feature Labels:** Inspired by [31], we use articulatory feature classes as ground truth labels. Specifically, we predict four articulatory attributes—*place*, *manner*, *height*, and *backness*—by employing four separate linear layers at the model’s output. The ground truth labels for each phoneme

¹https://huggingface.co/datasets/SpireLab/SPIRE_EMA_CORPUS

Table 1: *Articulatory feature labels for pretraining*

Feature	Possible Values
Place	bilabial, labiodental, dental, alveolar, postalveolar, palatal, velar, glottal, vowel, silence
Manner	stop, affricate, fricative, nasal, approximant, lateral, vowel, silence
Height	close, nearclose, closemid, mid, openmid, nearopen, open, consonant, silence
Backness	front, central, back, consonant, silence

are derived from the International Phonetic Alphabet (IPA) chart², with details provided in Table 1. As this is a classification task, we use Cross-Entropy loss for optimization.

- **Critical-articulator Labels:** Following [32], we identify critical articulators for 13 specific phonemes. Using this information, we construct a 12-dimensional binary vector, termed the *critical-articulator label*, for each input frame. Each element in this vector is set to 1 for articulators that are critical for the given phoneme and 0 otherwise. For phonemes outside the set of 13, all values in the vector are set to 0. The model is trained to predict this 12-dimensional vector using a sigmoid activation function, and we optimize the predictions using Binary Cross-Entropy loss. Additionally, for frames corresponding to phonemes outside the pre-defined set, we mask the loss to avoid penalizing the model for missing articulatory information.

After pretraining, we removed the final output layer of the model, which was used to predict the pretraining targets, and replaced it with a 12-dimensional linear layer for predicting EMA trajectories in the AAI task. We adopt a non-autoregressive transformer-based neural network architecture [33], similar to the model used in [14].

4. Experimental Setup

4.0.1. Pretraining Configurations

We explore different pretraining configurations by combining various target representations. The configurations are as follows:

- **ACP-T:** Pretraining is conducted using all three target types—*Articulatory Feature Labels*, *Critical-articulator Labels*, and *Phoneme Labels*.
- **AC-T:** This configuration utilizes *Articulatory Feature Labels* and *Critical-articulator Labels* for pretraining.
- **AP-T:** Pretraining is performed with *Articulatory Feature Labels* and *Phoneme Labels*.
- **CP-T:** This setup employs *Critical-articulator Labels* and *Phoneme Labels* as pretraining targets.
- **P-T:** Only *Phoneme Labels* are used for pretraining.
- **C-T:** Pretraining is conducted exclusively with *Critical-articulator Labels*.
- **A-T:** Only *Articulatory Feature Labels* are utilized for pretraining.

4.0.2. Input Configurations

We experiment with two different input feature types:

- **MFCC:** We use 13-dimensional MFCC as input to the model.
- **TERA:** We use 768-dimensional features extracted from the

²<https://www.internationalphoneticassociation.org/content/full-ipa-chart>

Table 2: *CC and RMSE for different configurations with MFCC inputs for seen speakers test set*

