Error-Informed Contrastive Learning for Dutch Personalized Dysarthric Phoneme Recognition

Master thesis report

Begüm Koç

Error-Informed Contrastive Learning for Dutch Personalized Dysarthric Phoneme Recognition

Thesis

to obtain the degree of Master of Science
in Data Science and Artificial Intelligence Technology
at Delft University of Technology,
to be defended publicly on Tuesday June 23rd, 2026.

by

Begüm KOÇ

Faculty of Electrical Engineering, Mathematics and Computer Science,
Delft University of Technology, Delft, Netherlands

Thesis committee:

Dr. Odette Scharenborg,	Technische Universiteit Delft
Dr. Catharine Oertel,	Technische Universiteit Delft
YuanYuan Zhang,	Technische Universiteit Delft

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

Contents

Abstract **ix**

Preface **xi**

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Research Questions . . . 3
1.3 Outline . . . 4

2 Background **5**
2.1 Fundamentals of Speech . . . 5
2.1.1 What is a Phoneme? . . . 5
2.1.2 The Phonetic Feature Space . . . 6
2.1.3 Phoneme Confusions and Confusion Matrices . . . 6
2.2 Automatic Speech Recognition . . . 6
2.2.1 Feature Extraction and Foundation Models . . . 7
2.2.2 Transfer Learning . . . 7
2.2.3 CTC and Forced Alignment . . . 8
2.2.4 Why Phoneme-Level Recognition? . . . 8
2.3 Dysarthric Speech . . . 9
2.3.1 Medical Context and Acoustic Characteristics . . . 9
2.3.2 Challenges for ASR . . . 9
2.3.3 Current Approaches in DSR . . . 10
2.4 Contrastive Learning . . . 10
2.4.1 Principles of Representation Learning . . . 10
2.4.2 Applications of Contrastive Learning in Speech and DSR . . . 11

3 Methodology **13**
3.1 Baseline Architecture . . . 13
3.1.1 Training Procedure . . . 14
3.2 Confusion Matrix Extraction . . . 15
3.2.1 Matrix Construction . . . 16
3.2.2 Filtering Systematic Confusions . . . 16
3.3 Triplet Extraction and Negative Sampling . . . 16
3.4 Contrastive Learning Pipeline . . . 18
3.4.1 Training Initializations . . . 18
3.4.2 Dynamic Phoneme Embedding Extraction . . . 18
3.4.3 Triplet Loss . . . 19
3.4.4 Projection Head . . . 19
3.4.5 Joint Optimization . . . 19

3.4.6 Sequential Branch Processing 20

4 Experimental Setup **21**
4.1 Dataset: The DysOne Corpus. 21
4.1.1 Speaker Profile 21
4.1.2 Recording Environment and Specifications 21
4.1.3 Corpus Composition and Data Splits 22
4.2 Data Preprocessing. 22
4.2.1 Phonemic Target Generation (Grapheme-to-Phoneme) 22
4.2.2 Phoneme Vocabulary Construction. 23
4.3 Baseline Training Configuration 23
4.4 Data Augmentation Strategy 23
4.5 Contrastive Learning Configuration 24
4.6 Software and Hardware Infrastructure 24
4.7 Evaluation Metrics 25
4.7.1 Phoneme Error Rate 25
4.7.2 Per-Phoneme Error Analysis 25
4.7.3 Statistical Significance Testing 25
4.8 Experimental Conditions 26

5 Results **27**
5.1 Baseline Performance 27
5.2 Confusion Matrices 28
5.3 Triplet Statistics 29
5.4 Contrastive Learning Results 29
5.4.1 Performance Against Baseline 29
5.4.2 Effect of Negative Sampling Strategy 30
5.4.3 Comparison Between Initialization Conditions 30
5.5 Phoneme-Level Error Analysis 31

6 Discussion **33**
6.1 RQ 1: To what extent does contrastive learning reduce phoneme error rate? 33
6.2 RQ 2: Does initialization strategy matter? 34
6.3 RQ 3: Does negative mining strategy matter? 34
6.4 RQ 4: Does confusion matrix construction method matter?. 35
6.5 Effectiveness of Personalized Contrastive Modeling 35
6.6 Limitations 36
6.7 Broader Implications 37

7 Conclusion **39**
7.1 Future Work 40

8 Use of Generative AI **43**

9 Appendix **45**
9.1 Scratch Initialization: 5-Epoch CTC Warmup. 45
9.2 Phoneme Vocabulary. 46
9.3 Empirical Triplet Counts 47

9.4 Per-Phoneme PER: Full Results 49

Abstract

Automatic speech recognition systems achieve near-human performance under standard conditions but perform poorly on dysarthric speech due to high acoustic variability resulting from neuromotor impairment. While speaker-specific adaptation can improve performance, limited training data restricts conventional learning approaches. Contrastive learning offers a promising alternative by encouraging more discriminative phoneme representations from limited data, but its effectiveness depends strongly on how negative examples are selected. This thesis investigates whether personalized contrastive learning can improve Dutch dysarthric phoneme recognition.

A Whisper-based encoder-DNN-CTC model is extended with a triplet-loss objective to improve phoneme-level discrimination. Four negative sampling strategies are compared: randomly selected, phonologically motivated, and two empirically derived from the model's own prediction errors, one estimated on the training set and one via cross-validation. Each is evaluated under two training regimes: contrastive fine-tuning of a pretrained model and training from scratch.

All contrastive approaches significantly outperform a CTC-only baseline. The strongest results are obtained with phonologically motivated and cross-validation-based empirical negatives when training from scratch, yielding up to a 10.7% relative reduction in phoneme error rate. Under fine-tuning, differences between sampling strategies are negligible. In contrast, when trained from scratch, the phonological and cross-validation-based empirical strategies significantly outperform randomly selected and training-set-based empirical negatives.

These findings suggest that, for this speaker, contrastive learning for dysarthric speech benefits from phonologically informed or empirically derived negative pairs rather than random selection. A practical trade-off emerges between the two strongest strategies: phonologically motivated sampling requires no speaker-specific preprocessing and is immediately applicable to new speakers, but generates a large number of triplets and is computationally expensive at training time. Cross-validation-based empirical sampling requires building a speaker-specific confusion matrix upfront, but produces fewer, more targeted triplets and trains more efficiently. Given comparable performance, the choice between them reduces to whether preprocessing overhead or training-time resources are the limiting constraint.

Index Terms: automatic speech recognition, dysarthria, contrastive learning, phoneme recognition

Preface

This thesis marks the end of a journey that began five years ago when I first arrived in Delft as a bachelor's student. Over the course of my bachelor's and master's studies here, I have grown both academically and personally, and I leave with far more than a degree.

Working on a topic that directly concerns the lives of people with dysarthria gave this research a sense of purpose that carried me through the more difficult stretches of the project. I have learned enormously along the way, and I hope to continue exploring this area long after this thesis. I also hope that this work contributes in some way to a field where there is still much to be done.

I am deeply grateful to my supervisor, Dr. Odette Scharenborg, for her guidance, critical feedback, and patience throughout this process. Her comments consistently pushed me to think more carefully and sharpen my work. I would also like to thank YuanYuan Zhang for her generous technical support and willingness to help. I learned a great deal from both of them. I am also grateful to Dr. Catharine Oertel for kindly agreeing to be part of my thesis committee.

I equally thank the Speech Lab Group for the warm atmosphere at the biweekly meetings. Being part of that group, hearing about others' research and sharing my own, was both very interesting and motivating.

Finally, I am grateful to my friends and family, and especially my parents, for their continuous support not only during this thesis but throughout my years in Delft. Their encouragement meant more than I can easily put into words. My time here will always remain one of my fondest memories.

Begüm Koç
Delft, June 2026

1

Introduction

1.1 Motivation

Automatic Speech Recognition (ASR) has undergone a significant transformation in recent years. Modern end-to-end neural systems can transcribe continuous speech with near-human performance across many languages, including Dutch [1, 2]. These advances have enabled widespread adoption in applications such as virtual assistants, automated captioning, and accessibility technologies. Yet these improvements have not benefited all speakers equally. State-of-the-art ASR systems are trained primarily on speech from neurologically healthy speakers under controlled or semi-controlled conditions. As a result, performance often drops sharply when these systems are applied to speech that deviates from these patterns [3, 4]. This includes atypical speech, which refers to speech affected by linguistic, physiological, developmental, or neuromotor variation [5]. For such speakers, error rates often reach levels that make current systems impractical for everyday use cases such as dictation or voice control [6].

Among the populations affected by this performance gap, speakers with dysarthria face significant challenges and stand to benefit greatly from reliable ASR technology [7, 8]. Dysarthria is a motor speech disorder caused by neurological damage and is characterized by impaired articulation, reduced respiratory control and high acoustic variability [9]. Because dysarthria is a neuromotor disorder that often co-occurs with broader physical mobility impairments, affected individuals may find using a keyboard or touchscreen challenging. In this context, voice-driven interfaces are not simply convenient, they are essential for enabling independence and ensuring digital accessibility [8]. Reliable ASR systems could support communication, device control and daily digital interaction for these users.

Despite this need, the performance of ASR systems on dysarthric speech remains inadequate. While modern systems achieve word error rates (WER) below 5% on standard benchmarks of typical speech, performance degrades substantially in Dysarthric Speech Recognition (DSR), with reported error rates frequently exceeding 30% [6]. For speakers with severe dysarthria, error rates can reach 80-90% [6, 10], making such systems impractical for real use.

1

The primary issue is the mismatch between training data and target speech. The same characteristics that make dysarthric speech difficult to produce, including irregular articulation, reduced intelligibility, and atypical speech rhythm and intonation, also make it difficult for standard ASR models to recognize. In addition, dysarthric speech varies substantially across speakers. A model that works well for one speaker may therefore perform poorly for another, which is why adapting models to individual speakers is necessary [6, 11].

Previous studies have reported that adapting models to individual speakers improves recognition accuracy, as such systems better capture speaker-specific acoustic and articulatory patterns [12–14]. However, such approaches rely on personalized training data, which limits their scalability and practical deployment.

Collecting dysarthric speech data is physically demanding for speakers and often results in only a few hours of available training material [15]. When high-capacity models are fine-tuned on such small datasets using standard training objectives, they tend to overfit. Instead of learning generalizable patterns, the models memorize the training data and fail to perform well on unseen utterances.

This motivates training objectives that can extract more discriminative information from small speaker-specific datasets without requiring additional annotated data. Contrastive learning has emerged as a promising methodology for this purpose. Unlike standard supervised training, which attempts to map acoustic inputs directly to text labels, contrastive frameworks incorporate a discriminative objective operating in the model's latent embedding space. The core mechanism trains the network to pull "positive" pairs (acoustically or phonetically similar representations) closer together while explicitly pushing "negative" pairs (dissimilar representations) apart. Recent research has explored the use of contrastive learning for DSR and reported improvements in the robustness and discriminability of learned speech representations [16, 17].

However, the effectiveness of contrastive learning depends strongly on how negative examples are selected. Existing methods commonly rely on either random negative sampling [16] or phonological sampling [17]. In random sampling, negative examples are chosen arbitrarily from other phoneme classes. This is often sufficient for typical speech, where phoneme categories are well separated and even arbitrary negatives provide a useful training signal.

For dysarthric speech, however, this assumption is weaker. Dysarthric speech is characterized by speaker-dependent misarticulations caused by neuromotor impairments [9, 18]. These errors are not arbitrary deviations, but recurring phonetic confusions linked to specific articulatory difficulties. Random sampling may therefore frequently select phoneme pairs that are already easily distinguishable for the speaker, while missing the contrasts that consistently break down.

Some approaches address this by using phonetic similarity measures to select harder negative examples [17]. These methods are more linguistically motivated because they prioritize phonemes that are close in articulatory or acoustic space and therefore more difficult to distinguish. However, they assume that phonetic proximity is determined by shared articulatory structure across all speakers, and do not account for the specific motor impairments that shape confusion patterns in dysarthric speech. In dysarthria, confusion patterns are shaped by the speaker's neuromotor impairment [9, 18], meaning

that a similarity measure grounded in universal articulatory features may be systematically misaligned with the speaker's actual error distribution. For example, a speaker with reduced tongue tip control may consistently confuse /t/ and /s/ not because these phonemes are universally close in articulatory space, but because the speaker's motor limitations make this particular contrast difficult to maintain. Whether such speaker-specific confusions provide a stronger training signal than articulatory similarity remains an open question.

This motivates the central hypothesis of this thesis: if DSR is inherently personalized, then the contrastive learning signal should also be personalized. Rather than selecting negative examples based on generic phonological similarity or random sampling, negative examples should reflect the specific phoneme contrasts that are difficult for an individual speaker.

This thesis proposes a personalized error-informed contrastive learning framework for Dutch DSR.[1] A speaker-specific phoneme confusion matrix, constructed from the model's prediction errors at the phoneme level, is used to identify contrasts that are repeatedly misclassified by the model. These empirically observed confusions are then incorporated into the contrastive objective through hard negative sampling, encouraging the model to learn more discriminative representations at precisely those boundaries where recognition errors occur.

1.2 Research Questions

The primary aim of this research is to improve phoneme recognition for Dutch dysarthric speech through personalized error-informed contrastive learning. The framework is studied along two experimental dimensions: (i) negative sampling strategy, comparing speaker-specific confusion-based sampling with random and phonologically motivated alternatives (RQ3, RQ4), and (ii) training integration, comparing contrastive training from scratch with contrastive fine-tuning of a pretrained Connectionist Temporal Classification (CTC) model [19] (RQ2). Together, these are evaluated against a CTC-only baseline (RQ1) using an encoder-based acoustic model with a CTC objective.

The main research question is:

- **Main RQ:** To what extent can error-informed contrastive representation learning improve phoneme recognition performance for Dutch dysarthric speech?

To answer this question, four sub-questions explore architectural and experimental choices.

- **RQ 1 (Effectiveness):** To what extent does integrating a joint CTC and contrastive triplet loss lower Phoneme Error Rate (PER) compared to a CTC-only baseline?
- **RQ 2 (Initialization strategy):** How does model initialization affect performance: joint training from scratch versus contrastive fine-tuning of a pretrained CTC-based model?
- **RQ 3 (Negative sampling strategy):** How does the choice of negative sampling strategy affect phoneme recognition performance? Specifically, how do error-informed

[1] The code is available at github.com/begumkoc/contrastive-dsr.

negatives derived from speaker-specific confusions compare to random and phonological sampling?

- **RQ 4 (Confusion matrix construction):** How does the method used to construct the confusion matrix affect overall recognition performance? Specifically, how do (i) the empirical training set approach, where confusions are derived from predictions on the training set, and (ii) the empirical cross-validation approach, where predictions are made on held-out data, differ in their impact on negative sampling and model performance?

