StuD: A Multimodal Approach for Stuttering Detection with RAG and Fusion Strategies

Pragya Khanna[1], Priyanka Kommagouni[1], Vamshiraghusimha Narasinga[1], Anil Kumar Vuppala[1]
[1]LTRC, International Institute of Information Technology, Hyderabad, India
{pragya.khanna, parvathipriyanka.b, narasinga.vamshi}@research.iiit.ac.in,
anil.vuppala@iiit.ac.in

Abstract

Stuttering is a complex speech disorder that challenges both ASR systems and clinical assessment. We propose a multimodal stuttering detection and classification model that integrates acoustic and linguistic features through a two-stage fusion mechanism. Fine-tuned Wav2Vec 2.0 and HuBERT extract acoustic embeddings, which are early fused with MFCC features to capture fine-grained spectral and phonetic variations, while Llama-2 embeddings from Whisper ASR transcriptions provide linguistic context. To enhance robustness against out-of-distribution speech patterns, we incorporate Retrieval-Augmented Generation or adaptive classification. Our model achieves state-of-the-art performance on SEP-28k and FluencyBank, demonstrating significant improvements in detecting challenging stuttering events. Additionally, our analysis highlights the complementary nature of acoustic and linguistic modalities, reinforcing the need for multimodal approaches in speech disorder detection.

1 Introduction

Stuttering is a multifactorial speech disorder characterized by disruptions in the normal flow of speech, including repetitions, prolongations, and silent blocks (Smith and Weber, 2017). Affecting approximately 1% of the global population (Yairi and Ambrose, 2013), it can lead to avoidance behaviors, social anxiety, and reduced participation in professional and educational settings (Craig et al., 2009; Koedoot et al., 2011). The severity and manifestation of stuttering vary across individuals, influenced by psychological, linguistic, and situational factors (Bloodstein et al., 2021).

From a speech technology perspective, stuttering presents a unique challenge for automatic speech recognition (ASR) systems, as it disrupts the temporal and lexical structure of speech, leading to higher word error rates (WERs) (Mendelev et al., 2021). The accurate detection and classification of stuttering events, leveraging both acoustic and lexical features, are essential for developing inclusive ASR systems and personalized therapy solutions. Additionally, ensuring robustness to out-of-distribution (OOD) scenarios—where models encounter speech patterns not seen during training—remains a critical challenge. Advancing stuttering detection systems capable of handling such variability holds significant promise for improving speech accessibility and communication equity.

Another fundamental hurdle in reliable stuttering detection is semantic ambiguity. Models must accurately distinguish between genuine dysfluent repetitions (e.g., 'I-I-I went') and semantically valid, fluent lexical repetitions or common speech patterns (e.g., 'this is a do-do' or 'He said, 'no, no, no!''). Relying solely on acoustic features or lexicon-agnostic approaches often leads to clinically unreliable false positives. Advancing stuttering detection systems capable of handling such complex variability, including semantic context, holds significant promise for improving speech accessibility and communication equity for millions. Dysfluency detection has traditionally relied heavily on acoustic features, given that various dysfluency types are distinctly observable in audio waveforms and spectrograms (Sheikh et al., 2022; Barrett et al., 2022). Early research predominantly employed traditional signal processing methods (Bayerl et al., 2020; Esmaili et al., 2017), extracting handcrafted features for classification via techniques such as Support Vector Machines (SVMs). However, the advent of deep neural networks has led to a significant performance leap over these traditional methods. Models like StutterNet (Sheikh et al., 2021) and similar architectures (Sheikh et al., 2022) have effectively showcased the power of deep learning for real-time dysfluency classification.

More recent advancements have capitalized on neural representations derived from self-supervised learning (SSL) models, including wav2vec 2.0 (Bayerl et al., 2022; Baevski et al., 2020). These SSL models are adept at capturing rich, contextualized speech features, markedly improving dysfluency detection without the need for extensive,

Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 698–707
December 20-24, 2025 ©2025 Association for Computational Linguistics

manually labeled datasets. Fine-tuning wav2vec 2.0 on stuttered speech has yielded promising results for detecting stuttering events at both utterance and frame levels. Nevertheless, a significant gap persists in fully exploring truly multimodal approaches that seamlessly integrate both acoustic and lexical features for comprehensive stuttering analysis.

Efforts to incorporate lexical features are a more recent but highly promising direction. Studies by Wagner et al. (Wagner et al., 2024) and Changawala Rudzicz (Changawala and Rudzicz, 2024) have demonstrated the potential of combining ASR transcriptions with large language models (LLMs) like Whisper (Radford et al., 2022) for dysfluency detection. Crucially, these methods have largely been evaluated on in-distribution data, leaving the critical challenge of Out-Of-Distribution (OOD) handling largely unaddressed. While Wong and Chen (Wong and Chen, 2024) attempted to tackle OOD issues through uncertainty estimation, their approach notably lacked the incorporation of lexical context, thereby limiting its robustness to novel stuttering patterns.