EMA data used for training (%)		6.25	12.5	25	50	75	100
Baseline	CC (Std)	0.7348 (0.1652)	0.7857 (0.1453)	0.8254 (0.1253)	0.8563 (0.1125)	0.8723 (0.1061)	0.8778 (0.1022)
	RMSE (Std)	1.4394 (0.3695)	1.3180 (0.3535)	1.1964 (0.3407)	1.0939 (0.3244)	1.0440 (0.3210)	1.0190 (0.3127)
ACP-T	CC (Std)	0.7811 (0.1518)	0.8112 (0.1370)	0.8379 (0.1251)	0.8616 (0.1127)	0.8731 (0.1082)	0.8797 (0.1026)
	RMSE (Std)	1.3535 (0.3580)	1.2620 (0.3515)	1.1694 (0.3371)	1.0850 (0.3300)	1.0850 (0.3300)	1.0135 (0.3183)
AC-T	CC (Std)	0.7779 (0.1552)	0.8121 (0.1358)	0.8380 (0.1231)	0.8592 (0.1150)	0.8741 (0.1059)	0.8810 (0.1010)
	RMSE (Std)	1.3538 (0.3585)	1.2602 (0.3468)	1.1719 (0.3388)	1.0920 (0.3280)	1.0429 (0.3260)	1.0125 (0.3188)
AP-T	CC (Std)	0.7782 (0.1522)	0.8095 (0.1403)	0.8389 (0.1221)	0.8620 (0.1118)	0.8728 (0.1073)	0.8804 (0.1028)
	RMSE (Std)	1.3550 (0.3529)	1.2749 (0.3529)	1.1770 (0.3389)	1.0809 (0.3269)	1.0490 (0.3210)	1.0084 (0.3165)
CP-T	CC (Std)	0.7774 (0.1548)	0.8101 (0.1384)	0.8380 (0.1256)	0.8616 (0.1116)	0.8742 (0.1052)	0.8804 (0.1037)
	RMSE (Std)	1.3619 (0.3490)	1.2530 (0.3449)	1.1670 (0.3389)	1.0900 (0.3339)	1.0400 (0.3240)	1.0131 (0.3228)
P-T	CC (Std)	0.7806 (0.152)	0.8097 (0.1361)	0.8372 (0.1245)	0.8624 (0.1121)	0.8733 (0.1074)	0.8794 (0.1026)
	RMSE (Std)	1.3539 (0.3580)	1.2602 (0.3472)	1.1710 (0.3400)	1.0880 (0.3359)	1.0430 (0.3260)	1.0163 (0.3210)
C-T	CC (Std)	0.7718 (0.1525)	0.8027 (0.1403)	0.8308 (0.1275)	0.8564 (0.1155)	0.8723 (0.1058)	0.8759 (0.1052)
	RMSE (Std)	1.3639 (0.3600)	1.2810 (0.3470)	1.1959 (0.3440)	1.1060 (0.3390)	1.0420 (0.3230)	1.0324 (0.3280)
A-T	CC (Std)	0.7773 (0.1547)	0.8101 (0.1377)	0.8345 (0.1259)	0.8614 (0.1137)	0.8728 (0.1088)	0.8787 (0.1037)
	RMSE (Std)	1.355 (0.3560)	1.2699 (0.3479)	1.1840 (0.3459)	1.0870 (0.3339)	1.0360 (0.3270)	1.0169 (0.3213)

Table 3: *CC and RMSE for different configurations with MFCC inputs for unseen speakers test set*

EMA data used for training (%)		6.25	12.5	25	50	75	100
Baseline	CC (Std)	0.6687 (0.2020)	0.6991 (0.1996)	0.7265 (0.1876)	0.7488 (0.1783)	0.7488 (0.1783)	0.7563 (0.1822)
	RMSE (Std)	1.5992 (0.3842)	1.5540 (0.4050)	1.4854 (0.3980)	1.4357 (0.3962)	1.4357 (0.3962)	1.4143 (0.4040)
ACP-T	CC (Std)	0.7259 (0.1872)	0.7399 (0.1814)	0.7469 (0.1875)	0.7616 (0.1796)	0.7653 (0.1803)	0.7689 (0.1823)
	RMSE (Std)	1.5241 (0.4001)	1.4779 (0.4041)	1.4514 (0.4043)	1.4160 (0.4060)	1.4160 (0.4059)	1.3848 (0.4154)
AC-T	CC (Std)	0.7318 (0.1806)	0.7271 (0.1890)	0.7448 (0.1889)	0.7611 (0.1769)	0.7652 (0.1791)	0.7667 (0.1808)
	RMSE (Std)	1.4690 (0.3801)	1.5075 (0.4056)	1.4436 (0.4156)	1.4040 (0.3889)	1.4029 (0.4090)	1.4054 (0.4194)
AP-T	CC (Std)	0.7271 (0.1833)	0.7313 (0.1903)	0.7490 (0.1824)	0.7633 (0.1798)	0.7652 (0.1791)	0.7683 (0.1776)
	RMSE (Std)	1.5010 (0.3889)	1.5019 (0.4029)	1.4589 (0.4129)	1.3899 (0.4029)	1.4029 (0.4090)	1.3986 (0.4133)
CP-T	CC (Std)	0.7269 (0.1871)	0.7412 (0.179)	0.7480 (0.191)	0.7588 (0.1839)	0.7641 (0.1793)	0.7594 (0.1861)
	RMSE (Std)	1.500 (0.3829)	1.4501 (0.3930)	1.4459 (0.4199)	1.4140 (0.4140)	1.4170 (0.4130)	1.4131 (0.4194)
P-T	CC (Std)	0.7229 (0.1897)	0.7358 (0.1800)	0.7432 (0.1865)	0.7567 (0.1809)	0.7686 (0.1804)	0.7687 (0.1764)
	RMSE (Std)	1.5180 (0.4079)	1.4850 (0.3957)	1.4579 (0.4059)	1.4450 (0.4120)	1.3869 (0.4150)	1.3949 (0.4086)
C-T	CC (Std)	0.7099 (0.1863)	0.7349 (0.1829)	0.7409 (0.1824)	0.7406 (0.1922)	0.7644 (0.1751)	0.7621 (0.1791)
	RMSE (Std)	1.5260 (0.3880)	1.4739 (0.3899)	1.4709 (0.4120)	1.4780 (0.4300)	1.4050 (0.4040)	1.4225 (0.4212)
A-T	CC (Std)	0.7324 (0.1819)	0.7367 (0.1816)	0.7510 (0.1839)	0.7547 (0.1809)	0.7626 (0.1814)	0.7652 (0.1769)
	RMSE (Std)	1.4900 (0.3910)	1.4939 (0.4110)	1.4459 (0.4079)	1.4349 (0.4110)	1.4080 (0.4150)	1.4021 (0.4041)

