L-Flow: Longitudinal Flow Matching for Progression-Aware Speech Biomarker Modeling

Enshi Zhang, Christian Poellabauer

Florida International University
{ezhan004,cpoellab}@fiu.edu

Abstract

Speech is an increasingly promising biomarker for neurodegenerative disorders due to its non-invasive nature, low cost, and suitability for frequent, longitudinal monitoring. However, existing clinical speech datasets are typically small, sparse, and irregularly sampled, limiting the ability to model continuous disease progression and develop robust biomarkers. We propose L-Flow, a progression-aware conditional speech generation framework for longitudinal speech trajectory completion. Given a patient's speech recordings, timestamps, and clinical progression labels, L-Flow learns a query-specific progression representation to synthesize speech at intermediate time points within the observed recording span. Experiments on two real-world longitudinal speech datasets demonstrate that L-Flow generates synthetic speech with stronger progression consistency than eight baseline augmentation methods. In addition, incorporating L-Flow–augmented data improves downstream regression performance to predict clinical severity, highlighting its utility for speech-based biomarker development.

1 Introduction

Neurodegenerative disorders, such as Alzheimer's disease (AD), Parkinson's disease (PD), and amyotrophic lateral sclerosis (ALS), are characterized by progressive neuronal loss that leads to a gradual decline in cognitive and motor function [1]. Among the affected domains, **speech** is particularly sensitive to disease progression, with patients exhibiting changes in speaking rate, pausing patterns, articulation, vocal intensity, and prosody [2, 3, 4]. These properties make speech a compelling candidate for biomarker development. Unlike neuroimaging, blood-based assays, or in-person clinical assessments, speech can be collected non-invasively, at low cost, and in remote settings. Previous research has shown that collecting speech and related behavioral measurements via mobile devices is feasible for remote emotional and health assessments [5]. This enables frequent, longitudinal sampling from the same individual, supporting fine-grained tracking of disease progression and more responsive clinical decision-making [6].

However, modeling the progression of neurodegenerative disorders from speech remains challenging in practice. Accessible clinical speech datasets are scarce due to privacy and data-sharing constraints [7, 8], and they are much smaller than datasets used in traditional speech tasks such as speech recognition (i.e., LibriSpeech [9]), audio recognition (i.e., AudioSet [10]), and those used to train modern speech foundation models (i.e., GigaSpeech [11], Common Voice [12]). Moreover, disease trajectories vary substantially across individuals, and data collection sessions are often irregularly spaced. As a result, longitudinal speech samples are sparse, unevenly sampled, and imbalanced across disease stages, limiting the robustness of progression modeling.

To address these challenges, we formulate longitudinal speech generation as a **trajectory completion problem**. Given a subject's speech recordings, timestamps, and ordinal clinical labels, the goal is

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

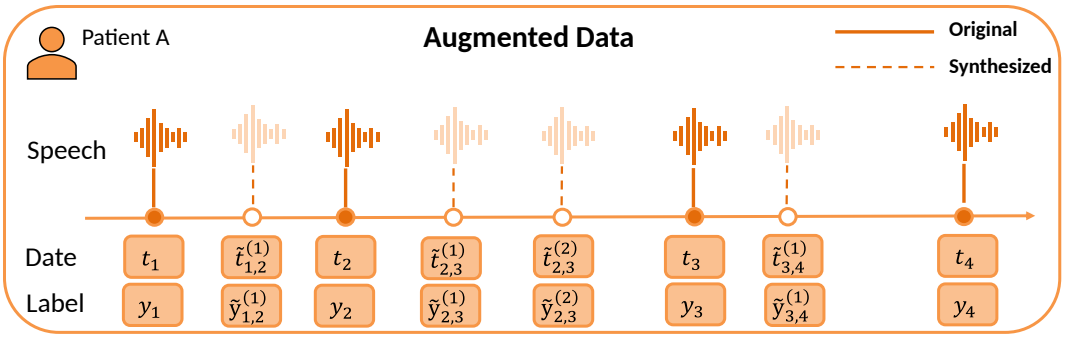

Figure 1: Illustration of augmenting sparse longitudinal speech recordings with synthesized intermediate samples. Solid circles indicate original recordings, while open circles represent synthesized samples with interpolated timestamps and labels.

to synthesize speech at user-specified intermediate time points within the observed temporal range. The generated samples preserve subject-specific vocal characteristics while remaining consistent with the expected disease progression across sessions. By densifying sparse and irregularly sampled trajectories, this approach provides additional training data for clinical speech modeling and helps mitigate data sparsity and stage imbalance in low-resource settings. Figure 1 illustrates this process of transforming sparse longitudinal recordings into continuous trajectories via intermediate speech synthesis.

In this work, we make two main contributions. First, we introduce **L-Flow**, a longitudinal speech generation framework that produces progression-consistent speech conditioned on time and clinical state. Second, we demonstrate that L-Flow–generated speech provides more effective data augmentation for downstream clinical modeling than signal-level augmentation, general-purpose text-to-speech (TTS) systems, and prior approaches for pathological speech generation. We evaluate our method on two real-world longitudinal clinical speech datasets spanning ALS and PD, and show consistent improvements in progression modeling and clinical severity prediction. The demo and code have been released[1].

2 Related Work

2.1 Longitudinal Data Analysis in Neurodegenerative Disorders

Longitudinal studies aim to understand how conditions evolve over extended periods, often months or years, by repeatedly collecting data from the same individual at multiple sessions. This setting is suitable for studying behavioral changes within neurodegenerative disorders, which can take years to show disease progression or advancement, in contrast to cross-sectional studies that typically observe each individual only once within a short time window. Prior work has largely focused on non-speech biomarkers, including neuroimaging, brain morphology, and fluid biomarkers [13, 14]. Large cohort resources such as ADNI [15] and OASIS [16] have enabled extensive research on disease trajectory modeling and future progression prediction [17, 18, 19].

Speech has emerged as a popular biomarker for research on neurodegenerative disorder. Although many studies and benchmark challenges have advanced speech-based analysis for disorders such as AD [20, 21, 22] and PD [23, 24, 25], most existing work uses a single speech sample per participant for classification or regression. Longitudinal speech modeling remains relatively underexplored, largely due to limited dataset availability. Recent work has begun modeling ALS progression from repeated speech recordings [26], and initiatives such as SpeechDx [27] and VOC-ALS [28] are collecting longitudinal speech data, although access remains limited and some collections are still ongoing.

Across modalities, longitudinal data analysis faces several common challenges: observations are sparse and irregularly sampled, progression trajectories vary substantially across individuals, and severe or rapidly progressing cases are often underrepresented [29, 30, 31, 32, 33].

[1] https://github.com/coolsoda/speech_synthesis

2.2 Speech Generation for Clinical Applications

Recent progress in speech generation has been driven by large-scale TTS models with increasingly strong conditional generation capabilities [34]. For instance, diffusion-based methods have demonstrated that high-quality, expressive speech can be synthesized in flexible conditioning settings [35, 36]. Non-autoregressive and codec-based generation frameworks have enhanced the scalability and controllability of zero-shot speech synthesis [37]. More recently, flow-matching-based TTS has emerged as an efficient alternative for generating high-fidelity speech, eliminating the need for complex alignment or duration modeling [38, 39, 40, 41, 42, 43, 44].

Speech synthesis has also been used as a data augmentation strategy for clinical applications, particularly when pathological speech datasets are limited or imbalanced. Prior studies have used TTS and voice conversion models to synthesize pathological or dysarthric speech samples, improving downstream recognition or classification tasks [45, 46, 47].

However, current speech generation research for clinical applications typically operates in static environments, where synthesis relies on a single observed state, such as a speaker's current style, emotional state, or disorder severity [8]. In contrast, our work treats disease progression as a generative condition. Rather than conditioning on a single data point, we synthesize speech from a patient's longitudinal history. This method allows us to fill in sparse longitudinal trajectories and provide valuable augmentation for subsequent clinical speech modeling.

3 Method

Following recent fully non-autoregressive flow-matching TTS systems, we build L-Flow on a text-guided speech infilling framework and extend it to the longitudinal setting. L-Flow consists of two main components: a **longitudinal progression encoder** and a **conditional flow-based speech generator**. The progression encoder maps a subject's historical recordings and a query time point to a query-specific latent progression state. This representation captures both subject-specific vocal characteristics and the temporal dynamics of disease progression. The conditional generator then synthesizes speech by conditioning on this progression state, enabling generation that is consistent with both the individual speaker and their expected clinical trajectory.

3.1 Training

For each subject i, we observe a sequence of speech recording sessions:

$$\mathcal{H}_i = \{(x_i^{(k)}, t_i^{(k)}, s_i^{(k)}, y_i^{(k)})\}_{k=1}^{T_i}, \tag{1}$$

where $x_i^{(k)}$ is the speech recording, $t_i^{(k)}$ is the recording time, $s_i^{(k)}$ is the transcript, and $y_i^{(k)} \in [0, 1]$ is the normalized clinical progression score.

During training, we sample one recording session q as the target and use the remaining sessions $\mathcal{H}_i \setminus \{q\}$ as longitudinal context. Let $m_i^{(q)} \in \mathbb{R}^{F \times N}$ denote the mel-spectrogram extracted from the target speech. Following conditional flow matching, we define a linear path between Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ and the target mel-spectrogram:

$$\psi_\sigma = (1 - \sigma)\epsilon + \sigma m_i^{(q)}, \quad \sigma \in [0, 1], \tag{2}$$

where σ denotes the flow step. The model is trained with masked speech reconstruction. It receives the partially observed target $(1 - \mathbf{m}) \odot m_i^{(q)}$, where $\mathbf{m} \in \{0, 1\}^{F \times N}$ is a binary mask and $\mathbf{m} = 1$ indicates the region to be reconstructed (i.e., masked out from the input).

For the text condition, the target transcript $s_i^{(q)}$ is converted into a character sequence and padded with filler tokens $\langle F \rangle$ to match the mel length, yielding $p_i^{(q)}$. In addition to this text-guided speech-infilling setup, L-Flow conditions on the subject's longitudinal context through a progression encoder:

$$z_{i,\text{prog}}^{(q)} = E_{\text{prog}}(\mathcal{H}_i \setminus \{q\}, t_i^{(q)}). \tag{3}$$

This representation summarizes the subject-specific progression state at the target recording time based on the available history. The reference speech $r_i^{(q)}$ is selected by temporal proximity:

$$k^* = \arg\min_k |t_i^{(q)} - t_i^{(k)}|, \quad r_i^{(q)} = x_i^{(k^*)}. \tag{4}$$

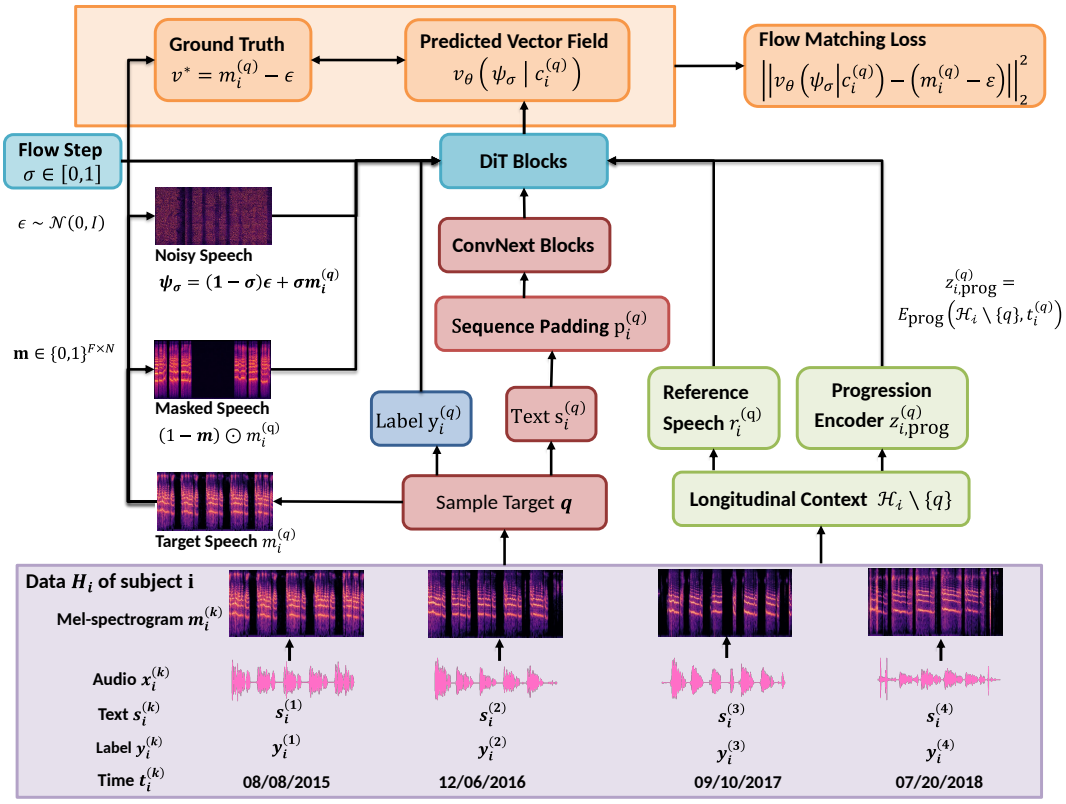

Figure 2: Overview of the L-Flow training; the $c_i^{(q)}$ in the predicted vector field represents all the conditions passed into the diffusion transformer (DiT) blocks.

