



# Personalized Fine-Tuning with Controllable Synthetic Speech from LLM-Generated Transcripts for Dysarthric Speech Recognition

Dominik Wagner<sup>1</sup>, Ilja Baumann<sup>1</sup>, Natalie Engert<sup>1</sup>, Seanie Lee<sup>2</sup>, Elmar Nöth<sup>3</sup>, Korbinian Riedhammer<sup>1</sup>, Tobias Bocklet<sup>1</sup>

<sup>1</sup>Technische Hochschule Nürnberg Georg Simon Ohm, Germany

<sup>2</sup>Korea Advanced Institute of Science & Technology, Republic of Korea

<sup>3</sup>Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

firstname.lastname@th-nuernberg.de, lsnfamily02@kaist.ac.kr, elmar.noeth@fau.de

## Abstract

In this work, we present our submission to the Speech Accessibility Project challenge for dysarthric speech recognition. We integrate parameter-efficient fine-tuning with latent audio representations to improve an encoder-decoder ASR system. Synthetic training data is generated by fine-tuning Parler-TTS to mimic dysarthric speech, using LLM-generated prompts for corpus-consistent target transcripts. Personalization with x-vectors consistently reduces word error rates (WERs) over non-personalized fine-tuning. AdaLoRA adapters outperform full fine-tuning and standard low-rank adaptation, achieving relative WER reductions of  $\sim 23\%$  and  $\sim 22\%$ , respectively. Further improvements ( $\sim 5\%$  WER reduction) come from incorporating wav2vec 2.0-based audio representations. Training with synthetic dysarthric speech yields up to  $\sim 7\%$  relative WER improvement over personalized fine-tuning alone.

**Index Terms:** speech recognition, dysarthric speech, personalization, low-rank adaptation, synthetic data

## 1. Introduction

Dysarthria is a motor speech disorder characterized by impaired control of the muscles involved in speech production, which can result from damage to the central or peripheral nervous system. This impairment leads to difficulties with articulation, phonation, respiration, resonance, and prosody [1]. Dysarthria manifests differently depending on the underlying neurological condition. For instance, hypokinetic dysarthria, commonly associated with Parkinson’s disease (PD), is characterized by monotony of pitch and loudness, reduced stress, imprecise consonants, a breathy or harsh voice, and rapid bursts of speech [1]. PD is a chronic and progressive neurodegenerative disorder that primarily affects the motor system due to the loss of dopamine-producing neurons in the substantia nigra region of the brain [2, 3]. About 70% of PD patients develop speech impairments, which tend to worsen over time [4, 5].

Dysarthria is also associated with other neurological conditions, such as amyotrophic lateral sclerosis (ALS) and cerebellar disorders like cerebral palsy. These disorders result in distinct patterns of speech dysfunction, such as mixed dysarthria in ALS and ataxic dysarthria in cerebellar disorders [1].

Despite advancements in automatic speech recognition (ASR) systems, they remain ill-suited for dysarthric speech due to the high variability of speech patterns across individuals and the limited availability of large, high-quality datasets [6–9]. Conventional ASR systems, optimized for typical speech, struggle to generalize to atypical speakers, resulting in poor recognition accuracy. This limitation is particularly detrimental for individuals with dysarthria, as speech often serves as a critical mode of interaction with devices like smartphones and smart home systems [10].

To address data scarcity, various data augmentation techniques have been proposed for training ASR systems for dysarthric speech. Audio-level augmentations, such as speed, pitch, tempo, volume, and vocal tract length perturbations, have proven effective in increasing training data diversity and robustness [11–14]. SpecAugment [15] and dynamic augmentation techniques inspired by it have also been shown to enhance ASR accuracy for dysarthric speech [8, 13, 16]. Domain-specific methods, such as Dysarthric SpecAugment, simulate impaired speech characteristics from typical speech using domain knowledge [13]. Text-to-speech (TTS) and voice conversion systems have also been employed to mitigate data scarcity by generating synthetic dysarthric speech [8, 16–19].

Speaker adaptation is a widely used method for personalizing ASR systems to dysarthric speakers. For example, in [20], feature-space maximum likelihood linear regression [21] and x-vectors [22] are used as auxiliary features for self-supervised models like wav2vec 2.0 [23] to capture speaker-specific characteristics. Other approaches include fine-tuning speaker-independent models on individual speakers [24–26] and leveraging latent features [27–30]. More recently, speaker-independent systems [31] and multi-task approaches with auxiliary classification tasks [32] have been used.

