Streaming Disfluency Detection for Cognitive Impairment Screening: BERT vs. LLMs on Korean Elderly Speech

Anonymous ACL submission

Abstract

Fine-grained disfluency detection is essential for cognitive impairment screening, and as speech-based screening tools grow in adoption, the demand for real-time classification is increasing. However, existing models require complete utterances before producing output, limiting their applicability to live clinical settings. We present the first streaming detection system covering eight disfluency categories for cognitively impaired Korean elderly speech. By learning to wait for further context before predicting a disfluency category, our model achieves competitive real-time detection accuracy at low latency. We also show the model's performance is clinically interpretable and generalizable to different screening tasks. Finally, we demonstrate the limitations of Large Language Models (LLMs) for fine-grained clinical disfluency classification.

1 Introduction

Language impairments such as semantic and syntactic deficits, substitutions of words and increased pauses all contribute to a decline in verbal fluency indicative of early cognitive change in Alzheimer's disease (AD) and mild cognitive impairment (MCI) (Leavitt et al., 2025; Mueller et al., 2018; Qi et al., 2023). Disfluent speech productions « *I-I, uh need to go to the store, uh no the pharmacy, uh to-tomorrow* » have therefore been regarded as clinically rich signal to screen for cognitive decline. Accordingly, previous works have leveraged disfluency features such as self-repairs, pauses, fillers, stutters, edit terms, word repetitions, substitutions and deletions to automatically predict diagnosis and cognitive scores in elders (Nasreen et al., 2021; Rohanian et al., 2020, 2021).

To facilitate by-hand disfluency annotation which can be work-intensive, the disfluency detection task has been tackled through end-to-end approaches operating directly on speech, using alignment-based detection frameworks (Lian et al., 2024; Zhou et al., 2024a) as well as sequence tagging discriminative models such as LSTM (Lou and Johnson, 2020; Zayats et al., 2016) or fine-tuned BERT (Bach and Huang, 2019; Rocholl et al., 2021; Romana et al., 2024). Such models are referred to as offline models, as they operate on full utterances, requiring the complete context before producing any output, with sequence tagging approaches achieving state-of-the-art performance on benchmark corpora such as Switchboard (Godfrey et al., 1992) and more recently extending to clinical populations (Farzana et al., 2022; Romana et al., 2022).

One limitation to offline models as demand in speech-based remote screening tools rises, is their inability to produce real-time disfluency detection for instant feedback or evaluation. To this end, streaming disfluency detection models offer predictions one word at a time, as the utterance progresses, allowing for real-time clinical decision support, adaptive task administration, and continuous session monitoring. Existing streaming disfluency models (Chen et al., 2022; Futami et al., 2023; Hough and Schlangen, 2017), however, are designed for downstream NLP tasks where disfluencies are noise to be removed, thereby operating under binary or simplified tagsets that discard the clinically relevant distinctions between disfluency types. This is fundamentally at odds with the clinical view, where disfluencies are the signal of interest, leaving fine-grained streaming detection for cognitive screening largely uninvestigated.

Another challenge in clinical disfluency detection is the computational and data cost of training discriminative models on limited clinical annotations, making it difficult to scale to new populations or disfluency categories. One alternative is to leverage the generative capabilities of Large Language Models (LLMs), which require no task-specific training. LLM-based disfluency detection

using parameter-efficient fine-tuning (Lian et al., 2024; Wagner et al., 2024) or prompting techniques (de Lima and Campelo, 2024; Teleki et al., 2025) has shown promise, but investigation on fine-grained clinical categories remains limited.

Our contributions are threefold. First, we propose the first fine-grained adaptive-wait streaming disfluency detection system for Korean elderly speech, covering eight linguistically motivated categories across a clinical population spanning AD, MCI, and cognitively intact speakers, and show that streaming training acts as a form of regularisation that benefits full-sequence detection. Second, we demonstrate that the model's learned wait behaviour is clinically interpretable: commitment latency tracks the structural complexity of disfluency categories, varies systematically across cognitive impairment stages, and generalises to spontaneous clinical speech. Third, we provide the first systematic comparison between discriminative fine-tuning and LLM prompting for fine-grained clinical disfluency classification, revealing the limits of generative approaches on rare and structurally complex categories.

