Leveraging Allophony in Self-Supervised Speech Models for Atypical Pronunciation Assessment

Anonymous ACL submission

Abstract

Allophony refers to the variation in the phonetic realization of a phoneme based on its phonetic environment. Modeling allophones is crucial for atypical pronunciation assessment, which involves distinguishing atypical from typical pronunciations. However, recent phoneme classifier-based approaches often simplify this by treating various realizations as a single phoneme, bypassing the complexity of modeling allophonic variation. Motivated by the acoustic modeling capabilities of frozen self-supervised speech model (S3M) features, we propose MixGoP, a novel approach that leverages Gaussian mixture models to model phoneme distributions with multiple subclusters. Our experiments show that MixGoP achieves state-of-the-art performance across four out of five datasets, including dysarthric and non-native speech. Our analysis further suggests that S3M features capture allophonic variation more effectively than MFCCs and Mel spectrograms, highlighting the benefits of integrating MixGoP with S3M features.[1]

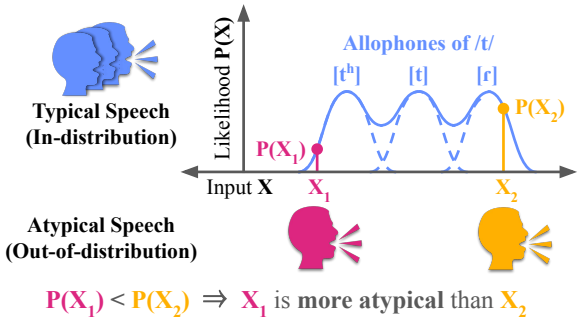

Figure 1: *Summary of our method, MixGoP. We model the likelihood of each phoneme using a Gaussian mixture, trained on typical speech (in-distribution), to capture allophonic variations. We then evaluate on atypical speech (out-of-distribution). The y-axis represents the log-likelihood of a phoneme, where lower values indicate greater atypicality.*

1 Introduction

A phoneme can be phonetically realized differently depending on its environment, a phenomenon known as *allophony* in phonology (Twaddell, 1952; Ladefoged, 1965; Collins et al., 2019). For instance, the English phoneme /t/ exhibits various allophonic realizations: [tʰ] (aspirated stop) in *tap*, [t] (unaspirated stop) in *stop*, [ɾ] (flap) in *butter*, and [ʔ] (glottal stop) in *kitten*. Accurately capturing these variations is crucial, as it reflects the full spectrum of phonetic realizations within a phoneme. It is particularly important for atypical pronunciation assessment (Twaddell, 1952; Jokisch et al., 2009; Vidal et al., 2019), as it has to distinguish atypical (out-of-distribution; OOD) from atypical (in-distribution) pronunciations (Yeo et al., 2023a).

Before the era of deep neural networks (DNNs), allophones were modeled for speech recognition (Sagayama, 1989; Lee et al., 1990; Young et al., 1994). However, DNN-based approaches (Hu et al., 2015b; Yeo et al., 2023a) depend on phoneme classifiers that treat speech segments from a single phoneme as a single cluster, avoiding the complexity of modeling allophones. This is partly due to DNN's strong classification capabilities, which rely on trained hidden features to model individual phonemes well.

In recent years, self-supervised speech models (S3Ms) have shifted the landscape of acoustic modeling. Unlike DNNs, S3Ms leverage their frozen features directly, without requiring additional training (Feng et al., 2023; Chang et al., 2024). Their effectiveness motivates us to revisit modeling allophones via Gaussian Mixture Models (GMMs) (Bilmes et al., 1998; Young et al., 1994). Hence, we propose MixGoP, a GMM-based approach that models each phoneme as a set of allophonic subclusters (see Figure 1). By integrating GMMs with S3M features, we aim to directly capture the al-

[1]The full codebase is available at https://anonymous.4open.science/r/acoustic-units-for-ood-3832

lophonic variations. We evaluate MixGoP with S3Ms in atypical pronunciation assessment with dysarthric and non-native speech.

Furthermore, we analyze the S3M features on how well they capture allophonic variation compared to Mel-frequency cepstral coefficients (MFCCs) and Mel spectrograms. While previous work has shown that S3Ms encode phonetic (Pasad et al., 2021; Wells et al., 2022; Abdullah et al., 2023; Choi et al., 2024b) and phonemic information (Martin et al., 2023; Choi et al., 2024a), a detailed investigation of how they capture allophony remains underexplored.

In summary, the contributions of our study are:

- MixGoP, a novel pronunciation-assessment approach that considers allophonic variation.
- Achieving state-of-the-art performance in four out of five dysarthric and nonnative datasets.
- Analysis of the utility of S3M features on MixGoP for capturing allophonic variations.

2 Method

In this section, we briefly review the conventional approach to pronunciation assessment, Goodness of Pronunciation (GoP) (Witt and Young, 2000). We highlight its limitations: (i) modeling a phoneme as a single cluster, and (ii) assuming atypical speech are in-distribution with respect to typical speech. We then introduce our method, MixGoP, which addresses these limitations by (i) modeling allophonic variation through a mixture distribution and (ii) relaxing in-distribution assumptions by removing the softmax function.

2.1 What is Goodness of Pronunciation?

GoP is a phoneme-level pronunciation score[2] that measures how much the acoustic output of atypical (dysarthric or nonnative) speech deviates from that of typical speech (healthy or native). GoP is measured by how likely a speech segment ($\mathbf{s}$) is to be the intended phoneme (p). Given the phoneme classifier $P_\theta(p|\mathbf{s})$ with trainable parameters θ, GoP is measured as the log phoneme posterior,[3]

$$\texttt{GoP}_p(s) = \log P_\theta(p|\mathbf{s}). \tag{1}$$

2.2 Limitations of GoP

Conventional phoneme classifiers used in GoP assume a *single cluster* for each phoneme. This is because logits $f_\theta(\mathbf{s})$ are often modeled with a speech encoder $\texttt{Enc}$ and a subsequent fully-connected (FC) layer with weights $\mathbf{W} \in \mathbb{R}^{|\mathcal{V}| \times F}$ (Xu et al., 2021; Yeo et al., 2023a):

$$f_\theta(\mathbf{s}) = \mathbf{W} \cdot \texttt{Enc}(\mathbf{s}) \tag{2}$$

where $\texttt{Enc}(\mathbf{s}) \in \mathbb{R}^F$, $|\mathcal{V}|$ denoting the vocabulary size (total number of phonemes), and F the output dimension of the encoder. If we consider a frozen encoder, the trainable parameter $\theta = \{\mathbf{W}\}$. Here, the weights $\mathbf{W}$ can be understood as a codebook, containing a F-dim centroid for each phoneme. It requires a *unimodal* (single peak) clustering of hidden features $\texttt{Enc}(\mathbf{s})$ for each phoneme. This limits the ability to capture allophonic variation, as allophones are represented as distinct acoustic subclusters within each phoneme.

Another limitation comes from the assumption that observed speech segments are *in-distribution* with respect to the training data. This comes from the phoneme classifier P_θ formulation,

$$P_\theta(p|\mathbf{s}) = \texttt{softmax}(f_\theta(\mathbf{s}))[p]. \tag{3}$$

With the phoneme classifier relying on the $\texttt{softmax}$ function, which models a categorical distribution, $\mathbf{s}$ is expected to be within phoneme distribution found in typical speech. However, this assumption is less suitable for atypical speech, which often exhibits substantial acoustic differences from typical speech (Yeo et al., 2023a; Korzekwa et al., 2021).

2.3 MixGoP: Modeling multiple subclusters within a single phoneme

To address the two limitations presented in Section 2.2, we introduce MixGoP, a mixture distribution-based GoP.