In this work, we aim to bridge these existing gaps by proposing a comprehensive, multimodal framework, StuD (Stutter Event Detection Model), that robustly leverages both advanced self-supervised speech models and state-of-the-art LLMs. Our contributions are specifically designed to enhance robustness against diverse and unseen stuttering patterns, significantly improve OOD handling through Retrieval-Augmented Generation (RAG), and ultimately achieve state-of-the-art performance in automated stuttering detection and classification on benchmark datasets.

Our work makes the following key contributions to the field of automated stuttering detection and classification:

1. **Multimodal Architecture with Two-Stage Fusion:** We propose a novel architecture that combines acoustic and linguistic features through early and late fusion mechanisms. The model leverages fine-tuned wav2vec 2.0 and HuBERT(Hsu et al., 2021) early fused with MFCCs for acoustic features and Llama-2(Touvron et al., 2023) embeddings derived from Whisper ASR transcriptions for linguistic features.

2. **Robust OOD Handling:** We incorporate Retrieval-Augmented Generation (RAG) to enhance system robustness against OOD samples. The model retrieves and compares embeddings of unfamiliar stuttering patterns with known training-set instances, allowing for adaptive classification and improved performance on diverse speech data.

3. **State-of-the-Art Performance:** Our model outperforms previous methods on SEP-28k and FluencyBank by 26% and 14% increase in average f1-score, respectively.

2 Corpora and Feature Representation

This study utilizes two publicly available datasets for stuttering event detection: SEP28k(Lea et al., 2021) for model training and FluencyBank(Bernstein Ratner and MacWhinney, 2018) for cross-corpora validation. These datasets contain labeled speech segments with explicitly defined types of disfluency.

SEP28k is a stuttering event detection dataset released by Apple in 2021. It consists of 28,177 three-second audio clips extracted from 265 publicly available podcasts, where speakers discuss stuttering. Each audio clip was labeled by three independent human annotators, who assigned one or more of the following disfluency types: Block, Interjection, Prolongation, Sound Repetition, and Word Repetition. Additional labels include Music, NoStutteredWords, and NoSpeech. The final annotation for each clip was determined using a majority voting system, where a label was assigned if at least two annotators agreed. In cases of full disagreement, all assigned labels were retained. Crucially, to ensure a robust evaluation of generalization across speakers, our 5-fold cross-validation strategy for SEP28k was implemented with strict speaker-independent splits. This ensures that data from any single speaker is confined entirely to either the training or validation set within each fold, mitigating potential data leakage and providing a more realistic assessment of model performance on unseen speakers.

This study utilizes the Adults Who Stutter portion of FluencyBank, which consists of 4,144 three-second labeled audio clips from 33 recorded sessions involving 23 male and 10 female speakers. The dataset follows a labeling scheme similar to SEP28k, allowing direct comparison between models trained on SEP28k and tested on FluencyBank. Unlike SEP28k, which primarily contains structured podcast speech, FluencyBank includes

elicited speech responses based on the Overall Assessment of the Speaker's Experience of Stuttering (OASES) protocol. The differences in recording environments, speaker demographics, and elicitation methods make FluencyBank a suitable dataset for evaluating cross-corpora generalization.

3 Proposed Methodology

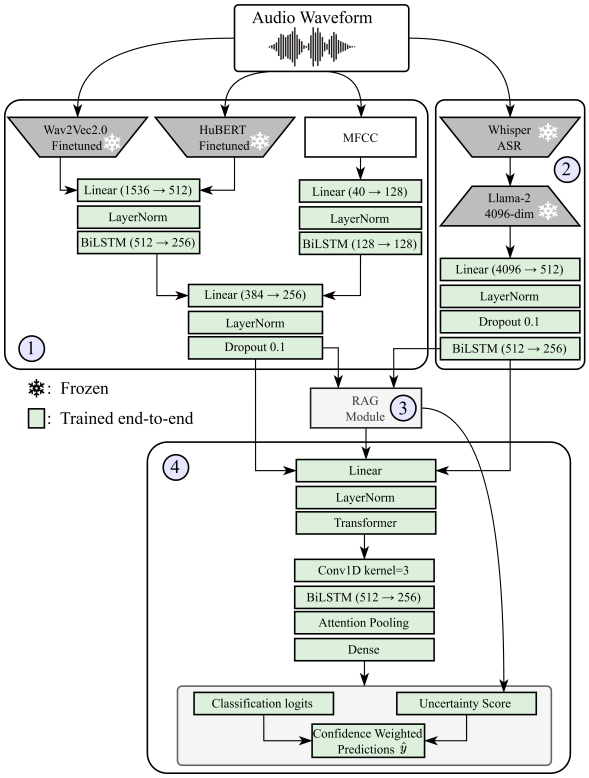

Figure 1: Overview of the StuD Model Architecture. Block 1 handles acoustic processing with Wav2Vec 2.0, HuBERT, and MFCC features. Block 2 processes text embeddings from Whisper ASR and LLaMA-2. Block 3 is the RAG module, retrieving contextual stutter patterns. Block 4 performs late fusion and classification, generating confidence-weighted predictions.

In this section, we detail StuD (Stutter Event Detection Model), a novel, comprehensive multimodal architecture designed to significantly enhance stuttering detection accuracy and generalization, particularly for challenging out-of-distribution (OOD) speech. StuD's architecture, visualized in Fig. 1, comprises distinct feature encoding modules for acoustic and linguistic information, a Retrieval-Augmented Generation (RAG) module for context enrichment, and a robust fusion network leading to confidence-weighted predictions.