When two sessions are equally close, we select the earlier recording, reflecting the ordinal progression assumption that later recordings are expected to show equal or more advanced impairment [48].

As shown in Figure 2, the model is trained to reconstruct the masked target region conditioned on the unmasked acoustic context, text, reference speech, and progression state:

$$P\big(\mathbf{m} \odot m_i^{(q)} \mid (1-\mathbf{m}) \odot m_i^{(q)}, p_i^{(q)}, r_i^{(q)}, z_{i,\text{prog}}^{(q)}, y_i^{(q)}\big). \tag{5}$$

Although training uses partial masks, inference performs full-sample generation by masking the entire target sequence.

3.2 Inference

At inference, the model is given a subject's observed history $\mathcal{H}_i$ and a query time $t_i^{(q)}$ within the temporal span of the observed recording sessions. Our primary use case is intermediate trajectory completion, where $t_i^{(a)} < t_i^{(q)} < t_i^{(b)}$ for two neighboring observed sessions a and b.

The progression encoder first produces a query-specific progression state:

$$z_{i,\text{prog}}^{(q)} = E_{\text{prog}}(\mathcal{H}_i, t_i^{(q)}). \tag{6}$$

The text condition $p_i^{(q)}$ is obtained from the predefined task transcript for fixed-task datasets, or provided as a prompt for spontaneous-speech datasets.

To generate speech, we start from Gaussian noise ϵ and solve the ODE defined by the learned vector field:

$$\frac{d\psi_\sigma}{d\sigma} = v_\theta(\psi_\sigma \mid p_i^{(q)}, r_i^{(q)}, z_{i,\text{prog}}^{(q)}, \tilde{y}_i^{(q)}), \quad \sigma : 0 \to 1. \tag{7}$$

The resulting mel-spectrogram ψ_1 is converted into waveform audio using a pre-trained HiFi-GAN vocoder [49]. The soft progression condition $\tilde{y}_i^{(q)}$ is computed by interpolation between neighboring observed sessions, as described in Eq. 9. The inference process of L-Flow is explained in Appendix C.

3.3 Progression-Aware Speech Generation

L-Flow builds on recent alignment-free flow-matching TTS systems [50, 40, 39]. Its speech generator uses a latent Diffusion Transformer (DiT) backbone [51, 52] with zero-initialized adaptive Layer Normalization. We introduce a longitudinal progression encoder that summarizes how a subject's speech evolves over time and infers a latent progression state for a queried time point. Each observed recording is first converted into a compact representation at the recording level that combines acoustic information, clinical severity, and relative temporal information. These representations are then aggregated by a 4-layer Transformer encoder The final hidden state of the query-time token is taken as the progression representation $z_{i,\mathrm{prog}}^{(q)}$.

The inferred progression state is continuous, which allows the model to represent intermediate disease states that may not correspond exactly to any observed recording. Observed clinical scores are used only at recorded sessions as ordinal anchors, while intermediate synthetic samples receive soft progression targets.

We generate synthetic samples within each interval between neighboring recording sessions $(t_i^{(a)}, t_i^{(b)})$. The number of generated samples is determined by

$$M(a,b) = \min\left(M_{\max}, \max\left(0, \left\lfloor \frac{t_i^{(b)} - t_i^{(a)}}{h} \right\rfloor - 1\right)\right), \tag{8}$$

where h is the desired temporal resolution and $M_{\max}$ limits excessive generation over large temporal gaps. For a queried time point $t_i^{(q)}$ between sessions a and b, the corresponding soft progression target is defined by temporal interpolation:

$$\tilde{y}_i^{(q)} = (1-\alpha_i^{(q)})y_i^{(a)} + \alpha_i^{(q)} y_i^{(b)}, \quad \alpha_i^{(q)} = \frac{t_i^{(q)} - t_i^{(a)}}{t_i^{(b)} - t_i^{(a)}}. \tag{9}$$

The model is trained with a combination of the standard flow-matching generation loss and progression-aware regularization:

$$\mathcal{L} = \mathcal{L}_{\mathrm{gen}} + \lambda_{\mathrm{prog}}\mathcal{L}_{\mathrm{prog}} + \lambda_{\mathrm{ord}}\mathcal{L}_{\mathrm{ord}}. \tag{10}$$

Here, $\mathcal{L}_{\mathrm{gen}}$ denotes the conditional flow-matching loss, $\mathcal{L}_{\mathrm{prog}}$ encourages temporally nearby progression states to remain close, and $\mathcal{L}_{\mathrm{ord}}$ encourages the latent states to preserve the ordinal clinical severity structure. Expanded details are discussed in Appendix A.

After training, L-Flow is applied only to the training split to generate intermediate speech samples, producing the augmented training set

$$\mathcal{D}_{\mathrm{aug}} = \mathcal{D}_{\mathrm{real}} \cup \mathcal{D}_{\mathrm{syn}}. \tag{11}$$

The same generation rule and augmentation budget are applied to all baseline generators for fair comparison.

4 Experimentation

4.1 Datasets

ALS-TDI. The ALS Therapy Development Institute (ALS-TDI) has collected a large-scale multimodal longitudinal dataset for studying ALS [53, 54]. Speech recordings were collected remotely via phone calls, where participants repeated the sentence "I owe you a yoyo today" five times. The dataset contains 19,557 recordings from 1,100 participants. Clinical labels are based on the ALSFRS-R speech subscore, a 5-point ordinal scale ranging from 4 (normal speech) to 0 (complete loss of useful speech) [55].

ParkCeleb. The Parkinson Celebrity (ParkCeleb) dataset consists of spontaneous speech samples from 80 public figures collected from online platforms for studying Parkinson's disease progression [56]. Each subject is associated with a documented year of diagnosis, with speech samples spanning up to 30 years before and 20 years after diagnosis. We focus on the 40 subjects with documented PD diagnoses. The dataset contains 8,552 recordings sampled at 16 kHz. Labels are defined according to diagnosis timing, with pre-diagnosis recordings labeled as 0 and post-diagnosis recordings labeled as 1.

4.2 Data Preprocessing

After removing incomplete utterances and samples with missing labels, we retained 5,605 recordings from 426 participants for ALS-TDI and 682 recordings from 40 PD subjects for ParkCeleb. ALS-TDI subjects contributed an average of 13.16 recordings over 407.86 days, with an average recording duration of 17.3 seconds. ParkCeleb subjects span longer trajectories, with a mean observation span of 9,368 days, an average gap of 826 days between recordings, and an average recording duration of 25.3 seconds. All audio samples are at 16 kHz, and both datasets have been fully anonymized by the providers, with no identity information shared.

We adopt an **80/20 subject-level split** to prevent leakage across repeated recordings from the same individual. Training subjects are used to train L-Flow and all generative baselines, while test subjects are held out entirely and used only for downstream evaluation.

To unify supervision across datasets, labels are mapped to $[0, 1]$, where larger values indicate greater impairment. For ALS-TDI, $y_i^* = (4 - y_i)/4$. For ParkCeleb, pre- and post-diagnosis recordings are mapped to 0 and 1, respectively. For synthesized samples between neighboring sessions a and b, we compute the soft target $\tilde{y}_i^{(q)}$ following Eq. 9. This enables consistent use of real and synthetic samples in downstream modeling.

4.3 Baselines

We compare L-Flow with eight baselines grouped into three categories. The first category includes traditional signal-level augmentation methods: speed perturbation [57] and vocal tract length perturbation (VTLP) [58]. These methods modify existing training recordings without modeling longitudinal disease progression.

The second category includes four state-of-the-art speech synthesis systems with strong voice cloning and text-to-speech capabilities: OmniVoice [59], Qwen3-TTS [60], VoxCPM2 [61], and XTTS-v2 [62]. These models are fine-tuned on the same training subjects as L-Flow but do not explicitly encode subject-specific longitudinal progression.

The third category includes pathology-oriented speech synthesis models: Text-to-Dysarthric-Speech (TTDS) [63] and Severity-Controllable Pathological TTS (SCP-TTS) [64], a pathology-aware extension of GradTTS. These baselines incorporate pathological or severity-related controls, but they do not model irregular longitudinal trajectories across multiple data collection sessions. Implementation and fine-tuning details for all baselines are provided in Appendix E.

4.4 Experimental Setup

We conduct two main experiments in this work. The first evaluates the quality of speech synthesis and data augmentation, focusing on whether the generated speech is intelligible, speaker-consistent, and progression-consistent. The second evaluates downstream utility, examining whether the synthetic speech generated by our method improves visit-level clinical prediction when used to augment the training set.

Synthetic speech is generated only for subjects in the training set. For each interval between consecutive recording sessions (t_a, t_b), the number of synthetic samples $M(a, b)$ follows Eq. 8. We set $(h, M_{\max})$ to $(30 \text{ days}, 2)$ for ALS-TDI and $(800 \text{ days}, 3)$ for ParkCeleb to ensure it aligns with the average gap between consecutive sessions within each dataset, and also to ensure not too many synthetic samples are generated, leading to error accumulation. Query timestamps are uniformly spaced within each interval, and soft targets are assigned by Eq. 9.

	ALS-TDI								ParkCeleb							
	WER ↓				SIM ↑				WER ↓				SIM ↑			
Method	Wsp	HuB	Vib	QASR	ECA	WLM	WSp	Tit	Wsp	HuB	Vib	QASR	ECA	WLM	WSp	Tit
Speed perturbation	**0.19**	**0.24**	**0.21**	**0.20**	**0.79**	**0.79**	**0.81**	0.80	0.31	0.39	0.31	0.32	0.70	0.67	0.69	0.65
VTLP	0.24	0.30	0.29	0.30	0.75	0.72	0.74	0.73	0.36	0.42	0.40	0.37	0.61	0.58	0.60	0.58
OmniVoice	0.30	0.36	0.32	0.31	0.69	0.67	0.68	0.66	0.30	0.35	0.32	0.31	0.62	0.59	0.61	0.60
Qwen3-TTS	0.24	0.29	0.25	0.24	0.74	0.71	0.73	0.70	**0.25**	**0.30**	0.27	**0.26**	0.65	0.62	0.63	0.64
VoxCPM2	0.25	0.30	0.26	0.25	0.73	0.70	0.72	0.69	0.27	0.32	0.29	0.28	0.66	0.63	0.65	0.61
XTTS-v2	0.33	0.39	0.35	0.34	0.63	0.61	0.67	0.60	0.35	0.41	0.38	0.36	0.55	0.54	0.57	0.55
TTDS	0.37	0.42	0.38	0.37	0.71	0.69	0.70	0.68	0.28	0.33	0.30	0.29	0.66	0.64	0.64	0.62
SCP-TTS	0.23	0.28	0.24	0.23	0.74	0.72	0.74	0.71	0.34	0.40	0.36	0.35	0.68	0.65	0.68	0.64
L-Flow	0.20	0.31	0.27	0.29	**0.79**	0.76	0.78	**0.81**	0.27	0.31	**0.25**	0.29	**0.71**	**0.68**	**0.71**	**0.69**

Table 1: Speech generation quality on ALS-TDI and ParkCeleb. We report word error rate (WER; lower is better) for linguistic content preservation and speaker similarity (SIM; higher is better) for speaker identity preservation. Note that these metrics evaluate basic generation validity rather than progression or clinical fidelity.

	Pairwise Acc. ↑		Kendall's τ ↑		Spearman's ρ ↑	
Method	**ALS-TDI**	**ParkCeleb**	**ALS-TDI**	**ParkCeleb**	**ALS-TDI**	**ParkCeleb**
Speed perturbation	0.55	0.51	0.13	0.09	0.18	0.14
VTLP	0.56	0.49	0.14	0.08	0.19	0.13
OmniVoice	0.61	0.57	0.20	0.16	0.28	0.23
Qwen3-TTS	0.64	0.62	0.24	0.21	0.33	0.30
VoxCPM2	0.63	0.60	0.22	0.19	0.31	0.27
XTTS-v2	0.59	0.54	0.18	0.13	0.26	0.20
TTDS	0.68	0.65	0.30	0.25	0.42	0.36
SCP-TTS	0.71	0.65	0.36	0.30	0.49	0.42
L-Flow	**0.79**	**0.74**	**0.49**	**0.40**	**0.64**	**0.56**

Table 2: Progression consistency of synthesized longitudinal speech on ALS-TDI and ParkCeleb. Pairwise Accuracy measures whether generated samples follow the correct relative progression order, while Kendall's tau and Spearman's rho measure agreement between the generated and reference longitudinal rankings; higher values are better for all metrics.