1.3 Outline

The remainder of this thesis is structured as follows. Chapter 2 provides the background, covering speech recognition, dysarthric speech, and contrastive learning. Chapter 3 presents the proposed method, including the baseline model, the construction of speaker-specific error statistics used to guide contrastive learning, and the formulation of the training pipeline. Importantly, the method explores two key design choices: how error statistics are estimated and how the contrastive objective is integrated into training. Chapter 4 describes the experimental setup, including the dataset, preprocessing, training configurations, and evaluation metrics. Chapter 5 presents the experimental results across all conditions, including baseline CTC performance, the effect of contrastive training under both initialization conditions, per-strategy comparisons across the four negative sampling approaches, phoneme-level error analysis, and bootstrap significance testing of key pairwise differences. Chapter 6 interprets the findings in relation to the research questions and prior work, while also addressing limitations and broader implications. Chapter 7 concludes the thesis and outlines future directions.

2 Background

This chapter introduces the technical concepts underlying this thesis. It begins with the fundamentals of speech and phonemes, then explains how speech is modeled computationally in ASR systems, and finally covers dysarthric speech and its challenges for recognition.

2.1 Fundamentals of Speech

Understanding how speech is structured at the phonetic and articulatory level helps in designing systems that can model and recognize it accurately. This section introduces phonemes, describes how they are organized in articulatory space, and explains how phoneme confusions arise.

2.1.1 What is a Phoneme?

Human speech is made up of discrete acoustic units called phonemes, which are the building blocks of spoken language. While a word represents a semantic concept, its phonetic transcription specifies the sequence of sounds required to produce it [20]. To standardize the transcription of these sounds across languages, linguists use the International Phonetic Alphabet (IPA), where each symbol represents a distinct articulatory gesture [21]. For example, the English word "cat" consists of three phonemes: /k/, /æ/, and /t/.

Phonemes are abstract categories representing the sound a speaker intends to produce. The actual physical realization of a phoneme, meaning how it sounds in practice, is referred to as a "phone". This distinction matters for acoustic modeling, since ASR systems operate on the continuous acoustic signal rather than the intended phoneme [20].

Phonemes can be described along articulatory dimensions. Consonants, which involve constricting airflow, are defined by three main features:

- **Voicing:** Whether the vocal folds vibrate during production (e.g., the voiced /z/ versus the voiceless /s/).
- **Place of articulation:** Where in the vocal tract airflow is constricted, ranging from the lips (bilabial sounds like /p/) to the throat (like /h/).

- **Manner of articulation:** How the airflow is manipulated, such as through complete closure and sudden release (plosives like /t/) or continuous turbulent airflow (fricatives like /f/).

Vowels are produced with a relatively open vocal tract and are defined by:

- **Height and backness:** The vertical and horizontal position of the tongue (e.g., /i/ is a high front vowel, /ɑ/ is a low back vowel).
- **Duration and tenseness:** The temporal length and muscular effort involved in production.

While these dimensions are universal, languages differ in how they use them. Dutch has approximately 40-50 phonemes, depending on whether loanwords from other languages are included [22]. In Dutch, for instance, both vowel duration and spectral shape together distinguish otherwise identical words, such as the short /ɑ/ in *man* versus the long /aː/ in *maan*.

2.1.2 The Phonetic Feature Space

The phonetic inventory can be viewed as a multidimensional feature space where each phoneme occupies a position defined by its articulatory features. Phonemes that differ by only a single feature, such as the Dutch /p/ and /b/ which share the same place and manner of articulation but differ only in voicing, are near neighbors in this space.

In typical speech, the acoustic differences between such neighbors are consistent and clear. Speakers reliably produce the articulatory gestures that separate one phoneme from another, so ASR systems can learn to distinguish them from data.

2.1.3 Phoneme Confusions and Confusion Matrices

A phoneme confusion occurs when a phoneme is misrecognized as another. A confusion matrix captures this structure: it is a square matrix where rows represent reference phonemes, columns represent predicted phonemes, and each entry records the frequency with which a given reference was predicted as a specific output. Off-diagonal entries reveal systematic confusions.

Confusion matrices have been used in ASR research to analyze error patterns and identify which phoneme contrasts are most difficult for a system [23, 24]. In dysarthric speech, such analyses show that misclassifications cluster around phoneme pairs that reflect each speaker's specific motor impairment [25, 26], making confusion matrices a useful tool for identifying which contrasts break down for a given speaker.

2.2 Automatic Speech Recognition

Having established the role of phonemes, this section examines how speech is modeled computationally.

ASR is the process of converting a continuous acoustic signal into a sequence of discrete linguistic units, such as phonemes or words. This task is inherently challenging due to variability in speech caused by differences between speakers, recording devices, and background noise [27].

2.2.1 Feature Extraction and Foundation Models

Before processing speech, an ASR model first transforms the raw audio waveform into a compact numerical representation. Most modern systems use a Log-Mel spectrogram [28]: a two-dimensional matrix where one axis represents time and the other represents frequency, warped to the Mel scale to approximate human auditory perception [29].

Historically, ASR systems used pipelines consisting of three separate components: an acoustic model (typically Hidden Markov Models combined with Gaussian Mixture Models, HMM-GMM), a pronunciation lexicon mapping words to phonemes, and an n-gram language model for predicting word sequences [27, 30]. While effective, these systems required significant domain expertise to build and carefully tune.

Modern ASR has shifted toward end-to-end architectures, which replace this pipeline with a single neural model that maps acoustic features directly to output sequences [1]. These models learn robust representations from large amounts of data, removing the need for handcrafted intermediate components, and achieve strong performance across many tasks [31].

Transformer architectures have become the most widely used approach in ASR, as they can model long-range dependencies in speech using self-attention mechanisms [32]. A prominent example is Whisper, a large-scale supervised model trained on approximately 680,000 hours of multilingual speech data [2]. It uses an encoder-decoder architecture: the encoder converts Log-Mel spectrograms into speech representations, and the auto-regressive decoder generates output tokens sequentially. In this thesis, only the Whisper encoder is used, with the decoder replaced by a CTC head. This avoids a known issue with auto-regressive decoders, which tend to favor linguistically probable outputs and can mask real pronunciation errors when the input deviates from typical speech [33].

Despite strong performance on typical speech, such models face challenges when applied to dysarthric speech, since the encoder's representations are optimized for typical articulatory patterns [34, 35].

2.2.2 Transfer Learning

Transfer learning is the practice of initializing a model with weights learned on a large source task and then adapting it to a smaller target task [36]. In ASR, the idea is that the lower layers of a foundation model have already learned general acoustic features, such as pitch and voicing, that transfer usefully across domains [37]. The higher layers, which are more specific to the source data, can then be fine-tuned to adapt to the target domain [38].

This approach is particularly valuable for DSR, where data scarcity makes training from scratch impractical [12]. A key risk during fine-tuning is catastrophic forgetting, where the model overwrites useful pretrained knowledge while adapting to new data [39]. Common strategies to mitigate this include using lower learning rates for the pretrained encoder than for the newly added classifier, or gradually unfreezing layers during training [40]. This distinction between fine-tuning a pretrained model and training from a random initialization directly motivates the comparison between initialization conditions examined in RQ2 (see Section 1.2).

2.2.3 CTC and Forced Alignment

Training a sequence model requires knowing how input frames align with output labels. For speech recognition, this means knowing which segments of audio correspond to which phoneme in the transcript. Obtaining such frame-level alignments by hand is extremely time-consuming, while automatic alignment requires an existing model, creating a circular dependency. CTC resolves this by marginalizing over all possible alignments during training, requiring only the input-output sequence pair and no frame-level labels.

CTC optimizes the network to predict a probability distribution over all possible valid alignments [19]. To achieve this, it introduces a "blank" token (ϵ), representing the absence of a phoneme. The network's vocabulary V is extended to $V' = V \cup \{\epsilon\}$. At each time step t, the network outputs a probability distribution over V'.

A path π is a sequence of tokens of length T, where a token is a single element from V' chosen at each time step. The probability of a specific path is computed assuming each time step is independent:

$$P(\pi|X) = \prod_{t=1}^{T} y_{\pi_t}^t \tag{2.1}$$

Because multiple paths can represent the same transcription (e.g., "a-a-b", "a-ϵ-b", and "ϵ-a-b" all collapse into "a-b" via a mapping function $\mathcal{B}$), the total probability of the target sequence Y is the sum of all valid paths:

$$P(Y|X) = \sum_{\pi \in \mathcal{B}^{-1}(Y)} P(\pi|X) \tag{2.2}$$

The CTC loss is the negative log-likelihood of this target sequence probability ($-\log P(Y|X)$). Minimizing this loss allows the network to implicitly align the sequences.

CTC forced alignment extends this by using the Viterbi algorithm to find the single most probable alignment path for a given transcript [41]. Unlike standard CTC, which sums over all valid paths, forced alignment constrains the search to paths that exactly match the target sequence, enabling precise identification of phoneme boundaries. In this work, forced alignment is applied dynamically during training: at each training step, the current model's frame-level CTC probabilities are used to recompute phoneme boundaries, ensuring that the boundaries stay accurate as the model's representations change during fine-tuning.

2.2.4 Why Phoneme-Level Recognition?

Most commercial ASR systems map acoustic input directly to words, relying on language models to predict likely outputs based on context. This can hide real pronunciation errors: the system may recognize the intended word correctly even when the sounds produced were quite different, similar to an autocorrect silently fixing a typo. This makes word-level systems less useful for studying how speech sounds are actually produced.

Phoneme-level recognition evaluates the acoustic signal more directly, without this contextual smoothing. It is also a more practical task in low-resource settings, such as speaker-dependent dysarthric datasets: because the set of phonemes is small and fixed, a model needs far less training data to cover the output space than a system with an open word vocabulary [4, 42]. For these reasons, phoneme-level recognition is better suited

than word-level systems to studying how dysarthric speech is actually produced, and more practical to train in the low-resource, speaker-dependent setting of this thesis.

2.3 Dysarthric Speech

While modern ASR systems perform well on typical speech, they are trained on typical speech and struggle in DSR. This section introduces dysarthria and the challenges it presents for recognition.

2.3.1 Medical Context and Acoustic Characteristics

Dysarthria is a motor speech disorder resulting from neurological injury or disease, such as cerebral palsy, brain trauma, neuropathy, or Parkinson's disease [9]. It directly impairs control over the muscles used for speech, including the lips, tongue, jaw, soft palate, larynx, and respiratory system.

The acoustic characteristics of dysarthric speech vary widely across subtypes and severity levels. Individuals may exhibit a broad range of speech abnormalities, including imprecise pronunciation, uncontrolled loudness, slurred articulation, slow speaking rates, abnormal pauses, and irregular pitch, rhythm, or breathing patterns [43]. These symptoms reduce speech intelligibility [9, 44], which negatively impacts social participation and quality of life [45, 46].

2.3.2 Challenges for ASR

While ASR technology has the potential to assist these individuals, standard systems are primarily trained on typical speech. As a result, they perform poorly on dysarthric speech due to a significant acoustic mismatch, with WER reaching 80% to 90% for speakers with low intelligibility [6, 10]. These limitations occur at the acoustic, linguistic, and data levels.

At the acoustic level, dysarthric speech sounds very different from the typical speech these models were trained on. This mismatch is heightened by two types of variability [47]. First, the same dysarthric speaker may produce the same word differently across attempts (intra-speaker variability). Second, different dysarthric speakers can sound very different from one another, even with the same diagnosis (inter-speaker variability). Because of this, a single general model cannot perform well across all speakers. The literature shows that speaker-dependent models outperform speaker-independent ones [6, 11, 12], which directly motivates the single-speaker approach taken in this thesis.

At the linguistic level, dysarthric speakers may produce systematic substitutions due to a limited phonemic repertoire [26]. For example, a speaker might consistently substitute or delete specific consonants. These recurring, speaker-specific error patterns are precisely what the confusion-based approach in this thesis is designed to capture.

At the data level, dysarthric speech datasets are scarce. Recording this type of speech is time-consuming and demanding for speakers due to fatigue and physical limitations, leaving the few publicly available datasets orders of magnitude smaller than typical ASR training sets [48]. This scarcity makes it difficult to train fully adapted models.

2.3.3 Current Approaches in DSR

The unique challenges of dysarthria have driven the development of specialized DSR systems. Early approaches adapted conventional architectures such as HMM-GMM models [4, 48], which allowed explicit phonetic alignment but struggled with the large acoustic variation in dysarthric speech.

The shift toward deep neural networks improved acoustic modeling. Researchers adopted various architectures, from early artificial neural networks to convolutional and recurrent neural networks [48–50]. To address the lack of data, these systems often use data augmentation techniques such as speed perturbation, noise addition, and audiovisual integration [50–52].

More recently, end-to-end architectures built on large pretrained models such as wav2vec 2.0 and Whisper have achieved state-of-the-art results on typical speech [2, 53]. However, because these models are pretrained on typical speech, their internal representations do not match the atypical patterns of dysarthric speakers, and DSR performance remains poor when applied without adaptation [54, 55].

Ultimately, DSR is limited by two related problems: data scarcity and high acoustic variability [15, 56].

2.4 Contrastive Learning

One approach to making better use of limited dysarthric speech data is to learn representations that are more discriminative, where similar sounds cluster together and different sounds stay apart in the model's internal space. This section introduces contrastive learning as a framework for achieving this, and discusses its application to speech and DSR.

2.4.1 Principles of Representation Learning

Contrastive learning is a representation learning approach designed to structure a model's embedding space. Unlike standard classification objectives such as Cross-Entropy or CTC, which optimize for absolute class probability, contrastive learning optimizes for relative distances between representations. The core idea is to pull the embeddings of similar samples closer together while simultaneously pushing the embeddings of dissimilar samples further apart [57].

Contrastive learning can be applied in both unsupervised and supervised settings. In unsupervised contrastive learning, similarity is defined through data augmentation [58]: two different augmentations of the same sample are treated as a positive pair without requiring class labels. In supervised contrastive learning, class labels determine similarity, where samples from the same class form positive pairs and samples from different classes form negative pairs [59].

In supervised settings, this is typically achieved using a triplet loss formulation. A triplet consists of an anchor (x_a), a positive example (x_p) from the same class as the anchor, and a negative example (x_n) from a different class. The loss encourages the distance between the anchor and the positive to be smaller than the distance between the anchor and the negative by at least a margin, m. The loss is defined as [60]:

$$L = \max(0, d(x_a, x_p) - d(x_a, x_n) + m) \tag{2.3}$$

where d is a distance function. In this work, d is cosine distance computed on L_2-normalized embeddings. L_2-normalization projects embeddings onto the unit hypersphere, making cosine distance a bounded, scale-invariant measure of similarity. The margin m controls how aggressively the loss separates positive and negative pairs: a larger margin produces a more discriminative embedding space, but setting it too large can destabilize training, while setting it too small may not separate dissimilar classes sufficiently [60].

The effectiveness of the contrastive objective depends heavily on the choice of the negative sample (x_n) [61]. Randomly selected negatives often produce easy comparisons (e.g., distinguishing a vowel from a consonant), where the model can already separate the samples correctly. As a result, the loss and gradient become zero, so the model weights are not updated and the network learns little from these samples. To avoid this, hard negative mining deliberately selects negatives that are already close to the anchor in the embedding space, forcing the model to make finer distinctions [60]. It is common practice to apply the contrastive objective in a separate embedding space rather than directly on the task-specific representations, to avoid degrading downstream performance [58].