Unlike prior works that often employ simpler fusion strategies or focus predominantly on in-distribution data (Romana and Koishida, 2023; Wagner et al., 2024), StuD integrates acoustic and linguistic information through a sophisticated two-stage fusion mechanism. This approach addresses the limitations of methods that may overlook granular acoustic descriptors like MFCCs or lack explicit mechanisms for handling novel speakers (Wong and Chen, 2024). The pipeline begins with specialized feature encoders processing audio waveforms to extract rich acoustic and linguistic representations. These are then fed into a hierarchical fusion strategy: an early fusion stage combines fine-grained acoustic features, followed by a late fusion stage integrating the consolidated acoustic representation with deep linguistic context. This design is crucial for capturing the multifaceted nature of stuttering, from subtle acoustic deviations to complex semantic disruptions. StuD is extensively trained on 18 hours of speech data from the SEP-28k dataset (Lea et al., 2021).

3.1 Feature Encoding

Effective stuttering detection relies on analyzing both acoustic and linguistic characteristics, as each provides distinct yet complementary insights into speech dysfluencies. Acoustic features are widely used to capture prolongations, repetitions, and other disruptions. Meanwhile, linguistic features offer a textual perspective on dysfluencies. Our architecture employs specific encoders for each modality, detailed below, to generate robust, synchronized frame-level representations.

3.1.1 Acoustic Feature Encoding

While transformer-based models like Wav2Vec 2.0 and HuBERT extract high-level contextual representations, Mel-frequency cepstral coefficients (MFCCs) provide fine-grained spectral details, ensuring a multi-scale acoustic representation. We adopt a multi-stage fine-tuning strategy for Wav2Vec 2.0 and HuBERT to enhance sensitivity to stuttering characteristics, ensuring that both models effectively capture the nuanced patterns of dysfluent speech while preserving general speech structure.

For Wav2Vec 2.0, we leverage the pretrained XLSR-53 model[1] with 315M parameters. During *Phase 1* of fine-tuning, the entire model is frozen except for the newly added classification head, focusing solely on the classification head to allow it to adapt to the task with minimal perturbation

[1] huggingface.co/docs/transformers/wav2vec2

to the base model. In *Phase 2*, the Feature Encoder (7 CNN blocks) and transformer layers 1-18 remain frozen, while layers 19-24 are unfrozen to refine higher-level speech patterns relevant to stuttering events. The classification head is continuously trained with a slightly higher learning rate than the transformer layers, ensuring synchronized adaptation across components. In the *final phase*, fine-tuning is confined to layers 19-24, with early stopping and weight decay mechanisms employed to prevent overfitting. Post-training, the classification head is removed, and 768-dimensional embeddings are extracted from the last layer. To achieve this, a linear projection layer was applied to reduce the original 1024-dimensional XLSR-53 output to the target 768 dimensions, ensuring synchronization across all features at 150 frames per 3-second segment.

For HuBERT, hubert-base-ls96[2] makes the backbone, and its fine-tuning process is divided into three progressively adaptive phases. In *Phase 1*, the CNN feature extractor and layers 1-8 are frozen to preserve foundational acoustic features. Layers 9-12 and the classification head are trained with a higher learning rate (2e-4) to promote rapid learning of high-level stutter patterns. *Phase 2* involves unfreezing layers 5-8, which handle speech rhythm and patterns, while the upper layers continue to adapt with a moderate learning rate. This phase focuses on refining the model's ability to capture temporal and rhythmic stuttering patterns. In *Phase 3*, layers 1-4 remain frozen to maintain basic speech knowledge, while layers 5-8 and upper layers are fine-tuned with reduced learning rates for precision tuning of stutter detection. Mixed-precision training, gradient clipping, and label smoothing are utilized throughout the process to optimize learning stability and mitigate class imbalances.

The output embeddings from the fine-tuned Wav2Vec 2.0 and HuBERT models, both 768-dimensional, are then passed to a Bi-directional Long Short-Term Memory (BiLSTM) network. While transformer-based models excel at capturing global contextual dependencies in speech, BiLSTMs are particularly adept at modeling local temporal patterns and fine-grained sequential dependencies within a sequence. By applying BiLSTMs post-transformer embeddings, we aim to extract more robust, stuttering-specific temporal dynamics (e.g., precise rhythm disruptions, subtle prolongations, or repeated phonetic segments) that complement the broader contextual understanding provided by the SSL models. This layered approach ensures that both macro- and micro-temporal features of dysfluent speech are effectively captured. The BiLSTMs reduce the dimensionality of both Wav2Vec 2.0 and HuBERT features to a hidden size of 256 each. Similarly, the 40-dimensional MFCC features are processed through their own BiLSTM with a hidden size of 128. These BiLSTMs are trained end-to-end as part of the overall StuD model, learning to extract relevant temporal features tailored for stuttering detection.