2 Related Work

2.1 Clinical disfluency detection models

While disfluency detection research on typical speech is varied, clinical populations remain underrepresented. Early end-to-end approaches operated purely on the audio signal: Kourkounakis et al. (2021) used STFT spectrograms with a ResNet-BiLSTM architecture on stuttered speech, and Bayerl et al. (2022) fine-tuned wav2vec 2.0 with multi-task learning for stuttering therapy monitoring. Recognising that transcripts carry complementary linguistic structure, subsequent work progressively introduced text information into the detection pipeline. Lian et al. (2024) and Zhou et al. (2024a) departed from this by introducing speech-text alignment matrices at the phoneme and word level, detecting disfluencies as structural anomalies in the alignment space on aphasia speech. Zhou et al. (2024b) further collapsed the speech-text boundary by training Whisper end-to-end with CTC loss on disfluency-tagged token targets, producing annotated transcripts directly from audio on both stuttered and aphasia speech.

On the transcript side, Romana et al. (2022) showed that out-of-domain adaptation of BERT pre-trained on the transcripts of the Switchboard corpus consistently outperforms LSTM architectures in a disfluency annotation task on pathological speech including stuttering and Parkinson's disease. Most similar to our setup, Farzana et al. (2022) leveraged a BERT model to automatically annotate four types of repetition and correction disfluencies in AD patients' speech transcripts. The authors further demonstrated that removing those disfluencies from transcripts to predict AD reduced classification performance, underlining the importance of disfluent signal in speech-based detection frameworks.

2.2 Streaming disfluency detection models

Although streaming disfluency detection offers clear advantages for live clinical deployment, the approach has attracted considerably less attention than its offline counterpart, as typical neural architectures require the full sequence before producing output. Early contributions from Hough and Schlangen (2015) showed the robustness of RNNs on streaming disfluency detection when paired with a Markov decoder. Hough and Schlangen (2017) further showed that jointly training LSTMs on streaming disfluency detection and utterance segmentation improves both accuracy and latency over separate task training.

Central to streaming disfluency detection is the lookahead window, allowing the model to consume future tokens before committing to a prediction (Baumann et al., 2011). Madureira and Schlangen (2020) were the first to show that bidirectional encoders such as BERT can operate incrementally by training on incomplete prefixes rather than full sequences, at a modest accuracy cost. They further showed that waiting for one or two additional tokens before committing to a prediction reduces errors, suggesting that right context is important and can be achieved through a fixed lookahead window. Similarly, Chen et al. (2020) proposed a Transformer encoder with a controllable self-attention mechanism that restricts each token's attention to a fixed number of future tokens L, jointly modelling real-time punctuation prediction and disfluency detection. Building on this, Rohanian and Hough (2021) fine-tuned $\text{BERT}_{\text{LARGE}}$ on successive prefixes of increasing length, while also exploring fixed lookahead as a decoding baseline for streaming disfluency detection. Their best-performing model replaced the fixed lookahead with GPT-2 generated continuations to simulate missing right context, combined with POS tags and word timings,

achieving accuracy competitive with offline models. Futami et al. (2023) jointly trained a streaming Transformer on ASR and disfluency detection using audio block processing with a fixed lookahead window, showing that joint modeling reduces latency and improves robustness to ASR errors compared to pipeline approaches.

Our work builds most directly on the recent contributions of Chen et al. (2022), in which authors used incremental fine-tuning on successive prefixes with a dynamic lookahead window on a small vocab BERT model. In addition to a disfluency prediction head, they added a wait-head mechanism which is trained to produce wait or emit prediction decisions based the model's hidden representations. Essentially, the model dynamically learns to directly output a prediction or wait until further right context has been shown to emit. Unlike their work, which focused on binary disfluent/fluent classification of tokens for post-processing disfluency removal, we adapt the training scheme to predict multiple disfluency categories for clinically relevant detection. Furthermore, we upgrade the architecture with additional components to improve both streaming accuracy and commitment latency.

3 Streaming and Offline Disfluency Detection

3.1 Discriminative Model

We leverage KR-BERT (Lee et al., 2020), a Korean $\text{BERT}_{\text{BASE}}$ model (Devlin et al., 2019) which we further adapt on utterances of Korean elderly speech via masked language modeling before training. Using this architecture, we extend the streaming sequence tagging framework of Chen et al. (2022) in three respects: the disfluency classification scheme is expanded from binary to eight-class fine-grained disfluency classification, the wait head is conditioned on the tag head's output distribution rather than operating independently, and a causal GRU (Chung et al., 2014) belief accumulator is introduced on top of the BERT encoder.