2.4.2 Applications of Contrastive Learning in Speech and DSR

Contrastive learning was first widely adopted in computer vision [58], and was subsequently applied to speech through self-supervised architectures such as Contrastive Predictive Coding (CPC) [62] and wav2vec [53], which use contrastive objectives to learn acoustic representations from large unlabeled datasets. In standard ASR, [63] showed that jointly training CTC and contrastive objectives improves phoneme discrimination, demonstrating that the two losses are complementary rather than competing.

In DSR, early contrastive work focused on utterance-level objectives. [16] used augmented dysarthric utterances as positive pairs to improve the stability of global speech representations. [64] extended this to word-level contrastive learning. However, neither approach addresses localized articulation errors at the phoneme level.

More recent work has moved to phoneme-level contrastive learning. DyPCL selects hard negatives based on articulatory feature distances – how similar two phonemes are in terms of how they are physically produced – and demonstrates that phoneme-level contrastive objectives are effective for dysarthric speech [17]. However, DyPCL selects negatives based on speaker-independent phonetic similarity measures. This thesis extends this line of work by using speaker-specific phoneme confusion patterns, derived from the model's own prediction errors, to select hard negatives that target the contrasts that actually break down for each individual speaker.

3

Methodology

This chapter details the proposed method that combines a CTC objective with an error-informed contrastive learning objective. Architectural design choices are described in terms of their functional purpose. All hyperparameter values are reported in Chapter 4.

3.1 Baseline Architecture

The baseline model, shown in Figure 3.1, is a Whisper-based CTC phoneme recognition system. It follows the standard encoder-DNN-CTC architecture used in the SpeechBrain ASR toolkit [65], adapted here for phoneme recognition in dysarthric speech.

Given an input waveform x with shape $[B, T_{\text{frames}}]$, where B is the batch size and T_{frames} is the number of audio samples, the model applies the following steps:

$$S = \text{LogMelSpectrogram}(x) \qquad [B \times 80 \times T_{\text{frames}}] \tag{3.1}$$

$$f = \text{WhisperEncoder}(S) \qquad [B \times T_{\text{sub}} \times d_{\text{encoder}}] \tag{3.2}$$

$$h = \text{DNN}(f) \qquad [B \times T_{\text{sub}} \times d_{\text{hidden}}] \tag{3.3}$$

$$z = \text{Linear}(h) \qquad [B \times T_{\text{sub}} \times |V|] \tag{3.4}$$

$$p = \text{LogSoftmax}(z) \qquad [B \times T_{\text{sub}} \times |V|] \tag{3.5}$$

First, the waveform is converted into an 80-channel Log-Mel spectrogram (3.1). The spectrogram is then passed through the pre-trained Whisper encoder (3.2), which produces frame-level acoustic representations. These representations are processed by a three-layer DNN (3.3) that adapts the encoder outputs for phoneme recognition. Finally, a linear projection (3.4) followed by a log-softmax layer (3.5) produces frame-level phoneme log-probabilities.

Here, T_{sub} denotes the number of frames after subsampling, determined by the Log-Mel feature extraction and the encoder subsampling factor. d_{encoder} represents the encoder hidden dimension, d_{hidden} is the hidden size of the DNN, and $|V|$ is the phoneme vocabulary size.

Each DNN layer (3.3) consists of a linear transformation followed by Layer Normalization, a LeakyReLU activation, and Dropout. Layer Normalization helps stabilize training by

normalizing activations within each layer, while Dropout reduces overfitting by randomly masking activations during training.

The model is trained using the CTC loss, which aligns frame-level predictions with the target phoneme sequence without requiring explicit frame-level annotations:

$$L_{\text{CTC}} = -\log P_{\text{CTC}}(y \mid x) = -\log \sum_{\pi \in \mathcal{B}^{-1}(y)} \prod_{t=1}^{T_{\text{sub}}} p_t(\pi_t \mid x) \tag{3.6}$$

where y is the target phoneme sequence, π represents a valid frame-level alignment path, and $\mathcal{B}^{-1}(y)$ denotes the set of all paths that collapse to y after removing repeated tokens and blank symbols.

3.1.1 Training Procedure

Fine-tuning a large-scale foundation model on a low-resource dataset introduces a risk of catastrophic forgetting, where limited and acoustically variable data causes the model to overwrite its pretrained representations. To mitigate this, the baseline adopts a two-phase differential optimization strategy from the SpeechBrain Whisper CTC fine-tuning recipe [65]:

- **Phase 1 (Warmup):** The Whisper encoder is frozen and only the DNN and CTC projection layers are updated. This allows the task-specific classifier to stabilize before the encoder is exposed to the learning signal.
- **Phase 2 (Fine-Tuning):** The encoder is unfrozen and trained jointly with the DNN and CTC head using two separate AdamW optimizers: one for the DNN (lr_{model}) and one for the encoder (lr_{whisper}), where $lr_{\text{whisper}} < lr_{\text{model}}$. This allows the model to adapt to the dysarthric speaker's phoneme patterns while preserving pretrained representations.

Both learning rates are adjusted using a NewBob scheduler: if validation loss does not improve by at least a threshold δ within a patience window, the learning rate is multiplied by a decay factor. All evaluations use greedy CTC decoding, taking the argmax at each frame.

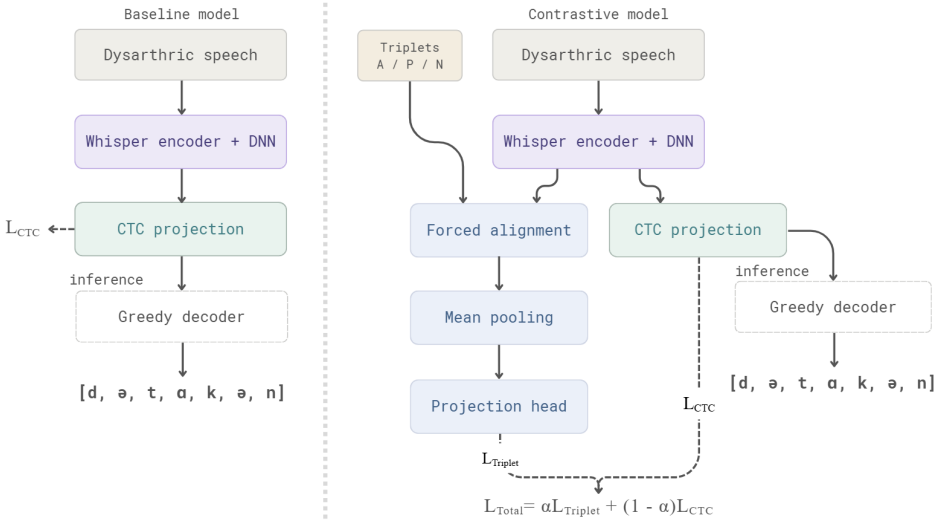

Figure 3.1: Overview of the baseline and contrastive model architectures. *Left:* The baseline model processes dysarthric speech through a Whisper encoder and DNN, producing frame-level log-probabilities via a CTC projection layer. At inference, a greedy decoder produces the phoneme sequence. *Right:* The contrastive model extends the baseline by processing triplets of anchor (A), positive (P), and negative (N) through the encoder and DNN. Forced alignment and mean pooling extract phoneme-level embeddings, which are passed through a projection head to compute the triplet loss. The total loss combines CTC and triplet objectives weighted by α. At inference, the projection head is discarded and only the CTC path is used.

3.2 Confusion Matrix Extraction

To identify phonemes that are systematically misrecognized, a confusion matrix is constructed from the baseline model's predictions. This matrix is used only to identify frequent substitution pairs, which serve as the basis for selecting hard negatives in contrastive training. Two approaches are evaluated to construct this confusion matrix:

Empirical Training Set:
The confusion matrix is constructed by running greedy inference on the full training set. Since the model has already been fine-tuned on these examples, the remaining substitution errors reflect phoneme pairs that are still difficult to distinguish after training exposure. These persistent confusions are therefore hypothesized to be the hardest cases for the model on this speaker's data, making them strong candidates for hard negative selection.

The validation set is not used for this purpose, as it is substantially smaller and low-frequency phonemes may not produce any substitution errors, leaving their corresponding rows empty. The test set is excluded to prevent data leakage, since the extracted confusion patterns are used to guide contrastive training.

Empirical Cross-Validation:
An alternative approach constructs the confusion matrix from predictions on held-out data. The training set is partitioned into five approximately equal folds. For each fold k, a separate baseline model is trained on the remaining four folds and used to decode the

held-out fold k. Substitution errors from all five held-out runs are then aggregated into a single global confusion matrix.

Because every prediction is made on utterances not seen during that model's training, this approach produces an unbiased estimate of phoneme-level error patterns across the training set. The resulting matrix captures a broader range of substitution errors than the training set approach.

Both approaches are compared to assess how the choice of estimation strategy affects the resulting negative sampling distribution and downstream model performance.

3.2.1 Matrix Construction

For both approaches, the confusion matrix M is constructed by comparing the model's greedy CTC-decoded output sequences $\hat{Y}$ against the corresponding ground-truth phoneme sequences Y. Since CTC produces unaligned output sequences, Levenshtein alignment is applied to each reference-hypothesis pair to extract edit operations, which fall into three categories: substitutions (a reference phoneme is replaced by a different phoneme), deletions (a reference phoneme is absent from the hypothesis), and insertions (the hypothesis contains a phoneme with no corresponding reference). Only substitution errors are retained: for each substitution in which a reference phoneme p_i is misclassified as p_j, the entry $M_{i,j}$ is incremented, yielding a matrix that captures the empirical frequency of phoneme-to-phoneme confusions. Insertions and deletions are excluded as they do not define a direct mapping between competing phonetic classes.

3.2.2 Filtering Systematic Confusions

Not all observed substitutions are equally informative. Low-frequency confusions may reflect random errors, isolated mispronunciations, or alignment noise rather than systematic articulatory difficulties. To address this, a minimum frequency threshold is applied, retaining only phoneme pairs whose substitution counts meet or exceed this threshold. The resulting filtered matrix provides a high-confidence approximation of the speaker's dysarthric error profile. The retained substitution pairs are used directly to define anchor-negative pairings in the empirical strategies described in the following section. The specific threshold value is reported in Chapter 4.

3.3 Triplet Extraction and Negative Sampling

As introduced in Section 2.4, contrastive learning requires triplets consisting of an anchor, a positive, and a negative example. The construction of the triplets proceeds in two stages:

Stage 1: Phoneme-Level Pairing

For each anchor phoneme class, the positive class is defined as the same phoneme occurring in a different utterance. The negative phoneme class is selected using one of four strategies, illustrated in Figure 3.2:

1. **Random:** The negative phoneme class is sampled uniformly from all phonemes different from the anchor. This serves as an uninformed baseline.

2. **Phonological:** The negative is selected as the phoneme with the minimum articulatory feature distance to the anchor, computed using the Panphon library [66].

Panphon represents each IPA phoneme as a vector of 22 binary articulatory features and computes a feature-weighted distance between pairs.

3. **Empirical Training Set:** Negatives are drawn from the filtered training-set confusion matrix described in Section 3.2. Every retained substitution pair defines an anchor-negative pairing; an anchor that is systematically confused with multiple phonemes is therefore paired with each of them.

4. **Empirical Cross-Validation:** The same procedure is applied to the five-fold cross-validated confusion matrix described in Section 3.2.

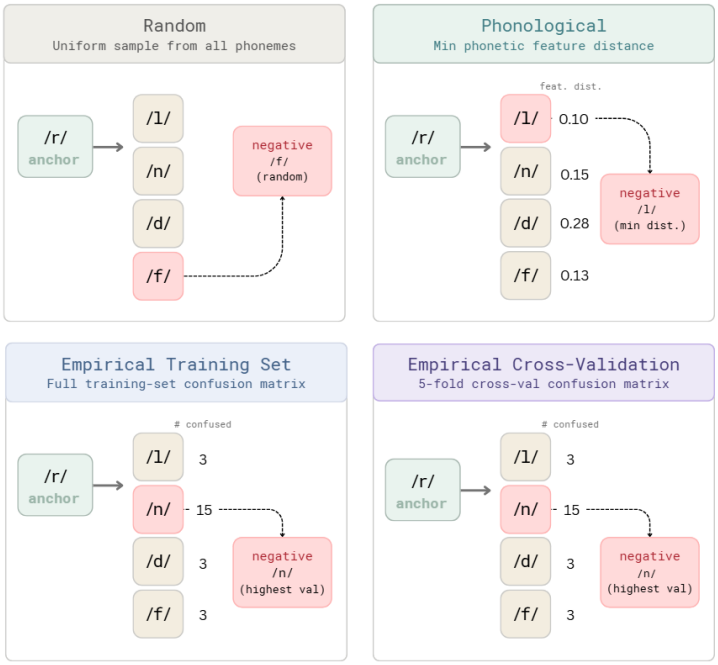

Figure 3.2: Illustration of the four negative sampling strategies for an example anchor phoneme /r/. *Random:* the negative is selected uniformly at random from other phoneme classes. *Phonological:* the negative is the phoneme with the minimum articulatory feature distance computed using Panphon library. *Empirical training set:* the negative is the most frequently confused phoneme in the full training-set confusion matrix. *Empirical cross-validation:* the same confusion-guided selection is applied to the five-fold cross-validated matrix.

Stage 2: Utterance-Level Triplet Construction

Once phoneme-level pairings are defined, training triplets are constructed offline. For each occurrence of an anchor phoneme in the training set, we sample: (i) a positive example from a different utterance containing the same phoneme class, and (ii) up to three negative examples from utterances containing the designated negative phoneme class. When fewer than three negatives are available, all candidates are used, resulting in fewer triplets for that anchor.

All samples are drawn from distinct utterances to prevent the model from exploiting shared recording conditions or local contextual cues. Every occurrence of each target phoneme is used as an anchor, ensuring full coverage of the available phonetic realizations. The resulting triplet dataset is shuffled prior to training.

Triplets are precomputed and stored on disk. Each entry specifies the audio file and the index of the target phoneme instance within the corresponding utterance for the anchor, positive, and negative examples.

During training, audio is loaded on the fly and passed through the model. Phoneme-level embeddings are then extracted via forced alignment, as described in Section 3.4.2. This step is performed dynamically because model parameters evolve during training, requiring recomputation of frame boundaries to reflect the current state of the model.

3.4 Contrastive Learning Pipeline

The proposed contrastive learning framework extends the baseline model with a triplet loss objective, as illustrated in Figure 3.1. The combined framework is evaluated under two initialization regimes: scratch and fine-tuning. In both settings, the Whisper encoder is optimized using the CTC objective while being regularized by the contrastive loss, enabling the model to improve phoneme-level recognition while preserving transcription performance.

3.4.1 Training Initializations

The two training regimes evaluated for integrating the contrastive objective differ only at the point at which contrastive learning is introduced. Both use the same model architecture, loss formulation, and training pipeline.