3.1.2 Linguistic Feature Encoding

To complement the acoustic embeddings, we incorporate ASR-derived word timing features from Whisper[3] and contextualized language representations from Llama 2[4]. Whisper (1.55 billion parameters) provides word-level timestamps that allow precise alignment of linguistic and acoustic features. While Whisper's internal acoustic features are robust, our architectural design specifically leverages fine-tuned Wav2Vec 2.0 and HuBERT as primary acoustic encoders to capture stuttering-specific acoustic nuances. Whisper's role is thus focused on providing high-fidelity, time-aligned ASR transcriptions, which serve as input for Llama-2 to extract semantically rich linguistic context. This alignment helps highlight different stutter events: *prolongations* manifest as extended word durations, *blocks* appear as silent pauses, and *repetitions* are identified through sequential text patterns. Llama 2 (13 billion parameters) generates 4096-dimensional embeddings that capture nuanced semantic relationships, enhancing detection of subtle stuttering variations.

To tailor these Llama-2 embeddings for the stuttering detection task and ensure compatibility with the acoustic modality, a trainable sequence of layers consisting of a linear projection ($4096 \rightarrow 512$), Layer Normalization, and a Dropout layer (0.1), followed by a BiLSTM with a hidden size of 256, reduces their dimensionality. This structured adaptation preserves the pre-trained knowledge of Llama-2 while optimizing it for stuttering-specific linguistic patterns. To ensure temporal alignment between modalities, we employ a frame-level expansion strategy, where each word embedding from Llama 2 is expanded to cover all frames within its

[2] huggingface.co/facebook/hubert-base-ls960

[3] github.com/openai/whisper

[4] huggingface.co/docs/transformers/en/model$_doc/llama2$

time span, with start and end frames determined by Whisper's ASR output. Frames without corresponding words are filled with zero vectors, creating frame-level linguistic features that are fully synchronized with the dimensions of the acoustic features.

3.2 Feature Fusion and RAG Integration

StuD's architecture integrates the extracted acoustic and linguistic features through a sophisticated two-stage fusion process, enhanced by a Retrieval-Augmented Generation (RAG) module. This approach ensures a comprehensive understanding of stuttering events by combining fine-grained acoustic details, high-level linguistic context, and a robust mechanism for handling out-of-distribution patterns.

3.2.1 Early Acoustic Fusion

The 256-dimensional features from the Wav2Vec 2.0 BiLSTM and the 256-dimensional features from the HuBERT BiLSTM are first concatenated to form a 512-dimensional representation. This combined SSL-based acoustic stream is then processed by a linear layer ($512 \rightarrow 256$), followed by Layer Normalization and a Dropout layer (0.1). Simultaneously, the 128-dimensional features from the MFCC BiLSTM undergo similar processing with a linear layer ($128 \rightarrow 128$), Layer Normalization, and Dropout (0.1).

These two processed acoustic streams (256-dim from SSL features and 128-dim from MFCCs) are then concatenated again to form a 384-dimensional early-fused acoustic feature. This early fusion step is critical as it allows the model to integrate fine-grained MFCC spectral information directly with the high-level contextual features from SSL models at an early stage. This mitigation of potential underrepresentation of MFCCs due to their lower dimensionality ensures that their distinct insights are fully leveraged before further processing.

3.3 Retrieval-Augmented Generation (RAG) Module

The RAG module, illustrated in detail in Fig. 2, serves as a bimodal context-enrichment system for stuttering detection, specifically designed to enhance robustness against OOD samples and rare stuttering patterns. It consists of three core components: reference database construction, similarity search, and context processing with cross-attention integration.

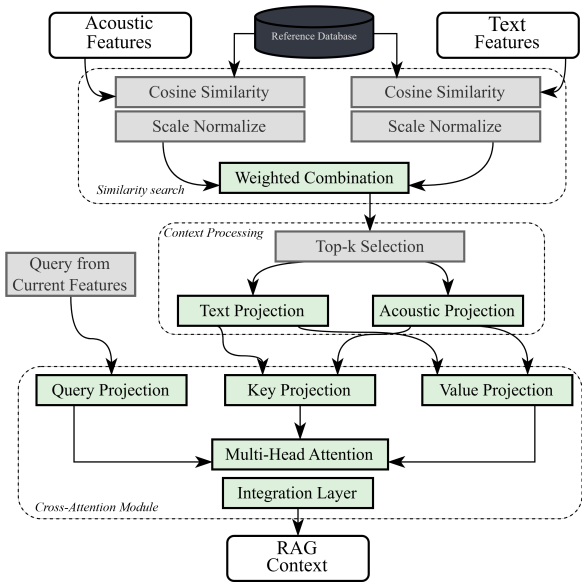

Figure 2: Detailed RAG Module.

3.3.1 RAG Reference Database Construction

The RAG reference database is constructed offline using the entire training partition of SEP-28k (Lea et al., 2021) and remains static throughout the main model training. For each utterance in the SEP-28k training set, a combined acoustic-linguistic embedding is generated by concatenating its 256-dimensional early-fused acoustic feature (output of the early fusion block) and its 256-dimensional linguistic feature (output of the Llama-2 processing block), resulting in a 512-dimensional vector.