Recent research shows the importance of personalization, synthetic data generation, and augmentation techniques in improving ASR performance for dysarthric speech. Building on these findings, this paper explores a lightweight personalization approach for large end-to-end ASR systems developed for small academic lecture corpora [33], leveraging a modified Whisper [34] architecture with speaker adaptation capabilities and parameter-efficient fine-tuning. Following previous works [8, 16, 17], we integrate SpecAugment into the training pipeline and explore the effectiveness of supplementing the relatively small amount of training data with artificial examples generated by a text-to-speech (TTS) synthesis system. Specifically, we fine-tune Parler-TTS [35], an autoregressive transformer [36] model operating on quantized audio units, to generate synthetic training data for dysarthric speech. In contrast to previous works [8, 17], Parler-TTS allows fine-grained control over attributes such as gender, noise level, and speaking rate through text descriptions, enabling the creation of diverse dysarthric speech datasets. Furthermore, Parler-TTS can be effectively fine-tuned using attributes such as pitch and speech quality derived from dysarthric speech data in combination with novel target transcripts generated by large language models (LLMs). Our experiments show that personalization with x-vectors consistently improves performance compared to non-personalized fine-tuning. AdaLoRA outperforms other training methods and further improvements can be achieved by incorporating additional audio representations into the ASR system.

We observe that the effectiveness of synthetic data varies across etiologies, with performance gains depending on both the specific condition and the TTS model used for generation.

## 2. Data

We employ the Speech Accessibility Project [37] (SAP) dataset for all experiments, with models submitted to the associated challenge.<sup>1</sup> Specifically, we use the SAP 2024-11-30 data package, a dataset curated for the development of dysarthric ASR systems, which includes English dysarthric speech from individuals with PD, ALS, cerebral palsy, down syndrome, and stroke. The 2024-11-30 version of the data contains speech recordings from a total of 430 participants, with 374 assigned to the training set and 56 to the development (*dev*) set. Each participant recorded an average of 350 utterances with an average duration of  $7.9 \pm 9.0$  seconds, drawn from virtual assistant commands, sentences from novels, and spontaneous speech. The combined duration of the recordings is  $\sim 333$  hours with  $\sim 290$  hours allocated to the training set and  $\sim 43$  hours allocated to the *dev* set. The majority of the data ( $\sim 74\%$ ) are recordings from participants with PD. The remainder is distributed between ALS ( $\sim 14\%$ ), cerebral palsy ( $\sim 6\%$ ), down syndrome ( $\sim 4\%$ ), and stroke ( $\sim 2\%$ ).

Most experimental results reported in our work are based on the *dev* set. We also report some results on the unseen *test1* set as part of the SAP challenge. However, due to upload restrictions and time constraints, we were unable to generate scores for every experiment.

## 3. Method

Figure 1 provides an overview of the components used in this work. The speech recognition system ① is based on the approach introduced in [33]. It employs Whisper [34] as the ASR backbone and a personalization component based on a small neural network ③, which projects audio representations obtained from a pre-trained and frozen embedding model (e.g. x-vectors [22]) ② into the latent space of the decoder. The model is trained to recognize dysarthric speech using either full fine-tuning (FFT) or parameter-efficient fine-tuning with LoRA [38] and AdaLoRA [39]. Synthetic speech data is generated with a fine-tuned Parler-TTS model [35] ④ using LLMs ⑤ to generate new target transcripts that closely align with the real data.

### 3.1. Model adaptation

Whisper employs a transformer-based encoder-decoder architecture [36]. Given log-Mel spectrogram features  $X_{1:T} = (x_1, \dots, x_T)$ , the encoder maps them to latent representations:

$$H_{1:M} = E_\theta(X_{1:T}), \quad H_{1:M} \in \mathbb{R}^{M \times b}, \quad (1)$$

where  $E_\theta$  is the encoder with parameters  $\theta$ , and  $M = T/2$  after subsampling. The decoder then estimates a probability distribution for the next token  $y_i$  over the vocabulary, given  $H_{1:M}$  and the previous tokens  $y_{<i} = (y_1, \dots, y_{i-1})$ :

$$P(y_i | y_{<i}, H_{1:M}) = D_\phi(y_{<i}, H_{1:M}), \quad \text{for } i \in [T'], \quad (2)$$

where  $D_\phi$  is the decoder with parameters  $\phi$  and  $[T'] = \{1, \dots, T'\}$ . The architecture is modified by incorporating audio representations. Given input features  $Z_{1:L} = (z_1, \dots, z_L)$ , multiple audio embedding models map them to a set of fixed-dimensional latent representations. Each embedding model  $f_{\psi_i}$  generates a sequence of  $U_i$  latent vectors. The sequence is averaged to obtain a single fixed-dimensional vector:

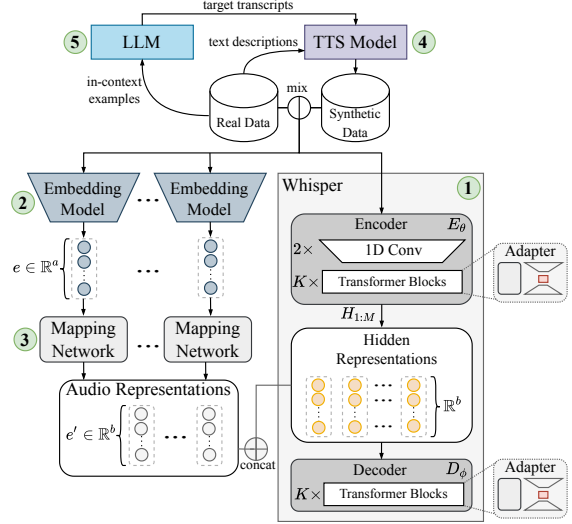


Figure 1: Overview of the proposed ASR system combining the incorporation of audio representations, fine-tuning, and synthetic training data generation.

$$e^{(i)} = \frac{1}{U_i} \sum_{u=1}^{U_i} f_{\psi_i}(Z_{1:L})_u \in \mathbb{R}^{a_i}, \quad \text{for } i \in [N], \quad (3)$$

where  $f_{\psi_i}(Z_{1:L})_u \in \mathbb{R}^{a_i}$  is the  $u$ -th row of  $f_{\psi_i}(Z_{1:L}) \in \mathbb{R}^{U_i \times a_i}$ . Each audio representation  $e^{(i)}$  is then projected into the decoder's latent space via a dedicated mapping network:

$$e'^{(i)} = W_2^{(i)} \sigma_\tau(W_1^{(i)} e^{(i)} + b_1^{(i)}) + b_2^{(i)}, \quad e'^{(i)} \in \mathbb{R}^b, \quad (4)$$

where  $\{(W_j^{(i)}, b_j^{(i)})\}_{j=1}^2$  are trainable parameters specific to each mapping network, and  $\sigma_\tau$  is Tanh. Finally, all the representations  $\{e'^{(i)}\}_{i=1}^N$  are concatenated with the decoder's input:

$$\tilde{H}_{1:N+M} = \text{concat}(e'^{(1)}, \dots, e'^{(N)}, H_{1:M}). \quad (5)$$

During training, only the adapter modules and mapping networks are updated, all other components remain frozen.

### 3.2. Synthetic dysarthric speech generation

We fine-tune the `parler-tts-mini-v1` version of Parler-TTS [40], a 880M parameter TTS model trained on  $\sim 45k$  hours of audio data. Parler-TTS is an open-source and open-weight implementation of [35], which is based on the MusicGen [41] architecture. The model has three core stages: First, a pre-trained and frozen text encoder based on FLAN-T5 [42] converts a text description of the target audio into a sequence of hidden representations. Second, a transformer language model acts as a decoder that autoregressively generates audio tokens. The model relies entirely on the text description for gender category, tone, and speech patterns. Cross-attention is used to condition the model on these text descriptions. The text prompts representing the spoken utterance are pre-pended to the decoder inputs. Finally, a neural audio codec is used to synthesize waveforms from the generated audio tokens. The open implementation of Parler-TTS employs the Describe Audio Codec (DAC) [43] instead of EnCodec [44] for audio synthesis.

#### 3.2.1. Target prompt generation

We use both Phi-3 [45] (`Phi-3-medium-4k-instruct`) and Llama 3.1 70B [46] (`Llama-3.1-70B-Instruct`) for generating English target transcripts. To seed the generation of diverse synthetic data that roughly follows the topic and phrasing distribution of the SAP dataset, we follow [47] and instruct

<sup>1</sup><https://eval.ai/web/challenges/challenge-page/2362/overview>

the LLM to generate new sentences based on ten randomly sampled utterances from the SAP training set. The goal is to increase data diversity in a controlled manner by generating sentences that closely align with the SAP dataset in terms of vocabulary, topics, and phrasing, while ensuring they are not identical. The SAP dataset does not include a gender attribute. However, we aim to make gender a controllable feature in Parler-TTS since the model has been pre-trained with binary gender information. To address this, we employ a pre-trained classifier to predict gender labels for the SAP data. The model is based on an ECAPA-TDNN [48] speaker verification model that was fine-tuned to predict binary gender labels using the VoxCeleb2 development set.