Scratch Initialization:
The model is randomly initialized and trained using the combined CTC and triplet loss from the start of training. Under this regime, transcription learning and representation learning are optimized jointly throughout the entire training process.

A variant of this setting was also explored in which the contrastive objective was introduced only after an initial CTC-only warmup phase, allowing the encoder to stabilize before contrastive regularization was applied. This approach did not outperform full joint training from scratch for the target speaker and is therefore reported only in the appendix (Appendix 9.1).

Fine-tuning Initialization:
The model is initialized from the best-performing baseline checkpoint and further fine-tuned using the combined CTC and triplet loss objective.

3.4.2 Dynamic Phoneme Embedding Extraction

To compute the triplet loss, a fixed-size embedding vector is extracted for each target phoneme from the continuous frame-level DNN output. Since CTC-based models produce frame-level probability distributions without explicit phoneme boundaries, Viterbi forced alignment is used to identify the relevant frame segments (see Chapter 2). The alignment is implemented using the *torchaudio* library [67].

Given the ground-truth phoneme sequence and the model's frame-level CTC log-probabilities, the Viterbi algorithm computes the most likely alignment between frames and phoneme labels. This assigns each phoneme a continuous block of audio frames.

The alignment is recomputed at every training step. As model parameters are updated during training, the underlying frame-level representations change, making static alignment boundaries from an earlier model inconsistent with the current state of the network. Dynamic recomputation ensures that phoneme boundaries remain aligned with the evolving representation space.

Once the start and end frames of a target phoneme are identified, the corresponding DNN outputs are mean-pooled over time to obtain a single embedding vector. Mean pooling is used as it produces a stable summary of the phoneme's acoustic characteristics across its duration. This embedding is then passed to the projection head for triplet loss computation.

3.4.3 Triplet Loss

A margin-based triplet loss encourages separation between phoneme representations:

$$L_{\text{triplet}} = \max(0, d(a,p) - d(a,n) + m) \tag{3.7}$$

where a is the anchor embedding, p is the positive embedding, n is the negative embedding, d is the cosine distance, and m is the margin. The loss penalizes cases where the anchor is not closer to the positive than to the negative by at least m.

Cosine distance is used because it is insensitive to vector magnitude, measuring only the angle between two vectors. Two instances of the same phoneme spoken at different volumes or durations are therefore treated as similar as long as their directional features are consistent.

3.4.4 Projection Head

Instead of applying the triplet loss directly to the DNN outputs used for CTC, a separate projection head maps these representations into a dedicated contrastive embedding space. This design is motivated by findings from SimCLR [58], which show that applying contrastive objectives directly to task-specific representations can distort them and degrade downstream performance. The projection head separates the representations used for CTC decoding from those used for the contrastive objective, reducing interference with the primary recognition task.

The projection head is implemented as a two-layer Multi-Layer Perceptron (MLP):

$$z = \text{Linear}(1024, 256) \rightarrow \text{ReLU} \rightarrow \text{Linear}(256, 128) \rightarrow L_2\text{-normalize}$$

L_2 normalization constrains all embeddings to lie on the unit hypersphere, making cosine similarity equivalent to a scaled Euclidean distance and improving the stability of triplet loss optimization. The projection head is randomly initialized and trained jointly with the contrastive objective. At inference time, the projection head is discarded and only the DNN outputs are used for CTC decoding.

3.4.5 Joint Optimization

Training with only a contrastive objective leads to representation collapse [58], where all embeddings converge to a single point regardless of input, causing the model to lose mean-

ingful acoustic-phonetic structure. Training is therefore formulated as a joint optimization:

$$L_{total} = \alpha L_{triplet} + (1 - \alpha) L_{CTC} \tag{3.8}$$

The CTC loss L_{CTC} ensures that the model maintains accurate phoneme-level transcription, while the triplet loss $L_{triplet}$ encourages separation between confusable phoneme embeddings. The hyperparameter $\alpha \in [0, 1]$ controls the trade-off between the two objectives. Setting $\alpha = 0$ corresponds to the baseline CTC model, while $\alpha = 1$ removes the transcription objective entirely.

3.4.6 Sequential Branch Processing

During contrastive training, the Whisper encoder is unfrozen and optimized jointly with the DNN. Processing the anchor, positive, and negative audio streams in parallel would exceed the available GPU memory, as it requires three simultaneous forward passes through the encoder. To address this constraint, the three branches are processed sequentially. A forward pass and forced alignment are computed independently for each branch, after which the resulting embeddings are combined to compute the triplet loss. Gradients from all three branches are accumulated before a single backward pass is performed.

4

Experimental Setup

This chapter details the empirical implementation of the methodology described in Chapter 3, specifying the dataset, preprocessing pipeline, training configurations, and evaluation metrics used across all experiments.

4.1 Dataset: The DysOne Corpus

All experiments use the DysOne corpus [68], a speech dataset recorded in collaboration with a single dysarthric speaker. Although the full corpus contains both native Dutch and non-native English speech, this thesis focuses exclusively on the Dutch portion. The English data are excluded so that the analysis remains focused on dysarthric speech patterns rather than those introduced by speaking a non-native language.

4.1.1 Speaker Profile

Because the characteristics of dysarthria vary substantially across individuals, a detailed clinical profile is critical for framing the acoustic evaluation. The participant is a 35-year-old native Dutch male who acquired dysarthria following brain damage from a high-impact trauma. The speaker self-identifies at Stage 3 of the disorder, indicating a noticeable reduction in overall speech intelligibility. The dataset was recorded across seven sessions to capture the natural day-to-day variability in the speaker's speech, which is common in dysarthria due to fatigue and fluctuating motor control.

4.1.2 Recording Environment and Specifications

All recordings were made in an acoustically isolated booth at 44.1 kHz using an Audio-Technica AT2020USB+ microphone positioned 20-30 cm from the speaker's mouth, with a pop filter to reduce plosive distortion. Although DysOne was originally collected as an audiovisual dataset, only the audio modality is used in this work. Because the dataset provides only word-level transcriptions, phoneme-level targets are derived during preprocessing, as described in Section 4.2.

4.1.3 Corpus Composition and Data Splits

The Dutch portion consists of read and spontaneous speech. The read speech was collected using two approaches. First, prompts were randomly selected from the Corpus Gesproken Nederlands (CGN) to provide a standardized set of utterances, with a maximum length of 15 words to limit fatigue. Second, additional prompts were created by the participant based on personal communication needs, primarily vocabulary from statistics lectures. The spontaneous speech consists of unscripted responses to open-ended questions.

The dataset was split into training, validation, and testing sets using a 70/10/20 split. A summary of the corpus composition by speech type and split is provided in Table 4.1.

Table 4.1: Utterance (Utts) counts and total speech durations for the training, validation, and testing splits of the DysOne corpus (Dutch partition).

Split	Read	Spontaneous	Total Utts	Duration
Training	1,021	141	1,162	138.0 min (2.3 hrs)
Validation	145	20	165	18.0 min (0.3 hrs)
Testing	290	40	330	38.9 min (0.6 hrs)
Total	**1,456**	**201**	**1,657**	**194.9 min (3.3 hrs)**

4.2 Data Preprocessing

This section describes the two preprocessing steps applied to the DysOne corpus prior to training: grapheme-to-phoneme conversion and vocabulary construction.

4.2.1 Phonemic Target Generation (Grapheme-to-Phoneme)

The DysOne corpus provides only word-level transcriptions. To obtain phoneme-level training targets, the transcriptions were converted into phoneme sequences using a grapheme-to-phoneme (G2P) system.

The conversion from word-level text to phoneme sequences was performed using the open-source speech synthesizer *espeak-ng* (v1.51) with its Dutch language ruleset. The output was generated in the International Phonetic Alphabet (IPA).

An important consideration during tokenization is the handling of multi-character phonemes. Dutch contains diphthongs (e.g., /ɛɪ/) and long vowels (e.g., /eː/), which represent single phonemic units despite consisting of multiple symbols. A naive character-level tokenization would incorrectly split these units (e.g., separating /ɛɪ/ into /ɛ/ and /ɪ/). To avoid this, *espeak-ng* was run with the separator flag (*–sep=_*), which inserts explicit delimiters between phonemes according to the synthesizer's internal rules.

After removing non-phonemic markers such as stress indicators and formatting artifacts, the resulting strings were split on these delimiters, preserving multi-character phonemes as single tokens. For example, the utterance "ergens schreeuwt een vogel" is converted to:

/ɛ r ɣ ə n s x r eʊ t ə n v oː ɣ ə l/

4.2.2 Phoneme Vocabulary Construction

To obtain a fixed output space for the classifier, a phoneme vocabulary was defined prior to training. Rather than adopting a predefined Dutch phoneme inventory, the vocabulary was constructed from all unique phoneme tokens occurring in the training data. This ensures that the vocabulary matches the symbol set produced by *espeak-ng*, including symbols that may occur in loanwords or less frequent pronunciations.

The resulting lexicon consisted of 49 distinct phoneme tokens. For the CTC loss function, an additional blank token (ϵ) was added, resulting in a final vocabulary size of $|V| = 50$. The blank token models silence and transitional acoustic states between phonemes. The full vocabulary is provided in Appendix 9.2.

4.3 Baseline Training Configuration

The baseline model was trained using the *whisper-small* encoder ($d_{\text{encoder}} = 768$), selected for its moderate model size relative to the limited training data (1,162 utterances, approximately 2.3 hours). The implementation follows the SpeechBrain Whisper CTC recipe [65], with hyperparameters tuned empirically for this dataset. The final configuration is summarized in Table 4.2.

Table 4.2: Baseline training hyperparameters.

Parameter	Value	Description
Encoder	*whisper-small*	$d_{\text{encoder}} = 768$
DNN Hidden Size (d_{hidden})	1024	Neurons per layer, 3 layers
Vocabulary ($\|V\|$)	50	49 Dutch IPA phonemes + CTC blank
Dropout (p)	0.3	Applied across all DNN layers
DNN LR (lr_{model})	8×10^{-4}	Base learning rate for classifier
Encoder LR (lr_{whisper})	1×10^{-5}	Constrained rate for Whisper encoder
Warmup	1000 steps	Whisper encoder frozen duration
Batch Size	16	Batch 8, gradient accumulation 2
Max Epochs	50	Absolute training cap

Training used separate optimizers for the encoder and the classification network. Each optimizer was paired with an independent NewBob learning rate scheduler with an improvement threshold of $\delta = 0.0025$. When the validation metric did not improve beyond this threshold, the learning rate was reduced by a factor of 0.5 for the classification network and 0.75 for the encoder. Early stopping was applied based on the validation PER. The final model parameters were selected from the epoch with the lowest validation PER.

4.4 Data Augmentation Strategy

Given the limited training data (2.3 hours), data augmentation was applied to increase acoustic variability during training using the SpeechBrain augmentation pipeline [65]. Only speed perturbation was used, randomly resampling audio at 95%, 100%, or 105% of the original speed. This introduces variation in speaking rate while preserving the spectral

characteristics of the signal. Spectral and noise-based augmentations were not applied, as this work focuses on dysarthric speech, where acoustic and phonetic properties already exhibit high variability across utterances.

4.5 Contrastive Learning Configuration

The fine-tuning condition was initialized from the best-performing baseline checkpoint, with all optimizer states reset prior to training. The scratch condition used random initialization while keeping all remaining hyperparameters unchanged. All hyperparameters were identical to those of the baseline configuration, with the exception of the DNN learning rate, which was manually reduced from 8×10^{-4} to 1×10^{-4} to prevent large gradient updates from disrupting the representations learned during baseline training.

The weight of the contrastive loss was set to $\alpha = 0.2$ and the triplet margin to $m = 0.3$. These values were selected empirically and not systematically tuned.

For all contrastive configurations, three negative samples per anchor were used where available. All fixed hyperparameters are summarized in Table 4.3.

Table 4.3: Fixed hyperparameters for the contrastive training phase, applied to both initialization conditions. The fine-tuning condition uses the baseline checkpoint; the scratch condition uses random initialization.

Parameter	Value	Description
Fine-tuning Init.	Baseline Checkpoint	Best baseline validation PER
Scratch Init.	Random	Randomly initialized weights
DNN LR (lr_{model})	1×10^{-4}	Reduced to prevent catastrophic forgetting
Encoder LR (lr_{whisper})	1×10^{-5}	Stable representation learning
Contrastive Weight (α)	0.2	Balance between CTC and triplet loss
Triplet Margin (m)	0.3	Minimum cosine distance threshold
Triplet Batch Size	2	6 audio files per step
Min. Confusion Threshold	5	Min. substitutions for pair retention

4.6 Software and Hardware Infrastructure

All experiments were implemented in Python 3.11 using PyTorch (v2.6.0) [69], building on the SpeechBrain Whisper CTC recipe [65]. Hyperparameters and training configurations were managed via SpeechBrain's YAML configuration system. The pre-trained Whisper encoder was loaded through the HuggingFace Transformers library. Phoneme sequences were generated using *espeak-ng* (v1.51), phonetic distances were computed using the Panphon library, and forced alignment was performed using *torchaudio* (v2.6.0) [67]. All experiments were run on the Delft AI Cluster (DAIC).

4.7 Evaluation Metrics

Model performance is evaluated using Phoneme Error Rate (PER) as the primary metric, supplemented by per-phoneme error analysis and pairwise bootstrap significance testing.

4.7.1 Phoneme Error Rate

The primary evaluation metric is Phoneme Error Rate (PER), computed using the Levenshtein edit distance between the predicted and reference phoneme sequences. A substitution occurs when a reference phoneme is replaced by a different phoneme, a deletion when a reference phoneme is omitted from the hypothesis, and an insertion when the hypothesis contains a phoneme with no corresponding reference phoneme. PER is defined as:

$$\text{PER} = \frac{S + D + I}{N} \times 100\% \tag{4.1}$$

where S, D, and I denote the numbers of substitutions, deletions, and insertions, respectively, and N is the total number of reference phonemes. PER is preferred over Word Error Rate (WER) for this task because word-level evaluation can obscure pronunciation errors through lexical and contextual constraints, as discussed in Section 2.2.4.

4.7.2 Per-Phoneme Error Analysis

While aggregate PER captures overall transcription accuracy, it does not indicate whether improvements are concentrated on the phoneme boundaries targeted by the contrastive objective. A system may achieve overall gains through small, distributed improvements across many phonemes without substantially affecting its targeted confusion pairs. To address this, per-phoneme PER is computed for each phoneme class as the error rate relative to the number of reference occurrences of that phoneme:

$$\text{PER}_{ph} = \frac{S_{ph} + D_{ph}}{N_{ph}} \times 100\% \tag{4.2}$$

where S_{ph}, D_{ph}, and N_{ph} denote substitutions, deletions, and reference counts for phoneme class ph respectively. Insertions are excluded because in the Levenshtein alignment they have no corresponding reference phoneme, and therefore cannot be attributed to any specific phoneme class. Per-phoneme PER thus measures how often the model fails to correctly produce a given reference phoneme, either by substituting it with another or deleting it entirely. Per-phoneme analysis is reported for the 44 of 49 vocabulary phonemes present in the test set reference transcriptions.