To evaluate whether synthetic data benefits different model families, we use Ridge Regression [65], XGBoost [66], and a two-layer MLP regressor [67]. Each model is trained on $\mathcal{D}_{\text{aug}} = \mathcal{D}_{\text{real}} \cup \mathcal{D}_{\text{syn}}$ and evaluated on held-out test subjects.

4.5 Evaluation Metrics

Speech Validity. We evaluate intelligibility using Word Error Rate (**WER**) and speaker preservation using speaker similarity (**SIM**). WER is computed by transcribing synthesized audio with Whisper-v3 (Wsp) [68], HuBERT-Large(Hub) [69], VibeVoice (Vib) [70], and Qwen3-ASR (QASR) [71], and comparing the outputs with the reference text condition used for generation. SIM is measured by the time-weighted cosine similarity between synthetic embeddings and the preceding and succeeding real recordings. Embeddings are extracted using four different modern speaker recognition models: ECAPA-VoxCeleb (ECA) [72], WavLM-Base+ (WLM) [73], WeSpeaker (WSp) [74], and TitaNet (Tit) [75].

Progression Consistency. We measure whether synthesized speech preserves longitudinal ordering using **Pairwise Accuracy**, **Kendall's Tau** (τ), and **Spearman's Rho** (ρ). To avoid overfitting, we use a 5-fold subject-level out-of-fold scheme on the training split: an XGBoost regressor is trained on real recordings from four folds and predicts normalized severity scores for synthesized samples in the held-out fold. We compare these predicted scores with the temporal ordering implied by query timestamps. Higher values indicate stronger progression consistency.

Method	Ridge Regression			XGBoost Regressor			MLP Regressor		
	MAE ↓	RMSE ↓	Spearman ↑	MAE ↓	RMSE ↓	Spearman ↑	MAE ↓	RMSE ↓	Spearman ↑
Original	0.35	0.42	0.48	0.32	0.40	0.51	0.34	0.41	0.50
Speed perturbation	0.33	0.40	0.52	0.31	0.39	0.54	0.30	0.38	0.56
VTLP	0.38	0.46	0.42	0.35	0.43	0.45	0.36	0.44	0.44
OmniVoice	0.28	0.36	0.59	0.24	0.32	0.62	0.29	0.37	0.58
Qwen3-TTS	0.22	0.30	0.65	0.20	0.28	0.68	0.23	0.31	0.64
VoxCPM2	0.25	0.33	0.61	0.22	0.30	0.64	0.26	0.34	0.60
XTTS-v2	0.31	0.39	0.54	0.28	0.36	0.57	0.32	0.40	0.53
TTDS	0.21	0.29	0.67	0.19	0.27	0.70	0.18	0.26	0.71
SCP-TTS	0.19	0.27	**0.72**	**0.16**	**0.24**	**0.75**	0.19	0.27	0.70
L-Flow	**0.17**	**0.25**	0.71	0.18	0.26	0.73	**0.16**	**0.24**	**0.76**

Table 3: The session-level regression performance on the held-out test set from the ALS TDI dataset. We evaluate Ridge Regression, XGBoost, and MLP under the same subject-level splits, reporting MAE and RMSE (lower is better) and Spearman correlation (higher is better).

Method	Ridge Regression			XGBoost Regressor			MLP Regressor		
	MAE ↓	RMSE ↓	Spearman ↑	MAE ↓	RMSE ↓	Spearman ↑	MAE ↓	RMSE ↓	Spearman ↑
Original	0.44	0.52	0.39	0.41	0.49	0.42	0.42	0.50	0.40
Speed perturbation	0.42	0.50	0.41	0.40	0.48	0.44	0.41	0.49	0.43
VTLP	0.46	0.55	0.35	0.43	0.51	0.38	0.45	0.53	0.37
OmniVoice	0.38	0.46	0.48	0.33	0.41	0.51	0.36	0.44	0.47
Qwen3-TTS	0.31	0.39	0.55	**0.25**	0.35	0.59	0.30	0.38	0.54
VoxCPM2	0.34	0.42	0.51	0.27	0.37	0.55	0.33	0.41	0.51
XTTS-v2	0.40	0.48	0.44	0.37	0.45	0.47	0.39	0.47	0.45
TTDS	0.29	0.37	0.57	0.28	0.36	0.58	0.26	0.34	0.60
SCP-TTS	0.30	0.38	0.56	0.30	0.38	0.56	0.28	0.36	0.58
L-Flow	**0.24**	**0.32**	**0.64**	0.29	**0.33**	**0.63**	**0.22**	**0.30**	**0.67**

Table 4: The session-level regression performance on the held-out test set from the ParkCeleb dataset.

Downstream Utility. We formulate downstream clinical modeling as a regression task in the normalized severity space and report Mean Absolute Error (**MAE**), Root Mean Squared Error (**RMSE**), and **Spearman Correlation**.

4.6 Results

Table 1 present the results for intelligibility and speaker similarity across both datasets. Signal-level augmentation methods, particularly speed perturbation, surprisingly yield strong WER and SIM, especially on the ALS TDI dataset. However, as the speech becomes more spontaneous, like in the ParkCeleb dataset, the performance tends to decline. Among all modern TTS systems, Qwen3 demonstrates competitive performance across both datasets, espeically on ASR. Overall, L-Flow achieves a competitive WER while consistently obtaining the highest speaker similarity scores among generative methods in both datasets.

Table 2 evaluates whether the synthesized speech adheres to the expected longitudinal progression ordering. L-Flow achieves the best results across both datasets and all three rank-based metrics. This trend is also observed in ParkCeleb.

By comparing the results from Table 1 and Table 2, we observe that while signal-level augmentation can generate samples that closely resemble real recordings, it does not effectively model how speech changes over time as a disorder progresses. Modern TTS systems can produce intelligible speech, but they lack specific mechanisms to account for individual disease progression. In contrast, pathology-aware systems such as TTDS and SCP-TTS outperform general TTS baselines, indicating that incorporating severity control is a valuable feature, even statically. Appendix I presents additional analyses of cross-validation stability and subject-level bootstrap significance.

Tables 3 and 4 report downstream regression performance on held-out real test subjects. We use **Original** to refer to the setting where downstream regressors are trained only on the original training recordings $\mathcal{D}_{\text{real}}$, without synthetic augmentation. Across both datasets, augmenting the training set as $\mathcal{D}_{\text{aug}} = \mathcal{D}_{\text{real}} \cup \mathcal{D}_{\text{syn}}$ generally improves downstream prediction compared to training on

	ALS-TDI		ParkCeleb	
Setting	$\rho\uparrow$	**MAE** $\downarrow$	$\rho\uparrow$	**MAE** $\downarrow$
Full model	**0.64**	**0.18**	**0.56**	**0.29**
w/o global progression modeling	0.38	0.22	0.31	0.35
w/o severity conditioning	0.51	0.18	0.44	0.29
w/o local speaker anchor	0.62	0.16	0.54	0.27
Reduced augmentation	0.59	0.20	0.50	0.32

Table 5: Main ablation results. We report progression consistency using Spearman's ρ (Table 2) and downstream utility using MAE on the held-out test set with XGBoost (Table 3 and Table 4) as the primary regressor.

$\mathcal{D}_{\text{real}}$ alone, although some augmentation methods can degrade performance. L-Flow achieves robust and consistent performance across different regression models and outperforms most baseline augmentation methods.

On ALS-TDI, L-Flow achieves the best performance under Ridge Regression and MLP, and remains competitive under XGBoost, where SCP-TTS performs slightly better. This pattern suggests that L-Flow-generated samples are broadly useful across different predictor families rather than benefiting only a single downstream model.

On ParkCeleb, L-Flow shows strong overall downstream utility. It achieves the best Ridge Regression and MLP results across MAE, RMSE, and Spearman correlation. Under XGBoost, Qwen3-TTS obtains the lowest MAE, while L-Flow achieves the best RMSE and Spearman correlation. This indicates that L-Flow may better preserve the relative ordering of progression, even when another baseline obtains a slightly lower pointwise error.

We further analyze progression-related acoustic characteristics, including articulation rate, pausing, jitter, and pitch variability, in Appendix H. Overall, these results suggest that, rather than simply increasing the number of audio samples, L-Flow-based augmentation provides additional temporal and severity-related variation, helping downstream models learn from augmented clinical speech data. Appendix J provides additional analyses of computational cost and negative augmentation effects.

4.7 Ablation Studies

We ablate the main components of L-Flow and evaluate each variant using two complementary metrics: **Progression Spearman (ρ)**, which measures temporal progression consistency, and **Downstream MAE**, which evaluates clinical prediction performance. Results are summarized in Table 5.

There are very interesting and valuable findings from the ablation studies. First, removing the progression encoder E_{prog} results in the largest degradation across both datasets, substantially reducing progression consistency and increasing downstream error. This confirms the importance of modeling the subject's longitudinal history in our cases. Second, removing ordinal severity labels reduces progression consistency, but with a limited effect on downstream MAE. This suggests that severity conditioning mainly provides ordering constraints that help align generated speech with disease progression, but is less critical for pointwise regression accuracy.

We also noticed that removing the reference speech $r_i^{(q)}$ slightly reduces progression consistency but improves downstream MAE. This suggests a possible trade-off: while the speaker anchor helps preserve subject-specific vocal characteristics, it may introduce additional variability that is less well aligned with the regression objective. Finally, reducing the augmentation budget to at most one intermediate sample per interval ($M_{\max} = 1$) weakens both progression consistency and downstream performance. This shows that denser synthetic trajectories provide more useful supervision for learning temporal and severity-related variation in our cases. Paired bootstrap tests further show the importance of global progression modeling in improving downstream prediction performance, as detailed in Appendix I.2.

5 Conclusions

In this work, we presented L-Flow, a longitudinal flow-matching framework for progression-aware clinical speech generation. By conditioning on each subject's historical speech recordings, irregular time gaps, and normalized clinical progression information, L-Flow synthesizes intermediate speech samples that preserve speaker characteristics and progression consistency. Experiments on ALS-TDI for amyotrophic lateral sclerosis and ParkCeleb for Parkinson's disease show that L-Flow achieves competitive speech validity and stronger progression consistency than traditional augmentation methods, modern speech synthesis systems, and pathology-specific speech generation baselines. Downstream results further show that L-Flow-generated augmentation improves regression of clinical severity across different model families. We hope this work provides an initial step toward controllable longitudinal speech generation for neurodegenerative disease modeling, where accumulated speech recordings over time may support more accurate health tracking and monitoring.

6 Limitations and Future Perspectives

This work has several limitations. First, due to data-sharing constraints and the limited availability of longitudinal clinical speech datasets for neurodegenerative disorders, our experiments are restricted to two English-language datasets, which may limit generalizability to other languages and clinical populations. Second, although we evaluate synthesized speech using linguistic, acoustic, and progression-oriented metrics, automatic evaluations cannot fully capture the perceptual and clinical realism of atypical speech. In particular, WER and speaker similarity primarily assess intelligibility and speaker preservation, rather than progression fidelity. A controlled human evaluation, potentially involving clinical experts, would provide complementary validation. Third, our generative model is trained from scratch, and exploring pretrained initialization may further improve speech quality but would require careful adaptation to preserve disease-related characteristics. Finally, while L-Flow focuses on interpolation within observed time ranges, reliable extrapolation to future disease states remains challenging and requires further validation with richer longitudinal data.

Looking forward, progression-aware speech synthesis may provide broader benefits beyond data augmentation. By generating speech variations that reflect disease progression, such models could support the development of more robust speech recognition and spoken language understanding systems for dysarthric or neurologically impaired speakers. As more diverse and multilingual longitudinal datasets become available, integrating pretrained generative models, progression-aware conditioning, and downstream clinical and assistive technologies represents a promising direction for future research.

Acknowledgments and Disclosure of Funding

This work was supported in part by the University Graduate School (UGS) Dissertation Fellowship at Florida International University. The authors declare no competing interests.

References

[1] Richard NL Lamptey, Bivek Chaulagain, Riddhi Trivedi, Avinash Gothwal, Buddhadev Layek, and Jagdish Singh. A review of the common neurodegenerative disorders: current therapeutic approaches and the potential role of nanotherapeutics. *International journal of molecular sciences*, 23(3):1851, 2022.

[2] Adolfo M García, Jessica de Leon, Boon Lead Tee, Damián E Blasi, and Maria Luisa Gorno-Tempini. Speech and language markers of neurodegeneration: a call for global equity. *Brain*, 146(12):4870–4879, 2023.

[3] Jonathan Heitz, Ines M Engler, and Nicolas Langer. Towards a speech-based digital biomarker for cognitive impairment: speech as a proxy for cognitive assessment. *npj Digital Medicine*, 2026.

[4] Siye Chen, Linghan Li, Shuyu Han, Wei Luo, Wenxia Wang, Yufan Yang, Xiaomeng Wang, Wenmin Zhang, Mo Chen, and Zhiwen Wang. Review of voice biomarkers in the screening of neurodegenerative diseases. *Interdisciplinary Nursing Research*, 3(3):190–198, 2024.

[5] Enshi Zhang, Rafael Trujillo, and Christian Poellabauer. The mersa dataset and a transformer-based approach for speech emotion recognition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13960–13970, 2024.