K-means clustering is then applied to these combined 512-dimensional embeddings to identify prototypical stuttering patterns for each dysfluency type. The number of clusters, k, for each stuttering type (Block, Interjection, Prolongation, Sound Repetition, Word Repetition) is determined proportionally to its empirical frequency in the SEP-28k training set. This proportional sampling strategy ensures adequate representation for less frequent stuttering manifestations, preventing their underrepresentation in the reference pool. From each cluster, the utterance whose combined embedding is closest to the cluster centroid (measured by Euclidean distance) is selected as the representative reference example. These selected examples, along with their associated acoustic and linguistic features, form the static RAG reference database.

3.3.2 Similarity Search and Dynamic Weighting

For an incoming query (current input utterance), its 256-dimensional early-fused acoustic feature

and 256-dimensional linguistic feature are compared against the static RAG reference database. Cosine similarity is computed independently between the query's acoustic features and all acoustic features in the database, and similarly for linguistic features. These raw similarity scores undergo L2-normalization and temperature scaling to ensure comparability across modalities.

The contributions of these acoustic and linguistic similarity scores are then dynamically balanced through learnable weights, $w_{acoustic}$ and $w_{linguistic}$, which are trainable parameters initialized for each modality ($w_{acoustic}+w_{linguistic} = 1$). The learning of these weights is driven by a contrastive loss function applied during training, alongside the main classification loss. This loss encourages the weights to emphasize the modality (acoustic or linguistic) that yields higher similarity for positive pairs (query and retrieved sample belonging to the same stuttering type) and lower similarity for negative pairs (different stuttering types). A learnable temperature parameter further sharpens the similarity distributions, refining the alignment between modalities. The specific formulation of the contrastive loss for weight optimization is detailed in Section 3.5.

The FAISS approximate nearest neighbor search algorithm (Johnson et al., 2017) is employed for efficient retrieval of the top-k ($k = 8$) most relevant samples from the database. To ensure diversity and prevent redundancy in the contextual information provided to the model, Maximum Marginal Relevance (MMR) is applied, balancing the similarity score with a diversity penalty during retrieval.

3.3.3 Context Processing and Cross-Attention Integration

The top-k retrieved samples undergo modality-specific transformations, followed by projection onto a shared latent space for unified representation within the RAG context. In the cross-attention integration mechanism, the current input's 256-dimensional early-fused acoustic feature and 256-dimensional linguistic feature are projected into query vectors (Q). The retrieved contextual features (both acoustic and linguistic, processed from the RAG database) are transformed into corresponding key (K) and value (V) pairs. Multi-head attention (with 8 attention heads, as specified in Section 3.5) is then applied, allowing the model to capture diverse aspects of the retrieved context in parallel. An integration layer then applies learned weights to combine the attended outputs, producing the final RAG Context embedding.

3.4 Late Fusion Network and Classification

In the final stage of StuD, the consolidated acoustic, linguistic, and RAG context embeddings are fused and processed through a pipeline designed to capture local temporal patterns and generate confidence-weighted predictions.

The 384-dimensional early-fused acoustic feature (output from Section 3.2.1), the 256-dimensional linguistic feature from Llama-2 (output from Section 3.1.2), and the RAG context embedding (output from Section 3.3.3) are concatenated to form a comprehensive multimodal representation. This fused representation is then passed through a series of layers designed to capture higher-level patterns and temporal dependencies, including: a Linear layer (input dimension determined by concatenated features), followed by Layer Normalization; a Transformer block; a Conv1D layer with a kernel size of 3 for local feature extraction; a BiLSTM with a hidden size of 256 for further temporal modeling; and finally, an Attention Pooling layer to aggregate information across the sequence.

The output of the attention pooling layer is fed into a Dense layer, which serves as the classification head. This head generates raw classification logits $\mathbf{p} \in \mathbb{R}^c$, where c is the number of stuttering classes: Block, Interjection, Prolongation, Sound Repetition, Word Repetition. Each class-specific uncertainty score $\mathbf{U}_{\text{class}}$ is defined as:

$$\mathbf{U}_{\text{class}} = [U_1, U_2, U_3, U_4, U_5] \in \mathbb{R}^c$$

For each stutter class i, the uncertainty U_i is computed as:

$$U_i = \alpha(1-\text{mean}(\mathbf{s}_i))+\beta H(\mathbf{d}_i)+\gamma(1-\text{mean}(\mathbf{c}_i))$$

Here, $\mathbf{s}_i$ represents similarity scores specifically for examples of class i, $\mathbf{d}_i$ is the label distribution in the local neighborhood of class i, and $\mathbf{c}_i$ denotes the context matching scores for examples of class i. The parameters α, β, and γ are learnable weights that sum to 1, optimized end-to-end as part of the overall network training. $H(\mathbf{d}_i)$ represents the entropy of the label distribution for class i.

Confidence-weighted predictions $\hat{y}$ are computed by adjusting the softmax of the classification logits $\mathbf{p} \in \mathbb{R}^c$ with the class-specific uncertainties via element-wise multiplication:

$$\hat{y} = \text{softmax}(\mathbf{p}) \odot (1 - \mathbf{U}_{\text{class}})$$

This class-specific uncertainty formulation improves the model's ability to identify and down-weight unreliable predictions, enhancing robustness when encountering OOD stutter patterns.