First, to overcome the unimodal assumption, MixGoP replaces phoneme classifier $P_\theta(p|\mathbf{s})$ in

[2]While Witt and Young (2000) uses the term "phone"-level pronunciation scores, we use the term "phoneme" to emphasize that the unit includes allophones. Note that Witt and Young (2000) also suggests that their use of "phone" roughly corresponds to a phoneme.

[3]GoP is traditionally calculated as the average log probability of phoneme p over the corresponding time frames: $\frac{1}{|F|}\sum_{f \in \mathbf{s}} \log P_\theta(p|f)$, where f refers to frames within the utterance $\mathbf{s}$. In our study, we simplify this by considering the GoP as the log probability of the phoneme as a whole, rather than averaging over frames.

eq. (1) with a Gaussian mixture model (GMM). The phoneme likelihood is formulated as:

$$P_\theta(\mathbf{s}|p) = \sum_{c=1}^{C} \pi_p^c \mathcal{N}(\mathtt{Enc}(\mathbf{s})|\boldsymbol{\mu}_p^c, \boldsymbol{\Sigma}_p^c) \quad (4)$$

where $\mathcal{N}$ denotes the multivariate Gaussian distribution, $\boldsymbol{\mu}_p^c \in \mathbb{R}^F$ and $\boldsymbol{\Sigma}_p^c \in \mathbb{R}^{F\times F}$ is the mean vector (centroid) and covariance matrix, and $\pi_p^c \in [0,1]$ is the mixing coefficient. Here, the trainable parameter $\theta = \{\boldsymbol{\mu}_p^c, \boldsymbol{\Sigma}_p^c, \pi_p^c\}_{c\in[C],p\in\mathcal{V}}$. Then, we can newly define our GoP score as:

$$\mathtt{MixGoP}_p(s) = \log P_\theta(\mathbf{s}|p). \quad (5)$$

Second, MixGoP removes the `softmax` function of eq. (3) by directly using the log-likelihood in eqs. (4) and (5). It relaxes the assumption of phonemes in atypical speech being in-distribution. The quadratic term inside each Gaussian:

$$-\frac{1}{2}(\mathtt{Enc}(\mathbf{s}) - \boldsymbol{\mu}_p^c)^T(\Sigma_p^c)^{-1}(\mathtt{Enc}(\mathbf{s}) - \boldsymbol{\mu}_p^c) \quad (6)$$

directly relates to the Mahalanobis distance, which is commonly used for OOD detection (Lee et al., 2018). By avoiding the `softmax`, MixGoP is likely to be more robust in handling OOD speech.

In summary, we train a total of $|\mathcal{V}|$ GMMs (one for each phoneme) where each GMM is composed of C subclusters. C is kept constant across all phonemes, as it is known that sufficiently large number of Gaussian mixtures can approximate any probability density (Nguyen et al., 2020). We use the k-means algorithm to determine the initial cluster centers and the expectation-maximization (EM) algorithm to optimize the parameters of the Gaussian mixtures, using `scikit-learn 1.4.1` (Pedregosa et al., 2011). By considering allophony in modeling, MixGoP is expected to better reflect the distribution of each phoneme.

3 Experiments

3.1 Datasets

We use five datasets: three dysarthric speech datasets (UASpeech (Kim et al., 2008), TORGO (Rudzicz et al., 2012), and SSNCE (TA et al., 2016)) and two non-native speech datasets (speechocean762 (Zhang et al., 2021) and L2-ARCTIC (Zhao et al., 2018)). In this paper, we use healthy or native speech as the training sets, and dysarthric and non-native speech as the test sets, in line with the OOD literature (Hendrycks and Gimpel, 2017). Refer to Appendix A for more details.

3.2 Feature extraction

For our experiments, we compare various speech feature extractors $\mathtt{Enc}(\mathbf{s})$ (eqs. (2) and (4)).

Traditional acoustic features. We use the Mel-Frequency Cepstral Coefficients (MFCCs) and Mel spectrograms as the baselines, using the default hyperparameters of librosa (McFee et al., 2015).

TDNN-F features. We compare with a factorized time-delay neural network (TDNN-F) model (Povey et al., 2018) for the speechocean762 dataset, as TDNN-F features have been often used as baselines (Zhang et al., 2021; Gong et al., 2022; Chao et al., 2022; Do et al., 2023).

S3M features. We employ two frozen S3Ms: XLS-R-300M (Babu et al., 2022) and WavLM-Large (Chen et al., 2022). XLS-R, trained cross-lingually, has demonstrated strong performance in ASR for low-resource languages (Babu et al., 2022) and dysarthric speech assessment (Yeo et al., 2023a). We also employ WavLM, a state-of-the-art model for various tasks, including phoneme recognition (Feng et al., 2023; Yang et al., 2021).

As different layers of S3Ms are known to encode different information (Pasad et al., 2021, 2023), we use features from each layer. Specifically, we extract convolutional features (denoted as layer index 0) and all consecutive Transformer features (denoted as layer indices 1 through 24).

Feature segmentation. We segment the features according to the start and end timestamps of each phoneme. Refer to the detailed time-alignment process in Appendix A. Then, we apply center pooling to extract one feature per segment.[4]

3.3 Baselines

We verify the effectiveness of our MixGoP by comparing it against various baselines (Yeo et al., 2023a; Sun et al., 2022; Shahin et al., 2024; Schölkopf et al., 2001). We evaluate on all the speech features listed in Section 3.2 across all the methods for fair comparison. These baselines are categorized into two groups: (i) phoneme classifier-based and (ii) OOD detector-based approaches.

[4]We chose center pooling over average pooling to reduce the impact of inaccurate phoneme alignments on GoP calculations. Both pooling methods are known to encode similar amounts of phonetic information for S3Ms (Pasad et al., 2024; Choi et al., 2024b).

Phoneme classifier-based approaches encompass conventional GoP formulations, which assume a unimodal distribution and in-distribution of phonemes, as discussed in Section 2.2. We employ four popular GoP formulations[5]: GMM-GoP (Witt and Young, 2000), NN-GoP (Hu et al., 2015b), DNN-GoP (Hu et al., 2015b), and MaxLogit-GoP (Yeo et al., 2023a). Note that all formulations use the same underlying phoneme classifier $P_\theta(p|\mathbf{s})$. They only differ by how to calculate the GoP scores. Refer to Yeo et al. (2023a) for more details.

OOD detector-based approaches calculate GoP by measuring how likely an input is to be an outlier. Our MixGoP is one of these approaches, as MixGoP models the likelihood $P_\theta(\mathbf{s}|p)$ with typical speech (eq. (4)) and identifies outliers (atypical speech) based on their likelihood (eq. (5)). We additionally test three baselines: k-nearest neighbors (kNN) (Sun et al., 2022), one-class support vector machine (oSVM) (Schölkopf et al., 2001), and phoneme-specific oSVM (p-oSVM) (Shahin and Ahmed, 2019). While kNN has been utilized for OOD detection (Sun et al., 2022), it has not previously been applied to dysarthric or non-native speech. Conversely, oSVM and p-oSVM have been applied to the evaluation of both disordered and non-native speech (Shahin and Ahmed, 2019; Shahin et al., 2024).

3.4 Training details

Phoneme classifier-based. The phoneme classifier in eq. (1) is trained on features from Section 3.2 with a single learnable FC layer (eq. (2)) with the default settings of Adam optimizer (Kingma and Ba, 2015) for a maximum of 500 iterations.

OOD detector-based. For kNN, we construct kNN model for each phoneme using the features from Section 3.2. Then, we use the maximum Euclidean distance between the input feature and the nearest 10% training data feature as the GoP score, following Sun et al. (2022). For oSVM, all phonemes is modeled with a single oSVM model (Shahin et al., 2024), while p-oSVM modeled each phoneme as a separate oSVM model. All oSVM models are trained with features using the default hyperparameters of `scikit-learn 1.4.1` (Pedregosa et al., 2011). Radial basis function was used for both oSVM and p-oSVM. We use the distance from the hyperplane as the GoP score.