## 4. Experiments and results

Our experiments were evaluated using the SAP challenge evaluation scheme, which is based on two metrics: WER and semantic score (SemScore). SemScore is a weighted sum of BERTscore [49], phonetic distance, and natural language inference probability [50]. For each experiment, we explored multiple hyperparameter configurations. The models trained with LoRA used learning rates  $\eta \in \{10^{-3}, 10^{-4}, 2 \cdot 10^{-5}\}$ , scaling parameters  $\alpha \in \{32, 64, 128\}$ , and ranks  $r \in \{8, 16, 32\}$ . Since AdaLoRA showed more stable behavior, we set  $\eta = 10^{-3}$  and  $\alpha = 32$  and explored rank configurations of  $r_{\text{initial}} \in \{12, 16, 24, 32, 64\}$  and  $r_{\text{target}} \in \{8, 12, 16, 24, 32\}$ . FFT was conducted using the learning rates  $\eta \in \{10^{-4}, 5 \cdot 10^{-5}, 5 \cdot 10^{-6}\}$ . We report the results under the best hyperparameter configuration on the *dev* set for each experiment in the subsequent tables. All experiments were conducted using the largest available Whisper model (*whisper-large-v3*) as the ASR backbone. Adapter modules were attached to the query and value matrices of each transformer block in the encoder and decoder. Speaker characteristics were incorporated via x-vectors extracted from a model trained on VoxCeleb 1 and 2 [51, 52]. All experiments used a dropout probability of 0.1 in the adapter modules and mapping networks. The models were trained for 15 epochs. The best model was chosen based on the lowest WER on a randomly sampled subset (~10%) of the *dev* set. Performance was evaluated every 2000 training steps. We used the AdamW optimizer ( $\lambda = 10^{-4}$ ,  $\epsilon = 10^{-8}$ ,  $\beta_1 = 0.99$ ,  $\beta_2 = 0.999$ ) with a linear rate schedule, a warm-up phase of 500 steps, and initial learning rates depending on the training method. Greedy decoding was used in all experiments.

We fine-tuned Parler-TTS for 10 epochs on the SAP training data using the AdamW optimizer ( $\lambda = 10^{-2}$ ,  $\epsilon = 10^{-8}$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.99$ ) with a constant learning rate of  $9.5 \cdot 10^{-4}$  and a warm-up phase of 2k steps. To generate synthetic dysarthric speech data, attributes characterizing the speaker were randomly sampled from the SAP training set and paired with LLM-generated prompts. Parler-TTS occasionally failed to produce intelligible speech for certain attribute-prompt combinations. To address these cases, we transcribed all synthetic utterances using Whisper (*whisper-medium*) and discarded those with a WER of 35 or higher.

### 4.1. Main results

The goal of the experiments in Table 1 is evaluating the impact of personalization and SpecAugment, while also comparing the performance of LoRA, AdaLoRA, and FFT. The first row (#1) shows the results when inference on the underlying base model (*whisper-large-v3*) is performed without any fine-tuning. Applying SpecAugment during training leads to WER and SemScore improvements with AdaLoRA and FFT

Table 1: Comparison of training methods, personalization and SpecAugment. Results in parentheses indicate the non-public *test1* set, all other results were computed on the *dev* set.

#	Training Method	Personalization	Spec-Augment	WER (↓)	SemScore (↑)
1	None	✗	✗	12.31	84.63
2	LoRA	✗	✗	10.87	88.92
3	LoRA	✗	✓	13.28	88.47
4	AdaLoRA	✗	✗	11.73	87.02
5	AdaLoRA	✗	✓	11.32	87.32
6	FFT	✗	✗	11.77	87.26
7	FFT	✗	✓	10.57	88.13
8	LoRA	✓	✗	10.52	88.89
9	LoRA	✓	✓	11.09	88.93
10	AdaLoRA	✓	✗	<b>8.05</b> (10.65)	91.29 (88.33)
11	AdaLoRA	✓	✓	8.12 (11.36)	<b>92.31</b> (88.93)
12	FFT	✓	✗	11.54	86.69
13	FFT	✓	✓	10.27	88.50

in the non-personalized setting (#2-7). However, for LoRA without personalization, it increases WER from 10.87 to 13.28. In the personalized setting (#8-13), AdaLoRA shows a slight degradation in WER from 8.05 to 8.12 but an improvement in SemScore from 91.29 to 92.31. Without personalization, FFT yields the lowest WER (10.57), while LoRA and AdaLoRA perform worse. However, in the personalized setting, AdaLoRA outperforms both LoRA and FFT, achieving the lowest WER (8.05 in #10) and highest SemScore (92.31 in #11).

The results of experiment #10 represent a ~31% reduction in WER and a ~6% increase in SemScore compared to the corresponding experiment without personalization (#4). On the non-public *test1* set, WER deteriorated more significantly with SpecAugment (#11) compared to the experiment without it (#10), while SemScore remained higher in #11. Even though, SpecAugment does not yield consistent performance gains, we found that it somewhat stabilizes training with respect to hyperparameter choices, particularly the initial and target ranks in AdaLoRA fine-tuning.