3.5 Training Details

The overall StuD model is trained end-to-end. The primary objective is to minimize a Binary Cross-Entropy Loss with logits (BCEWithLogitsLoss), suitable for multi-label classification, which assesses the discrepancy between predicted logits and true labels for each stuttering class. Additionally, the contrastive loss function (as introduced in Section 3.3.2) is incorporated to guide the learning of dynamic weights within the RAG's similarity search. The total loss function is a weighted sum of these components: $L_{total} = L_{BCE} + \lambda L_{contrastive}$, where λ is a hyperparameter balancing the two losses, empirically set to 0.1. The specific formulation of the contrastive loss, targeting the RAG module's dynamic weights, is a triplet-based loss. For a given anchor embedding A, a positive sample P (same stuttering type), and a negative sample N (different stuttering type), the loss encourages the distance between A and P to be smaller than the distance between A and N by a margin m:

$$L_{contrastive} = \max(0, d(A, P) - d(A, N) + m)$$

where $d(\cdot, \cdot)$ is the Euclidean distance between weighted similarity scores, and m is a predefined margin. Positive and negative pairs are sampled dynamically from the current batch and the RAG reference database.

The model was trained with a batch size of 32. Optimization was performed using the AdamW optimizer (base learning rate: 2e-4) and a linear learning rate schedule that included 5% warmup steps. Early stopping was employed with a patience of 5 epochs, triggered by the lowest development set loss to prevent overfitting.

The model architecture included BiLSTMs for early-stage SSL and MFCC processing, each with hidden sizes of 512 and 128, respectively, and dropout rates of 0.1. Cross-attention modules and late-fusion transformers used 8 attention heads and a hidden size of 512. Training followed an end-to-end strategy, leveraging mixed precision (fp16) for computational efficiency and gradient clipping (max norm 1.0) to stabilize training. Training and feature extraction were distributed across 4 NVIDIA RTX 2080 Ti GPUs, leveraging 40 virtual CPU cores for data loading and preprocessing.

4 Experiments and Results

Table 1: F1 Scores for Multilabel Stutter Detection on SEP28k with 5-Fold Cross-Validation

No.	Acoustic Features	Text Embeddings	INJ	BLK	PRO	SND	WRD	AVG F1
1	Bayerl et al. (Bayerl et al., 2023)		0.32	0.77	0.53	0.53	0.64	0.56
2	Wagner et al. (Wagner et al., 2024)		0.57	0.74	0.56	0.54	0.64	0.61
3	✓ (ALL)	✗ (Without)	0.56	0.75	0.55	0.58	0.60	0.60
4	✗ (Without)	✓ (Llama-2)	0.23	0.31	0.24	0.29	0.35	0.28
5	✓ (Except Wav2Vec2.0)	✓ (Llama-2)	0.59	0.69	0.61	0.52	0.59	0.60
6	✓ (Except HuBERT)	✓ (Llama-2)	0.62	0.63	0.51	0.47	0.43	0.53
7	✓ (Except MFCC)	✓ (Llama-2)	0.79	0.71	0.59	0.69	0.72	0.70
8	✓ (ALL)	✓ (Llama-2)	**0.90**	**0.78**	**0.81**	**0.78**	**0.83**	**0.82**

Table 2: FluencyBank F1 scores for model trained on SEP28k

No.	Method	INJ	BLK	REP	PRO	AVG F1
1	Lea et al. (Lea et al., 2021)	0.82	0.56	0.66	0.67	0.68
2	Bayerl et al. (Bayerl et al., 2023)	0.84	0.33	0.51	0.60	0.57
3	Sheikh et al. (Sheikh et al., 2023)	0.64	0.04	0.22	0.42	0.33
4	Changawala et al. (Changawala and Rudzicz, 2024)	0.86	0.43	0.73	0.63	0.66
5	Proposed (w/o RAG)	0.84	0.66	0.72	0.64	0.71
6	Proposed	**0.91**	**0.73**	**0.80**	**0.82**	**0.81**

This section details the empirical evaluation of the proposed StuD model's performance on both in-distribution and out-of-distribution stuttering detection. Experiments validated our multimodal fusion, individual feature contributions, and the impact of the Retrieval-Augmented Generation (RAG) module for generalizing to unseen speech patterns. All evaluations utilized the F1 score, a critical metric in clinical and educational contexts for balancing precision and recall. Detailed training parameters are in Section 3.5.

4.1 Preliminary Feature Effectiveness

Initial experiments revealed that while lexical features alone underperformed acoustic-only features, their combination consistently outperformed either modality individually. This underscores the complementary nature of acoustic and linguistic information for comprehensive stuttering detection. Furthermore, Llama-2 consistently outperformed frozen BERT embeddings (768-dim) for linguistic feature extraction, demonstrating superior contextual understanding from its larger-scale pretraining on conversational data. This justified Llama-2's selection for our architecture.