[6] Michael Neumann, Hardik Kothare, and Vikram Ramanarayanan. Multimodal speech biomarkers for remote monitoring of als disease progression. *Computers in Biology and Medicine*, 180:108949, 2024.

[7] Johnny Tam, Christine Weaver, Oliver Watts, Siddharthan Chandran, Suvankar Pal, Rowling Speech Consortium, et al. Anne rowling neurological speech corpus: clinically annotated longitudinal dataset for developing speech biomarkers in neurodegenerative disorders. In *Proc. Interspeech 2025*, pages 5693–5697, 2025.

[8] Si-Ioi Ng, Lingfeng Xu, Ingo Siegert, Nicholas Cummins, Nina R Benway, Julie Liss, and Visar Berisha. An end-to-end overview of clinical speech ai. *IEEE Transactions on Audio, Speech and Language Processing*, 2026.

[9] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE, 2015.

[10] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 776–780. IEEE, 2017.

[11] Guoguo Chen, Shuzhou Chai, Guanbo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, et al. Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio. *arXiv preprint arXiv:2106.06909*, 2021.

[12] Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus. In *Proceedings of the twelfth language resources and evaluation conference*, pages 4218–4222, 2020.

[13] Hyuna Cho, Ziquan Wei, Seungjoo Lee, Tingting Dan, Guorong Wu, and Won Hwa Kim. Conditional diffusion with ordinal regression: Longitudinal data generation for neurodegenerative disease studies. In *The Thirteenth International Conference on Learning Representations*, 2025.

[14] Martina Valletta, Nils Briel, Idil Yuksekel, Michelle Barboure, Anna Coward, Julie FH De Houwer, Ayesha Fawad, Alberto González-Mayoral, Gianmarco Iaccarino, Francisco Martínez-Dubarbie, et al. Fluid biomarkers for neurodegenerative diseases: a comprehensive update. *Alzheimer's Research & Therapy*, 2025.

[15] Susanne G Mueller, Michael W Weiner, Leon J Thal, Ronald C Petersen, Clifford Jack, William Jagust, John Q Trojanowski, Arthur W Toga, and Laurel Beckett. The alzheimer's disease neuroimaging initiative. *Neuroimaging Clinics*, 15(4):869–877, 2005.

[16] Pamela J LaMontagne, Tammie LS Benzinger, John C Morris, Sarah Keefe, Russ Hornbeck, Chengjie Xiong, Elizabeth Grant, Jason Hassenstab, Krista Moulder, Andrei G Vlassenko, et al. Oasis-3: longitudinal neuroimaging, clinical, and cognitive dataset for normal aging and alzheimer disease. *medrxiv*, pages 2019–12, 2019.

[17] Etienne Maheux, Igor Koval, Juliette Ortholand, Colin Birkenbihl, Damiano Archetti, Vincent Bouteloup, Stéphane Epelbaum, Carole Dufouil, Martin Hofmann-Apitius, and Stanley Durrleman. Forecasting individual progression trajectories in alzheimer's disease. *Nature Communications*, 14(1):761, 2023.

[18] Suixia Zhang, Jing Yuan, Yu Sun, Fei Wu, Ziyue Liu, Feifei Zhai, Yaoyun Zhang, Judith Somekh, Mor Peleg, Yi-Cheng Zhu, et al. Machine learning on longitudinal multi-modal data enables the understanding and prognosis of alzheimer's disease progression. *Iscience*, 27(7), 2024.

[19] Nana Nyarko Brenya Appiah Kubi and Sajid Nazir. Dementia prediction with multimodal clinical and imaging data. *International Journal of Information Technology*, 17(1):5–16, 2025.

[20] Saturnino Luz, Fasih Haider, Davida Fromm, Ioulietta Lazarou, Ioannis Kompatsiaris, and Brian MacWhinney. An overview of the adress-m signal processing grand challenge on multilingual alzheimer's dementia recognition through spontaneous speech. *IEEE Open Journal of Signal Processing*, 5:738–749, 2024.

[21] Kaichen Jia, Jinpeng Li, Ke Li, and Wei-Qiang Zhang. Whisper-based multilingual alzheimer's disease detection and improvements for low-resource language. In *Proc. Interspeech*, pages 549–553, 2025.

[22] Chanwoo Park, Anna Seo Gyeong Choi, Sunghye Cho, and Chanwoo Kim. Reasoning-based approach with chain-of-thought for alzheimer's detection using speech and large language models. *arXiv preprint arXiv:2506.01683*, 2025.

[23] Terry Yi Zhong, Esther Janse, Cristian Tejedor-Garcia, Louis ten Bosch, and Martha Larson. Evaluating the usefulness of non-diagnostic speech data for developing parkinson's disease classifiers. *arXiv preprint arXiv:2505.18722*, 2025.

[24] Emmy Postma and Cristian Tejedor-Garcia. Evaluating the effectiveness of pre-trained audio embeddings for classification of parkinson's disease speech data. *arXiv preprint arXiv:2506.02078*, 2025.

[25] David Gimeno-Gómez, Rubén Solera-Ureña, Anna Pompili, Carlos-D Martínez-Hinarejos, Rita Cardoso, Isabel Guimarães, Joaquim J Ferreira, and Alberto Abad. On the relevance of clinical assessment tasks for the automatic detection of parkinson's disease medication state from speech. *arXiv preprint arXiv:2505.15378*, 2025.

[26] Liming Wang, Yuan Gong, Nauman Dawalatabad, Marco Vilela, Katerina Placek, Brian Tracey, Yishu Gong, Alan Premasiri, Fernando Vieira, and James Glass. Automatic prediction of amyotrophic lateral sclerosis progression using longitudinal speech transformer. *arXiv preprint arXiv:2406.18625*, 2024.

[27] Lampros Kourtis. Speechdx: A gold-standard speech-and-language dataset for prognostic ad biomarker development. *Alzheimer's & Dementia*, 21:e104638, 2025.

[28] Raffaele Dubbioso, Myriam Spisto, Laura Verde, Valentina Virginia Iuzzolino, Gianmaria Senerchia, Elena Salvatore, Giuseppe De Pietro, Ivanoe De Falco, and Giovanna Sannino. Voice signals database of als patients with different dysarthria severity and healthy controls. *Scientific Data*, 11(1):800, 2024.

[29] Garrett Fitzmaurice, Marie Davidian, Geert Verbeke, and Geert Molenberghs. *Longitudinal data analysis*. CRC press, 2008.

[30] Peter Diggle. *Analysis of longitudinal data*. Oxford university press, 2002.

[31] Kristen A Severson, Lana M Chahine, Luba A Smolensky, Murtaza Dhuliawala, Mark Frasier, Kenney Ng, Soumya Ghosh, and Jianying Hu. Discovery of parkinson's disease states and disease progression modelling: a longitudinal data study using machine learning. *The Lancet Digital Health*, 3(9):e555–e564, 2021.

[32] Jianchang Hu and Silke Szymczak. A review on longitudinal data analysis with random forest. *Briefings in bioinformatics*, 24(2):bbad002, 2023.

[33] Enshi Zhang, Rafael M Trujillo, and Christian Poellabauer. Participant engagement and data quality: Lessons learned from a mental wellness crowdsensing study. *Proceedings of the ACM on Human-Computer Interaction*, 9(2):1–30, 2025.

[34] Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*, 2023.

[35] Kai Shen, Zeqian Ju, Xu Tan, Yanqing Liu, Yichong Leng, Lei He, Tao Qin, Sheng Zhao, and Jiang Bian. Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers. *arXiv preprint arXiv:2304.09116*, 2023.

[36] Keon Lee, Dong Won Kim, Jaehyeon Kim, Seungjun Chung, and Jaewoong Cho. Ditto-tts: Diffusion transformers for scalable text-to-speech without domain-specific factors. *arXiv preprint arXiv:2406.11427*, 2024.

[37] Sanyuan Chen, Shujie Liu, Long Zhou, Yanqing Liu, Xu Tan, Jinyu Li, Sheng Zhao, Yao Qian, and Furu Wei. Vall-e 2: Neural codec language models are human parity zero-shot text to speech synthesizers. *arXiv preprint arXiv:2406.05370*, 2024.

[38] Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, et al. Voicebox: Text-guided multilingual universal speech generation at scale. *Advances in neural information processing systems*, 36:14005–14034, 2023.

[39] Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, JianZhao JianZhao, Kai Yu, and Xie Chen. F5-tts: A fairytaler that fakes fluent and faithful speech with flow matching. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6255–6271, 2025.

[40] Shivam Mehta, Ruibo Tu, Jonas Beskow, Éva Székely, and Gustav Eje Henter. Matcha-tts: A fast tts architecture with conditional flow matching. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11341–11345. IEEE, 2024.

[41] Wenhao Guan, Qi Su, Haodong Zhou, Shiyu Miao, Xingjia Xie, Lin Li, and Qingyang Hong. Reflow-tts: A rectified flow model for high-fidelity text-to-speech. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10501–10505. IEEE, 2024.

[42] Yiwei Guo, Chenpeng Du, Ziyang Ma, Xie Chen, and Kai Yu. Voiceflow: Efficient text-to-speech with rectified flow matching. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11121–11125. IEEE, 2024.

[43] Sefik Emre Eskimez, Xiaofei Wang, Manthan Thakker, Canrun Li, Chung-Hsien Tsai, Zhen Xiao, Hemin Yang, Zirun Zhu, Min Tang, Xu Tan, et al. E2 tts: Embarrassingly easy fully non-autoregressive zero-shot tts. In *2024 IEEE spoken language technology workshop (SLT)*, pages 682–689. IEEE, 2024.

[44] Xiaohui Sun, Ruitong Xiao, Jianye Mo, Bowen Wu, Qun Yu, and Baoxun Wang. F5r-tts: Improving flow-matching based text-to-speech with group relative policy optimization. *arXiv preprint arXiv:2504.02407*, 2025.

[45] Wing-Zin Leung, Heidi Christensen, and Stefan Goetze. Text-to-dysarthric-speech generation for dysarthric automatic speech recognition: is purely synthetic data enough? In *International Conference on Speech and Computer*, pages 203–216. Springer, 2025.

[46] Yuki Matsuzaka, Ryoichi Takashima, Chiho Sasaki, and Tetsuya Takiguchi. Data augmentation for dysarthric speech recognition based on text-to-speech synthesis. In *2022 IEEE 4th Global Conference on Life Sciences and Technologies (LifeTech)*, pages 399–400. IEEE, 2022.

[47] Mohammad Soleymanpour, Michael T Johnson, Rahim Soleymanpour, and Jeffrey Berry. Synthesizing dysarthric speech using multi-speaker tts for dysarthric speech recognition. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7382–7386. IEEE, 2022.

[48] Neil P Oxtoby, Louise-Ann Leyland, Leon M Aksman, George EC Thomas, Emma L Bunting, Peter A Wijeratne, Alexandra L Young, Angelika Zarkali, Manuela MX Tan, Fion D Bremner, et al. Sequence of clinical and neurodegeneration events in parkinson's disease progression. *Brain*, 144(3):975–988, 2021.

[49] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in neural information processing systems*, 33:17022–17033, 2020.

[50] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[52] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[53] Fernando G Vieira, Subhashini Venugopalan, Alan S Premasiri, Maeve McNally, Aren Jansen, Kevin McCloskey, Michael P Brenner, and Steven Perrin. A machine-learning based objective measure for als disease severity. *NPJ digital medicine*, 5(1):45, 2022.

[54] Anoopum S Gupta, Siddharth Patel, Alan Premasiri, and Fernando Vieira. At-home wearables and machine learning sensitively capture disease progression in amyotrophic lateral sclerosis. *Nature communications*, 14(1):5080, 2023.

[55] Jesse M Cedarbaum, Nancy Stambler, Errol Malta, Cynthia Fuller, Dana Hilt, Barbara Thurmond, Arline Nakanishi, Bdnf Als Study Group, 1A complete listing of the BDNF Study Group, et al. The alsfrs-r: a revised als functional rating scale that incorporates assessments of respiratory function. *Journal of the neurological sciences*, 169(1-2):13–21, 1999.

[56] Anna Favaro, Ankur Butala, Thomas Thebaud, Jesús Villalba, Najim Dehak, and Laureano Moro-Velázquez. Unveiling early signs of parkinson's disease via a longitudinal analysis of celebrity speech recordings. *npj Parkinson's Disease*, 10(1):207, 2024.

[57] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. Audio augmentation for speech recognition. In *Interspeech*, volume 2015, page 3586, 2015.

[58] Navdeep Jaitly and Geoffrey E Hinton. Vocal tract length perturbation (vtlp) improves speech recognition. In *Proc. ICML workshop on deep learning for audio, speech and language*, volume 117, page 21, 2013.

[59] Han Zhu, Lingxuan Ye, Wei Kang, Zengwei Yao, Liyong Guo, Fangjun Kuang, Zhifeng Han, Weiji Zhuang, Long Lin, and Daniel Povey. Omnivoice: Towards omnilingual zero-shot text-to-speech with diffusion language models. *arXiv preprint arXiv:2604.00688*, 2026.