### 4.2. Generalization beyond Whisper

To evaluate the generalization ability of x-vector-based personalization beyond Whisper, we applied it to HuBERT [53] as well. Specifically, we used the largest available HuBERT model (*hubert-xlarge-1160k*; 1B parameters) as the base encoder, followed by a bidirectional long short-term memory (BiLSTM) [54] block and a connectionist temporal classification (CTC) objective [55]. Personalization was integrated by concatenating the outputs of the HuBERT model with speaker vectors, before being fed into the BiLSTM layer. Without personalization, the model achieved a WER of 9.31 and a SemScore of 91.06 on the *dev* set. When incorporating personalization, the model improved, achieving a WER of 8.79 and a SemScore of 91.42 on the *dev* set.

### 4.3. Adding additional audio representations

Table 2: Results with *wav2vec 2.0* audio representations. All experiments were conducted using AdaLoRA w/o SpecAugment.

#	Personalization	Audio Representation	WER (↓)	SemScore (↑)
1.1	✓	base (layer 2)	7.80	91.16
1.2	✓	base (layer 6)	7.86	91.31
1.3	✓	base (layer 12)	<b>7.68</b>	<b>91.38</b>
1.4	✓	etiology	8.07 (10.78)	91.17 (88.21)
1.5	✓	category	8.09 (10.67)	91.27 (88.35)
1.6	✗	base (layer 2)	9.30	89.36
1.7	✗	base (layer 6)	9.53	88.78
1.8	✗	base (layer 12)	8.09	91.21
1.9	✗	etiology	11.55	87.13
1.10	✗	category	8.83	90.70

Since incorporating speaker vectors during training improved performance (cf. Table 1), a natural extension is to explore the effect of additional audio representations. Inspired by [56, 57], we trained linear classifiers on top of a pre-trained wav2vec 2.0 model (*wav2vec2-base*; 94M parameters) to predict metadata from the SAP dataset. Specifically, we trained classifiers to infer etiology labels and utterance categories. We then extracted latent representations at the last transformer layer of the underlying wav2vec 2.0 backbone and used them as additional input features to the ASR system. Additionally, we extracted representations from the non-fine-tuned wav2vec 2.0 base model at layers 2, 6, and 12.

We extended experiment #10 from Table 1 (AdaLoRA with personalization, without SpecAugment) due to the disproportionate WER increase on the *test1* set observed in experiment #11. Table 2 presents the results of integrating additional audio representations. In the upper half (#1.1-1.5), where personalization is applied, several configurations outperform experiment #10 from Table 1. However, using representations fine-tuned for etiology and category classification embeddings did not yield improvements (#1.4 and #1.5). The best-performing experiment is AdaLoRA with personalization and audio representations extracted at layer 12 of the base model, achieving 7.68 WER and 91.38 SemScore. These results indicate that lower-layer representations (e.g., layer 2 and layer 6) are suboptimal in this context, achieving slightly higher WERs and lower SemScores. This aligns with previous findings that deeper layers in wav2vec 2.0 encode more phonetic and word content [58], which may be more beneficial for ASR. The lower half of the table, where personalization is not used, shows that audio embeddings alone do not improve performance. However, when combined with personalization, synergies emerge, leading to overall performance gains.

#### 4.4. Adding synthetic speech data

The effect of supplementing the training data with increasing amounts of synthetic speech are shown in Table 3. Following [8], we report the results for ratios between 10% and 100% of additional synthesized training data. In the setup without SpecAugment (#2.1-2.3), adding Parler-TTS-generated synthetic speech from LLM-generated target prompts improves performance over experiment #10 from Table 1. At 10% synthetic data, WER drops to 7.71, and SemScore increases to 91.58. Increasing the synthetic ratio to 50% further reduces WER to 7.57 and boosts SemScore to 92.25. At 100%, the best result is achieved, with a WER of 7.47 and SemScore of 92.39, both improvements over experiment #10 in Table 1. Experiments #2.4-2.6 show that SpecAugment does not improve performance in this setting. However, WERs also decrease with increasing amounts of synthetic data. Experiments #2.7-2.9 use transcripts from the SAP training set as prompts for the TTS system instead of LLM-generated prompts. These experiments show worse performance compared to speech synthesis based on the more diverse LLM-generated prompts (#2.1-2.3). Experiments #2.10-2.12 follow experiment #1.3 from Table 2, incorporating additional audio representations obtained from the last layer of a pre-trained wav2vec 2.0 system. While effective without synthetic data, results did not improve when synthetic data was used.

In addition to Parler-TTS, we used OpenVoice [59] in a zero-shot setting by selecting reference speakers from randomly sampled utterances in the SAP training set to extract style information. The target transcripts were based on the same LLM-generated prompts used in Parler-TTS. However, in our exper-

Table 3: Results with varying portions of synthetic training data using AdaLoRA fine-tuning. All experiments employ x-vector based personalization.