4.7.3 Statistical Significance Testing

Pairwise bootstrap significance tests are used for all system comparisons. Utterances are resampled with replacement over $n = 10{,}000$ iterations, and aggregate PER is computed for each resample by pooling all phonemes across the resampled utterances. This avoids bias introduced by averaging per-utterance PER values, where utterances of different lengths contribute equally regardless of token count.

The bootstrap distribution is centered under the null hypothesis of no difference between systems. The p-value is the proportion of null bootstrap samples that produce a

difference greater than or equal to the observed difference (two-sided test). Confidence intervals are computed from the unshifted bootstrap distribution using the 2.5th and 97.5th percentiles. Significance thresholds are $p < 0.05$ (*), $p < 0.01$ (**), and $p < 0.001$ (***).

4.8 Experimental Conditions

Two initialization strategies are evaluated in combination with four negative sampling strategies, yielding a total of eight contrastive systems. Table 4.4 summarizes all experimental conditions. The zero-shot and baseline systems serve as reference points.

Table 4.4: Overview of all evaluated systems. Contrastive models are evaluated under two initialization regimes: fine-tuning (FT) from a pretrained CTC baseline and scratch initialization.

System	Init	Description
Zero-shot Whisper	—	No fine-tuning; applied directly to dysarthric speech.
CTC Baseline	—	Whisper encoder + DNN, CTC objective only.
Random	FT / Scratch	Joint CTC + triplet loss, random negatives.
Phonological	FT / Scratch	Joint CTC + triplet loss, negatives by minimum Panphon feature distance.
Empirical Training Set	FT / Scratch	Joint CTC + triplet loss, negatives from training-set confusion matrix.
Empirical Cross-Validation	FT / Scratch	Joint CTC + triplet loss, negatives from cross-validated confusion matrix.

5

Results

This chapter presents the experimental findings across all systems.

5.1 Baseline Performance

Training was configured for a maximum of 50 epochs, with early stopping based on validation PER. The best validation performance was achieved at epoch 31. This model is referred to as the baseline.

The zero-shot model prior to fine-tuning achieved a PER of 982.7%, rendering it effectively unusable for meaningful ASR tasks in our use case. This result highlights the necessity of per-speaker fine-tuning in a dysarthric speech context. After fine-tuning, PER decreased to 35.4%, demonstrating substantial adaptation to the target domain.

A notable shift in the error profile is observed between the zero-shot and fine-tuned settings. The zero-shot model is heavily dominated by insertion errors, whereas after fine-tuning the error distribution becomes more balanced, with substitutions constituting the largest proportion of errors (47%). This indicates that while fine-tuning substantially reduces spurious insertions, residual errors are primarily due to phoneme-level confusions, which are expected to be better addressed through contrastive learning.

Table 5.1: Zero-shot and fine-tuned baseline performance on the test set (330 utterances, 10,508 reference phoneme tokens).

			Error Breakdown		
System	**PER (%)**	**Total Errors**	**Ins**	**Del**	**Sub**
Zero-shot	982.7	105,812	96,595	5	9,212
Baseline	35.4	3,722	713	1,246	1,763

5.2 Confusion Matrices

Table 5.2 lists the top-20 substitution pairs from the Empirical Training Set and Empirical Cross-Validation confusion matrices.

Table 5.2: Top-20 substitution pairs from Empirical confusion matrices (Training vs Cross-Validation).

Training Set

Rank	Target	Prediction	Count
1	/d/	/t/	38
2	/aː/	/ə/	17
3	/ɛ/	/ə/	15
4	/r/	/n/	15
5	/eː/	/ə/	12
6	/f/	/p/	7
7	/f/	/v/	7
8	/ɪ/	/ə/	6
9	/ə/	/ɛɪ/	5
10	/n/	/ə/	5
11	/ɔ/	/ə/	5
12	/ɣ/	/k/	5
13	/ɛ/	/eː/	4
14	/n/	/r/	4
15	/ɪ/	/ɛ/	4
16	/aː/	/ɑ/	4
17	/ɔ/	/ɔː/	4
18	/m/	/n/	4
19	/ɵ/	/ə/	4
20	/i/	/ə/	4

Cross-Validation

Rank	Target	Prediction	Count
1	/d/	/t/	55
2	/aː/	/ə/	37
3	/eː/	/ə/	31
4	/eː/	/ɛ/	25
5	/ɛ/	/ə/	20
6	/ɔ/	/ə/	16
7	/r/	/n/	14
8	/ɑ/	/ə/	10
9	/f/	/p/	9
10	/a/	/ɵ/	9
11	/eː/	/ɵ/	8
12	/aː/	/ɑ/	8
13	/ɪ/	/ə/	8
14	/ə/	/ɛɪ/	8
15	/ɔ/	/ɔː/	7
16	/ɛɪ/	/ɛ/	7
17	/ɛɪ/	/ə/	7
18	/eʊ/	/eː/	7
19	/ə/	/ɛ/	7
20	/ɵ/	/a/	7

The substitution pairs in both matrices are not random: they cluster into a small set of recurring error types. The most frequent single error in both matrices is /d/ → /t/, a voicing confusion where a sound produced with vocal fold vibration is replaced by its unvoiced counterpart. A broader voicing pattern also appears at rank 7 of training-set derived confusions, where /f/ → /v/ shows the reverse direction. The most pervasive pattern overall is vowel centralization: multiple vowels that are produced with the tongue in a peripheral position in the mouth (/aː/, /ɛ/, /eː/, /ɪ/, /ɔ/, /ɑ/, /ɛɪ/) are substituted by the neutral central vowel schwa /ə/. This is consistent with reduced articulatory range, where the speaker's tongue does not reach the extremes of the vowel space. A third pattern involves vowel length: /ɛ/ → /eː/, /aː/ → /ɑ/, and /ɔ/ → /ɔː/ reflect confusion between short and long versions of the same vowel. Finally, a smaller set of manner errors appear in both tables, including /r/ → /n/ and /f/ → /p/. These patterns are consistent with articulatory error profiles reported for dysarthric speakers in prior work [70].

The two matrices agree on the dominant error types but differ in scope. Counts are higher in the cross-validated matrix because its predictions are made on held-out utterances:

errors that the fully trained model no longer makes on its own training data remain visible, capturing a broader range of confusions.

5.3 Triplet Statistics

Table 5.3 summarizes triplet counts and phoneme-pair coverage for each negative sampling strategy. Random and Phonological cover the full phoneme inventory, while the Empirical strategies target only the pairs the baseline model actually confuses. Empirical Cross-Validation yields substantially broader coverage than Empirical Training Set, reflecting its use of held-out predictions to estimate the error distribution.

Table 5.3: Triplet summary. Coverage refers to the set of phoneme pairs used as anchor-negative combinations.

System	Total Triplets	Unique Pairs	Coverage
Random	112,011	131	Full inventory
Phonological	112,011	131	Full inventory
Empirical Training Set	18,417	12	Confusion-based subset
Empirical Cross-Validation	35,626	31	Confusion-based subset

Full per-pair triplet counts for the Empirical strategies are provided in Appendix 9.3.

5.4 Contrastive Learning Results

Two initialization strategies were evaluated: contrastive fine-tuning from the trained baseline checkpoint, and contrastive training from scratch with the contrastive objective active from epoch 0. Table 5.4 reports PER, error breakdown, and significance against the baseline for all systems under both conditions.

5.4.1 Performance Against Baseline

All contrastive systems significantly outperform the baseline across both initialization conditions, as shown in the **Sig.** column of Table 5.4. The only exception is Random under scratch initialization, which reaches significance only at $p = 0.012$ (*) rather than the $p < 0.001$ level achieved by every other system. This confirms that integrating a contrastive objective improves phoneme recognition regardless of initialization strategy or negative sampling method.

Under fine-tuning initialization, absolute PER reductions range from 2.0 percentage points (Random) to 2.4 percentage points (Empirical Training Set), with substitution errors decreasing by 75 to 111 instances across systems. Under scratch initialization the spread is wider: Phonological and Empirical Cross-Validation achieve the largest reductions of 3.8 percentage points each, corresponding to a 10.7% relative PER reduction, with substitution reductions of 191 and 190 instances respectively, while Empirical Training Set and Random show smaller gains.

Table 5.4: PER and error counts for all contrastive systems under both initialization conditions (330 utterances, 10,508 reference phoneme tokens). Δ reports absolute PER reduction versus the baseline. The **Sig.** column reports pairwise bootstrap significance against the baseline ($n = 10{,}000$ resamplings). Significance thresholds: $^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$.

						Error Breakdown		
Init	**System**	**PER (%)**	**Δ**	**Sig.**	**Total**	**Ins**	**Del**	**Sub**
—	Baseline	35.4	—	—	3,722	713	1,246	1,763
Fine-tune	Random	33.4	−2.0	***	3,512	640	1,198	1,674
	Phonological	33.1	−2.3	***	3,480	633	1,159	1,688
	Empirical Train	33.0	−2.4	***	3,467	680	1,132	1,655
	Empirical CV	33.2	−2.2	***	3,491	623	1,216	1,652
Scratch	Random	34.5	−0.9	*	3,625	753	1,180	1,692
	Phonological	31.6	−3.8	***	3,319	654	1,093	1,572
	Empirical Train	33.3	−2.1	***	3,499	631	1,181	1,687
	Empirical CV	31.7	−3.8	***	3,326	702	1,051	1,573

5.4.2 Effect of Negative Sampling Strategy

The two initialization conditions differ sharply in whether the choice of negative sampling strategy matters. Under fine-tuning initialization, pairwise bootstrap comparisons among the four contrastive systems reveal no significant differences between any pair (all $p > 0.13$), indicating that the choice of negative sampling strategy does not meaningfully affect performance when initializing from the trained baseline.

Under scratch initialization, by contrast, clear differences emerge. Table 5.5 reports pairwise comparisons for this condition. Phonological and Empirical Cross-Validation are not significantly different from each other ($p = 0.869$), while both significantly outperform Empirical Training Set and Random ($p < 0.001$). Empirical Training Set in turn significantly outperforms Random ($p < 0.001$). The observed ranking is:

$$\text{Phonological} \approx \text{Empirical Cross-Validation} > \text{Empirical Training Set} > \text{Random}$$

where $\approx$ denotes no statistically significant difference.

5.4.3 Comparison Between Initialization Conditions

For Phonological and Empirical Cross-Validation, scratch initialization yields lower PER than fine-tuning, and these differences are statistically significant ($p < 0.001$). For Random, fine-tuning outperforms scratch ($p < 0.01$). For Empirical Training Set, the difference between conditions is not statistically significant ($p = 0.397$). Results are summarized in Table 5.6.

Table 5.5: Pairwise bootstrap significance tests between contrastive systems under scratch initialization ($n = 10{,}000$ resamplings). Δ = difference in PER (B−A) in percentage points. Significance thresholds: $^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$, ns = not significant.

System A	System B	Δ (pp)	95% CI	Sig.
Random	Empirical Train	−1.2	[−1.9, −0.5]	***
Random	Empirical CV	−2.9	[−3.6, −2.1]	***
Random	Phonological	−2.9	[−3.7, −2.1]	***
Empirical Train	Empirical CV	−1.7	[−2.3, −1.0]	***
Empirical Train	Phonological	−1.7	[−2.4, −1.0]	***
Empirical CV	Phonological	+0.1	[−0.7, +0.8]	ns

Table 5.6: Bootstrap significance tests comparing initialization conditions per strategy ($n = 10{,}000$ resamplings). Δ = PER difference (scratch − fine-tune) in percentage points; negative values favor scratch. Significance thresholds: $^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$, ns = not significant.

Strategy	Δ (pp)	95% CI	Sig.
Random	+1.08	[+0.32, +1.81]	**
Phonological	−1.53	[−2.31, −0.74]	***
Empirical Train	+0.30	[−0.40, +1.03]	ns
Empirical CV	−1.57	[−2.39, −0.73]	***

5.5 Phoneme-Level Error Analysis

Figure 5.1 shows the number of phonemes improved or degraded relative to the baseline for each system under both initialization conditions. A 2 percentage point (pp) threshold is chosen to distinguish substantive phoneme-level changes from minor fluctuations.

Under fine-tuning initialization, all systems show broadly similar improvement profiles. Under scratch initialization, Empirical Cross-Validation shows the highest number of improved phonemes, followed closely by Phonological, which also shows no degradations.

Figure 5.2 presents per-phoneme PER changes for the eight phonemes whose substitution pairs appear in both confusion matrices (Section 5.2). These represent the most consistently observed confusion boundaries and are the phonemes for which the Empirical strategies are most expected to show improvement. Per-phoneme PER changes for all phonemes are provided in Appendix 9.4.

Under fine-tuning initialization, most targeted phonemes improve across all systems with broadly similar magnitudes. Under scratch initialization, the pattern is more differentiated: /eː/ and /r/ show the largest reductions under Phonological and Empirical Cross-Validation. The most prominent degradation is /f/ under Random scratch initialization; Phonological and Empirical Cross-Validation improve it instead. The phonemes /aː/ and /ɔ/ improve more substantially under Phonological and Empirical Cross-Validation than under Empirical Training Set and Random under scratch initialization.

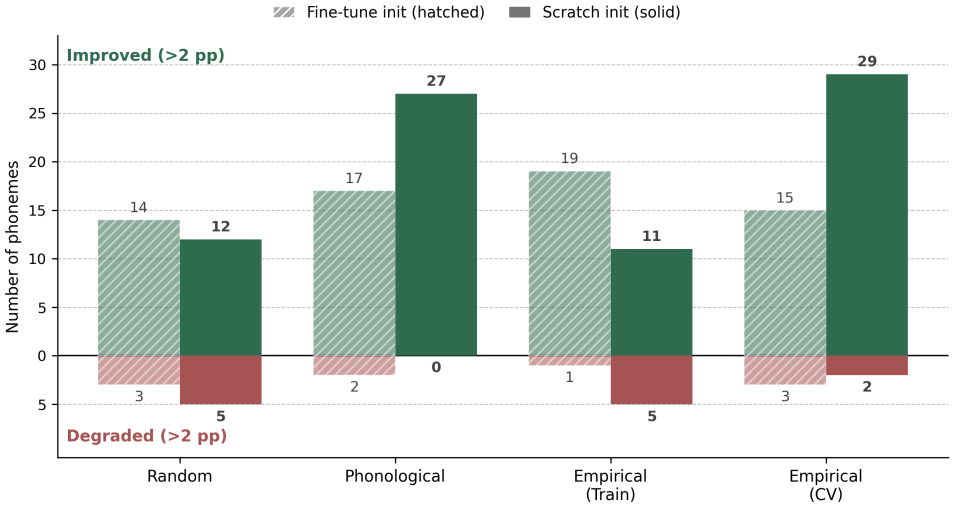

Figure 5.1: Number of phonemes improved or degraded by more than 2 pp relative to the baseline for each contrastive system. A 2 pp threshold is adopted as a conservative convention for distinguishing substantive phoneme-level changes from minor fluctuations. Hatched bars correspond to fine-tuning initialization and solid bars to scratch initialization; each system was evaluated over all 44 phonemes present in the test set.