MixGoP. In our MixGoP framework, we apply random subsampling of 512 features per phoneme. Empirical analysis indicates that subsampling does not necessarily degrade performance (See Section 5.2). The number of subclusters for each phoneme-wise Gaussian mixture is set to 32. A detailed investigation into the effect of the number of subclusters on GoP performance is discussed in Appendix C.2.

3.5 Evaluation

As described in Section 3.2, we segment the spoken utterance $\mathbf{x}$ phoneme-wise: $\mathbf{x} = \{(p_1, \mathbf{s_1}), (p_2, \mathbf{s_2}), \cdots, (p_N, \mathbf{s_N})\}$, where p_i is the phoneme label, $\mathbf{s}_i$ is the observed speech segment, and N is the total number of phonemes within the utterance. Following Yeo et al. (2023a), we define the pronunciation score of an utterance $\mathbf{x}$ as:

$$\texttt{Pronunciation}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} \texttt{GoP}_p(s), \quad (7)$$

where the definition of the GoP is different per each method. That is, the GoP scores are averaged across the utterance.

Similar to Yeo et al. (2023a), we evaluate performance using the Kendall-tau correlation coefficient between the utterance-level pronunciation scores and the ground truth dysfluency/disfluency scores provided by the dataset. While Yeo et al. (2023a) used additional training data, our setting only uses the aformentioned datasets.

Unlike other datasets, L2-ARCTIC contains only phoneme-wise mispronunciation detection labels (0 for correct, and 1 for mispronounced). Therefore, we directly measure the correlation between the predicted phoneme-wise pronunciation scores and the mispronunciation detection labels.

3.6 Results

Table 1 presents the experimental results, where we report the best-performing layer's results for S3Ms. Refer to Appendix C.1 for the performance of individual layers. Table 1 shows that our proposed MixGoP method achieved the state-of-the-art performance across all the datasets except L2-ARCTIC. As for the L2-ARCTIC dataset, the best performance was observed with NN-GoP.

[5] GoP formulations were named after the underlying models (Zhang et al., 2021; Yeo et al., 2023a), leading to names like GMM-GoP. However, GMM-GoP scoring method (eq. (1)) does not necessarily rely on GMM.

Table 1: *Kendall-tau correlation coefficient between the pronunciation scores and the dysfluency/disfluency (absolute value). Bigger is better. For S3Ms, the best performance across layers is displayed.*

Dataset	Feature	Phoneme classifier-based				Out-of-distribution detector-based			
		GMM-GoP	NN-GoP	DNN-GoP	MaxLogit-GoP	kNN	oSVM	p-oSVM	MixGoP (Proposed)
UASpeech	MFCC	0.4284	0.4095	0.3609	0.4299	0.4180	0.1066	0.1050	0.1818
	MelSpec	0.2093	0.1718	0.2422	0.2141	0.1012	0.0993	0.0863	0.0394
	XLS-R	0.5516	0.5528	0.5480	0.5471	0.5585	0.3538	0.2467	0.6017
	WavLM	0.5675	0.5679	0.5464	0.5579	0.6057	0.5369	0.3269	**0.6233**
TORGO	MFCC	0.4064	0.3447	0.3905	0.4062	0.3465	0.1686	0.1054	0.2817
	MelSpec	0.2707	0.2619	0.1497	0.2711	0.2869	0.2114	0.2405	0.1959
	XLS-R	0.6769	0.6735	0.6405	0.6752	0.7044	0.5856	0.5362	**0.7125**
	WavLM	0.6821	0.6807	0.6328	0.6807	0.7030	0.6705	0.6211	0.7071
SSNCE	MFCC	0.2653	0.2539	0.2666	0.2727	0.0445	0.1944	0.0756	0.0819
	MelSpec	0.1828	0.1609	0.0513	0.1869	0.1537	0.1139	0.1064	0.1739
	XLS-R	0.5424	0.5416	0.4985	0.5435	0.5027	0.1926	0.1667	0.5414
	WavLM	0.5469	0.5468	0.4860	0.5474	0.5230	0.3583	0.2336	**0.5533**
speechocean762	MFCC	0.3900	0.3754	0.2554	0.4046	0.3216	0.2021	0.1113	0.1256
	MelSpec	0.2135	0.0641	0.2318	0.2290	0.1111	0.1090	0.0709	0.0038
	TDNN-F	0.4002	0.3560	0.2428	0.3595	0.3608	0.0990	0.0005	0.1970
	XLS-R	0.5329	0.5314	0.3716	0.5356	0.4430	0.3124	0.1574	0.4994
	WavLM	0.5348	0.5332	0.3800	0.5342	0.4320	0.3945	0.1732	**0.5387**
L2-ARCTIC	MFCC	0.1363	0.1409	0.1193	0.1193	0.0419	0.0035	0.0344	0.0433
	MelSpec	0.0491	0.0389	0.0319	0.0319	0.0222	0.0034	0.0269	0.0103
	XLS-R	0.2425	**0.3122**	0.1913	0.1913	0.1681	0.0369	0.0672	0.1521
	WavLM	0.2401	0.2693	0.1964	0.1964	0.1892	0.0824	0.0781	0.1821

Comparison between features. We observe that all other features generally underperform compared to both S3Ms, further highlighting the general effectiveness of frozen S3Ms (Yang et al., 2021). Interestingly, MFCCs show non-negligible performance for phoneme classification-based methods, while the performance is limited for MixGoP and kNN. It aligns with the results of Choi and Yeo (2022), where S3Ms, unlike MFCCs, stores information as a relative distance between the features so that the Mahalanobis distance of MixGoP (Equation (6)) or the Euclidean distance of kNN (Section 3.4) can be effective. We explore this discussion in detail in Section 4.

Comparison between datasets. We observe that MixGoP tends to be effective on dysarthric datasets, whereas NN-GoP tends to perform well on non-native datasets. We suspect this is due to dysarthric test sets being more OOD than non-native test sets, so accurate likelihood estimation becomes more important. Dysarthric train/test datasets are split by healthy and dysarthric speakers. However, non-native datasets only contain non-native speakers, where we split train/test using utterance-wise or phoneme-wise pronunciation scores. (Details in Appendix A.) Hence, it is likely that the train/test difference of non-native datasets is less severe than that of dysarthric datasets. Inherent acoustic differences between dysarthric and non-native speech may have further widen the dataset differences (Yeo et al., 2023a; Korzekwa et al., 2021).

Comparison within the same groups. For *phoneme classification-based* baselines, performance greatly differs across methods, despite all four methods being based on the same classifier. This supports the importance of selecting an appropriate equation for uncertainty quantification in phoneme class-based methods, aligning with the findings of Yeo et al. (2023a). Regarding *OOD detector-based* baselines, SVM-based methods usually perform the worst across all datasets. In contrast, kNN achieves performance comparable to our proposed MixGoP on the TORGO and L2-ARCTIC datasets, while MixGoP delivers the best overall performance.

4 Allophony of S3M features

Section 3 empirically demonstrates that leveraging S3M features with MixGoP helps enhance downstream performance compared to other features, such as MFCCs and Mel spectrograms. This section aims to further verify the suitability of S3Ms for representing individual phonemes with allophonic variations. First, we examine S3M features at the phoneme-level, using the dimensionality reduction technique in Section 4.1. Next, we design a metric to quantify the ability of capturing allophonic variations, comparing S3M features to MFCCs and Mel spectrogram in Section 4.2. For our analyses, we used the healthy speech recordings from the TORGO dataset (Rudzicz et al., 2012), which includes gold-standard phonemic transcriptions and alignments.