#	TTS Model	LLM Prompt	Spec-augment	Audio Repr.	Synth Data	WER (↓)	SemScore (↑)
2.1	Parler-TTS	✓	✗	✗	10%	7.71	91.58
2.2	Parler-TTS	✓	✗	✗	50%	7.57	92.25
2.3	Parler-TTS	✓	✗	✗	100%	<b>7.47</b>	<b>92.39</b>
2.4	Parler-TTS	✓	✓	✗	10%	9.28 (11.68)	90.65 (87.32)
2.5	Parler-TTS	✓	✓	✗	50%	8.49 (11.68)	91.65 (88.62)
2.6	Parler-TTS	✓	✓	✗	100%	7.92	91.17
2.7	Parler-TTS	✗	✗	✗	10%	8.44	91.83
2.8	Parler-TTS	✗	✗	✗	50%	7.84	91.50
2.9	Parler-TTS	✗	✗	✗	100%	8.43	90.03
2.10	Parler-TTS	✓	✗	✓	10%	8.17	90.66
2.11	Parler-TTS	✓	✗	✓	50%	8.15	90.93
2.12	Parler-TTS	✓	✗	✓	100%	8.27	91.79
2.13	OpenVoice	✓	✗	✗	10%	8.24	91.56
2.14	OpenVoice	✓	✗	✗	50%	8.35 (12.61)	90.48 (86.13)
2.15	OpenVoice	✓	✗	✗	100%	9.14	89.25

iments (#2.13–2.15), OpenVoice-generated synthetic data resulted in higher WERs, with 8.24 at 10% synthetic data compared to 7.71 for Parler-TTS. Performance further degrades as more synthetic data is added, with WER rising to 9.14 at 100%.

To further assess the impact of synthetic data augmentation, we compared the best results from Table 3 (#2.1–2.3) to experiment #10 from Table 1 at the etiology level. Table 4 shows that the effectiveness of synthetic data varies by etiology, with the largest gains observed for ALS, while improvements for other conditions were smaller in relative terms.

Table 4: Comparison of results by etiology.

#	ALS		Cerebral Palsy		Down Syndrome		Parkinson		Stroke	
	WER	SemScore	WER	SemScore	WER	SemScore	WER	SemScore	WER	SemScore
10	6.54	90.14	18.68	78.19	22.95	74.73	7.01	94.23	9.04	93.42
2.1	5.27	90.72	17.47	78.31	23.59	74.02	6.62	94.44	8.93	93.50
2.2	4.73	92.11	16.26	79.52	21.73	75.83	6.68	94.86	9.01	94.00
2.3	4.91	92.52	16.47	79.74	22.10	75.76	6.50	94.97	9.18	93.62

#### 4.5. Limitations

Although, our approach improves dysarthric ASR through personalized fine-tuning and synthetic data augmentation, several limitations remain. First, the effectiveness of synthetic dysarthric speech varies by etiology, indicating that Parler-TTS and its textual description of voice attributes may not fully capture the complexity of real dysarthric speech patterns. Moreover, due to time and resource constraints, we were only able to obtain the best results with synthetic data after the challenge had already concluded. As a result, we were unable to score our best models on the unseen *test1* set, leaving uncertainty about how well these models would generalize to this dataset.

## 5. Conclusions

This work explored improving dysarthric speech recognition through personalized fine-tuning with speaker vectors, synthetic speech augmentation, and the integration of additional audio representations. We fine-tuned Parler-TTS to generate controllable synthetic dysarthric speech using LLM-generated transcripts and incorporated personalization via x-vectors. Our results show that personalization with AdaLoRA significantly reduces WER, achieving up to a ~31% improvement over non-personalized fine-tuning. Additionally, integrating wav2vec 2.0-based audio representations further reduced WER by ~5%. Synthetic dysarthric speech contributed up to ~7% additional improvement on the overall dataset, but its effectiveness varied by etiology and TTS model.