TERA model [34]. Previous studies [21, 18], which utilize a subset of the dataset used in this work, have demonstrated that TERA features yield the best results for AAI tasks. We extract TERA features using the s3prl toolkit³.

4.0.3. Training Data Configurations

To assess the effectiveness of the proposed methodology in low-resource settings, we conduct experiments using different fractions of the SpireEMA training dataset: **6.25%, 12.5%, 25%, 50%, 75%, and 100%**. We ensure that within each subset, the same utterances are included across all speakers to maintain consistency.

For pretraining, we use the full 100-hour LibriSpeech dataset along with a specified fraction of the SpireEMA dataset. During fine-tuning for the AAI task, only the selected percentage of the SpireEMA dataset is utilized.

4.0.4. Model Configuration and Evaluation

We use two transformer encoders with four layers each, a single attention head per layer, and input, output, and feedforward dimensions of 256, with an attention dimension of 32. The model has 7.6M parameters.

The training uses the Adam optimizer [35] with a 0.0001 learning rate and early stopping based on validation loss. The implementation, based on PyTorch, is open-sourced⁴.

³<https://github.com/s3prl/s3prl>

⁴<https://github.com/coding-phoenix-12/Multi-Target-Pretraining-AAI>

The evaluation uses Correlation Coefficient (CC) and Root Mean Squared Error (RMSE), averaged across utterances and subjects. The baseline model is trained with the same AAI data but without pretraining.

4.1. Results and Discussions

4.1.1. Impact of Pretraining on Seen Speakers

Table 2 shows that pretraining consistently improves performance across all data fractions, with the largest gains in low-resource conditions (6.25% and 12.5%). Models that underwent pretraining achieved better performance, demonstrating the benefits of leveraging auxiliary targets.

- 100% training data:** The AC-T configuration achieves the highest CC (0.8810), while AP-T achieves the lowest RMSE (1.0084). This indicates that even with full data, pretraining can provide slight performance improvements.
- 6.25% training data:** ACP-T performs best (CC: 0.7811, RMSE: 1.3535), highlighting that incorporating all three pretraining targets—articulatory labels, critical-articulator labels, and phoneme labels—yields the most robust initialization when training data is limited.

4.1.2. Impact of Pretraining on Unseen Speakers

Table 3 presents results for unseen speakers, where neither the speakers nor the utterances were seen during training. Similar to the seen speakers’ case, pretraining consistently improves performance across all data percentages, with a larger gap in

Table 4: CC and RMSE for Baseline and ACP-T configurations with MFCC and TERA inputs for seen speakers test set