4.1 Motivating Observation

S3Ms are trained to reconstruct masked signals using surrounding information. Hence, we hypothesize that this will allow the S3Ms to capture local acoustic characteristics, including allophones from various phonetic environments. To verify such phenomena, we observed the final layer features of WavLM for each phoneme, which have generally shown the best performance across datasets (See Figure 5). Specifically, we use UMAP dimensionality reduction (McInnes et al., 2018) with the cosine distance metric to visualize the features, similar to Choi and Yeo (2022). We also extract the four utterances closest to each of the ten centroids to observe the phonetic environments of each cluster.

Figure 2 demonstrates one example, with the distribution of /ʌ/ (/AH/ in ARPABET) and its environments of the healthy subset of TORGO. We observed multiple clusters for each phoneme, each with phonetically similar environments, which motivates the metric for quantifying allophony ability.

4.2 Quantifying Allophony

Previous studies have found that S3M features model the phoneme distributions with multiple clusters (Wells et al., 2022; Martin et al., 2023). However, there has been limited analysis on directly quantifying the relationship between the S3M feature subclusters and allophony. To this end, we design a setting that measures the mutual information between the S3M feature subcluster indices and the surrounding phonetic environment of each phoneme, which is an indicator of allophones.

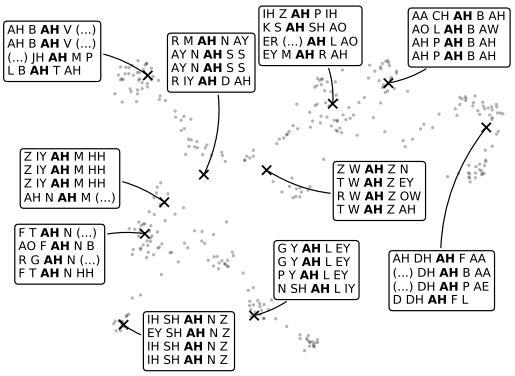

Figure 2: *Visualization of WavLM-Large features of the* /AH/ *phoneme in the TORGO healthy subset. Phonemes are indicated using ARPABET. We observe that* /AH/ *consists of subclusters, each reflecting allophones from different surrounding phonetic environments.*

First, to obtain the subclusters within MFCCs, Mel spectrograms, and S3M features, we apply the k-means algorithm with $k = 32$ clusters to the features of each phoneme $V \in \mathcal{V}$. Then, each utterance has the designated k-means cluster index I. For a dataset with a total of $|\mathcal{V}|$ phonemes, we train $|\mathcal{V}|$ different k-means models.

Note that our MixGoP uses k-means clusters as the initializer. Also, we observed few to no EM optimization steps due to high dimensionality (Wang et al., 2015). As a result, the initialized cluster centroids will likely be similar to the final centroids $\boldsymbol{\mu}_p^c$ in eq. (4) for calculating phoneme likelihood $P_\theta(\mathbf{s}|p)$.

We then compare the utterance-wise cluster indices with their allophony. Since the TORGO dataset does not provide phonetic transcriptions, we utilize the surrounding phonetic environment, which is closely linked to allophonic variation. For simplicity, we define the environment E as the natural class of the preceding and following phonemes, similar to phoneme environment clustering (Sagayama, 1989). We use the height, backness, and roundness for the vowels and the place and manner of the consonants for the natural class. For example, each /i/ and /k/ is represented as `close-front-unrounded` and `velar-plosive`, respectively. If the phoneme is word-initial or word-final, we also include them as the environment information. Therefore, with $|\mathcal{C}|$ number of natural classes, the number of all the possible environments is $(|\mathcal{C}|+1)^2$.

To quantify allophonic information within each subclusters, we measure the mutual information $\mathrm{MI}(I; E)$ between the cluster indices I and the

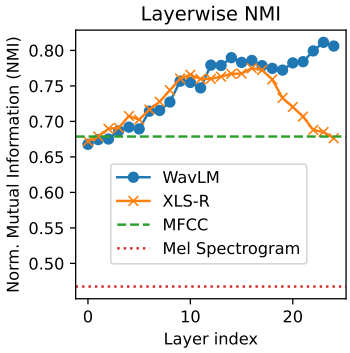

Figure 3: *Normalized Mutual Information $MI(I;E)/H(E)$ between the k-means cluster indices I and the phonetic environment E on the TORGO healthy subset. We show the layerwise NMI for S3Ms and absolute value for MFCC and Mel spectrogram.*

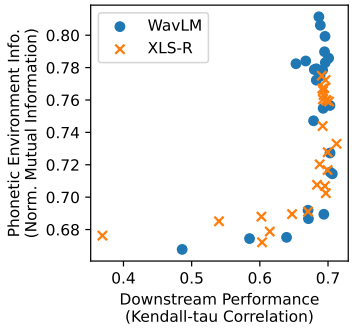

Figure 4: *Comparing phonetic environment information with the downstream task performance.*

environment E. We normalize the value by the environment entropy $H(E)$ so that the resulting value is between 0 and 1. We call the metric *Allophone environment-Normalized Mutual Information* (ANMI) $\text{MI}(I;E)/H(E)$. The actual calculation is nearly identical to Phoneme Normalized Mutual Information (PNMI) (Hsu et al., 2021), except we replaced phoneme V to environment E.

Results. Figure 3 shows the phonetic environment information inside cluster indices for MFCCs, Mel spectrograms, XLS-R, and WavLM. For S3M models, we plot across different layers. We can observe that S3Ms contain more information on the phonetic environment compared to traditional features, implying that S3Ms successfully capture allophony. This finding introduces an interesting implication regarding the effect of varying cluster sizes when applying k-means to S3M features for discrete units (Chang et al., 2024). It is known that different cluster sizes lead to varying levels of granularity, with smaller cluster size capturing phoneme information while larger cluster size capturing speaker information (Sicherman and Adi, 2023). Our results suggest that cluster size in between may capture allophonic variations.

5 Analysis

5.1 Does capturing phonetic environment leads to better downstream performance?

It is crucial to examine whether capturing phonetic environments (or allophones) actually improves downstream performance. To assess this, we compare the amount of phonetic environment information inside S3Ms and the actual downstream performance on the pronunciation assessment. In Figure 4, we observe that the downstream performance positively correlates until around NMI < 0.72, where the downstream performance saturates even if the amount of phonetic environment information increases. We suspect this behavior is due to S3Ms capturing more surrounding information, which may not be useful for the pronunciation assessment task. Our hypothesis aligns with the previous empirical observation of Pasad et al. (2023) and Choi et al. (2024b) that S3Ms have non-negligible word-level modeling abilities, which requires a larger temporal receptive field. Moreover, the layerwise trends of Figure 3, *i.e.*, WavLM persistently increasing and XLS-R peaking in the middle, are also similar to previous empirical observations on word-level layerwise information (Pasad et al., 2023).

5.2 Sample efficiency of MixGoP

We randomly subsampled training set samples to check the influence of training data size. To train the GMM for each phoneme, we can either use all the occurrences in the dataset or limit the maximum number of samples. For optimal performance, we searched for the maximum number of samples between 64, 128, 256, 512, or using the full dataset. For example, if we set the maximum as 64, and /a/ and /i/ each have a total of 100 and 50 samples, to train the GMM of /a/, we randomly subsample 64 samples. On the other hand, since there are only 50 samples for /i/, all 50 samples are used.