Tag and Wait heads Two lightweight linear classifiers produce per-token tag and wait predictions on top of the BERT encoder. The tag head produces per-token disfluency logits over 8 fine-grained categories:

$$f(\mathbf{x}_t) = \mathbf{W}_{\text{tag}}\mathbf{x}_t + \mathbf{b}_{\text{tag}} \in R^8 \quad (1)$$

The wait head makes a binary emit/wait decision at each token position, determining whether the

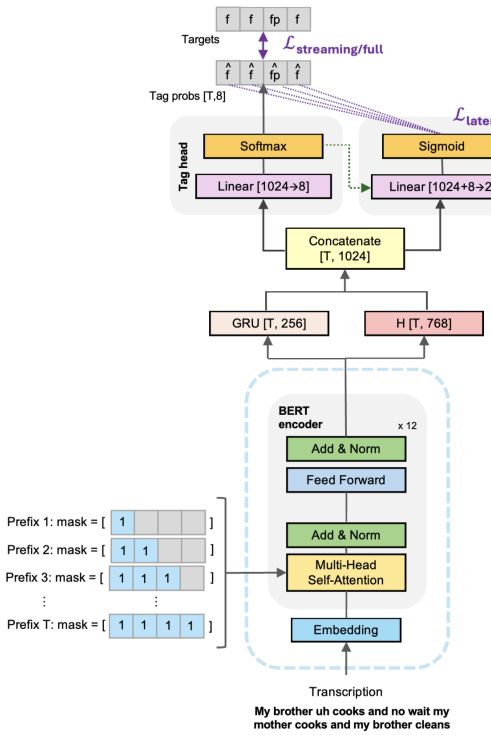

Figure 1: The architecture of our streaming BERT model. The input transcription is fed as T successive causal prefixes, each masked to expose only the first i tokens to the BERT encoder. The BERT hidden states H are passed through a causal GRU belief accumulator that maintains a summary of disfluency history across token positions. The GRU output and BERT hidden state are concatenated into a combined representation. The tag head (left) produces per-token probability distributions over eight disfluency categories. The wait head (right) receives both the encoder representation and the tag head's softmax output, and emits a binary commit/wait decision at each token position.

model should commit to a prediction given the tokens observed so far or defer until more context is available. In Chen et al. (2022), this head operates directly on the encoder representation, independently of the tag head. We depart from this by conditioning the wait head on the tag head's current output distribution, concatenating the softmax tag probabilities to the encoder representation before the wait decision is made:

$$g(\mathbf{x}_t,\ f(\mathbf{x}_t)) = \mathbf{W}_{\text{wait}}[\mathbf{x}_t;\ \sigma(f(\mathbf{x}_t))] + \mathbf{b}_{\text{wait}} \in R^2 \quad (2)$$

The intuition is that the decision of whether to defer should be sensitive to the model's current disfluency belief: a token assigned high probability to *fluent* warrants rapid commitment, whereas uncertainty across disfluency categories should encourage the model to wait for additional context before

committing.

Training objective Let (x, y) be an input-output sequence pair with prefixes $x_1, x_2, \ldots, x_{|x|}$, $f(x)$ the tag head logits, and $g(x)$ the wait head logits. The full-sequence loss anchors the tag head to the complete utterance:

$$\mathcal{L}_{\text{full}}(x, y) = H(\sigma(f(x)),\, y) \tag{3}$$

where $H(\cdot, \cdot)$ is cross-entropy and $\sigma(\cdot)$ is softmax. This term alone constitutes the offline training objective. In the streaming setting, the model must additionally learn to classify correctly from incomplete prefixes and to decide at each token whether to emit a prediction or wait for more context. The streaming objective therefore introduces two additional terms computed over the intermediate prefixes $x_1, \ldots, x_{|x|-1}$. The streaming loss supervises the tag head at each prefix step, but only at positions where the wait head has committed to a prediction:

$$\begin{aligned}\mathcal{L}_{\text{streaming}}(x, y) = \sum_{i=1}^{|x|-1} m(\sigma(g(x_i))) \\ \circ\, H(\sigma(f(x_i)),\, y_i)\end{aligned} \tag{4}$$

where $m(\sigma(g(x_i)))$ is a binary mask that zeros out positions at and beyond the first wait prediction in prefix x_i. Importantly, the model could trivially minimise $\mathcal{L}_{\text{streaming}}$ by always waiting and never committing. The latency penalty addresses this by penalising the expected number of tokens deferred before emission:

$$\mathcal{L}_{\text{latency}}(x) = \sum_{i=1}^{|x|-1} \sum_{j=1}^{i} (i - j)\, \sigma(g(x_i)_j) \tag{5}$$