[60] Hangrui Hu, Xinfa Zhu, Ting He, Dake Guo, Bin Zhang, Xiong Wang, Zhifang Guo, Ziyue Jiang, Hongkun Hao, Zishan Guo, et al. Qwen3-tts technical report. *arXiv preprint arXiv:2601.15621*, 2026.

[61] Yixuan Zhou, Guoyang Zeng, Xin Liu, Xiang Li, Renjie Yu, Ziyang Wang, Runchuan Ye, Weiyue Sun, Jiancheng Gui, Kehan Li, et al. Voxcpm: Tokenizer-free tts for context-aware speech generation and true-to-life voice cloning. *arXiv preprint arXiv:2509.24650*, 2025.

[62] Edresson Casanova, Kelly Davis, Eren Gölge, Görkem Göknar, Iulian Gulea, Logan Hart, Aya Aljafari, Joshua Meyer, Reuben Morais, Samuel Olayemi, et al. Xtts: a massively multilingual zero-shot text-to-speech model. *arXiv preprint arXiv:2406.04904*, 2024.

[63] Wing-Zin Leung, Mattias Cross, Anton Ragni, and Stefan Goetze. Training data augmentation for dysarthric automatic speech recognition by text-to-dysarthric-speech synthesis. *arXiv preprint arXiv:2406.08568*, 2024.

[64] Bence Mark Halpern, Wen-Chin Huang, Lester Phillip Violeta, and Tomoki Toda. Severity-controllable pathological text-to-speech synthesis for clinical applications. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 34:573–582, 2026.

[65] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[66] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[67] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[68] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR, 2023.

[69] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460, 2021.

[70] Zhiliang Peng, Jianwei Yu, Yaoyao Chang, Zilong Wang, Li Dong, Yingbo Hao, Yujie Tu, Chenyu Yang, Wenhui Wang, Songchen Xu, et al. Vibevoice-asr technical report. *arXiv preprint arXiv:2601.18184*, 2026.

[71] Xian Shi, Xiong Wang, Zhifang Guo, Yongqi Wang, Pei Zhang, Xinyu Zhang, Zishan Guo, Hongkun Hao, Yu Xi, Baosong Yang, et al. Qwen3-asr technical report. *arXiv preprint arXiv:2601.21337*, 2026.

[72] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. *arXiv preprint arXiv:2005.07143*, 2020.

[73] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, 2022.

[74] Hongji Wang, Chengdong Liang, Shuai Wang, Zhengyang Chen, Binbin Zhang, Xu Xiang, Yanlei Deng, and Yanmin Qian. Wespeaker: A research and production oriented speaker embedding learning toolkit. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[75] Nithin Rao Koluguri, Taejin Park, and Boris Ginsburg. Titanet: Neural model for speaker representation with 1d depth-wise separable convolutions and global context. In *ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 8102–8106. IEEE, 2022.

[76] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[77] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

A Training Objective

L-Flow is trained with a conditional flow-matching objective together with two progression-aware regularization terms. For a target recording session q, let $m_i^{(q)}$ denote the target mel-spectrogram and $\epsilon \sim \mathcal{N}(0, I)$ denote Gaussian noise with the same shape. We define the linear probability path:

$$\psi_\sigma = (1-\sigma)\epsilon + \sigma m_i^{(q)}, \quad \sigma \sim \mathcal{U}[0,1]. \tag{12}$$

The corresponding ground-truth vector field is:

$$v^* = \frac{d\psi_\sigma}{d\sigma} = m_i^{(q)} - \epsilon. \tag{13}$$

The conditional generation loss trains the model to predict this vector field:

$$\mathcal{L}_{\text{gen}} = \mathbb{E}_{\sigma,\epsilon,\mathbf{m}}\left[\left\| v_\theta(\psi_\sigma \mid c_i^{(q)}) - (m_i^{(q)} - \epsilon)\right\|_2^2\right], \tag{14}$$

where the conditioning set is:

$$c_i^{(q)} = \{(1-\mathbf{m}) \odot m_i^{(q)}, p_i^{(q)}, r_i^{(q)}, z_{i,\text{prog}}^{(q)}, y_i^{(q)}\}. \tag{15}$$

Here, $(1-\mathbf{m}) \odot m_i^{(q)}$ is the visible acoustic context, $p_i^{(q)}$ is the padded text condition (to align with length of mel), $r_i^{(q)}$ is the reference speech, $z_{i,\text{prog}}^{(q)}$ is the query-specific progression state, and $y_i^{(q)}$ is the observed normalized clinical score during training. For synthetic generation, $y_i^{(q)}$ is replaced by the interpolated soft progression target $\tilde{y}_i^{(q)}$.

The progression smoothness loss encourages nearby query times from the same subject to have similar latent disorder-progression states:

$$\mathcal{L}_{\text{prog}} = \sum_i \sum_{q,q'} w_{q,q'} \left\| z_{i,\text{prog}}^{(q)} - z_{i,\text{prog}}^{(q')} \right\|_2^2, \tag{16}$$

where $w_{q,q'} = \exp(-|t_i^{(q)} - t_i^{(q')}|/\tau)$ assigns larger weights to temporally closer pairs, and τ is a temperature hyperparameter.

The ordinal regularization loss encourages latent progression states to preserve the ordering implied by observed clinical severity scores. For two observed sessions q and q' from the same subject, we define:

$$\mathcal{L}_{\text{ord}} = \sum_i \sum_{q,q'} \max\left(0, \gamma - \text{sign}(y_i^{(q)} - y_i^{(q')}) \cdot g(z_{i,\text{prog}}^{(q)}, z_{i,\text{prog}}^{(q')})\right), \tag{17}$$

where $g(\cdot,\cdot)$ is a learnable scoring function that compares two progression states, and γ is a margin. This term is applied only to observed recording sessions with available clinical labels.

The final objective is:

$$\mathcal{L} = \mathcal{L}_{\text{gen}} + \lambda_{\text{prog}}\mathcal{L}_{\text{prog}} + \lambda_{\text{ord}}\mathcal{L}_{\text{ord}}. \tag{18}$$

In our experiments, λ_{prog} and λ_{ord} are selected on the validation split and kept fixed across datasets.

B Training Setup

All experiments were implemented in PyTorch [76] and trained on our server with two NVIDIA L40S GPUs, each with 48GB VRAM. The driver version is 555.42.06 and CUDA 12.5. The software environment included PyTorch 2.5.1+cu121, Torchaudio 2.5.1+cu121, Torchvision 0.20.1+cu121, Transformers 4.46.3, TRL 0.10.1, PEFT 0.12.0, BitsAndBytes 0.49.2, Accelerate 1.13.0, and Datasets 4.8.4.

Audio recordings were sampled at 16 kHz by default and converted to mel-spectrograms before training. During training, one observed recording session was randomly sampled as the target session, and the remaining sessions from the same subject were used as longitudinal context. A random contiguous temporal region of the target mel-spectrogram was masked for speech-infilling training.

The model was optimized using AdamW [77] with a learning rate of 2×10^{-4} and a batch size of 16, and was trained for 200k steps. We applied a linear learning rate warmup over the first 5k steps, followed by cosine decay. Gradient clipping with a maximum norm of 1.0 was used to stabilize training. Mixed-precision training was enabled for efficiency.

The loss weights were set to $\lambda_{\text{prog}} = 0.1$ and $\lambda_{\text{ord}} = 0.05$, selected based on validation performance and kept fixed across datasets. For all datasets, we used an 80/20 subject-level split. Synthetic samples were generated only for training subjects after L-Flow was trained. No synthetic samples from held-out test subjects were used during model training, baseline fine-tuning, or downstream regressor training.

To evaluate the utility of synthetic data, we train three types of downstream regressors: Ridge Regression, XGBoost, and a Multilayer Perceptron (MLP). All models are trained on the augmented dataset $\mathcal{D}_{\text{aug}}$ and evaluated on held-out real test subjects. The same feature extraction pipeline and train/test split are used across all methods to ensure fair comparison.

Validation and Hyperparameter Selection. We adopt an 80/20 subject-level train-test split, ensuring that recordings from the same subject do not appear in both partitions. For L-Flow hyperparameter selection, we randomly reserve 10% of the training subjects as a subject-disjoint validation set. We conduct a grid search over $\lambda_{\text{prog}} \in \{0.05, 0.1, 0.2\}$ and $\lambda_{\text{ord}} \in \{0.01, 0.05, 0.1\}$, selecting the configuration that best balances generation fidelity, measured by linguistic content and speaker preservation, and progression consistency, measured by progression-state agreement and longitudinal ordering. This yields $\lambda_{\text{prog}} = 0.1$ and $\lambda_{\text{ord}} = 0.05$.

Separately, we perform five-fold subject-level cross-validation within the training partition for downstream predictor and model selection. We assign all recordings from each subject to the same fold to prevent subject-level leakage. The held-out test subjects are excluded from both selection procedures and are used only for final evaluation.

Ridge Regression. Ridge Regression is used as a linear baseline. We use ℓ_2 regularization with coefficient $\alpha = 1.0$. The model is trained using the closed-form solution as implemented in scikit-learn. All input features are standardized to zero mean and unit variance.

XGBoost. We use XGBoost as a non-linear tree-based regressor. The model is configured with the following hyperparameters:

- Number of trees: 500
- Maximum tree depth: 6
- Learning rate: 0.05
- Subsample ratio: 0.8
- Column sampling ratio: 0.8
- Minimum child weight: 1
- Objective: `reg:squarederror`

Early stopping with a patience of 20 rounds is applied using a validation split from the training data.

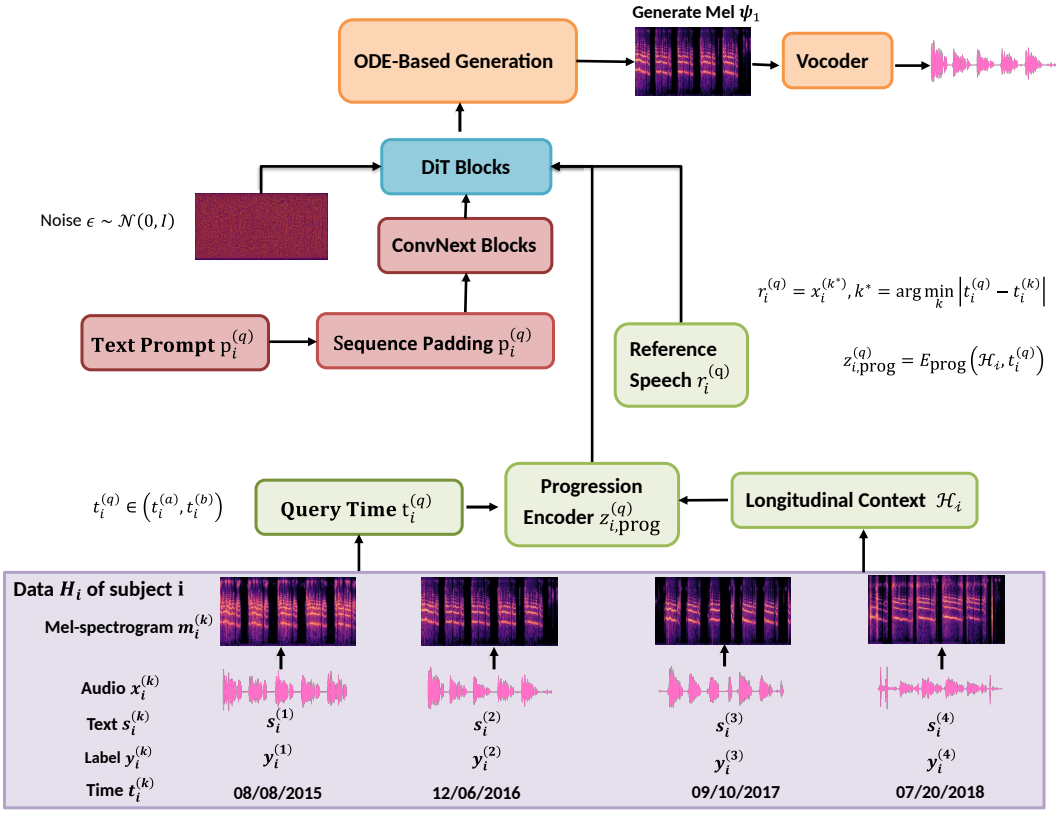

Figure 3: Overview of the L-Flow inference process.

Multilayer Perceptron (MLP). We use a lightweight two-layer MLP as a neural baseline. The architecture consists of:

- Input dimension: d (same as embedding dimension)
- Hidden layer: 256 units with ReLU activation
- Output layer: 1 unit (regression)

The model is trained using Adam with learning rate 1×10^{-3} and batch size 32 for 50 epochs. Dropout with rate 0.2 is applied to the hidden layer to prevent overfitting.