Table 2 shows that increasing the number of training samples generally improves performance. However, performance plateaus as the sample size increases, with 512 samples yielding the best results across various datasets, except for L2-ARCTIC. This suggests MixGoP performs well even with a relatively small number of sam-

Table 2: *Ablation on random subsampling.*

Dataset	64	128	256	512	Full
UASpeech	0.6145	0.6202	**0.6235**	0.6233	0.6200
TORGO	0.7039	0.7093	0.7124	**0.7125**	**0.7125**
SSNCE	0.5392	0.5448	0.5487	**0.5533**	0.5475
speechocean762	0.5021	0.5157	0.5283	**0.5387**	0.5364
L2-ARCTIC	0.1816	0.1775	0.1809	0.1821	**0.1965**

ples (fewer than 100), which is advantageous for dysarthric and nonnative speech, where data is often limited. However, this also indicates that adding more data does not necessarily lead to further improvements, indicating that the model's performance may be constrained by data scalability.

6 Related works

6.1 Phoneme-level Pronunciation Assessment

Witt and Young (2000) first introduced GoP to estimate the log posterior probability of a phoneme using a Hidden Markov Model (HMM). Later improvements replaced HMMs with deep neural networks (Hu et al., 2015a,b; Li et al., 2016) and S3Ms (Xu et al., 2021; Yeo et al., 2023a; Cao et al., 2024). GoP has also been enhanced by considering additional factors, such as HMM transition probabilities (Sudhakara et al., 2019; Shi et al., 2020) and phoneme duration (Shi et al., 2020).

Section 2.2 emphasizes the usefulness of framing the pronunciation assessment of atypical speech as the OOD detection task. Yeo et al. (2023a) also addressed this by not using `softmax` for phoneme classifiers improved performance. However, the use of `softmax` during training introduced in-distribution bias. Cheng et al. (2020) modeled input probability with latent representations, but their method still depended on assessment scores to train the prediction model. Our approach improves by directly modeling phoneme likelihood instead of relying on phoneme classifiers. Furthermore, our approach explicitly accounts for allophonic variation within phonemes.

6.2 S3M Feature Analysis

Previous literature on the phonetics and phonology of S3Ms often compared downstream task performance of different layers (Martin et al., 2023; Pasad et al., 2021, 2023, 2024; Choi et al., 2024a). Linear probes (Martin et al., 2023; Choi et al., 2024a) or canonical correlation analysis (Pasad et al., 2021, 2023, 2024) are often used to measure the amount of information. Our work is complementary as previous works focus on the existence of the information, whereas we further investigate on how the information is structured within the S3M features.

Also, discrete speech units from k-means clustering of S3M features have been used as the tokenizer for speech (Chang et al., 2024). Its underlying assumption comes from the feature structure being useful, *i.e.*, similar-sounding segments are close to each other. Choi et al. (2024b) showed that phonetically similar words are close to each other. Also, Baevski et al. (2020); Hsu et al. (2021); Liu et al. (2023) demonstrated that phonemes and S3M cluster indices strongly correlate with each other. Sicherman and Adi (2023); Abdullah et al. (2023) showed that natural classes are also well-clustered. Finally, Wells et al. (2022) showed that the dynamic nature of a single phoneme articulation is captured by a stream of cluster indices. Extending previous works, we focus on the multimodal nature of phonemes and demonstrate that allophones are construct subclusters within the single phoneme.

7 Conclusion

We demonstrated that improved modeling of allophony can enhance performance in OOD detection for the assessment of atypical speech, and that leveraging S3M features can further improve this performance. Specifically, our novel approach, MixGoP, addresses the limitations of uni-modality and in-distribution assumptions by employing Gaussian mixtures, which effectively model allophones and eliminate the need for softmax probabilities. Additionally, we show that utilizing S3M features further enhances OOD detection performance. Our results also confirm that S3M features capture allophonic variation more effectively than traditional features, validating the extension of our approach to include S3Ms. We evaluated eight methods across five dysarthric and nonnative speech datasets, with MixGoP achieving state-of-the-art performance on four of the datasets.

Our work provides a deeper understanding of how S3M representations can be hierarchically structured, from allophones to phonemes. Further, it sheds new light on the acoustic modeling perspective of speech, expanding the existing k-means-based speech discretization. It shows the possibility of using atypical speech as a benchmark to measure the quality of S3M features, especially regarding OOD robustness.

Limitations

First, a key limitation is the restricted generalizability of our findings across languages. Although we aim for our work to benefit a wide range of atypical speakers, including both dysarthric and non-native speakers, our research primarily focuses on English (four English datasets and one Tamil dataset). This limitation stems from the availability of publicly accessible datasets, but we recognize the need for broader cross-linguistic research in future work to ensure that our findings are applicable across diverse languages.

Additionally, we employed different methods for forced alignment across datasets, as outlined in Appendix A. Time alignments were either provided by the dataset or automatically generated using the Montreal Forced Aligner (McAuliffe et al., 2017). However, we did not verify the quality of these alignments in our study. This introduces the possibility that variations in alignment quality could have impacted the GoP scores, potentially affecting the overall results. While we do not primarily focus on comparing performance across datasets, future work could benefit from verifying alignment quality to ensure more reliable GoP scores, and cross-dataset comparisons.

We also acknowledge that our allophony analysis was primarily based on the TORGO dataset, which provided time alignments that were manually annotated by linguists. Extending this analysis to other datasets with similarly verified time alignments would further support the generalizability of our findings.

Finally, the method used to calculate utterance-level (Equation (7)) pronunciation scores can be improved. In our current approach, we simply averaged phoneme probabilities across each utterance; however, it is well-known that certain phonemes have a greater impact on overall pronunciation scores. While our initial analysis, as presented in Appendix C.3, provides a preliminary exploration of this issue, further investigation is needed to identify more robust approaches. Expanding upon this analysis could lead to improved techniques that more accurately evaluate atypical speech.

Ethics Statement

The risk of atypical pronunciation assessment research primarily pertains to data handling and the potential for unintended consequences in use.

Firstly, while we used publicly available datasets that have undergone prior ethical review, it is important to recognize that these datasets still contain sensitive information, particularly speakers' voices. Since no additional anonymization processes were applied in this study, we strongly recommend that any replication of this work prioritize the protection of participants' rights and privacy to the greatest extent possible.

Secondly, concerns arise regarding the potential usage of atypical speech assessment scores. These assessments may unintentionally reinforce negative stereotypes or stigmas associated with speech disorders or non-native accents. If the results are interpreted as evaluations of an individual's language ability or intelligence, they could further marginalize dysarthric or non-native speakers. Also, there is a risk in placing too much emphasis on 'correctness' in phoneme-level pronunciation assessment. Focusing heavily on accurate phoneme production prescribes a rigid, normative standard of speech, potentially penalizing linguistic diversity and variation. For both dysarthric and non-native speakers, such an emphasis might overshadow more functional measures of communication success, which may be more meaningful in real-world contexts. Despite these concerns, which warrant careful consideration, we want to emphasize that our work is intended to have a significant positive impact from an ethical perspective.

Finally, we note that ChatGPT was employed for grammatical refinement and to improve the clarity of English usage in the manuscript. We also state that every sentence generated by ChatGPT was reviewed by the authors.

References

B. M Abdullah, M. M. Shaik, B. Möbius, and D. Klakow. 2023. An information-theoretic analysis of self-supervised discrete representations of speech. In *Proc. Interspeech*.

A. Babu, C. Wang, A. Tjandra, et al. 2022. XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale. In *Proc. Interspeech*.

A. Baevski, Y. Zhou, A. Mohamed, and M. Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Proc. NeurIPS*.

J.A. Bilmes et al. 1998. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International Computer Science Institute*.

M. Blondel, O. Teboul, Q. Berthet, and J. Djolonga. 2020. Fast differentiable sorting and ranking. In *Proc. ICML*.

X. Cao, Z. Fan, T Svendsen, and G. Salvi. 2024. A framework for phoneme-level pronunciation assessment using ctc. In *Proc. Interspeech*.