## 6. References

- [1] F. L. Darley, A. E. Aronson, and J. R. Brown, "Differential diagnostic patterns of dysarthria," *Journal of Speech and Hearing Research*, vol. 12, no. 2, pp. 246–269, 1969.
- [2] A. E. Lang and A. M. Lozano, "Parkinson's disease," *New England Journal of Medicine*, vol. 339, no. 15, pp. 1044–1053, 1998.
- [3] J. M. Fearnley and A. J. Lees, "Ageing and parkinson's disease: substantia nigra regional selectivity," *Brain*, vol. 114, no. 5, pp. 2283–2301, 10 1991.
- [4] L. Hartelius and P. Svensson, "Speech and swallowing symptoms associated with parkinson's disease and multiple sclerosis: A survey," *Folia Phoniatrica et Logopaedica*, vol. 46, no. 1, 12 2009.
- [5] H. L. Klawans, "Individual manifestations of parkinson's disease after ten or more years of levodopa," *Movement Disorders*, vol. 1, no. 3, pp. 187–192, 1986.
- [6] M. Moore, H. Venkateswara, and S. Panchanathan, "Whistle-blowing ASRs: Evaluating the need for more inclusive speech recognition systems," in *Interspeech*, 2018, pp. 466–470.
- [7] F. Xiong, J. Barker, and H. Christensen, "Phonetic analysis of dysarthric speech tempo and applications to robust personalised dysarthric speech recognition," in *ICASSP*, 2019, pp. 5836–5840.
- [8] W.-Z. Leung, M. Cross, A. Ragni, and S. Goetze, "Training data augmentation for dysarthric automatic speech recognition by text-to-dysarthric-speech synthesis," in *Interspeech*, 2024.
- [9] H. Christensen, M. B. Aniol, P. Bell, P. D. Green, T. Hain, S. King, and P. Swietojanski, "Combining in-domain and out-of-domain speech data for automatic recognition of disordered speech," in *Interspeech*, 2013.
- [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, Sep. 2019.
- [11] B. Vachhani, C. Bhat, and S. K. Kopparapu, "Data augmentation using healthy speech for dysarthric speech recognition," in *Interspeech*, 2018.
- [12] M. Geng, X. Xie, S. Liu, J. Yu, S. Hu, X. Liu, and H. Meng, "Investigation of data augmentation techniques for disordered speech recognition," in *Interspeech*, 2020, pp. 696–700.
- [13] C. Bhat, A. Panda, and H. Strik, "Improved ASR performance for dysarthric speech using two-stage data augmentation," in *Interspeech*, 2022, pp. 46–50.
- [14] F. Xiong, J. Barker, and H. Christensen, "Phonetic analysis of dysarthric speech tempo and applications to robust personalised dysarthric speech recognition," in *ICASSP*, 2019, pp. 5836–5840.
- [15] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Interspeech*, 2019.
- [16] H. Wang, T. Thebaud, J. Villalba, M. Sydnor, B. Lammers, N. Dehak, and L. Moro-Velazquez, "DuTa-VC: A Duration-aware Typical-to-atypical Voice Conversion Approach with Diffusion Probabilistic Model," in *Interspeech*, 2023, pp. 1548–1552.
- [17] M. Soleymanpour, M. T. Johnson, R. Soleymanpour, and J. Berry, "Synthesizing dysarthric speech using multi-speaker TTS for dysarthric speech recognition," in *ICASSP*, 2022, pp. 7382–7386.
- [18] Z. Jin, X. Xie, M. Geng, T. Wang, S. Hu, J. Deng, G. Li, and X. Liu, "Adversarial data augmentation using vae-gan for disordered speech recognition," in *ICASSP*, 2023, pp. 1–5.
- [19] J. Harvill, D. Issa, M. Hasegawa-Johnson, and C. Yoo, "Synthesis of new words for improved dysarthric speech recognition on an expanded vocabulary," in *ICASSP*, 2021, pp. 6428–6432.
- [20] M. K. Baskar, T. Herzog, D. Nguyen, M. Diez, T. Polzehl, L. Burget, and J. Čermocký, "Speaker adaptation for wav2vec2 based dysarthric ASR," in *Interspeech*, 2022, pp. 3403–3407.
- [21] M. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech & Language*, vol. 12, no. 2, pp. 75–98, 1998.
- [22] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-Vectors: Robust DNN Embeddings for Speaker Recognition," in *ICASSP*, 2018, pp. 5329–5333.
- [23] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," in *NeurIPS*, 2020, pp. 12 449–12 460.
- [24] R. Takashima, T. Takiguchi, and Y. Ariki, "Two-step acoustic model adaptation for dysarthric speech recognition," in *ICASSP*, 2020, pp. 6104–6108.
- [25] K. Tomanek, K. Seaver, P.-P. Jiang, R. Cave, L. Harrell, and J. R. Green, "An analysis of degenerating speech due to progressive dysarthria on asr performance," in *ICASSP*, 2023, pp. 1–5.
- [26] J. Shor *et al.*, "Personalizing ASR for dysarthric and accented speech with limited data," in *Interspeech*, 2019, pp. 784–788.
- [27] M. Geng *et al.*, "On-the-fly feature based rapid speaker adaptation for dysarthric and elderly speech recognition," in *Interspeech*, 2023.
- [28] S. Wang, S. Zhao, J. Zhou, A. Kong, and Y. Qin, "Enhancing dysarthric speech recognition for unseen speakers via prototype-based adaptation," in *Interspeech*, 2024, pp. 1305–1309.