EMA data used for training (%)			6.25	12.5	25	50	75	100
Inputs	Models							
MFCC	Baseline	CC (Std)	0.7348 (0.1652)	0.7857 (0.1453)	0.8254 (0.1253)	0.8563 (0.1125)	0.8723 (0.1061)	0.8778 (0.1022)
		RMSE (Std)	1.4394 (0.3695)	1.318 (0.3535)	1.1964 (0.3407)	1.0939 (0.3244)	1.0440 (0.3210)	1.0190 (0.3127)
	ACP-T	CC (Std)	0.7811 (0.1518)	0.8112 (0.1370)	0.8379 (0.1251)	0.8616 (0.1127)	0.8731 (0.1082)	0.8797 (0.1026)
		RMSE (Std)	1.3535 (0.3580)	1.2620 (0.3515)	1.1694 (0.3371)	1.0850 (0.3300)	1.0850 (0.3300)	1.0135 (0.3183)
TERA	Baseline	CC (Std)	0.7722(0.1564)	0.8045 (0.1401)	0.8362 (0.1237)	0.8629 (0.1105)	0.8770 (0.1044)	0.8812 (0.1025)
		RMSE (Std)	1.3680 (0.3671)	1.2657 (0.3558)	1.1649 (0.3389)	1.0759 (0.3289)	1.0220 (0.3190)	1.0103 (0.3228)
	ACP-T	CC (Std)	0.7870 (0.1549)	0.8102 (0.1412)	0.8378 (0.1279)	0.8639 (0.1153)	0.8754 (0.1073)	0.8826 (0.1025)
		RMSE (Std)	1.3250 (0.3600)	1.2530 (0.3499)	1.1710 (0.3459)	1.0770 (0.3320)	1.3600 (0.4140)	0.9950 (0.3210)

Table 5: CC and RMSE for Baseline and ACP-T configurations with MFCC and TERA inputs for unseen speakers test set

EMA data used for training (%)			6.25	12.5	25	50	75	100
Inputs	Models							
MFCC	Baseline	CC (Std)	0.6687 (0.2020)	0.6991 (0.1996)	0.7265 (0.1876)	0.7488 (0.1783)	0.7488 (0.1783)	0.7563 (0.1822)
		RMSE (Std)	1.5992 (0.3842)	1.5540 (0.4050)	1.4854 (0.3980)	1.4357 (0.3962)	1.4357 (0.3962)	1.4143 (0.4040)
	ACP-T	CC (Std)	0.7259 (0.1872)	0.7399 (0.1814)	0.7469 (0.1875)	0.7616 (0.1796)	0.7653 (0.1803)	0.7689 (0.1823)
		RMSE (Std)	1.5241 (0.4001)	1.4779 (0.4041)	1.4514 (0.4043)	1.4160 (0.4060)	1.4160 (0.4059)	1.3848 (0.4154)
TERA	Baseline	CC (Std)	0.7325 (0.1834)	0.7396 (0.1890)	0.7540 (0.1731)	0.7664 (0.1782)	0.7777 (0.1725)	0.7717 (0.1769)
		RMSE (Std)	1.4788 (0.4027)	1.4515 (0.4113)	1.4160 (0.4040)	1.3860 (0.4070)	1.3774 (0.4120)	1.3878 (0.4154)
	ACP-T	CC (Std)	0.7561 (0.1697)	0.7562 (0.1782)	0.7621 (0.1768)	0.7755 (0.1718)	0.7818 (0.1719)	0.7810 (0.1758)
		RMSE (Std)	1.3999 (0.3770)	1.4110 (0.3889)	1.4110 (0.3970)	1.3849 (0.4070)	1.3600 (0.4140)	1.3619 (0.4210)

performance for low-resource conditions (6.25% to 25%). This suggests that pretraining enhances the generalization ability of AAI models.

- 100% training data:** The ACP-T configuration achieves the highest performance (CC: 0.8612, RMSE: 1.1023), indicating that incorporating all three auxiliary targets provides the best generalization.
- 6.25% training data:** The A-T configuration performs best (CC: 0.7324, RMSE: 1.4900), suggesting that when extremely limited data is available, using only articulatory labels during pretraining still provides a strong foundation.
- Interestingly, many pretraining configurations fine-tuned on just **50% of the training data outperforms the baseline model trained on the full dataset**, reinforcing the efficiency of pretraining in reducing the reliance on large labelled datasets.

4.1.3. Comparing MFCC and TERA Inputs

Tables 4 and 5 compare the impact of input types—MFCC vs. TERA—under the ACP-T pretraining configuration.