4.2 Ablation Study on Acoustic Features (SEP-28k)

An ablation study on the SEP-28k dataset, using 5-fold speaker-independent cross-validation (as detailed in Section 2), evaluated the contribution of each acoustic feature, with results in Table 1. Table 1 highlights the differential importance of each acoustic feature. Wav2Vec 2.0's removal (row 5) most significantly impacted Interjections (INJ) and

Prolongations (PRO), dropping F1s from 0.82 to 0.60 and 0.61 respectively, likely due to its strength in capturing fine-grained temporal and prosodic patterns. Conversely, HuBERT's absence (row 4) most degraded performance for Prolongations (PRO), Sound Repetitions (SND), and Word Repetitions (WRD) (e.g., WRD from 0.83 to 0.43), aligning with its effectiveness in identifying repeated phonetic units. MFCC removal (row 3) caused a general, less severe, F1 decrease across all types, reflecting its foundational role in spectral information without targeting specific patterns.

A notable observation from Table 1 is the counter-intuitive behavior of MFCCs: while the full model (row 8) performs best with all three acoustic features, including MFCCs with only a single SSL model (rows 4, 5) degrades performance compared to using just Wav2Vec 2.0 and HuBERT (row 3). This suggests a complex interplay. We hypothesize that MFCCs provide unique low-level spectral information that becomes synergistic only when both high-level SSL features (Wav2Vec 2.0 and HuBERT) are present. Otherwise, MFCCs may introduce redundancy or interference, leading to a poorer fused representation. This dynamic underscores the necessity of a truly multi-scale acoustic input for optimal stuttering detection.

4.3 Out-of-Distribution Performance

To rigorously assess StuD's generalization capabilities on unseen speech patterns and speakers, we evaluated its performance on the FluencyBank dataset, which remained entirely unseen during training on SEP-28k. As discussed in Section 2, FluencyBank's distinct recording environments, speaker demographics, and elicitation methods (OASES protocol) make it an ideal cross-corpora dataset for evaluating OOD robustness. Table 2 presents the F1 scores for StuD compared to existing methods.

The results in Table 2 unequivocally demonstrate the significant impact of the RAG module on OOD generalization. Comparing "Proposed (w/o RAG)" (row 5) with "StuD (with RAG)" (row 6), the RAG module contributes to a substantial increase in average F1 score from 0.71 to 0.81, representing a 14% relative improvement on FluencyBank. This improvement is consistent across all dysfluency types, highlighting RAG's effectiveness in enhancing robustness by comparing new, unfamiliar stuttering patterns to known examples from the reference database and adaptively adjusting confidence based on similarity. This mechanism is crucial for improving generalization to unseen data, a primary goal of our research.

StuD also significantly outperforms all compared baseline methods on FluencyBank, achieving an average F1 score of 0.81. This represents a substantial leap in cross-corpora performance, particularly for challenging categories like Blocks and Prolongations, where prior methods often struggled.

4.4 Overall Performance and Context

Our comprehensive evaluation confirms that StuD achieves state-of-the-art performance on both in-distribution (SEP-28k, average F1 of 0.82) and out-of-distribution (FluencyBank, average F1 of 0.81) datasets. This consistent high performance across diverse datasets and stuttering types, particularly the significant improvements observed with the RAG module on OOD data, validates the efficacy of our multimodal architecture and two-stage fusion mechanism. The complementary nature of acoustic (Wav2Vec 2.0, HuBERT, MFCC) and linguistic (Llama-2 from Whisper) modalities is strongly reinforced by these results, emphasizing the necessity of multimodal approaches for accurate and robust speech disorder detection.

5 Conclusion and Future Work

In this work, we presented a novel approach to stuttered speech classification - StuD - wherein integration of RAG not only improved performance on known categories but also increased robustness when handling speech from unfamiliar speakers. Our fine-tuning procedure further refined the model by emphasizing the most salient features of stuttered speech, resulting in a significant performance boost on benchmark datasets.

Our work bridges the gap between controlled experimental settings and real-world speech variability, laying the groundwork for more adaptive and clinically relevant speech disorder diagnostic tools.

Future research can explore multimodal integration, such as incorporating video data, though the scarcity of annotated multimodal datasets poses a challenge. Addressing this limitation, along with leveraging transfer or unsupervised learning, could further enhance model robustness and generalizability for real-world use.

Limitations

While the StuD model demonstrates significant advancements in multimodal stuttering detection and OOD generalization, this work also highlights several inherent challenges in the field, which serve as crucial directions for future research.

First, our current framework exclusively focuses on acoustic and linguistic modalities. While these provide robust and complementary insights into stuttering phenomena, incorporating additional modalities, such as video data (e.g., facial expressions, head movements, or articulatory gestures), holds substantial promise. Such visual cues could offer further discriminative information, particularly for subtle or covert stuttering events. However, the scarcity of large-scale, annotated multimodal datasets for stuttered speech currently poses a significant challenge for integrating these additional data streams. Future work shall explore strategies for leveraging limited multimodal data, potentially through transfer learning or weakly supervised approaches, to develop more comprehensive diagnostic tools.

Second, while our model aims to differentiate genuine dysfluencies from semantically valid repetitions, and our analysis provides a broad understanding of performance, a more granular and clinically-oriented error analysis could further refine its utility. Understanding specific patterns of false positives (e.g., fluent speech misclassified as stuttered repetitions due to natural linguistic patterns like 'do-do' or emphatic phrases) and false negatives (e.g., missed subtle blocks or prolongations) is crucial for clinical deployment. Future efforts will involve detailed qualitative error analysis, potentially incorporating expert human review, to pinpoint the precise types of dysfluencies the model struggles with. This will guide targeted improvements, especially in enhancing the model's semantic grounding and lexical dependency.