- [29] A. Hernandez, P. A. Pérez-Toro, E. Nöth, J. R. Orozco-Arroyave, A. Maier, and S. H. Yang, "Cross-lingual self-supervised speech representations for improved dysarthric speech recognition," in *Interspeech*, 2022.
- [30] S. Hu, X. Xie, M. Geng, Z. Jin, J. Deng, G. Li, Y. Wang, M. Cui, T. Wang, H. Meng, and X. Liu, "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.
- [31] S. Singh, Q. Wang, Z. Zhong, C. Mendes, M. Hasegawa-Johnson, W. Abdulla, and S. R. Shahamiri, "Robust cross-etiology and speaker-independent dysarthric speech recognition," 2025. [Online]. Available: <https://arxiv.org/abs/2501.14994>
- [32] X. Zheng, B. Phukon, and M. Hasegawa-Johnson, "Fine-tuning automatic speech recognition for people with parkinson's: An effective strategy for enhancing speech technology accessibility," in *Interspeech*, 2024.
- [33] D. Wagner, I. Baumann, T. Ranzenberger, K. Riedhammer, and T. Bocklet, "Personalizing large sequence-to-sequence speech foundation models with speaker representations," in *SLT*, 2024.
- [34] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust Speech Recognition via Large-Scale Weak Supervision," 2022. [Online]. Available: <https://arxiv.org/abs/2212.04356>
- [35] D. Lyth and S. King, "Natural language guidance of high-fidelity text-to-speech with synthetic annotations," 2024. [Online]. Available: <https://arxiv.org/abs/2402.01912>
- [36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *NeurIPS*, 2017.
- [37] M. Hasegawa-Johnson *et al.*, "Community-supported shared infrastructure in support of speech accessibility," *Journal of Speech, Language, and Hearing Research*, vol. 67, no. 11, 2024.
- [38] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," in *ICLR*, 2022.
- [39] Q. Zhang, M. Chen, A. Bukharin, P. He, Y. Cheng, W. Chen, and T. Zhao, "Adaptive Budget Allocation for Parameter-Efficient Fine-Tuning," in *ICLR*, 2023.
- [40] Y. Lacombe, V. Srivastav, and S. Gandhi, "Parler-TTS," 2024. [Online]. Available: <https://github.com/huggingface/parler-tts>
- [41] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Defossez, "Simple and controllable music generation," in *NeurIPS*, vol. 36, 2023, pp. 47 704–47 720.
- [42] H. W. Chung *et al.*, "Scaling instruction-finetuned language models," 2022. [Online]. Available: <https://arxiv.org/abs/2210.11416>
- [43] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, "High-fidelity audio compression with improved RVQGAN," in *NeurIPS*, 2023.
- [44] A. Defossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *TMLR*, 2023.
- [45] M. Abdin *et al.*, "Phi-3 technical report: A highly capable language model locally on your phone," 2024. [Online]. Available: <https://arxiv.org/abs/2404.14219>
- [46] A. Grattafiori *et al.*, "The Llama 3 herd of models," 2024. [Online]. Available: <https://arxiv.org/abs/2407.21783>
- [47] Y. Li, S. Bubeck, R. Eldan, A. D. Giorno, S. Gunasekar, and Y. T. Lee, "Textbooks are all you need II: phi-1.5 technical report," 2023. [Online]. Available: <https://arxiv.org/abs/2309.05463>
- [48] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," in *Interspeech*, 2020, pp. 3830–3834.
- [49] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with BERT," in *ICLR*, 2020.
- [50] Y. Chen and S. Eger, "MENLI: Robust evaluation metrics from natural language inference," *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 804–825, 2023.
- [51] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A Large-Scale Speaker Identification Dataset," in *Interspeech*, 2017.
- [52] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Interspeech*, 2018, pp. 1086–1090.
- [53] W. Hsu, B. Bolte, Y. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units," *IEEE TASLP*, 2021.
- [54] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm networks," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, 2005, pp. 2047–2052.
- [55] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006, p. 369–376.
- [56] D. Gimeno-Gómez, C. Botelho, A. Pompili, A. Abad, and C.-D. Martínez-Hinarejos, "Unveiling interpretability in self-supervised speech representations for parkinson's diagnosis," 2024. [Online]. Available: <https://arxiv.org/abs/2412.02006>
- [57] D. Wagner, S. P. Bayerl, I. Baumann, E. Nöth, K. Riedhammer, and T. Bocklet, "Large language models for dysfluency detection in stuttered speech," in *Interspeech*, 2024, pp. 5118–5122.
- [58] A. Pasad, J.-C. Chou, and K. Livescu, "Layer-wise analysis of a self-supervised speech representation model," in *ASRU*, 2021, pp. 914–921.
- [59] Z. Qin, W. Zhao, X. Yu, and X. Sun, "Openvoice: Versatile instant voice cloning," 2024. [Online]. Available: <https://arxiv.org/abs/2312.01479>