C Inference Procedure

Figure 3 illustrates the flow of inference. At inference time, L-Flow generates speech at a queried time point using the subject's observed recording history. Given a query time $t_i^{(q)}$ within the temporal span of the observed recording sessions, we first compute the soft progression condition $\tilde{y}_i^{(q)}$ by temporal interpolation between the two neighboring observed sessions, as described in Eq. 9. The reference speech $r_i^{(q)}$ is selected as the observed recording with the closest timestamp to $t_i^{(q)}$; if two recordings are equally close, the earlier recording is used.

The observed history $\mathcal{H}_i$ and query time $t_i^{(q)}$ are passed to the progression encoder to obtain the query-specific progression state $z_{i,\mathrm{prog}}^{(q)}$. The text condition $p_i^{(q)}$ is obtained from the fixed task transcript when available, or from a user-provided text prompt for spontaneous-speech settings. As in training, the text is converted into a character sequence and padded with filler tokens to match the desired mel length.

Generation starts from Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$. We solve the learned flow ODE:

$$\frac{d\psi_\sigma}{d\sigma} = v_\theta \left(\psi_\sigma \mid p_i^{(q)}, r_i^{(q)}, z_{i,\text{prog}}^{(q)}, \tilde{y}_i^{(q)} \right), \quad \sigma : 0 \to 1. \tag{19}$$

We use an Euler ODE solver with 32 function evaluations and uniformly spaced flow steps. The resulting mel-spectrogram ψ_1 is converted into waveform audio using the same pre-trained HiFi-GAN vocoder for all methods.

For synthetic augmentation, inference is performed only for subjects in the training split. We generate intermediate samples between neighboring recording sessions according to the generation budget defined in Eq. 8. The held-out test subjects are never used for synthetic generation.

D Theoretical Justification

We provide theoretical justification for three design choices in L-Flow: the conditional flow-matching target, the soft progression interpolation rule, and the temporal smoothness regularization.

D.1 Conditional Flow-Matching Target

Assumptions. Let $m \in \mathbb{R}^{F \times N}$ represent a target mel-spectrogram sampled from the data distribution, and let $\epsilon \sim \mathcal{N}(0, I)$ denote Gaussian noise of the same shape. For $\sigma \in [0, 1]$, define the linear interpolation path

$$\psi_\sigma = (1 - \sigma)\epsilon + \sigma m. \tag{20}$$

Assume that m and ϵ are fixed when differentiating with respect to σ.

Proposition 1. Under the linear probability path above, the ground-truth vector field is

$$v^*(\psi_\sigma) = \frac{d\psi_\sigma}{d\sigma} = m - \epsilon. \tag{21}$$

Proof. By definition,

$$\psi_\sigma = (1 - \sigma)\epsilon + \sigma m. \tag{22}$$

Taking the derivative with respect to σ gives

$$\frac{d\psi_\sigma}{d\sigma} = -\epsilon + m = m - \epsilon. \tag{23}$$

Therefore, for the linear flow path, the vector field that transports ϵ to m is exactly $m - \epsilon$. This is the target used in the conditional flow-matching loss.

□

D.2 Validity of Soft Progression Interpolation

Assumptions. Consider two neighboring observed recording sessions a and b for subject i, with recording times satisfying

$$t_i^{(a)} < t_i^{(q)} < t_i^{(b)}. \tag{24}$$

Let the normalized clinical labels satisfy $y_i^{(a)}, y_i^{(b)} \in [0, 1]$. Define

$$\alpha_i^{(q)} = \frac{t_i^{(q)} - t_i^{(a)}}{t_i^{(b)} - t_i^{(a)}} \tag{25}$$

and

$$\tilde{y}_i^{(q)} = (1 - \alpha_i^{(q)}) y_i^{(a)} + \alpha_i^{(q)} y_i^{(b)}. \tag{26}$$

Proposition 2. The interpolated soft target $\tilde{y}_i^{(q)}$ lies in $[0, 1]$.

Proof. Since $t_i^{(a)} < t_i^{(q)} < t_i^{(b)}$, we have

$$0 < t_i^{(q)} - t_i^{(a)} < t_i^{(b)} - t_i^{(a)}. \tag{27}$$

Dividing by the positive quantity $t_i^{(b)} - t_i^{(a)}$ gives

$$0 < \alpha_i^{(q)} < 1. \tag{28}$$

Thus, $\tilde{y}_i^{(q)}$ is a convex combination of $y_i^{(a)}$ and $y_i^{(b)}$. Since both endpoints lie in $[0, 1]$, any convex combination of them also lies in $[0, 1]$. Therefore,

$$\tilde{y}_i^{(q)} \in [0, 1]. \tag{29}$$

□

D.3 Temporal Smoothness Regularization

Assumptions. For subject i, let $z_i(t)$ represent the latent progression state inferred at time t. In practice, this is represented by $z_{i,\text{prog}}^{(q)}$ for queried times $t_i^{(q)}$. Assume that temporal smoothness is encouraged by the regularizer

$$\mathcal{L}_{\text{prog}} = \sum_i \sum_{q,q'} w_{q,q'} \left\| z_{i,\text{prog}}^{(q)} - z_{i,\text{prog}}^{(q')} \right\|_2^2, \tag{30}$$

where $w_{q,q'} \geq 0$ and larger weights are assigned to temporally closer pairs.

Proposition 3. For any pair (q, q') with $w_{q,q'} > 0$, minimizing $\mathcal{L}_{\text{prog}}$ penalizes large changes in latent progression states between temporally nearby query points.

Proof. Each pairwise term in $\mathcal{L}_{\text{prog}}$ is

$$w_{q,q'} \left\| z_{i,\text{prog}}^{(q)} - z_{i,\text{prog}}^{(q')} \right\|_2^2. \tag{31}$$

Because $w_{q,q'} \geq 0$, this term is nonnegative. For a fixed positive weight $w_{q,q'} > 0$, the term increases quadratically with the Euclidean distance between the two latent states. Therefore, reducing this term requires reducing

$$\left\| z_{i,\text{prog}}^{(q)} - z_{i,\text{prog}}^{(q')} \right\|_2. \tag{32}$$

When $w_{q,q'}$ is larger for temporally closer query points, the objective places stronger penalties on large latent changes over short time intervals. Thus, minimizing $\mathcal{L}_{\text{prog}}$ encourages the learned latent progression trajectory to vary smoothly over time. □

D.4 Summary

Proposition 1 justifies the vector-field target used in the flow-matching loss. Proposition 2 shows that the interpolated soft targets remain valid normalized progression scores. Proposition 3 explains how the progression regularizer discourages abrupt latent-state changes between nearby time points, which aligns with the goal of synthesizing temporally consistent intermediate speech samples.

E Baseline Implementation Details

We compare L-Flow with eight baseline augmentation methods. To ensure fairness, all methods use the same subject-level training split, the same generation budget $M(a, b)$, the same target text condition, and the same downstream train/test protocol. Synthetic samples are generated only for subjects in the training split. For each generated sample, the query time $t_i^{(q)}$ and soft target $\tilde{y}_i^{(q)}$ are assigned using the same interpolation rule as L-Flow.

E.1 Traditional Signal-Level Augmentation

Speed Perturbation. Speed perturbation modifies the temporal rate of the original speech waveform while preserving the transcript and speaker identity. For each selected training recording, we generate augmented samples using speed factors $\{0.9, 1.1\}$. The waveform is resampled after perturbation to 16 kHz. Since this method directly transforms existing recordings rather than synthesizing speech at a queried intermediate time, we assign each augmented sample the same clinical label as the source recording. For downstream augmentation, samples are selected to match the same augmentation budget as L-Flow.

Vocal Tract Length Perturbation (VTLP). VTLP simulates vocal tract variation by applying frequency-axis warping to the acoustic representation. We apply a piecewise linear frequency warping to the mel-spectrogram with warp factors sampled from $\{0.9, 1.1\}$. The perturbed mel-spectrograms are converted back to waveform using the same HiFi-GAN vocoder used by L-Flow. Similar to speed perturbation, VTLP modifies existing training recordings and does not model intermediate disease progression. Therefore, augmented samples inherit the label of the source recording and are subsampled to match the L-Flow augmentation budget.

E.2 General-Purpose Speech Synthesis Baselines

For all general-purpose TTS baselines, we use the same input text prompt as L-Flow. For ALS-TDI, the fixed task transcript ('I owe you a yoyo today' * 5) is used. For ParkCeleb, the text prompt associated with the target generation setting is used. To preserve speaker identity, each model is provided with the same reference speech selected by temporal proximity:

$$k^* = \arg\min_k |t_i^{(q)} - t_i^{(k)}|.$$

If two recordings are equally close, the earlier recording is selected. Since these models do not explicitly model longitudinal progression, they are conditioned only on text and reference speech. For downstream evaluation, each synthesized sample is paired with the same interpolated soft target $\tilde{y}_i^{(q)}$ used by L-Flow.

All TTS baselines are fine-tuned on the training subjects only. We fine-tune each model for 50k steps using AdamW with learning rate 1×10^{-5}, batch size 8, weight decay 1×10^{-2}, and gradient clipping at 1.0. Mixed-precision training is enabled for efficiency. We use early stopping based on validation WER and speaker similarity. During inference, all generated mel-spectrograms are converted to waveform using the same HiFi-GAN vocoder when the baseline does not provide its own waveform decoder.

OmniVoice. OmniVoice is used as a general voice-cloning baseline. During fine-tuning, the model is trained on pairs of reference speech and target transcript from the training subjects. At inference, the temporally nearest observed recording is used as the speaker prompt, and the target text prompt is used as the linguistic condition. We use a maximum generated duration of 30 seconds, temperature 0.7, and top-p sampling with $p = 0.9$ when stochastic decoding is supported. For deterministic settings, we use greedy decoding.

Qwen3-TTS. Qwen3-TTS is fine-tuned as a text-conditioned speech synthesis baseline with reference audio prompting. We format each training example using the reference speech, its transcript, and the target transcript. During inference, the reference speech is selected using the same temporal proximity rule as L-Flow. We use temperature 0.7, top-$p = 0.9$, and a maximum generation length matched to the target prompt duration. For fixed-task ALS-TDI generation, the sentence-level prompt is fixed across samples. For ParkCeleb, the prompt is provided by the spontaneous-speech generation setting.

VoxCPM2. VoxCPM2 is used as a voice-cloning TTS baseline. The model is fine-tuned on training-subject speech using the same subject-level split. At inference, VoxCPM2 receives the same text prompt and reference speech as L-Flow. We use the default acoustic decoding configuration with temperature 0.7 and top-$p = 0.9$ when applicable. The maximum output duration is clipped to match the duration range of the corresponding dataset.

XTTS-v2. XTTS-v2 is used as a multilingual zero-shot TTS and voice-cloning baseline. We fine-tune the model on training-subject recordings using the same transcript and audio pairs. At inference, the temporally nearest recording is used as the speaker reference. We use a temperature of 0.75, length penalty of 1.0, repetition penalty of 2.0, and top-$p = 0.85$. Generated audio is resampled to 16 kHz before downstream evaluation.

E.3 Pathology-Oriented Speech Synthesis Baselines

Text-to-Dysarthric-Speech (TTDS). TTDS is fine-tuned using the training recordings and their corresponding transcripts. We provide the normalized severity score as an auxiliary control variable when supported by the model; otherwise, recordings are grouped into severity bins and the bin index is used as a control token. We use five severity bins corresponding to intervals $[0, 0.2)$, $[0.2, 0.4)$, $[0.4, 0.6)$, $[0.6, 0.8)$, and $[0.8, 1.0]$. For synthesized intermediate samples, the interpolated target $\tilde{y}_i^{(q)}$ is mapped to the corresponding severity bin. TTDS is fine-tuned for 50k steps using AdamW with learning rate 1×10^{-5} and batch size 8.

Severity-Controllable Pathological TTS (SCP-TTS). We fine-tune the model on training-subject speech with normalized clinical severity scores as continuous control inputs. For ALS-TDI, the ALSFRS-R speech subscore is transformed to $[0, 1]$ as described in the main text. For ParkCeleb, pre-diagnosis and post-diagnosis samples are mapped to 0 and 1, respectively. At inference, SCP-TTS receives the target text, reference speech, and interpolated soft severity target $\tilde{y}_i^{(q)}$. The model is fine-tuned for 50k steps using AdamW with learning rate 1×10^{-5}, batch size 8, and gradient clipping at 1.0.

E.4 Fairness Controls Across Baselines

To ensure that improvements are attributable to progression-aware generation rather than differences in data volume or evaluation protocol, we apply the following controls. First, all methods generate synthetic speech only for training subjects. Second, all methods use the same number of generated samples per interval, determined by Eq. 8. Third, all generated samples are assigned soft targets using Eq. 9. Fourth, all downstream regressors are trained and evaluated using the same feature extraction pipeline and held-out real test subjects. All methods use the same vocoder and 16 kHz output sampling rate.

F Model Architecture Details

L-Flow consists of two main modules: a longitudinal progression encoder and a conditional flow-based speech generator. The progression encoder summarizes a subject's observed recording history into a query-specific latent state, while the generator synthesizes speech conditioned on text, reference speech, clinical severity, and the inferred progression state.