Finally, while the Retrieval-Augmented Generation (RAG) module significantly enhances OOD robustness, its current retrieval mechanism is not end-to-end differentiable within the main optimization loop. This means that the process of selecting relevant reference examples is not directly optimized by the final classification loss. Future research could investigate differentiable retrieval architectures or alternative forms of RAG integration to allow for a more seamless, gradient-based optimization of the entire system, potentially leading to further performance gains and a deeper learned synergy between retrieval and classification.

Acknowledgments

The authors would like to thank I-Hub Data, IIIT Hyderabad, India, for their support during this research.

References

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Preprint*, arXiv:2006.11477.

Liam Barrett, Junchao Hu, and Peter Howell. 2022. Systematic review of machine learning approaches for detecting developmental stuttering. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 30:1160–1172.

Sebastian P. Bayerl, Florian Hönig, Joelle Reister, and Korbinian Riedhammer. 2020. Towards automated assessment of stuttering and stuttering therapy. *Preprint*, arXiv:2006.09222.

Sebastian P. Bayerl, Dominik Wagner, Ilja Baumann, Florian Hönig, Tobias Bocklet, Elmar Nöth, and Korbinian Riedhammer. 2023. A stutter seldom comes alone – cross-corpus stuttering detection as a multi-label problem. In *Interspeech 2023*, pages 1538–1542.

Sebastian Peter Bayerl, Dominik Wagner, Elmar Noeth, and Korbinian Riedhammer. 2022. Detecting dysfluencies in stuttering therapy using wav2vec 2.0. In *Interspeech 2022*, pages 2868–2872.

Nan Bernstein Ratner and Brian MacWhinney. 2018. Fluency bank: A new resource for fluency research and practice. *Journal of Fluency Disorders*, 56:69–80.

O. Bloodstein, N.B. Ratner, and S.B. Brundage. 2021. *A Handbook on Stuttering, Seventh Edition*. A Handbook on Stuttering. Plural Publishing, Incorporated.

Vrushank Changawala and Frank Rudzicz. 2024. Whister: Using whisper's representations for stuttering detection. In *Interspeech 2024*, pages 897–901.

Ashley Craig, Elaine Blumgart, and Yvonne Tran. 2009. The impact of stuttering on the quality of life in adults who stutter. *Journal of Fluency Disorders*, 34(2):61–71.

Iman Esmaili, Nader Jafarnia Dabanloo, and Mansour Vali. 2017. An automatic prolongation detection approach in continuous speech with robustness against speaking rate variations. *Journal of Medical Signals and Sensors*, 7:1 – 7.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *Preprint*, arXiv:2106.07447.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *Preprint*, arXiv:1702.08734.

Caroline Koedoot, Clazien Bouwmans, Marie-Christine Franken, and Elly Stolk. 2011. Quality of life in adults who stutter. *Journal of Communication Disorders*, 44(4):429–443.

Colin Lea, Vikramjit Mitra, Aparna Joshi, Sachin Kajarekar, and Jeffrey Bigham. 2021. Sep-28k: A dataset for stuttering event detection from podcasts with people who stutter. In *ICASSP*.

Valentin Mendelev, Tina Raissi, Guglielmo Camporese, and Manuel Giollo. 2021. Improved robustness to disfluencies in rnn-transducer based speech recognition. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6878–6882.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *Preprint*, arXiv:2212.04356.

Amrit Romana and Kazuhito Koishida. 2023. Toward a multimodal approach for disfluency detection and categorization. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Shakeel A. Sheikh, Md Sahidullah, Fabrice Hirsch, and Slim Ouni. 2021. Stutternet: Stuttering detection using time delay neural network. *Preprint*, arXiv:2105.05599.

Shakeel A. Sheikh, Md Sahidullah, Fabrice Hirsch, and Slim Ouni. 2022. Machine learning for stuttering identification: Review, challenges and future directions. *Neurocomputing*, 514:385–402.

Shakeel A Sheikh, Md Sahidullah, Fabrice Hirsch, and Slim Ouni. 2023. Advancing stuttering detection via data augmentation, class-balanced loss and multi-contextual deep learning. *IEEE Journal of Biomedical and Health Informatics*, 27(5):2553–2564.

Anne Smith and Christine Weber. 2017. How stuttering develops: The multifactorial dynamic pathways theory. *Journal of Speech, Language, and Hearing Research*, 60(9):2483–2505.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Dominik Wagner, Sebastian P. Bayerl, Ilja Baumann, Elmar Noeth, Korbinian Riedhammer, and Tobias Bocklet. 2024. Large language models for dysfluency detection in stuttered speech. In *Interspeech 2024*, pages 5118–5122.

Jeremy H. M. Wong and Nancy F. Chen. 2024. Distilling distributional uncertainty from a gaussian process. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 9956–9960.

Ehud Yairi and Nicoline Ambrose. 2013. Epidemiology of stuttering: 21st century advances. *Journal of Fluency Disorders*, 38(2):66–87.