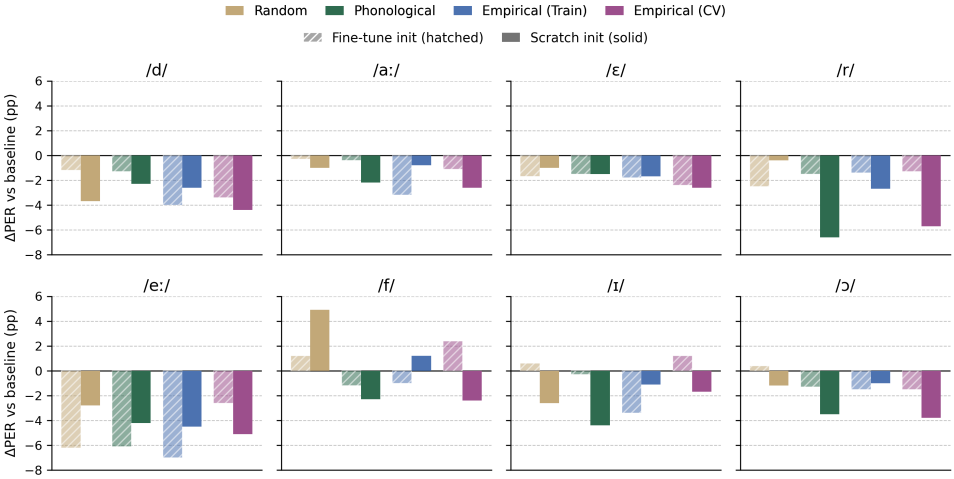

Figure 5.2: Per-phoneme PER change relative to the baseline (ΔPER, percentage points) for the eight phonemes whose substitution pairs appear in both confusion matrices. Each panel shows one phoneme, with bars grouped by negative sampling strategy; hatched bars correspond to fine-tuning initialization and solid bars to scratch initialization. Negative values indicate improvement.

6

Discussion

This chapter interprets the experimental findings presented in Chapter 5. Across the four research questions, the results consistently show that contrastive learning improves phoneme recognition in a speaker-dependent dysarthric setting, but that its effectiveness depends on how negative examples are constructed and how training is initialized. While all configurations outperform the CTC-only baseline, the largest gains are observed when negative sampling encodes meaningful phonological or confusion structure, particularly under joint training from scratch. Gains from more naive or training-set-derived strategies are less stable and depend more strongly on the presence of a pretrained representation.

6.1 RQ 1: To what extent does contrastive learning reduce phoneme error rate?

Both initialization conditions produce significant PER reductions relative to the CTC-only baseline, confirming that error-informed contrastive learning improves phoneme recognition in a speaker-dependent dysarthric setting. The baseline PER of 35.4% is consistent with reported ranges for speaker-dependent models on severe dysarthric speech [6, 10], and aligns with the speaker's Stage 3 intelligibility profile, suggesting the baseline reflects a typical severity level rather than an unusually weak starting point.

The strongest contrastive configurations achieve a relative PER reduction of approximately 10.7%. The most directly comparable work, DyPCL [17], reports a 22% relative WER reduction on the UASpeech corpus using multi-speaker training, dynamic curriculum-based negative sampling, and access to healthy speech references. This thesis operates under more constrained conditions without curriculum scheduling or external canonical speech supervision, while focusing on single-speaker adaptation. Despite these differences, the observed improvements indicate that phoneme-level contrastive objectives remain effective in low-resource, speaker-specific settings.

Improvements are primarily driven by reductions in substitution errors, which constitute the dominant error type in the baseline. This is consistent with the design of the contrastive objective, which explicitly increases separation between embeddings of frequently confused phoneme pairs, and with the confusion patterns identified in Section 5.2,

where voicing errors and vowel centralization account for the largest share of baseline errors. Prior analyses of dysarthric speech similarly show that errors concentrate along systematic articulatory dimensions [70], suggesting that the observed reductions reflect improved phoneme-level recognition rather than changes in sequence-level decoding behavior.

6.2 RQ 2: Does initialization strategy matter?

Initialization strategy affects performance, but its impact depends on the negative mining strategy, as will be discussed further in Section 6.3. For Phonological and Empirical Cross-Validation, scratch training achieves substantially lower PER than fine-tuning from a CTC-pretrained encoder. For Random, the pattern reverses: fine-tuning outperforms scratch training. For Empirical Training Set, the difference between conditions is not significant.

Both settings optimize a joint CTC and contrastive objective, differing only in whether training begins from a pretrained checkpoint or random initialization. A pretrained CTC initialization may already impose structure on the phoneme representation space, limiting how freely the contrastive objective can reshape phoneme boundaries. This is consistent with findings on representation rigidity in fine-tuned models, where pretrained representations resist restructuring even under new objectives [71]. When training begins from scratch, both objectives jointly shape representation learning from the start, which appears most beneficial when the contrastive signal is informative, as seen for Phonological and Empirical Cross-Validation. It is also worth noting that the contrastive loss weight $\alpha = 0.2$ was fixed across both initialization conditions; part of the observed difference may therefore reflect suboptimal objective balancing in the fine-tuning setting rather than a fundamental property of the initialization strategy (see Section 6.6).

Initialization also influences sensitivity to contrastive signal quality. Under fine-tuning-based initialization, no statistically significant differences are observed between strategies, suggesting that pretrained representations partially buffer weaker contrastive signals. Under scratch initialization, differences in negative quality translate more directly into performance gaps. Scratch training therefore provides a more sensitive setting for evaluating contrastive learning design choices.

These findings indicate that sequential CTC pretraining followed by contrastive fine-tuning is not universally optimal. When informative negatives are available, joint optimization from scratch yields better performance.

6.3 RQ 3: Does negative mining strategy matter?

The impact of negative mining strategy depends strongly on initialization, as established in Section 6.2. Under fine-tuning-based initialization, no significant differences are observed between strategies. Under scratch initialization, a clear ranking emerges: Phonological $\approx$ Empirical Cross-Validation $>$ Empirical Training Set $>$ Random.

Random negatives provide weak learning signals as they are not aligned with systematic phoneme confusability. Empirical Training Set negatives reflect a compressed view of the error space due to overconfident predictions on familiar utterances (see Section 6.4). Empirical Cross-Validation and Phonological provide more structured signals: the former

captures held-out confusion patterns, the latter encodes articulatory similarity. Both yield more informative contrastive signals, consistent with prior findings that hard negative mining is most effective when negatives encode meaningful structure [61].

The similar performance of Phonological and Empirical Cross-Validation is notable given their different origins. Empirical Cross-Validation reflects empirical speaker-specific confusions, whereas Phonological is derived from a population-level articulatory feature space [66]. Although their selected pairs only partially overlap, both methods produce negatives that reflect meaningful phonological structure, suggesting that what matters is not the exact identity of negatives but whether they encode genuine phonological difficulty.

This finding has broader implications. For speakers whose error patterns are broadly consistent with articulatory feature space, speaker-independent phonological priors may be sufficient substitutes for empirically derived confusion data. This is practically significant in clinical deployment settings where collecting sufficient speech data to estimate reliable confusion matrices may not be feasible.

6.4 RQ 4: Does confusion matrix construction method matter?

Empirical Cross-Validation outperforms Empirical Training Set, particularly under scratch initialization. The assumption underlying Empirical Training Set, that substitution patterns on training data reflect the most informative confusions, is not fully supported. Because the model has already been optimized on the same utterances, it produces more confident predictions on familiar data, and many genuine confusion patterns are suppressed in the resulting matrix. Cross-validated matrices, derived from held-out predictions, better approximate out-of-sample generalization errors and capture a broader range of substitution patterns, as reflected in their wider coverage (31 pairs versus 12 pairs from the Empirical Training Set matrix, reported in Section 5.3).

Phoneme-level analysis further supports this. The vowels /oː/ and /ɛɪ/ are absent from the Empirical Training Set matrix but improve substantially under Empirical Cross-Validation in the scratch setting, as they receive no targeted contrastive signal in the former.

6.5 Effectiveness of Personalized Contrastive Modeling

This work adopts a personalized approach in which the contrastive objective targets phoneme boundaries derived from an individual speaker's error patterns, contrasting with speaker-independent approaches that apply a single fixed model regardless of articulatory characteristics [6, 11].

Both confusion-based strategies improve significantly over the baseline, and per-phoneme analysis shows improvements concentrated at the intended confusability boundaries. At the same time, the equivalence between Phonological and Empirical Cross-Validation suggests that explicitly modeling speaker-specific errors does not always yield additional benefit over a strong articulatory prior. For this speaker, the dominant error structure is broadly consistent with general phonological dimensions such as voicing,

manner of articulation, and vowel centralization.

This does not contradict the motivation that dysarthric speech can exhibit personalized confusability patterns shaped by individual motor impairment [9, 18]. Rather, it suggests that for this speaker these patterns remain largely constrained within articulatory feature space. However, this conclusion is limited to a single-speaker setting and may not generalize to speakers whose errors are less phonologically structured. In such cases, speaker-specific confusion data may provide additional benefit. More broadly, the decision between phonological and empirical negative selection should be informed by the degree to which a speaker's error patterns deviate from standard articulatory structure, which could be estimated from a small pilot recording before committing to full confusion matrix construction.

The two approaches involve different practical trade-offs. Phonological sampling requires no pretrained model or confusion matrix estimation, making it applicable to new speakers, but produces approximately three times as many triplets as confusion-based strategies, resulting in higher training-time computational cost. Confusion-based strategies require additional preprocessing but produce smaller, more targeted triplet sets and more efficient training. Given comparable performance, the choice between them depends on whether preprocessing overhead or training-time resources are the limiting constraint.

6.6 Limitations

Single speaker. All experiments are conducted on a single Dutch dysarthric speaker. The effectiveness of the proposed pipeline cannot be established beyond this individual case, and further evaluation on speakers with different dysarthria profiles and severity levels is required to assess broader applicability.

Dataset size. The training data consists of approximately 2.3 hours of speech. This limits the statistical power of comparisons between strategies, particularly for overall PER. Larger datasets would enable more robust estimation of strategy differences, and would allow investigation of whether the Phonological and Empirical Cross-Validation equivalence holds with more data, and whether Empirical Training Set would benefit where model overconfidence is less of a problem.

Fixed hyperparameters. The contrastive loss weight, triplet margin, number of negatives per anchor, and maximum triplets per confusion pair were fixed across all conditions and not tuned per initialization regime or strategy. Reported differences may therefore reflect suboptimal objective balancing in some configurations. Additionally, phonemes serving as negatives for multiple anchor classes simultaneously, such as /n/ for both /r/ and /d/, may receive conflicting gradient updates that destabilize their representations. Constraining the number of anchors a phoneme can serve as a negative for could mitigate this effect.

Static anchor-negative pairings. While phoneme-level frame boundaries are recomputed dynamically during training via forced alignment, the anchor-negative pairings themselves are fixed prior to training based on a single confusion matrix estimate. As

the model's representations evolve, the paired negatives may no longer reflect the most confusable phonemes at later training stages. Periodically updating the confusion matrix and resampling triplets during training could provide a more accurate signal, though this would introduce additional computational overhead.

Representational analysis. Representational analysis was attempted to directly evaluate whether phoneme embeddings were more separated in the contrastive models than in the baseline. Cluster separation scores were near zero across all conditions, and projection head embeddings were unrecoverable due to the SpeechBrain checkpointer not registering the projection head module during saving. As a result, no conclusions could be drawn about the geometry of the learned embedding space, and PER serves as the sole indicator of improved phoneme discrimination. Future work should incorporate embedding extraction explicitly during training rather than relying on post-hoc checkpoint recovery.

6.7 Broader Implications

This thesis sits at the intersection of several developments that currently shape Data Science and Artificial Intelligence (AI) Technology. The first is the growing role of foundation models. Although new architectures and models continue to appear at a rapid pace, much of applied AI research now builds on large pretrained models such as Whisper rather than training systems from scratch, and an increasingly important question is how to adapt such models to settings they were not trained for. This thesis is an instance of that question: it adapts a foundation model, trained on hundreds of thousands of hours of typical speech, to a single speaker with 2.3 hours of atypical speech.

Within this landscape, this research extends two areas in particular. The first is contrastive learning for dysarthric speech. Despite its success in computer vision and self-supervised speech modeling, contrastive learning has seen relatively little application in DSR, with only a handful of studies exploring it to date [16, 17]. The results of this thesis add to that limited body of evidence, supporting the usability of phoneme-level contrastive objectives in low-resource dysarthric settings and showing that the choice of negative examples is a meaningful design dimension rather than an implementation detail.

The second, and arguably more distinctive, contribution concerns the level at which adaptation takes place. Personalization is a broad trend across AI, from medicine to recommendation systems, but research that fine-tunes a model to a single individual, using that individual's own error patterns as a training signal, remains rare. Most adaptation work in speech recognition targets groups of speakers or severity categories rather than one person. This thesis demonstrates that such concentrated, individual-level adaptation is both feasible and effective, which is relevant beyond speech: the same principle could apply in any domain where errors are systematic and user-specific.

Finally, this work connects to the discussion on fairness and inclusivity in AI. The performance gap between typical and atypical speech is a concrete example of how systems trained on majority data underserve minority populations. At the same time, the single-speaker scope of this thesis illustrates a persistent tension in the field: the populations that most need adapted models are often those for whom large-scale data collection is least feasible.

7

Conclusion

This thesis investigated whether personalized error-informed contrastive learning can improve phoneme recognition for a single Dutch dysarthric speaker. A joint CTC and triplet loss objective was applied to a Whisper-based encoder-DNN-CTC model, comparing four negative sampling strategies: Random, Phonological, Empirical Training Set, and Empirical Cross-Validation, under both fine-tuning and scratch initialization conditions. All contrastive configurations significantly outperform the CTC-only baseline, with the strongest systems, Phonological and Empirical Cross-Validation under scratch initialization, achieving up to a 10.7% relative reduction in phoneme error rate, demonstrating that contrastive representation learning improves phoneme-level discrimination in a low-resource, speaker-dependent setting.

The impact of both initialization and negative sampling strategy is conditional. Under fine-tuning-based initialization, performance differences between strategies are minimal, suggesting that pretrained representations reduce sensitivity to the choice of negative samples. Under scratch initialization, clearer differences emerge: Phonological and Empirical Cross-Validation negatives outperform both Empirical Training Set and Random sampling, indicating that negative sample quality becomes more important when no pretrained phonetic structure is available.

For this speaker, Phonological negatives perform competitively with Empirical Cross-Validation negatives, despite not being derived from speaker-specific error statistics. The two approaches do, however, differ in where their costs fall: Phonological sampling requires no preprocessing but incurs higher training-time computational cost due to its larger triplet set, whereas Empirical Cross-Validation requires additional preprocessing steps but enables more targeted and efficient training. Under scratch initialization, Empirical Cross-Validation tends to outperform Empirical Training Set, suggesting that held-out predictions capture a broader and more reliable estimate of phoneme confusability than in-sample predictions, though differences between the two empirical strategies are not statistically significant in all conditions.