Recording-level representation. For each recording session, the waveform is converted into a mel-spectrogram $m_i^{(k)} \in \mathbb{R}^{F \times N_k}$. Since recordings have varied durations, we use a lightweight acoustic encoder E_{sp} to obtain a fixed-dimensional representation:

$$h_i^{(k)} = E_{\text{sp}}(m_i^{(k)}), \quad h_i^{(k)} \in \mathbb{R}^d. \tag{33}$$

In our implementation, E_{sp} is a 3-layer 1D convolutional network with GELU activations and temporal mean pooling, followed by a linear projection. We set $d = 512$ to match the hidden dimension of the progression encoder.

Severity and temporal embeddings. The normalized clinical score $y_i^{(k)} \in [0, 1]$ is projected into a severity embedding:

$$e_i^{(k)} = E_{\text{sev}}(y_i^{(k)}), \quad e_i^{(k)} \in \mathbb{R}^d. \tag{34}$$

We fuse acoustic and severity information into a clinical-acoustic state:

$$a_i^{(k)} = \text{MLP}_{\text{state}}\left([h_i^{(k)}; e_i^{(k)}]\right), \quad a_i^{(k)} \in \mathbb{R}^d. \tag{35}$$

To encode irregular recording schedules, we represent time in days using relative temporal features:

$$\tau_i^{(k)} = t_i^{(k)} - t_i^{(1)}, \quad \Delta_i^{(k)} = \begin{cases} 0, & k = 1, \\ t_i^{(k)} - t_i^{(k-1)}, & k > 1. \end{cases} \tag{36}$$

Here, $\tau_i^{(k)}$ denotes the relative position of a session within the subject's observed trajectory, and $\Delta_i^{(k)}$ denotes the gap from the previous session. These features are mapped to a temporal embedding:

$$e_{t,i}^{(k)} = \mathrm{MLP}_t\left([\tau_i^{(k)}, \Delta_i^{(k)}]\right), \quad e_{t,i}^{(k)} \in \mathbb{R}^d. \tag{37}$$

The final recording-level token is:

$$u_i^{(k)} = a_i^{(k)} + e_{t,i}^{(k)}. \tag{38}$$

Progression encoder. For a query time $t_i^{(q)}$, we construct a query-time embedding using its relative position from the first observed session:

$$e_{q,i}^{(q)} = \mathrm{MLP}_q\left(t_i^{(q)} - t_i^{(1)}\right), \quad e_{q,i}^{(q)} \in \mathbb{R}^d. \tag{39}$$

The query embedding is prepended to the sequence of recording-level tokens, and the output corresponding to the query token is used as the progression state:

$$z_{i,\mathrm{prog}}^{(q)} = \mathrm{TransformerEnc}\big([e_{q,i}^{(q)}; u_i^{(1)}; \ldots; u_i^{(T_i)}]\big)_{\mathrm{query}}. \tag{40}$$

The progression encoder is implemented as a 4-layer Transformer encoder with 8 attention heads, hidden dimension 512, feed-forward dimension 1024, and dropout 0.1.

Conditional speech generator. The speech generator follows an alignment-free flow-matching TTS backbone based on a latent Diffusion Transformer. The padded character sequence $p_i^{(q)}$ is first refined by ConvNeXt V2 blocks before being fused with speech features. The generator predicts the vector field:

$$v_\theta\left(\psi_\sigma \mid p_i^{(q)}, r_i^{(q)}, z_{i,\mathrm{prog}}^{(q)}, y_i^{(q)}\right), \tag{41}$$

where $r_i^{(q)}$ is the reference speech selected from the observed history and $y_i^{(q)}$ is the observed severity score during training. During synthetic generation, $y_i^{(q)}$ is replaced by the interpolated soft target $\tilde{y}_i^{(q)}$.

The progression state $z_{i,\mathrm{prog}}^{(q)}$ is projected into the same conditioning space as the flow-step embedding and injected into the DiT blocks through adaptive layer normalization.

G Additional Progression Fidelity Evaluation

We conduct additional evaluations to examine whether L-Flow captures progression-related changes in speech beyond the longitudinal ordering metrics reported in the main paper. Specifically, we compare L-Flow with direct latent interpolation, analyze phoneme-level intelligibility across ALS severity levels, and assess the acoustic distributional similarity between synthesized and real speech at corresponding progression stages.

G.1 Comparison with Linear Latent Interpolation

As described in Eq. 9, L-Flow assigns a soft clinical progression target to each intermediate query time through linear interpolation between neighboring observed clinical scores. Importantly, this interpolation applies only to the scalar progression target; it does not impose a linear trajectory on the generated speech, its acoustic features, or the learned longitudinal representation.

To distinguish the contribution of learned longitudinal dynamics from deterministic interpolation, we introduce a linear latent interpolation baseline (*Linear-Interp*). Given two temporally neighboring

Dataset	Method	Pairwise Acc. ↑	Kendall's τ ↑	Spearman's ρ ↑	MMD ↓
ALS-TDI	Linear-Interp	0.70	0.39	0.52	0.31
	SCP-TTS	0.71	0.36	0.49	0.25
	L-Flow	**0.79**	**0.49**	**0.64**	**0.22**
ParkCeleb	Linear-Interp	0.66	0.31	0.43	0.29
	SCP-TTS	0.65	0.30	0.42	0.26
	L-Flow	**0.74**	**0.40**	**0.56**	**0.20**

Table 6: Comparison of linear latent interpolation and progression-aware generation on ALS-TDI and ParkCeleb. Pairwise Accuracy, Kendall's τ, and Spearman's ρ evaluate longitudinal progression consistency (higher is better), while acoustic MMD measures distributional discrepancy between real and synthesized speech (lower is better).

observations at times t_a and t_b, we encode their speech into 256-dimensional utterance-level acoustic representations, denoted by h_a and h_b. For a query time t_q, we compute

$$h_q = (1 - \alpha_q)h_a + \alpha_q h_b, \qquad \alpha_q = \frac{t_q - t_a}{t_b - t_a}. \tag{42}$$

The interpolated representation h_q is then serves as the acoustic condition for the same generation backbone used by L-Flow.

Linear-Interp uses the same neighboring observations and query time as L-Flow but omits the longitudinal progression encoder and progression-aware training objectives. Consequently, it explicitly constrains the intermediate acoustic representation to a linear path between the two observed states.

As shown in Table G.1, Linear-Interp achieves non-trivial progression consistency on both datasets, indicating that explicit temporal interpolation provides a useful progression structure. However, L-Flow consistently outperforms Linear-Interp across all three progression-ordering metrics and achieves lower acoustic MMD. These results suggest that linear interpolation alone cannot reproduce the progression-related acoustic distributions captured by L-Flow. Whereas Linear-Interp assumes a straight trajectory between observed acoustic representations, L-Flow learns query-dependent speech characteristics from the subject's longitudinal history without explicitly constraining their evolution to be linear.

G.2 Severity-Stratified Phoneme Intelligibility

To complement the word error rate (WER) evaluation, we conduct a severity-stratified phoneme-level analysis on ALS-TDI. Because participants repeat a predefined sentence across recording sessions, this dataset lets us examine changes in phonetic intelligibility across different levels of speech impairment.

We calculate phoneme error rate (PER) using the normalized phoneme-level Levenshtein distance between the reference and recognized phoneme sequences. We report the mean utterance-level PER for each ALSFRS-R speech score, ranging from 4 (normal speech) to 0 (loss of useful speech). Lower PER indicates greater phonetic intelligibility; however, PER should not be interpreted as a direct measure of clinical fidelity.

Table G.2 shows that PER generally increases as ALS speech impairment becomes more severe. General-purpose TTS systems maintain relatively low PER across severity levels, with substantially smaller increases toward advanced impairment stages than those observed in real recordings. This suggests that these models tend to produce speech that remains comparatively intelligible even when conditioned on more severe disease states.

In contrast, L-Flow exhibits a more pronounced increase in PER across severity levels, broadly following the deterioration observed in original recordings. Nevertheless, its PER at the most severe stage exceeds that of the original recordings, indicating that closer agreement in the direction of change does not necessarily imply better absolute intelligibility or clinical fidelity. We therefore interpret these results together with the acoustic distribution analyses below.

Method	4	3	2	1	0	Mean
Original	0.08	0.12	0.19	0.31	0.47	0.23
Speed perturbation	0.08	0.13	0.20	0.32	0.48	0.24
VTLP	0.11	0.16	0.24	0.36	0.51	0.28
OmniVoice	0.13	0.15	0.17	0.20	0.24	0.18
Qwen3-TTS	0.09	0.12	0.15	0.18	0.22	0.15
VoxCPM2	0.10	0.13	0.16	0.19	0.23	0.16
XTTS-v2	0.15	0.17	0.20	0.23	0.27	0.20
TTDS	0.13	0.17	0.23	0.31	0.42	0.25
SCP-TTS	0.10	0.14	0.20	0.28	0.38	0.22
L-Flow	0.09	0.14	0.22	0.35	0.52	0.26

Table 7: Severity-stratified utterance-level phoneme error rate (PER) on ALS-TDI. Scores range from 4 (normal speech) to 0 (loss of useful speech). Lower values mean greater phonetic intelligibility, but not necessarily greater clinical realism.

Method	4	3	2	1	0	Mean
Speed perturbation	0.16	0.22	0.30	0.39	0.49	0.31
VTLP	0.19	0.26	0.35	0.45	0.55	0.36
OmniVoice	0.18	0.27	0.38	0.50	0.61	0.39
Qwen3-TTS	0.15	0.24	0.35	0.47	0.58	0.36
VoxCPM2	0.16	0.25	0.36	0.48	0.59	0.37
XTTS-v2	0.21	0.30	0.41	0.53	0.64	0.42
TTDS	0.15	0.21	0.28	0.36	0.44	0.29
SCP-TTS	**0.13**	0.18	0.24	0.31	0.38	0.25
L-Flow	0.14	**0.16**	**0.20**	**0.26**	**0.32**	**0.22**

Table 8: Severity-conditioned acoustic fidelity on ALS-TDI, measured by MMD between real and synthesized speech distributions. Lower values show closer acoustic agreement within each ALSFRS-R speech severity level.

G.3 Severity- and Time-Conditioned Acoustic Fidelity

To evaluate whether synthesized speech reproduces acoustic characteristics associated with disease progression, we compare the acoustic feature distributions of real and generated speech at corresponding progression stages.

For each utterance, we extract acoustic features covering prosody, phonation, articulation, and pausing, including the mean and standard deviation of fundamental frequency (F_0), intensity, jitter, shimmer, harmonic-to-noise ratio, the first two formants (F_1 and F_2), speaking rate, articulation rate, pause ratio, and mean pause duration. Each feature is standardized independently for each speaker to reduce inter-speaker variability.

We measure the discrepancy between real and synthesized acoustic feature distributions using Maximum Mean Discrepancy (MMD) with a radial basis function (RBF) kernel. The kernel bandwidth is selected using the median heuristic. Lower MMD indicates closer agreement with the acoustic characteristics of real speech within the corresponding progression stratum.

ALS-TDI. For ALS-TDI, recordings are grouped according to the ALSFRS-R speech score, from 4 (normal speech) to 0 (loss of useful speech).

As shown in Table G.3, acoustic discrepancies generally increase as speech impairment becomes more severe, particularly for general-purpose TTS systems. L-Flow achieves the lowest average MMD of 0.22, compared with 0.25 for SCP-TTS, and yields the lowest discrepancy at severity scores 3 through 0. These results suggest that L-Flow better preserves severity-dependent acoustic distributions, especially at more advanced stages of impairment.

ParkCeleb. Unlike ALS-TDI, ParkCeleb provides diagnosis timing rather than a multi-level clinical severity score. We therefore stratify recordings into four intervals relative to PD diagnosis: more than

Method	< -5 y	-5–0 y	0–5 y	> 5 y	Mean
Speed perturbation	0.21	0.27	0.36	0.47	0.33
VTLP	0.25	0.32	0.41	0.52	0.38
OmniVoice	0.23	0.31	0.42	0.54	0.38
Qwen3-TTS	0.21	0.29	0.39	0.50	0.35
VoxCPM2	0.22	0.30	0.40	0.51	0.36
XTTS-v2	0.27	0.35	0.46	0.58	0.42
TTDS	0.19	0.25	0.33	0.41	0.30
SCP-TTS	0.17	0.22	0.29	0.36	0.26
L-Flow	**0.13**	**0.17**	**0.22**	**0.28**	**0.20**

Table 9: Time-stratified acoustic fidelity on ParkCeleb, measured by MMD between real and synthesized speech distributions. Recordings are grouped by time relative to PD diagnosis rather than measured clinical severity. Lower values indicate closer acoustic agreement.

five years before diagnosis, zero to five years before diagnosis, zero to five years after diagnosis, and more than five years after diagnosis.

Table G.3 shows that L-Flow achieves the lowest MMD across all four diagnosis-relative time intervals, with an average of 0.20 compared with 0.26 for SCP-TTS. The differences are particularly pronounced in the post-diagnosis intervals, where modeling heterogeneous speech characteristics becomes more challenging. These results complement the progression-ordering evaluation and suggest that L-Flow better matches the acoustic distributions of real longitudinal speech across different periods relative to diagnosis.