- TERA consistently outperforms MFCC**, confirming that SSL features provide richer speech representations, leading to better articulatory inversion performance.
- However, **MFCC with pretraining achieves performance close to or better than the baseline TERA model**, especially in low-resource scenarios (6.25%–25%).
- In Table 4 (seen speakers test set), for **6.25% to 25% of the training data, ACP-T with MFCC input outperforms the baseline model with TERA input**, further reinforcing the effectiveness of pretraining in enhancing model performance even with simpler input representations.

Overall, these results highlight the significant benefits of pretraining for AAI, particularly in low-resource conditions, where it improves both seen and unseen speaker performance while enabling more efficient data utilization.

5. Conclusions

In this work, we explored the impact of pretraining on the Articulatory-to-Acoustic Inversion (AAI) task, investigating

different pretraining targets and input types. Our experiments demonstrated that pretraining significantly improves model performance, particularly in low-resource settings.

Our key findings are as follows:

- Pretraining consistently enhances AAI performance**, particularly in low-resource conditions, where it significantly boosts CC and reduces RMSE.
- Pretraining improves generalization to unseen speakers**, with larger performance gains observed in this scenario compared to seen speakers.
- Pretraining with MFCC input achieves performance comparable to or better than SSL-based models in certain cases**, reinforcing the value of our approach. This suggests that pretraining allows models to learn meaningful articulatory representations without requiring computationally expensive self-supervised feature extractors at inference time.
- Models trained with TERA input still outperform those trained with MFCC input in most cases**, confirming the advantages of SSL-based features for AAI. However, the ability of pretraining to bridge this performance gap highlights its effectiveness in reducing reliance on complex feature extractors.
- By reducing computational overhead without sacrificing accuracy**, our method presents a more efficient alternative for AAI, making it particularly suitable for deployment in real-time and resource-constrained environments.

These results emphasize the importance of pretraining in AAI models, particularly for low-resource settings and robust generalization across speakers. In future work, we plan to explore additional pretraining strategies and target representations to further enhance AAI performance. Investigating diverse pretraining objectives could improve generalization across speakers and data conditions, making AAI systems more adaptable and effective.

6. Acknowledgement

We thank the Department of Science and Technology (DST), Government of India, for supporting this work.