The combined objective is:

$$\mathcal{L}_{\text{train}} = \mathcal{L}_{\text{full}} + \gamma\, \mathcal{L}_{\text{streaming}} + \lambda\, \mathcal{L}_{\text{latency}} \tag{6}$$

At inference time, the model receives tokens one by one and must produce meaningful tag and wait decisions from every incomplete observation of the utterance. To teach the model this behaviour during training, we simulate the streaming scenario by exposing it to every possible prefix of each training sequence. Concretely, a sequence of length $|x|$ gives rise to $|x|$ training instances, where instance i contains only the first i tokens $x_1, \ldots, x_i$ and is blind to all future tokens. This is implemented efficiently by constructing an attention mask $M_{b,i,j} = \mathbf{1}[j \leq i] \cdot \text{mask}_{b,j}$ that restricts self-attention to the first i positions. All $|x|$ masked instances are stacked into an expanded batch of shape $BT \times T$ and processed in a single forward pass, with the loss computed jointly over all prefix states.

Casual GRU belief accumulator Although the prefix-masked BERT forward passes enforce causality during training, each prefix is encoded independently: BERT at step i has no recollection of the representations computed at step $i - 1$. This is a meaningful limitation for disfluency detection, where many phenomena are inherently sequential and history-dependent. For instance, repetitions and restarts are only identifiable as disfluent relative to what was said earlier in the utterance. To address this, we append a single-layer unidirectional GRU that operates over the BERT hidden states. Within each forward pass the GRU maintains a running belief state $\mathbf{g}_t$ updated incrementally from left to right, encoding a compressed summary of the disfluency history seen so far. The BERT hidden state and GRU output are concatenated to form the combined representation:

$$\mathbf{C}_t = [\mathbf{h}_t;\, \mathbf{g}_t] \in R^{1024} \tag{7}$$

$\mathbf{h}_t$ contributes rich local contextualisation from the encoder, while $\mathbf{g}_t$ contributes accumulated sequential belief over the unfolding utterance.

3.2 Generative Model

We select Qwen3-30B-A3B (Yang et al., 2025) for its scale and multilingual pretraining which provide strong Korean language understanding without task-specific adaptation, and its instruction-tuned variant which enables zero- and few-shot prompting through natural language task descriptions. Unlike BERT, Qwen3 is a decoder-only autoregressive model that generates output token-by-token conditioned on the full input context. Rather than directly modelling $P(\text{tag} \mid \text{context})$ over a fixed label set, it models $P(\text{next token} \mid \text{context})$ over the complete vocabulary, and disfluency tag prediction is achieved by prompting the model to generate integer sequences corresponding to tag labels. This formulation has important implications: the model has no native mechanism for streaming prediction, no explicit uncertainty representation over tag classes, and no training signal for latency-aware commitment. We address each of these through prompt design and a chain-of-thought (CoT) streaming simulation described below.

Zero/Few-shot prompting We evaluate three offline prompting conditions of increasing informativeness. In the *zero-shot minimal condition*, the

model receives only a compact tag legend alongside the transcript, with no defintions or structural rules. In the *zero-shot with rules condition*, the prompt is augmented with full tag definitions, repair relation summaries, and explicit linguistic constraints. In the *few-shot with rules condition*, six annotated examples drawn from our training data are supplemented as conversation turns, covering all major disfluency patterns in addition to the rules.

Chain-of-Thought (CoT) prompting We propose a streaming simulation that draws a functional parallel to the tag-conditioned wait head of the discriminative model. At each token position t, the model receives only the left context $x_{1:t+w}$ where $w \in \{0, \ldots, W\}$ is the number of additional words consumed before committing, and is prompted to reason explicitly about whether right context is necessary before predicting the tag at position t. This reasoning follows a three-step chain-of-thought format:

1. *Disfluency likelihood*: high or low
2. *Right context needed*: yes or no
3. *Output*: tag in $[0, 7]$ or *wait*

If the model outputs *wait*, the context window is extended by one word and the query is repeated until the model comits to a tag. This CoT streaming simulation is applied to each prompting condition zero/few-shot prompting condition allowing us to assess whether prompt informativeness affects not only offline classification accuracy but also the model's streaming behaviour and latency.