H Progression-Related Acoustic Feature Analysis

To further investigate whether L-Flow captures interpretable acoustic changes associated with disease progression, we analyze four progression-relevant speech features on ALS-TDI: articulation rate, pause ratio, jitter, and fundamental frequency (F_0) variability. These features characterize complementary aspects of speech production, including articulatory timing, pausing behavior, phonatory stability, and prosodic variation.

We standardize each feature independently for each speaker to reduce inter-speaker variability. We report the mean standardized feature value at each ALSFRS-R speech severity level, ranging from 4 (normal speech) to 0 (loss of useful speech). We compare L-Flow with real recordings, Linear-Interp, and SCP-TTS to assess how closely the generated speech reproduces severity-dependent acoustic trends.

As shown in Table H, the original recordings exhibit systematic acoustic changes with increasing speech impairment. Articulation rate and F_0 variability decrease as severity progresses, whereas pause ratio and jitter increase. All three synthesis methods reproduce the general direction of these changes, suggesting that they capture some severity-related acoustic information.

However, Linear-Interp produces more compressed feature trajectories, particularly at advanced impairment levels. SCP-TTS more closely follows the original trends, but still underestimates the magnitude of several acoustic changes. In comparison, L-Flow consistently produces feature values closer to those of the original recordings across severity levels, with particularly noticeable differences in pause ratio, jitter, and F_0 variability at the most severe stages.

These findings complement the progression consistency and acoustic distribution evaluations in Appendix G. While rank-based metrics assess whether synthesized speech follows the expected temporal ordering, the feature-level analysis provides more interpretable evidence that L-Flow captures progression-associated changes in speech production.

I Statistical Significance and Robustness

We conduct additional analyses to assess the stability and statistical reliability of the experimental results. Specifically, we report fold-level validation performance and estimate confidence intervals and statistical significance using subject-level bootstrap resampling.

Feature	Method	4	3	2	1	0
Articulation rate	Original	0.68	0.39	0.08	-0.31	-0.72
	Linear-Interp	0.61	0.34	0.06	-0.22	-0.55
	SCP-TTS	0.63	0.35	0.03	-0.27	-0.62
	L-Flow	0.66	0.38	0.06	-0.29	-0.68
Pause ratio	Original	-0.62	-0.35	-0.06	0.34	0.76
	Linear-Interp	-0.55	-0.30	-0.04	0.23	0.51
	SCP-TTS	-0.57	-0.31	0.00	0.28	0.62
	L-Flow	-0.60	-0.34	-0.04	0.32	0.71
Jitter	Original	-0.49	-0.34	-0.08	0.27	0.71
	Linear-Interp	-0.43	-0.25	-0.06	0.15	0.39
	SCP-TTS	-0.45	-0.27	-0.02	0.20	0.49
	L-Flow	-0.47	-0.31	-0.06	0.25	0.65
F_0 variability	Original	0.51	0.32	0.11	-0.18	-0.57
	Linear-Interp	0.44	0.27	0.09	-0.10	-0.32
	SCP-TTS	0.46	0.28	0.08	-0.13	-0.40
	L-Flow	0.49	0.30	0.10	-0.17	-0.52

Table 10: Progression-related acoustic features across ALSFRS-R speech severity levels on ALS-TDI. Values represent mean speaker-standardized features. Scores range from 4 (normal speech) to 0 (loss of useful speech).

Dataset	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean $\pm$ SD
ALS-TDI	0.77	0.80	0.78	0.82	0.79	0.79 ± 0.02
ParkCeleb	0.70	0.74	0.71	0.76	0.73	0.73 ± 0.02

Table 11: Five-fold validation Pairwise Accuracy of L-Flow on ALS-TDI and ParkCeleb. Results use subject-disjoint folds within the training partition. Higher values represent better progression-ordering consistency.

I.1 Cross-Validation Stability

As described in Appendix B, we use a subject-level 80/20 train-test split, excluding held-out test subjects from model selection. For downstream predictor selection, we perform five-fold subject-level cross-validation within the training partition, ensuring that all recordings from the same subject remain in the same fold.

To examine the stability of progression consistency across folds, we report the validation Pairwise Accuracy of L-Flow on both datasets in Table I.1.

The results show relatively consistent progression-ordering performance across folds, with mean Pairwise Accuracy of 0.79 on ALS-TDI and 0.73 on ParkCeleb, and a standard deviation of approximately 0.02 on both datasets. These findings indicate limited variation in progression consistency across the evaluated subject-disjoint validation folds.

I.2 Subject-Level Bootstrap Significance Analysis

To quantify uncertainty in the final evaluation, we perform 10,000 rounds of subject-level bootstrap resampling on the held-out test subjects. We obtain 95% confidence intervals (CIs) for the primary evaluation metrics and conduct paired bootstrap tests to compare L-Flow with competing augmentation methods and ablated model variants. The paired comparisons use the same resampled subjects to account for the dependence between methods evaluated on the same test set. Table 12 summarizes the results for progression consistency, downstream prediction, and ablation analysis.

Progression Consistency. L-Flow achieves statistically significant improvements in Spearman's rank correlation over SCP-TTS on both ALS-TDI ($p = 0.006$) and ParkCeleb ($p = 0.018$). These results statistically support the progression-consistency improvements observed in the main experiments.

Experiment	Dataset	Metric	L-Flow [95% CI]	Baseline [95% CI]	Baseline method	p-value
Progression	ALS-TDI	$\rho \uparrow$	0.64 [0.59, 0.69]	0.49 [0.43, 0.55]	SCP-TTS	**0.006**
	ParkCeleb	$\rho \uparrow$	0.56 [0.49, 0.63]	0.42 [0.34, 0.50]	SCP-TTS	**0.018**
Downstream	ALS-TDI	MAE $\downarrow$	0.18 [0.16, 0.20]	0.16 [0.14, 0.19]	SCP-TTS	0.112
	ParkCeleb	MAE $\downarrow$	0.29 [0.25, 0.33]	0.25 [0.21, 0.30]	Qwen3-TTS	0.084
Ablation	ALS-TDI	MAE $\downarrow$	0.18 [0.16, 0.20]	0.23 [0.20, 0.26]	w/o progression	**0.021**
	ParkCeleb	MAE $\downarrow$	0.29 [0.25, 0.33]	0.35 [0.30, 0.40]	w/o progression	**0.028**

Table 12: Subject-level bootstrap analysis on held-out test subjects using 10,000 resampling iterations. Values are reported with 95% confidence intervals. Progression is evaluated using Spearman's ρ (higher is better), while downstream prediction and ablation are evaluated using XGBoost MAE (lower is better).

Downstream Prediction. For the selected XGBoost MAE comparisons, L-Flow obtains numerically higher errors than the strongest competing augmentation baselines on ALS-TDI and ParkCeleb. However, neither difference reaches statistical significance ($p = 0.112$ and $p = 0.084$, respectively). Therefore, these individual comparisons do not establish an advantage for L-Flow in pointwise prediction accuracy. The downstream results should instead be interpreted across the full set of predictors and evaluation metrics reported in the main paper.

Ablation Analysis. Removing global progression modeling significantly increases downstream prediction error on both ALS-TDI ($p = 0.021$) and ParkCeleb ($p = 0.028$). These findings support the contribution of subject-specific longitudinal progression modeling to the downstream utility of generated speech and complement the ablation results reported in the main paper.

Overall, the statistical analysis supports L-Flow's improvements in progression consistency and the contribution of its progression modeling component. In contrast, the selected downstream comparisons against strong TTS augmentation baselines show no statistically significant differences, highlighting the importance of distinguishing progression fidelity from pointwise prediction accuracy.

J Computational Efficiency and Augmentation Effects

We provide additional analysis of the computational cost of speech augmentation and its impact on downstream prediction. These experiments examine the trade-off between augmentation efficiency and progression-related utility, including cases where synthetic augmentation degrades predictive performance.

J.1 Computational Cost of Speech Generation

We conducted all experiments using two NVIDIA L40S GPUs, each with 48 GB of memory. To ensure fair downstream comparisons, we use the same augmentation budget across all methods, such that each method contributes an equal number of additional speech samples to the downstream training set.

We measure the time required to generate the complete augmented ALS-TDI training set. As shown in Table J.1, signal-level transformations are substantially faster than neural speech-generation methods. L-Flow requires 11.2 hours, comparable to OmniVoice (10.7 hours) and Qwen3-TTS (12.5 hours). The remaining neural generation baselines require approximately 6–8 hours. ParkCeleb shows similar relative trends, with slightly shorter absolute generation times.

These results indicate that L-Flow incurs substantially greater generation cost than simple signal-level transformations, but operates within a comparable computational range to other neural speech synthesis approaches. The additional cost should therefore be considered alongside its improvements in progression consistency and acoustic fidelity.

Method	Generation time
Speed perturbation	1.17 h
VTLP	1.17 h
Other neural baselines	6–8 h
OmniVoice	10.7 h
L-Flow	11.2 h
Qwen3-TTS	12.5 h

Table 13: Time required to generate the complete augmented ALS-TDI training set. All methods use the same augmentation budget.

Dataset	Training data	Ridge	XGBoost	MLP
ALS-TDI	Original	0.3	1.5	3.7
	Augmented	1.5	3.6	11.3
ParkCeleb	Original	0.2	1.4	3.3
	Augmented	0.3	1.7	4.5

Table 14: Downstream predictor training time (hours) with and without speech augmentation. Augmented training sets contain the same number of additional samples across generation methods.

J.2 Downstream Training Cost

After generating the augmented speech, we train the downstream predictors using either the original training recordings or the augmented training set. Since all augmentation methods contribute the same number of synthetic samples, the downstream training cost is comparable across augmentation methods for a given predictor.

Table J.2 reports the training times for Ridge Regression, XGBoost, and MLP on both datasets. As expected, incorporating additional synthetic samples increases downstream training time. The increase is more pronounced on ALS-TDI, particularly for the MLP predictor, whose training time rises from 3.7 to 11.3 hours. On ParkCeleb, the corresponding increase is smaller across all three predictors.

Importantly, these additional training costs arise from the larger augmented datasets rather than method-specific changes to the downstream prediction architecture. Thus, the principal computational distinction between augmentation methods lies in the speech-generation stage.

J.3 Negative Effects of Speech Augmentation

Although synthetic speech augmentation can improve downstream progression prediction, its benefits are not guaranteed. In particular, we observe cases where signal-level augmentation degrades performance relative to training on the original recordings alone.

Table J.3 highlights the effect of Vocal Tract Length Perturbation (VTLP) on Ridge Regression. We extract the results from the downstream evaluations reported in the main paper.

On both datasets, VTLP augmentation increases MAE and RMSE while reducing Spearman correlation compared with the original-data baseline. This indicates that adding synthetic samples does not necessarily improve downstream prediction, even when the augmented samples preserve some characteristics of the original speech.

One possible explanation is that signal-level transformations modify spectral characteristics without explicitly accounting for the subject's longitudinal disease trajectory. Consequently, the resulting acoustic variations may not align with the intended progression targets. While these results do not establish the precise cause of performance degradation, they highlight the importance of evaluating the progression-related utility of augmented speech rather than relying solely on sample quantity or conventional speech-quality metrics.

Together, these findings illustrate the trade-offs involved in longitudinal speech augmentation. L-Flow requires greater computational resources than conventional signal-level transformations, but its progression-aware generation mechanism provides additional benefits in longitudinal consistency

Dataset	Method	MAE ↓	RMSE ↓	ρ ↑
ALS-TDI	Original	0.35	0.42	0.48
	VTLP	0.38	0.46	0.42
ParkCeleb	Original	0.44	0.52	0.39
	VTLP	0.46	0.55	0.35

Table 15: Examples of negative augmentation effects using VTLP and Ridge Regression. MAE and RMSE are lower-is-better metrics, while Spearman's ρ is higher-is-better.

and acoustic distribution matching. The negative augmentation cases further emphasize that effective clinical speech augmentation depends not only on generating additional recordings, but also on preserving task-relevant progression information.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction align with the goals and contributions of this work.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of the work in detail in Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Theoretical justifications can be found in Appendix D.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The detailed settings for training and inference can be found in Section 4.4. Appendix B and C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The data sources and acquisition have been discussed in detail in Section 4.1. The link to the anonymized code is provided at the end of Section 1 on page 2.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: They are discussed in detail in Sections 4.4 and Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: They are included in sections 4.2, 4.4, and 4.5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In Section B, and Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: It is in line with the NeurIPS code of conduct.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: In Section 5 and Appendix 6.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper has no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The sources of the datasets used are given in Section 4.1, and previous works (TTS models) are cited throughout the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We explicitly mentioned the release of the code and model at the end of Section 1.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The datasets used in this work were collected by other organizations, to which we were granted access upon formal request.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The datasets used in this work were collected by other organizations, and these projects are IRB approved. Before we were given access, the data had been fully anonymized.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are only used for editing and formatting purposes.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.