X. Chang, B. Yan, K. Choi, et al. 2024. Exploring speech recognition, translation, and understanding with discrete speech units: A comparative study. In *Proc. ICASSP*.

F-A. Chao, T-H. Lo, T-I. Wu, et al. 2022. 3m: An effective multi-view, multi-granularity, and multi-aspect modeling approach to english pronunciation assessment. In *Proc. APSIPA ASC*.

S. Chen, C. Wang, Z. Chen, et al. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE JSTSP*.

S. Cheng, Z. Liu, L. Li, et al. 2020. ASR-Free Pronunciation Assessment. In *Proc. Interspeech*.

K. Choi, J-W. Jung, and S. Watanabe. 2024a. Understanding probe behaviors through variational bounds of mutual information. In *Proc. ICASSP*.

K. Choi, A. Pasad, T. Nakamura, et al. 2024b. Self-supervised speech representations are more phonetic than semantic. In *Proc. Interspeech*.

K. Choi and E-J. Yeo. 2022. Opening the black box of wav2vec feature encoder. *arXiv preprint arXiv:2210.15386*.

B. Collins, I. M Mees, and P. Carley. 2019. Phoneme, allophone and syllable. In *Practical English Phonetics and Phonology*. Routledge.

H. Do, Y. Kim, and G-G. Lee. 2023. Hierarchical pronunciation assessment with multi-aspect attention. In *Proc. ICASSP*.

P. Enderby. 1980. Frenchay dysarthria assessment. *British Journal of Disorders of Communication*.

T-H. Feng, A. Dong, C-F Yeh, et al. 2023. SUPERB @ SLT 2022: Challenge on generalization and efficiency of self-supervised speech representation learning. In *Proc. SLT*.

Y. Gong, Z. Chen, I-H. Chu, P. Chang, and J. Glass. 2022. Transformer-based multi-aspect multi-granularity non-native english speaker pronunciation assessment. In *Proc. ICASSP*.

D. Hendrycks and K. Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *Proc. ICLR*.

A. Hernandez, E.J. Yeo, S. Kim, and M. Chung. 2020. Dysarthria detection and severity assessment using rhythm-based metrics. In *Proc. Interspeech*.

W-N. Hsu, B. Bolte, Yao-Hung H. Tsai, et al. 2021. HuBERT: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM TASLP*.

W. Hu, Y. Qian, and F. K. Soong. 2015a. An improved dnn-based approach to mispronunciation detection and diagnosis of l2 learners' speech. In *SLaTE*.

W. Hu, Y. Qian, F. K Soong, and Y. Wang. 2015b. Improved mispronunciation detection with deep neural network trained acoustic models and transfer learning based logistic regression classifiers. *Speech Communication*.

O. Jokisch, A. Wagner, R. Sabo, et al. 2009. Multilingual speech data collection for the assessment of pronunciation and prosody in a language learning system. In *Proc. of SPECOM*.

H. Kim, M. Hasegawa-Johnson, A. Perlman, et al. 2008. Dysarthric speech database for universal access research. In *Proc. Interspeech*.

H. Kim, K. Martin, M. Hasegawa-Johnson, and A. Perlman. 2010. Frequency of consonant articulation errors in dysarthric speech. *Clinical linguistics & phonetics*.

D. P. Kingma and J. Ba. 2015. Adam: A method for stochastic optimization. In *Proc. ICLR*.

J. Kominek and A. W. Black. 2004. The cmu arctic speech databases. In *Proc. Speech Synthesis Workshop*.

D. Korzekwa, J. Lorenzo-Trueba, S. Zaporowski, S. Calamaro, T. Drugman, and B. Kostek. 2021. Mispronunciation detection in non-native (l2) english with uncertainty modeling. In *Proc. ICASSP*.

P. Ladefoged. 1965. The nature of general phonetic theories. *Monograph Series on Languages and Linguistics*, pages 27–42.

J. Lee, K. Lee, H. Lee, and J. Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Proc. NeurIPS*.

K-F. Lee, S. Hayamizu, H-W. Hon, et al. 1990. Allophone clustering for continuous speech recognition. In *Proc. ICASSP*.

K. Li, X. Qian, and H. Meng. 2016. Mispronunciation detection and diagnosis in l2 english speech using multidistribution deep neural networks. *IEEE/ACM TASLP*.

O. D. Liu, H. Tang, and S. Goldwater. 2023. Self-supervised predictive coding models encode speaker and phonetic information in orthogonal subspaces. In *Proc. Interspeech*.

L. MacKenzie and D. Turton. 2020. Assessing the accuracy of existing forced alignment software on varieties of british english. *Linguistics Vanguard*, 6(s1):20180061.

K. Martin, J. Gauthier, C. Breiss, and R. Levy. 2023. Probing Self-supervised Speech Models for Phonetic and Phonemic Information: A Case Study in Aspiration. In *Proc. Interspeech*.

M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger. 2017. Montreal forced aligner: Trainable text-speech alignment using kaldi. In *Proc. Interspeech*.

B. McFee, C. Raffel, D. Liang, et al. 2015. librosa: Audio and music signal analysis in python. In *Proc. Python in Science Conference*.

L. McInnes, J. Healy, N. Saul, and L. Großberger. 2018. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*.

B. K. Ng and P. S. Chiew. 2023. L1 influence on stop consonant production: A case study of malaysian mandarin-english bilinguals. *Asian Englishes*.

T. Nguyen, H. D. Nguyen, F. Chamroukhi, and G. J. McLachlan. 2020. Approximation by finite mixtures of continuous density functions that vanish at infinity. *Cogent Mathematics & Statistics*.

A. Pasad, C-M. Chien, S. Settle, et al. 2024. What Do Self-Supervised Speech Models Know About Words? *TACL*.

A. Pasad, J-C Chou, and K. Livescu. 2021. Layer-wise analysis of a self-supervised speech representation model. In *Proc. ASRU*, pages 914–921.

A. Pasad, B. Shi, and K. Livescu. 2023. Comparative layer-wise analysis of self-supervised speech models. In *Proc. ICASSP*.

F. Pedregosa, G. Varoquaux, A. Gramfort, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*.

D. Povey, G. Cheng, Y. Wang, et al. 2018. Semi-orthogonal low-rank matrix factorization for deep neural networks. In *Proc. Interspeech*, pages 3743–3747.

F. Rudzicz, A. K. Namasivayam, and T. Wolff. 2012. The torgo database of acoustic and articulatory speech from speakers with dysarthria. *Language Resources and Evaluation*.

S. Sagayama. 1989. Phoneme environment clustering for speech recognition. In *Proc. ICASSP*.

B. Schölkopf, J. C Platt, J. Shawe-Taylor, et al. 2001. Estimating the support of a high-dimensional distribution. *Neural computation*.

M. Shahin and B. Ahmed. 2019. Anomaly detection based pronunciation verification approach using speech attribute features. *Speech Communication*.

M. Shahin, J. Epps, and B. Ahmed. 2024. Phonological level wav2vec2-based mispronunciation detection and diagnosis method. In *Proc. Interspeech*.

J. Shi, N. Huo, and Q. Jin. 2020. Context-aware goodness of pronunciation for computer-assisted pronunciation training. In *Proc. Interspeech*.

A. Sicherman and Y. Adi. 2023. Analysing discrete self supervised speech representation for spoken language modeling. In *Proc. ICASSP*.

S. Sudhakara, M. K. Ramanathi, C. Yarra, and P. K. Ghosh. 2019. An improved goodness of pronunciation (gop) measure for pronunciation evaluation with dnn-hmm system considering hmm transition probabilities. In *Proc. Interspeech*.