These results demonstrate that personalized contrastive learning is effective for low-resource dysarthric speech recognition, and that the choice of negative sampling strategy matters when no pretrained phonetic structure constrains the representation space. The

central practical implication is not that speaker-specific confusion data is strictly necessary, but that negative examples must encode genuine phonological difficulty to produce a meaningful training signal. For new speakers where confusion matrix construction is feasible, cross-validation-based empirical sampling offers targeted and efficient training. Where it is not, phonologically motivated sampling provides a competitive alternative with no speaker-specific preprocessing required. Whether these findings generalize beyond a single speaker remains an open question, and multi-speaker evaluation is a necessary next step before drawing broader conclusions about the framework's clinical applicability.

7.1 Future Work

Based on this thesis, several directions for future work are identified across three angles: evaluation scope, contrastive learning design, and training configuration choices.

Evaluation scope

This thesis evaluates a single Dutch dysarthric speaker at Stage 3 intelligibility. A key open question is whether the proposed speaker-specific contrastive learning framework generalizes beyond this setting, and how its behavior changes across different dysarthria profiles and data conditions.

- **Multi-speaker evaluation.** Extending the evaluation to speakers with varying severity levels and impairment characteristics would help clarify whether the personalized confusion-based pipeline leads to consistent improvements across individuals, and whether the observed equivalence between phonological and confusion-based negatives still holds when a speaker's error patterns are less phonologically structured.
- **Cross-dataset and cross-lingual evaluation.** Applying the method to other dysarthric speech corpora and languages would further test robustness to differences in phonetic inventory, recording conditions, and dataset composition.

Contrastive learning design

These directions focus on how contrastive signals are defined and updated.

- **Healthy speech as reference positives.** In the present framework, both anchor and positive examples are drawn from the dysarthric speaker's own utterances. An alternative would be to use canonical healthy speech pronunciations of the same phoneme as positives, providing a stable reference target for phonemes where the speaker's own productions vary substantially across utterances. The framework would remain personalized, as the negative mining strategy continues to be derived from the speaker's specific confusion patterns.
- **Phoneme-level loss weighting.** In the present framework, all triplets contribute equally to the contrastive loss regardless of how frequently the corresponding phoneme pair is confused. A possible extension would be to weight each triplet's contribution by the confusion frequency of its anchor-negative pair, so that more persistently confused boundaries (for example, in this case /d/ and /t/), contribute a stronger learning signal than rare confusions.

- **Multiple positives per anchor.** The present framework uses a single positive example per anchor instance. Using multiple positive instances for each anchor could reduce sensitivity to individual utterance variability.
- **Curriculum-based negative scheduling.** This thesis evaluates each negative sampling strategy independently. An interesting extension would be to apply curriculum learning by beginning training with easier negatives such as Random or Phonological sampling, and gradually introducing confusion-based hard negatives as the model stabilizes into a more structured representation space.

Training configuration

These directions concern optimization and architectural choices that were held fixed across all conditions in this study.

- **Hyperparameter optimization.** The contrastive loss weight α, triplet margin, and number of negatives per anchor were fixed for all experiments. Under fine-tuning-based initialization, where representations are already structured, a lower α may reduce unnecessary interference with the CTC objective. For confusion-based strategies, which target a small number of phoneme pairs, increasing the number of negatives per anchor could allow denser coverage of the targeted confusion boundaries. Systematic tuning of these parameters could improve the balance between transcription accuracy and phoneme discrimination.
- **Alternative encoder architectures.** This thesis uses the Whisper encoder, which was pretrained on large-scale supervised multilingual speech recognition. Replacing Whisper with self-supervised encoders such as wav2vec 2.0 or HuBERT would help determine whether the trends observed are specific to Whisper's supervised pretraining regime, or reflect a more general property of the framework.

8

Use of Generative AI

During the preparation of this thesis, generative AI tools were used in the following ways. Claude (Anthropic) and ChatGPT (OpenAI) were used to support academic writing by improving the clarity and readability of drafted text, checking grammar and style, and assisting with LaTeX formatting of tables. Claude was additionally used for code support, in the form of debugging suggestions and generating plotting scripts used to visualize experimental results. AI was not used to generate research ideas or to interpret results. All AI-assisted content was critically reviewed, verified, and revised where necessary before inclusion.

9

Appendix

9.1 Scratch Initialization: 5-Epoch CTC Warmup

Table 9.1 reports results for the scratch initialization condition with a 5-epoch CTC warmup, in which five epochs of CTC-only training precede contrastive fine-tuning.

Table 9.1: PER and error breakdown for the scratch initialization with 5-epoch CTC warmup condition (330 utterances, 10,508 reference phoneme tokens).

			Error Breakdown		
System	**PER (%)**	**Total**	**Ins**	**Del**	**Sub**
Random	34.1	3,579	638	1,244	1,697
Phonological	31.6	3,315	639	1,099	1,577
Empirical (Train)	34.2	3,596	675	1,236	1,685
Empirical (CV)	31.9	3,351	642	1,101	1,608

9.2 Phoneme Vocabulary

Table 9.2 lists the full phoneme inventory used during training, along with each phoneme's index in the CTC output layer.

Table 9.2: Full phoneme vocabulary (49 phonemes + CTC blank token at index 0). † Phoneme does not occur in the test set reference transcriptions.

Index	Phoneme	Index	Phoneme	Index	Phoneme
0	⟨blank⟩	17	/p/	34	/ɔː/
1	/a/†	18	/r/	35	/ə/
2	/aː/	19	/s/	36	/ɛ/
3	/b/	20	/t/	37	/ɛɪ/
4	/d/	21	/tʃ/†	38	/ɛː/
5	/eʊ/	22	/u/	39	/ɣ/
6	/eː/	23	/v/	40	/ɪ/
7	/f/	24	/w/	41	/ɪː/
8	/h/	25	/x/	42	/ɲ/†
9	/i/	26	/y/	43	/ɵ/
10	/iː/†	27	/yʊ/	44	/ɾ/
11	/j/	28	/z/	45	/ʃ/
12	/k/	29	/ʌː/	46	/ʊ/
13	/l/	30	/ŋ/	47	/əʊ/
14	/m/	31	/ɔʏ/	48	/ʒ/
15	/n/	32	/ɑ/	49	/ð/†
16	/oː/	33	/ɔ/		

9.3 Empirical Triplet Counts

Tables 9.3 and 9.4 list all anchor-negative pairs and triplet counts for the Empirical Training Set and Empirical Cross-Validation strategies respectively.

Table 9.3: Triplet counts per phoneme pair for the Empirical Training Set strategy (18,417 total triplets, 12 unique pairs).

Anchor	Negative	Triplets
/ə/	/ɛɪ/	4,167
/n/	/ə/	3,694
/r/	/n/	2,500
/d/	/t/	1,559
/aː/	/ə/	1,193
/ɛ/	/ə/	1,173
/ɪ/	/ə/	1,114
/ɔ/	/ə/	831
/eː/	/ə/	790
/ɣ/	/k/	766
/f/	/v/	315
/f/	/p/	315
Total unique pairs		12
Total triplets		18,417

Table 9.4: Triplet counts per phoneme pair for the Empirical Cross-Validation strategy (35,626 total triplets, 31 unique pairs).

Anchor	Negative	Triplets
/ə/	/ɛ/	4,167
/ə/	/ɛɪ/	4,167
/r/	/ə/	2,500
/r/	/n/	2,500
/d/	/t/	1,559
/l/	/n/	1,431
/ɑ/	/oː/	1,420
/ɑ/	/ʌː/	1,420
/ɑ/	/ə/	1,420
/aː/	/ə/	1,193
/aː/	/ɑ/	1,193
/ɛ/	/eː/	1,173
/ɛ/	/ə/	1,173
/ɪ/	/ə/	1,114
/i/	/ə/	857
/ɔ/	/ə/	831
/ɔ/	/ɔː/	831
/ɔ/	/oː/	831
/eː/	/ʌː/	790
/eː/	/ə/	790
/eː/	/ɛ/	790
/z/	/s/	754
/oː/	/ə/	662
/ɛɪ/	/ɛ/	647
/ɛɪ/	/ə/	647
/f/	/p/	315
/ʌː/	/ɑ/	270
/əʊ/	/oː/	152
/w/	/y/	18
/eʊ/	/eː/	8
/tʃ/	/t/	3
Total unique pairs		31
Total triplets		35,626

9.4 Per-Phoneme PER: Full Results

Tables 9.5 and 9.6 report per-phoneme PER and absolute change relative to the baseline for all 44 phonemes present in the test set reference transcriptions, under fine-tuning and scratch initialization respectively.

Table 9.5: Per-phoneme PER (%) and Δ relative to baseline for all 44 phonemes under fine-tuning initialization. Phonemes sorted by baseline PER in descending order. Negative Δ values indicate improvement. Per-phoneme PER is computed as (substitutions + deletions) / reference count.

Ph.	Ref	Baseline	Δ Ran.	Δ Phon.	Δ Emp. Tr.	Δ Emp. CV
/ʌː/	8	100.0%	−12.5	−25.0	−12.5	−12.5
/ɛː/	1	100.0%	+0.0	+0.0	+0.0	+0.0
/ɾ/	2	100.0%	+0.0	+0.0	+0.0	+0.0
/yʊ/	2	100.0%	+0.0	+0.0	+0.0	+0.0
/eʊ/	3	100.0%	+0.0	+0.0	+0.0	+0.0
/ʒ/	4	100.0%	+0.0	+0.0	+0.0	+0.0
/ʃ/	9	77.8%	−33.3	−44.4	−66.7	−55.6
/w/	3	66.7%	−33.3	−33.3	−33.3	+0.0
/f/	82	63.4%	+1.2	−1.2	−1.2	+2.4
/ɪː/	35	62.9%	+0.0	+0.0	+0.0	−2.9
/y/	38	60.5%	−5.3	−7.9	+0.0	−5.3
/ɵ/	75	60.0%	+1.3	−1.3	+0.0	−4.0
/ŋ/	71	57.7%	−1.4	−2.8	−5.6	−4.2
/ɔʏ/	28	50.0%	−3.6	−3.6	−3.6	+0.0
/əʊ/	49	49.0%	−4.1	−6.1	−2.0	−2.0
/ɔː/	32	43.8%	+12.5	+6.2	+9.4	+9.4
/h/	263	42.2%	−3.0	−2.7	−3.0	−3.0
/j/	89	39.3%	−1.1	+1.1	−1.1	−1.1
/r/	699	38.8%	−2.4	−1.6	−1.3	−1.4
/oː/	174	36.8%	−5.7	−6.3	−8.6	−4.6
/ɛ/	332	36.7%	−1.8	+0.0	−2.1	−3.3
/p/	141	36.2%	+4.3	−2.1	+0.7	+1.4
/b/	155	34.2%	+1.3	−1.9	−0.6	−2.6
/n/	1036	33.2%	−0.2	−0.8	+0.0	+0.4
/eː/	230	33.0%	−6.1	−6.1	−7.0	−2.6
/l/	439	30.1%	−0.2	−2.1	−2.7	−1.4
/ɪ/	328	27.7%	+0.6	−0.3	−3.4	+1.2
/d/	442	26.9%	−1.4	−2.0	−4.1	−3.4
/ɑ/	407	26.3%	−1.0	−1.2	−1.5	−1.0
/k/	383	25.8%	−4.4	−5.0	−5.2	−3.9
/z/	219	25.6%	−1.4	−0.5	−0.5	−1.4
/ɔ/	231	25.5%	+0.4	−1.3	−1.7	−1.7
/ʊ/	204	25.0%	+0.0	−1.0	−2.0	−1.5
/i/	232	23.7%	−3.4	−5.2	−5.2	−1.7
/ə/	1215	23.0%	−2.7	−2.1	−2.3	−2.0
/aː/	364	22.5%	−0.3	−0.3	−3.3	−1.1
/u/	80	22.5%	+2.5	+3.8	−3.8	+5.0
/ɛɪ/	169	22.5%	+0.0	+0.0	+0.0	−0.6

Ph.	Ref	Baseline	Δ Ran.	Δ Phon.	Δ Emp. Tr.	Δ Emp. CV
/t/	856	21.6%	+0.4	−0.1	−0.6	+0.0
/m/	267	19.5%	−0.4	+1.9	−0.4	−0.4
/s/	434	18.9%	−2.8	−2.8	−1.8	−3.0
/x/	170	17.1%	−1.2	+0.6	+1.2	+1.2
/v/	292	16.4%	−1.0	+1.0	−2.7	−1.0
/ɣ/	215	16.3%	+0.5	−0.5	−1.4	+0.9

Table 9.6: Per-phoneme PER (%) and Δ relative to baseline for all 44 phonemes under scratch initialization. Phonemes sorted by baseline PER in descending order. Negative Δ values indicate improvement. Per-phoneme PER is computed as (substitutions + deletions) / reference count.

Ph.	Ref	Baseline	Δ Ran.	Δ Phon.	Δ Emp. Tr.	Δ Emp. CV
/ɑː/	8	100.0%	−12.5	−25.0	−25.0	−25.0
/ɛː/	1	100.0%	+0.0	+0.0	+0.0	+0.0
/ɾ/	2	100.0%	+0.0	+0.0	+0.0	+0.0
/yʊ/	2	100.0%	+0.0	+0.0	+0.0	+0.0
/eʊ/	3	100.0%	+0.0	+0.0	+0.0	+0.0
/ʒ/	4	100.0%	+0.0	+0.0	+0.0	+0.0
/ʃ/	9	77.8%	−11.1	−22.2	−22.2	−11.1
/w/	3	66.7%	−33.3	+0.0	−33.3	+0.0
/f/	82	63.4%	+4.9	−2.4	+1.2	−2.4
/ɪː/	35	62.9%	+0.0	−2.9	−8.6	−8.6
/y/	38	60.5%	+10.5	−10.5	+5.3	−2.6
/ɵ/	75	60.0%	−5.3	−8.0	−4.0	−9.3
/ŋ/	71	57.7%	−1.4	−8.5	−4.2	−2.8
/ɔʏ/	28	50.0%	−7.1	−7.1	−3.6	−7.1
/əʊ/	49	49.0%	+0.0	−4.1	+2.0	−8.2
/ɔː/	32	43.8%	+9.4	+0.0	+12.5	+9.4
/h/	263	42.2%	+0.4	−6.8	−0.4	−2.7
/j/	89	39.3%	+0.0	−1.1	+1.1	+0.0
/r/	699	38.8%	−0.4	−6.6	−2.9	−5.7
/oː/	174	36.8%	+1.1	−5.2	−1.7	−6.9
/ɛ/	332	36.7%	−0.9	−3.3	−0.9	−3.0
/p/	141	36.2%	+3.5	+0.7	+3.5	+2.1
/b/	155	34.2%	−5.2	−7.7	−5.2	−1.3
/n/	1036	33.2%	+1.4	−1.8	+0.9	−2.0
/eː/	230	33.0%	−3.0	−4.3	−4.8	−5.2
/l/	439	30.1%	−0.7	−3.0	+0.5	−3.4
/ɪ/	328	27.7%	−3.0	−4.6	−1.2	−1.8
/d/	442	26.9%	−3.8	−1.8	−1.8	−4.5
/ɑ/	407	26.3%	−3.2	−3.4	−2.2	−2.7
/k/	383	25.8%	+0.3	−6.5	−1.3	−4.4
/z/	219	25.6%	−0.5	−0.9	−1.4	−2.7
/ɔ/	231	25.5%	−1.3	−3.5	−0.4	−3.9
/ʊ/	204	25.0%	+0.5	+0.5	−1.0	−0.5
/ɪ/	232	23.7%	+1.7	−2.6	+1.7	−2.6

Ph.	Ref	Baseline	Δ Ran.	Δ Phon.	Δ Emp. Tr.	Δ Emp. CV
/ə/	1215	23.0%	−2.1	−3.2	−1.6	−2.9
/aː/	364	22.5%	−2.2	−2.2	−0.5	−3.0
/u/	80	22.5%	+1.2	−3.8	+0.0	−3.8
/ɛɪ/	169	22.5%	+4.1	+0.0	+2.4	−2.4
/t/	856	21.6%	−0.5	−1.5	−0.1	−1.9
/m/	267	19.5%	+1.5	−1.1	+0.7	+0.7
/s/	434	18.9%	−1.2	−3.2	−1.2	−0.2
/x/	170	17.1%	−1.8	−3.5	−1.2	−4.1
/v/	292	16.4%	+0.3	−2.1	−0.3	−2.1
/ɣ/	215	16.3%	+1.4	+0.5	+0.5	−2.8

References

[1] D. Wang, X. Wang, and S. Lv, "An overview of end-to-end automatic speech recognition," *Symmetry*, vol. 11, no. 8, 2019. [Online]. Available: https://www.mdpi.com/2073-8994/11/8/1018

[2] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International conference on machine learning*. PMLR, 2023, pp. 28 492–28 518.