7. References

- [1] P. K. Ghosh and S. S. Narayanan, "A subject-independent acoustic-to-articulatory inversion," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4624–4627.
- [2] V. Mitra, H. Nam, C. Y. Espy-Wilson, E. Saltzman, and L. Goldstein, "Articulatory information for noise robust speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 1913–1924, 2010.
- [3] Z.-H. Ling, K. Richmond, J. Yamagishi, and R.-H. Wang, "Integrating articulatory features into hmm-based parametric speech synthesis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1171–1185, 2009.
- [4] M. Li, J. Kim, A. Lammert, P. K. Ghosh, V. Ramanarayanan, and S. Narayanan, "Speaker verification based on the fusion of speech acoustics and inverted articulatory signals," *Computer speech & language*, vol. 36, pp. 196–211, 2016.
- [5] A. Suemitsu, J. Dang, T. Ito, and M. Tiede, "A real-time articulatory visual feedback approach with target presentation for second language pronunciation learning," *The Journal of the Acoustical Society of America*, vol. 138, no. 4, pp. EL382–EL387, 2015.
- [6] J. N. Larar, J. Schroeter, and M. M. Sondhi, "Vector quantization of the articulatory space," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 12, pp. 1812–1818, 1988.
- [7] K. Richmond, "Mixture density networks, human articulatory data and acoustic-to-articulatory inversion of continuous speech," in *Proc. Workshop on Innovation in Speech Processing*, 2001, pp. 259–276.
- [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, no. 3, pp. 215–227, 2008.
- [9] L. Zhang and S. Renals, "Acoustic-articulatory modeling with the trajectory hmm," *IEEE Signal Processing Letters*, vol. 15, pp. 245–248, 2008.
- [10] A. Illa and P. K. Ghosh, "Representation learning using convolution neural network for acoustic-to-articulatory inversion," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5931–5935.
- [11] A. S. Shahrehabaki, S. M. Siniscalchi, G. Salvi, and T. Svendsen, "Sequence-to-sequence articulatory inversion through time convolution of sub-band frequency signals," *database*, vol. 1, p. 5, 2020.
- [12] Y. M. Siriwardena, A. A. Attia, G. Sivaraman, and C. Espy-Wilson, "Audio data augmentation for acoustic-to-articulatory speech inversion using bidirectional gated rnns," *arXiv preprint arXiv:2205.13086*, 2022.
- [13] P. Liu, Q. Yu, Z. Wu, S. Kang, H. Meng, and L. Cai, "A deep recurrent approach for acoustic-to-articulatory inversion," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4450–4454.
- [14] S. Udupa, A. Roy, A. Singh, A. Illa, and P. K. Ghosh, "Estimating Articulatory Movements in Speech Production with Transformer Networks," in *Interspeech*, 2021, pp. 1154–1158.
- [15] J. Wang, J. Liu, X. Li, M. Yu, J. Gao, Q. Fang, and L. Liu, "Two-stream joint-training for speaker independent acoustic-to-articulatory inversion," in *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5.
- [16] Y. M. Siriwardena, G. Sivaraman, and C. Espy-Wilson, "Acoustic-to-articulatory speech inversion with multi-task learning," *arXiv preprint arXiv:2205.13755*, 2022.
- [17] J. Wang, J. Liu, L. Zhao, S. Wang, R. Yu, and L. Liu, "Acoustic-to-articulatory inversion based on speech decomposition and auxiliary feature," in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4808–4812.
- [18] J. Bandekar, S. Udupa, and P. K. Ghosh, "Exploring a classification approach using quantised articulatory movements for acoustic to articulatory inversion," in *Proc. Interspeech 2023*, pp. 5147–5151.
- [19] A. Illa and P. K. Ghosh, "Low resource acoustic-to-articulatory inversion using bi-directional long short term memory," in *Interspeech*, 2018, pp. 3122–3126.
- [20] A. S. Shahrehabaki, N. Olfati, S. M. Siniscalchi, G. Salvi, T. Svendsen *et al.*, "Transfer learning of articulatory information through phone information," in *Interspeech*, 2020, pp. 2877–2881.
- [21] S. Udupa, C. Siddarth, and P. K. Ghosh, "Improved acoustic-to-articulatory inversion using representations from pretrained self-supervised learning models," in *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5.
- [22] S. K. Maharana, K. K. Adidam, S. Nandi, and A. Srivastava, "Acoustic-to-articulatory inversion for dysarthric speech: Are pre-trained self-supervised representations favorable?" in *2024 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, pp. 408–412.
- [23] Y. Hao, R. Amooie, W. de Vries, T. Tienkamp, R. van Noord, and M. Wieling, "Exploring self-supervised speech representations for cross-lingual acoustic-to-articulatory inversion," in *Interspeech 2024*. ISCA, 2024, pp. 4603–4607.
- [24] W.-J. Chung and H.-G. Kang, "Patient-aware acoustic-to-articulatory inversion through speech ssl layer analysis," in *2024 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, pp. 1–3.
- [25] Y. Sun, Y. Xie, J. Zhang, and D. Ke, "Arti-invar: A pre-trained model for enhancing acoustic-to-articulatory inversion performance," in *2024 IEEE 14th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pp. 154–158.
- [26] J. Bandekar, S. Udupa, and P. K. Ghosh, "Articulatory synthesis using representations learnt through phonetic label-aware contrastive loss," in *Proc. Interspeech*, 2024, pp. 427–431.
- [27] A. Wrench, "A multichannel articulatory speech database and its application for automatic speech recognition," in *Proc. 5th seminar on speech production: models and data*, 2000.
- [28] "3d electromagnetic articulograph," available online: <http://www.articulograph.de/>, last accessed: 4/2/2020.
- [29] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2015, pp. 5206–5210.
- [30] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.
- [31] M. Morshed and M. Hasegawa-Johnson, "Cross-lingual articulatory feature information transfer for speech recognition using recurrent progressive neural networks," *Proceedings of Interspeech*, 2022.
- [32] J. Kim, A. Toutios, S. Lee, and S. S. Narayanan, "A kinematic study of critical and non-critical articulators in emotional speech production (running title: criticality of articulators and emotion)," *The Journal of the Acoustical Society of America*, 2015.
- [33] A. Vaswani, "Attention is all you need," *NeurIPS*, 2017.
- [34] A. T. Liu, S.-W. Li, and H.-y. Lee, "Tera: Self-supervised learning of transformer encoder representation for speech," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2351–2366, 2021.
- [35] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *ICLR*, 2015.