Y. Sun, Y. Ming, X. Zhu, and Y. Li. 2022. Out-of-distribution detection with deep nearest neighbors. In *Proc. ICML*.

M. C. TA, T. Nagarajan, and P. Vijayalakshmi. 2016. Dysarthric speech corpus in tamil for rehabilitation research. In *Proc. TENCON*.

William F Twaddell. 1952. Phonemes and allophones in speech analysis. *Journal of the Acoustical Society of America*.

J. Vidal, L. Ferrer, and L. Brambilla. 2019. Epadb: A database for development of pronunciation assessment systems. In *Proc. Interspeech*.

Z. Wang, Q. Gu, Y. Ning, and H. Liu. 2015. High dimensional em algorithm: Statistical optimization and asymptotic normality. In *Proc. NeurIPS*.

D. Wells, H. Tang, and K. Richmond. 2022. Phonetic Analysis of Self-supervised Representations of English Speech. In *Proc. Interspeech*.

S. M. Witt and S. J. Young. 2000. Phone-level pronunciation scoring and assessment for interactive language learning. *Speech Communication*, 30:95–108.

X. Xu, Y. Kang, S. Cao, et al. 2021. Explore wav2vec 2.0 for Mispronunciation Detection. In *Proc. Interspeech*.

S-W. Yang, P-H. Chi, Y-S. Chuang, et al. 2021. SUPERB: Speech Processing Universal PERformance Benchmark. In *Proc. Interspeech*, pages 1194–1198.

E-J. Yeo, K. Choi, S. Kim, and M. Chung. 2023a. Speech Intelligibility Assessment of Dysarthric Speech by using Goodness of Pronunciation with Uncertainty Quantification. In *Proc. Interspeech*.

E-J. Yeo, H. Ryu, J. Lee, et al. 2023b. Comparison of l2 korean pronunciation error patterns from five l1 backgrounds by using automatic phonetic transcription. In *Proc. ICPhS*.

S. J. Young, J. J. Odell, and P. C. Woodland. 1994. Tree-based state tying for high accuracy modelling. In *Proc. HLT*.

J. Zhang, Z. Zhang, Y. Wang, et al. 2021. speechocean762: An open-source non-native english speech corpus for pronunciation assessment. In *Proc. Interspeech*.

G. Zhao, S. Sonsaat, A. Silpachai, et al. 2018. L2-arctic: A non-native english speech corpus. In *Proc. Interspeech*.

A Datasets

In our study, we utilize read speech at the sentence level. While the TORGO and SSNCE datasets include both word- and sentence-level data, we focused exclusively on sentences to ensure consistency, as the non-native datasets contain only sentence-level speech. Although the UASpeech dataset consists solely of word-level materials, we included it for comparison with prior work that applied GoP to dysarthric speech (Yeo et al., 2023a).

In contrast to non-native speech datasets, which provide utterance-level scores, dysarthric speech datasets offer intelligibility scores at the speaker level. Therefore, we applied these speaker-level scores to each utterance in the dysarthric datasets.

All the datasets are publicly available, with licenses that allow academic use. We used the datasets for exclusively academic purposes.

A.1 Dysarthric Speech Datasets

UASpeech (Kim et al., 2008) comprises recordings from 25 English speakers, of whom 14 have dysarthria and 11 are healthy. The severity of dysarthria was assessed using the Frenchay Dysarthria Assessment (FDA) (Enderby, 1980), categorizing four speakers as having high-intelligibility, three as mid-intelligibility, three as low-intelligibility, and four as very low-intelligibility. This study analyzes common and uncommon words selected for their diverse phonemetic sequences, which are crucial for evaluating the pronunciation of phonemes in varied contexts. Although the dataset includes recordings from an 8-microphoneme array, we utilize only the fifth microphoneme for computational efficiency. In total, 6,589 utterances from healthy speakers and 8,370 utterances from dysarthric speakers are used. For time alignment information, we apply the Montreal Forced Aligner (MFA) (McAuliffe et al., 2017).

TORGO (Rudzicz et al., 2012) consists of recordings from 15 English speakers, including 8 with dysarthria and 7 healthy individuals. The severity was determined using FDA scores, classifying two speakers as mild, one as mild-to-moderate, one as moderate-to-severe, and four as severe. To balance the classes, mild-to-moderate and moderate-to-severe speakers were merged into a single moderate category. A total of 156 healthy and 413 dysarthric utterances are used. Similar to UASpeech, we apply MFA for time alignment in-

formation. Then, we integrate alignments that were manually adjusted by two linguists, as outlined by Hernandez et al. (2020). These manually refined alignments will be made publicly accessible in our repository.[1]

SSNCE (TA et al., 2016) comprises Tamil speech recordings from 20 dysarthric and 10 healthy speakers. Severity is categorized based on intelligibility scores rated on a 7-point Likert scale by two speech pathologists, resulting in 7 mild (scores 1-2), 10 moderate (scores 3-4), and 3 severe (scores 5-6) speakers. Each speaker recorded 260 distinct sentences, totaling 2,600 healthy and 5,200 dysarthric utterances. We use the time stamps provided with the datasets.

A.2 Nonnative Speech Datasets

speechocean762 (Zhang et al., 2021) consists of 5000 utterances from 250 Mandarin-speaking nonnative children and adult speakers. English proficiency levels' ratio maintains a 2:1:1 ratio for good, medium, and poor, ensuring representation across different proficiencies. Our analysis focuses on total scores at the sentence level, pronunciation quality graded on a scale from 0 to 10. We use scores 9 and 10 as training data and the rest as test data. We employ forced alignments generated using the Kaldi recipe, following the experimental setup of Zhang et al. (2021). This allows for a direct comparison of S3M and Kaldi features, as the quality of phonems alignment can affect the evaluation results (MacKenzie and Turton, 2020).

L2-ARCTIC (Zhao et al., 2018) is a non-native English corpus comprising recordings from 24 speakers, whose first languages (L1) include Hindi, Korean, Mandarin, Spanish, Arabic, and Vietnamese. Each speaker contributes approximately one hour of read speech derived from CMU ARCTIC prompts (Kominek and Black, 2004). For this study, we use 150 utterances per speaker, where manual annotations were available for phonememic errors such as substitutions, deletions, and additions. We use the existing data split as the train/test split, where we further exclude mispronounced utterances in the training data. Unlike other datasets, L2-ARCTIC only contains phoneme-wise mispronunciation detection labels (0/1). Therefore, we used the GoP scores and the label at the phoneme level, not at the utterance level. We use the time stamps provided with the datasets.

Table 3: *Ablation on the number of clusters for Gaussian mixtures.*

Dataset	4	8	16	32	64
UASpeech	0.6130	0.6201	0.6209	**0.6233**	0.6224
TORGO	0.7043	0.7027	0.7060	**0.7125**	N/A
SSNCE	0.5439	0.5483	0.5532	**0.5533**	N/A
speechocean762	**0.5448**	0.5426	0.5444	0.5387	0.5367
L2-ARCTIC	0.1745	0.1784	0.1759	**0.1821**	0.1794

B Computational cost

Extracting S3M features for all the datasets takes less than one day for each S3M on a single NVIDIA V100 GPU. Running the experiments on all the datasets takes less than a day on a 64-CPU 128GB-memory machine for the default hyperparameter settings.

C Additional analysis

C.1 Layerwise analysis of downstream performance

We further compare the layerwise trend of XLS-R and WavLM in Figure 5 to provide a guideline for which layer to use for each downstream task. WavLM tends to have decent or even the best performance in the final layer, while XLS-R features often suffer a great decrease in performance in the final layer. This indicates that the choice of which layer to use is more critical when using XLS-R compared to WavLM, whereas the last layer of WavLM generally shows decent performance.