[3] M. Benzeghiba, R. De Mori, O. Deroo, S. Dupont, T. Erbes, D. Jouvet, L. Fissore, P. Laface, A. Mertins, C. Ris, R. Rose, V. Tyagi, and C. Wellekens, "Automatic speech recognition and speech variability: A review," *Speech Communication*, vol. 49, no. 10, pp. 763–786, 2007, intrinsic Speech Variations. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167639307000404

[4] H. Christensen, S. Cunningham, C. Fox, P. Green, and T. Hain, "A comparative study of adaptive, automatic recognition of disordered speech," in *Interspeech 2012*, 2012, pp. 1776–1779.

[5] Y. Zhang, A. Herygers, T. Patel, Z. Yue, and O. Scharenborg, "Exploring data augmentation in bias mitigation against non-native-accented speech," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2023, pp. 1–8.

[6] J. R. Green, R. L. MacDonald, P.-P. Jiang, J. Cattiau, R. Heywood, R. Cave, K. Seaver, M. A. Ladewig, J. Tobin, M. P. Brenner *et al.*, "Automatic speech recognition of disordered speech: Personalized models outperforming human listeners on short phrases." in *Interspeech*, vol. 2021, 2021, pp. 4778–4782.

[7] K. Rosen and S. Yampolsky, "Automatic speech recognition and a review of its functioning with dysarthric speech," *Augmentative and Alternative Communication*, vol. 16, no. 1, pp. 48–60, 2000. [Online]. Available: https://doi.org/10.1080/07434610012331278904

[8] M. S. Hawley, S. P. Cunningham, P. D. Green, P. Enderby, R. Palmer, S. Sehgal, and P. O'Neill, "A voice-input voice-output communication aid for people with severe speech impairment," *IEEE Transactions on neural systems and rehabilitation engineering*, vol. 21, no. 1, pp. 23–31, 2012.

[9] J. R. Duffy *et al.*, *Motor speech disorders: Substrates, differential diagnosis, and management*. Elsevier Health Sciences, 2012.

[10] L. De Russis and F. Corno, "On the impact of dysarthric speech on contemporary asr cloud platforms," *Journal of Reliable Intelligent Environments*, vol. 5, no. 3, pp. 163–172, 2019.

[11] M. J. Kim, J. Yoo, and H. Kim, "Dysarthric speech recognition using dysarthria-severity-dependent and speaker-adaptive models," in *Proc. Interspeech 2013*, 2013, pp. 3622–3626.

[12] J. Shor, D. Emanuel, O. Lang, O. Tuval, M. Brenner, J. Cattiau, F. Vieira, M. McNally, T. Charbonneau, M. Nollstadt *et al.*, "Personalizing asr for dysarthric and accented speech with limited data," in *Proc. Interspeech 2019*, 2019, pp. 784–788.

[13] S. R. Shahamiri, V. Lal, and D. Shah, "Dysarthric speech transformer: A sequence-to-sequence dysarthric speech recognition system," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 31, pp. 3407–3416, 2023.

[14] S. Sehgal and S. Cunningham, "Model adaptation and adaptive training for the recognition of dysarthric speech," in *proceedings of SLPAT 2015: 6th workshop on speech and language processing for assistive technologies*, 2015, pp. 65–71.

[15] J. Yi, "Research and application analysis on key problems of automatic speech recognition for dysarthria," *Applied and Computational Engineering*, vol. 115, pp. 110–116, 2024.

[16] L. Wu, D. Zong, S. Sun, and J. Zhao, "A sequential contrastive learning framework for robust dysarthric speech recognition," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 7303–7307.

[17] W. Lee, S. Im, H. Do, Y. Kim, J. Ok, and G. Lee, "DyPCL: Dynamic phoneme-level contrastive learning for dysarthric speech recognition," in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, L. Chiruzzo, A. Ritter, and L. Wang, Eds. Albuquerque, New Mexico: Association for Computational Linguistics, Apr. 2025, pp. 4701–4712. [Online]. Available: https://aclanthology.org/2025.naacl-long.240/

[18] R. D. Kent, G. Weismer, J. F. Kent, and J. C. Rosenbek, "Toward phonetic intelligibility testing in dysarthria," *Journal of Speech and Hearing Disorders*, vol. 54, no. 4, pp. 482–499, 1989.

[19] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[20] L. Lim, "Peter ladefoged, a course in phonetics (4th edn.). orlando, fl: Harcourt, inc., 2001. pp. xiv+ 289. isbn 0-15-507319-2." *Journal of the International Phonetic Association*, vol. 32, no. 1, pp. 79–87, 2002.

[21] International Phonetic Association, *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press, 1999.

[22] G. Booij, *The phonology of Dutch*. Oxford University Press, 1999.

[23] A. Buzo, H. Cucu, C. Burileanu, M. Pasca, and V. Popescu, "Word error rate improvement and complexity reduction in automatic speech recognition by analyzing acoustic

model uncertainty and confusion," in *2011 6th Conference on Speech Technology and Human-Computer Dialogue (SpeD)*. IEEE, 2011, pp. 1–8.

[24] S. Bhatt, A. Dev, and A. Jain, "Confusion analysis in phoneme based speech recognition in hindi," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 10, pp. 4213–4238, 2020.

[25] E. Hermann and M. Magimai-Doss, "Handling acoustic variation in dysarthric speech recognition systems through model combination." in *Interspeech*, 2021, pp. 4788–4792.

[26] S. O. C. Morales, S. J. Cox *et al.*, "Modelling confusion matrices to improve speech recognition accuracy, with an application to dysarthric speech." in *INTERSPEECH*, 2007, pp. 1565–1568.

[27] S. Alharbi, M. Alrazgan, A. Alrashed, T. Alnomasi, R. Almojel, R. Alharbi, S. Alharbi, S. Alturki, F. Alshehri, and M. Almojil, "Automatic speech recognition: Systematic literature review," *IEEE Access*, vol. 9, pp. 131 858–131 876, 2021.

[28] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, and T. Sainath, "Deep learning for audio signal processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 206–219, 2019.

[29] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[30] G. Mark and Y. Steve, "The application of hidden markov models in speech recognition," *Foundations and Trends® in Signal Processing*, vol. 1, no. 3, pp. 195–304, 2024.

[31] R. Prabhavalkar, T. Hori, T. N. Sainath, R. Schlüter, and S. Watanabe, "End-to-end speech recognition: A survey," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 325–351, 2023.

[32] L. Dong, S. Xu, and B. Xu, "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 5884–5888.

[33] A. Aboeitta, A. Sharshar, Y. Nafea, and S. Shehata, "Bridging asr and llms for dysarthric speech recognition: Benchmarking self-supervised and generative approaches," in *Proc. Interspeech 2025*, 2025, pp. 2123–2127.

[34] V. Raja, A. V. Ganesan, A. Syamkumar, R. Banerjee, and H. Schwartz, "Idiosyncratic versus normative modeling of atypical speech recognition: Dysarthric case studies," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025, pp. 33 514–33 525.

[35] S. Hu, X. Xie, M. Geng, Z. Jin, J. Deng, G. Li, Y. Wang, M. Cui, T. Wang, H. Meng *et al.*, "Self-supervised asr models and features for dysarthric and elderly speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 3561–3575, 2024.

[36] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, p. 9, 2016.

[37] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *Advances in neural information processing systems*, vol. 27, 2014.

[38] A. Pasad, J.-C. Chou, and K. Livescu, "Layer-wise analysis of a self-supervised speech representation model," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 914–921.

[39] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[40] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 328–339.

[41] L. Kürzinger, D. Winkelbauer, L. Li, T. Watzel, and G. Rigoll, "Ctc-segmentation of large corpora for german end-to-end speech recognition," in *International Conference on Speech and Computer*. Springer, 2020, pp. 267–278.

[42] L. Besacier, E. Barnard, A. Karpov, and T. Schultz, "Automatic speech recognition for under-resourced languages: A survey," *Speech communication*, vol. 56, pp. 85–100, 2014.

[43] L. Rampello, L. Rampello, F. Patti, and M. Zappia, "When the word doesn't come out: A synthetic overview of dysarthria," *Journal of the Neurological Sciences*, vol. 369, pp. 354–360, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022510X16305391

[44] W. Xue, R. van Hout, C. Cucchiarini, and H. Strik, "Assessing speech intelligibility of pathological speech in sentences and word lists: The contribution of phoneme-level measures," *Journal of Communication Disorders*, vol. 102, p. 106301, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021992423000011

[45] K. Spencer, C. Friedlander, and K. Brown, "Predictors of health-related quality of life and communicative participation in individuals with dysarthria from parkinson's disease," *International Journal of Neurodegenerative Disorders*, vol. 3, no. 1, p. 014, 2020.

[46] S. Sapir and A. E. Aronson, "The relationship between psychopathology and speech and language disorders in neurologic patients," *Journal of Speech and Hearing Disorders*, vol. 55, no. 3, pp. 503–509, 1990.

[47] C. Bhat, "Recognition of dysarthric speech using voice parameters for speaker adaptation and multi-taper spectral estimation," *Interspeech 2016*, 2016.

[48] Z. Qian and K. Xiao, "A survey of automatic speech recognition for dysarthric speech," *Electronics*, vol. 12, no. 20, 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/20/4278

[49] B. F. Zaidi, S. A. Selouani, M. Boudraa, and M. Sidi Yakoub, "Deep neural network architectures for dysarthric speech analysis and recognition," *Neural Computing and Applications*, vol. 33, no. 15, pp. 9089–9108, 2021.

[50] K. Bharti and P. K. Das, "A survey on asr systems for dysarthric speech," in *2022 4th International Conference on Artificial Intelligence and Speech Technology (AIST)*, 2022, pp. 1–6.

[51] B. Vachhani, C. Bhat, and S. K. Kopparapu, "Data augmentation using healthy speech for dysarthric speech recognition." in *Interspeech*, 2018, pp. 471–475.

[52] P. Sapkota, H. K. Kathania, S. R. Kadiri, and S. Narayanan, "Improving end-to-end speech recognition for dysarthric speech through in-domain data augmentation," in *2024 58th Asilomar Conference on Signals, Systems, and Computers*, 2024, pp. 345–349.

[53] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[54] S. Rathod, M. Charola, and H. A. Patil, "Transfer learning using whisper for dysarthric automatic speech recognition," in *International conference on speech and computer*. Springer, 2023, pp. 579–589.

[55] S. Hu, X. Xie, Z. Jin, M. Geng, Y. Wang, M. Cui, J. Deng, X. Liu, and H. Meng, "Exploring self-supervised pre-trained asr models for dysarthric and elderly speech recognition," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[56] J. Wilson, Bronagh Blaney, "Acoustic variability in dysarthria and computer speech recognition," *Clinical Linguistics & Phonetics*, vol. 14, no. 4, pp. 307–327, 2000.

[57] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1735–1742.

[58] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PmLR, 2020, pp. 1597–1607.

[59] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *Advances in neural information processing systems*, vol. 33, pp. 18 661–18 673, 2020.

[60] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[61] J. Robinson, C.-Y. Chuang, S. Sra, and S. Jegelka, "Contrastive learning with hard negative samples," in *International Conference on Learning Representations (ICLR)*, 2021.

[62] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[63] C. Talnikar, T. Likhomanenko, R. Collobert, and G. Synnaeve, "Joint masked cpc and ctc training for asr," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 3045–3049.

[64] S. Wang, S. Zhao, J. Zhou, A. Kong, and Y. Qin, "Enhancing dysarthric speech recognition for unseen speakers via prototype-based adaptation," in *Proc. Interspeech 2024*, 2024, pp. 1305–1309.

[65] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio, "SpeechBrain: A general-purpose speech toolkit," 2021, arXiv:2106.04624.

[66] D. R. Mortensen, P. Littell, A. Bharadwaj, K. Goyal, C. Dyer, and L. S. Levin, "Panphon: A resource for mapping IPA segments to articulatory feature vectors," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. ACL, 2016, pp. 3475–3484.

[67] J. Hwang, M. Hira, C. Chen, X. Zhang, Z. Ni, G. Sun, P. Ma, R. Huang, V. Pratap, Y. Zhang, A. Kumar, C.-Y. Yu, C. Zhu, C. Liu, J. Kahn, M. Ravanelli, P. Sun, S. Watanabe, Y. Shi, Y. Tao, R. Scheibler, S. Cornell, S. Kim, and S. Petridis, "Torchaudio 2.1: Advancing speech recognition, self-supervised learning, and audio processing components for pytorch," 2023.

[68] Y. Zhang, M. J. Valkering, O. Scharenborg, and Z. Yue, "DysOne: A dutch and non-native english dysarthric audio-video dataset," 2026, manuscript in preparation.

[69] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[70] F. Rudzicz, A. K. Namasivayam, and T. Wolff, "The torgo database of acoustic and articulatory speech from speakers with dysarthria," *Language resources and evaluation*, vol. 46, no. 4, pp. 523–541, 2012.

[71] B. Neyshabur, H. Sedghi, and C. Zhang, "What is being transferred in transfer learning?" *Advances in neural information processing systems*, vol. 33, pp. 512–523, 2020.