C.2 Impact of the number of subclusters

We explored the optimal number of subclusters for the downstream task. We conducted a grid search with cluster sizes of 4, 8, 16, 32, and 64, while keeping the best model and layer index from Section 3.6 fixed. For the TORGO and SSNCE datasets, we did not test 64 clusters due to insufficient phoneme samples to train the Gaussian mixtures.

As shown in Table 3, increasing the number of clusters often results in marginal improvements in downstream performance. We attribute this to the fact that better distribution modeling tends to enhance performance. This reinforces our hypothesis that the number of subclusters does not need to exactly match the number of allophones, as sufficiently large number of Gaussian mixtures can approximate any probability density (Nguyen et al., 2020). The best performance was consistently achieved with a cluster size of 32 across datasets,

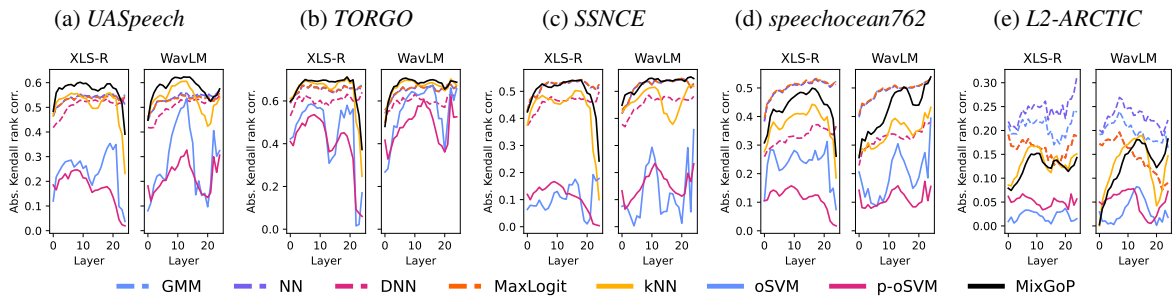

Figure 5: *Kendall-tau correlation coefficient when features are extracted from different layers of S3M models.*

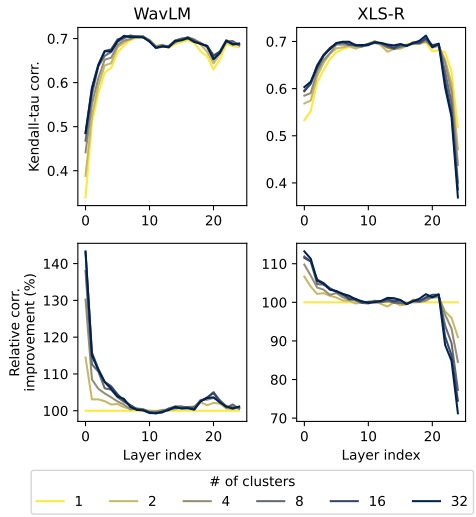

Figure 6: *Downstream performances with varying number of clusters and S3M layer index. Relative performance improvement is obtained by dividing the Kendall-tau correlation with that of cluster size 1. Except for the later layers of XLS-R with extreme performance degradation, having bigger number of clusters yield better performances.*

with the exception of the speechocean762 dataset. For speechocean762, the highest performance was attained with the smallest cluster size, 4, although performances with cluster sizes of 8 and 16 were also similar.

To further analyze the behaviors of different clusters, we performed additional layerwise analysis on TORGO, the smallest dataset, in Figure 6. We can clearly observe that a larger number of clusters leads to better downstream performance across different layers. Especially on the best-performing layer (layer index 19 of XLS-R and 6 for WavLM), it shows bigger differences per different number of clusters. We can also observe that the number of clusters 16 and 32 is the most similar, potentially indicating the performance saturation with respect to the number of clusters.

C.3 Learnable phoneme-wise attention

In dysarthric speech, pronunciation scores for certain phonemes are known to exert more influence on speech intelligibility (Yeo et al., 2023a). Relevant factors include their place and manner of articulation, and articulatory complexity (Kim et al., 2010). Non-native speakers, on the other hand, make different pronunciation mistakes based on their native language (Ng and Chiew, 2023; Yeo et al., 2023b).

To model the phoneme-wise importance, we design a learnable attention module $\alpha \in \mathbb{R}^{|\mathcal{V}|}$ to satisfy two conditions: (i) bounded attention weights ($0 \leq \alpha[p] \leq 1$ for any phoneme p), and (ii) the weights sum up to one ($\sum_{i=1}^{N} \alpha[p_i] = 1$):

$$\alpha[p_i] = \frac{e^{\mathbf{w}_{p_i}}}{\sum_{i=1}^{N} e^{\mathbf{w}_{p_i}}}, \tag{8}$$

where the phoneme-wise logits $\mathbf{w} \in \mathbb{R}^{|\mathcal{V}|}$ is a learnable parameter with the vocabulary size. The formulation is nearly the same as the `softmax` function with the minor difference: if the same phoneme occurs multiple times within the utterance, it shares the same weights.

We can extend our MixGoP by combining the Gaussian mixtures (Equation (4)) and the attention module (Equation (8)):

$$\texttt{MixGoPAttn}(\mathbf{x}) := \sum_{i=1}^{N} \alpha[p_i] \cdot \log P_\theta(\mathbf{s}|p). \tag{9}$$

We improve upon the phoneme-wise GoP of Equation (7) by (i) replacing the phoneme classifier $P_\theta(p|\mathbf{s})$ by the phoneme density estimator $P_\theta(\mathbf{s}|p)$ and (ii) replacing the uniform importance $1/N$ with the learnable weight $\alpha[p_i]$.

We train the phoneme-wise attention module by directly maximizing the Spearman's rank correlation coefficient between $\texttt{MixGoPAttn}(\mathbf{x})$

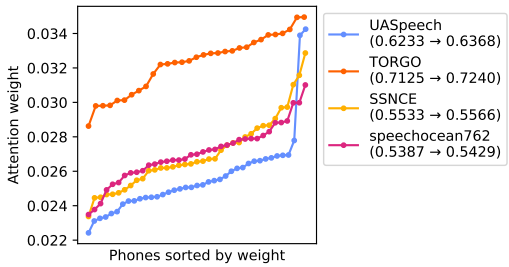

Figure 7: *Learned attention scores and the performance improvement on UASpeech, TORGO, SSNCE, and speechocean762.*

and the pronunciation score y (degree of dysfluency/disfluency for $\mathbf{x}$):

$$\mathcal{L} = -\texttt{spearman}(\text{MixGoPAttn}(\mathbf{x}), y), \qquad (10)$$

where we freeze the Gaussian mixture models and only train the logits $\mathbf{w}$. The sorting operation within `spearman` is made differentiable by soft sorting (Blondel et al., 2020), following Blondel et al. (2020)'s implementation.

We applied 5-fold cross-validation to obtain accurate evaluation results. The phoneme attention module is trained on four datasets, excluding L2-ARCTIC as the dataset does not provide utterance-level scores. Note that this setting differs from the experiments demonstrated in Table 1, as we use the test set to train the attention module.

As demonstrated in Figure 7, applying the attention module resulted in small yet consistent performance improvements across all datasets. Furthermore, the attention weight differences between the least and most influential phonemes varied by up to 1.5 times, indicating variability in their contributions. This highlights the importance of considering the differing influence of each phoneme when calculating utterance-level pronunciation scores, which is crucial for optimizing pronunciation assessment performance.

However, we could not identify a consistent pattern across datasets regarding which phonemes consistently received higher or lower attention scores. This suggests that the variation in attention weights may be influenced by various factors, such as speakers, and phoneme distribution. Further investigation into the underlying mechanisms driving these variations is necessary to gain a deeper understanding of the differing impact of phonemes in pronunciation assessment.[6]

[6]Full list of attention weights can be found in https://anonymous.4open.science/r/attn-C316/