4 Experimental Setup

4.1 Data

Our experiments draw on two complementary corpora. The AIHub Elderly Speech corpus [1] is a large-scale Korean elderly speech dataset collected from speakers aged 60 and above. Since the corpus contains no disfluency annotations, we apply a rule-based augmentation pipeline described below, yielding 2,609,656 utterances separated into train, validation, and test sets following an 8:1:1 ratio. The Voxgnosis corpus is an in-house clinical dataset in which 141 Korean elderly participants spanning three diagnostic groups (AD $n = 32$, MCI $n = 47$, and cognitively normal (CN) controls $n = 62$). Each participant produced 18 spoken responses to a word-ordering task specifically designed to elicit disfluency-rich speech by placing simultaneous demands on lexical retrieval and syntactic planning. Disfluency annotations were produced by two trained speech professionals following the Switchboard corpus annotation manual (Meteer and Taylor, 1995), from which eight categories applicable to our clinical population were derived: fluent, filled pause, edit term, restart, repetition, stutter, substitution, and insertion. Full category definitions are provided in Appendix A.1. The annotated corpus yields 2,443 utterances partitioned into train and validation sets following an 8:2 ratio stratified by diagnosis group. Word-level statistics for both corpora are summarised in Table 1.

[1] https://www.aihub.or.kr/aihubdata/data/view.do?dataSetSn=107

	AIHub (aug.)	Voxgnosis
# Utterances	2,609,656	2,443
# Words	39,734,253	16,491
Fluent	75.13%	86.97%
Filled pause	0.88%	0.73%
Edit term	1.98%	1.04%
Restart	8.60%	4.68%
Repetition	7.44%	2.64%
Stutter	2.77%	1.75%
Substitution	0.93%	0.66%
Insertion	2.26%	1.54%

Table 1: Word-level statistics for the AIHub augmented and Voxgnosis corpora. Tag percentages reflect the proportion of words assigned to each disfluency category.

Data augmentation Since the AIHub corpus consists of fluent transcripts, we apply a rule-based augmentation procedure to inject synthetic disfluencies prior to training (see Appendix A.2 for the full pipeline). Unlike prior augmentation approaches that inject disfluencies uniformly at random (Lian et al., 2024; Zhou et al., 2024b), our pipeline is fully grounded in the clinical distribution. Tag injection probabilities are derived empirically from the Voxgnosis word-level disfluency distribution and injection rules are constrained by bigram patterns extracted from Voxgnosis annotations. Substitutions are generated using FastText (Bojanowski et al., 2017)-based semantic similarity.

4.2 Evaluation Metrics

We evaluate models along two axes: classification performance and latency. All models are evaluated on precision, recall, and macro-F1 over the eight

disfluency categories, computed at the token level. Macro-averaging is preferred over weighted averaging given the severe class imbalance, as it treats each disfluency category equally regardless of frequency.

Offline metrics For offline models, which observe the complete utterance before producing predictions, we report macro-F1, precision, and recall computed over all valid token positions.

Streaming metrics At inference time, the model processes the utterance prefix by prefix, receiving one additional token at each step. The moment the wait head emits for token t, the tag head prediction at that step is locked in as the final prediction for that token. Streaming F1, precision, and recall are computed over these committed predictions using standard token-level formulas, where each prediction was produced from an incomplete prefix rather than the full utterance. Differently from Chen et al. (2022), who credit any prefix step at which the correct label appears, we score only the prediction at the moment of commitment, more directly reflecting deployment conditions where committed predictions cannot be revised. Average Wait Tokens (AWT) (based on Chen et al. (2022)) measures the average number of tokens consumed between first seeing token t and first emitting a prediction for it. Tokens To Detection (TTD) (based on Hough and Schlangen (2017)) is computed exclusively on correctly detected disfluent tokens and measures the average number of tokens consumed after position t before the model first emits a correct disfluency label. For offline models, which have no wait mechanism and commit immediately, both AWT and TTD are zero by construction. Detailed formulas for all metrics are provided in Appendix A.3.

5 Results

5.1 Detection Performance

Table 2 reports streaming and offline disfluency detection performance evaluated on the Voxgnosis test set. Results on the AIHub test set are provided in Appendix A.4.

When trained with the streaming objective, the model achieves F1 = 0.63 in the streaming evaluation while maintaining offline F1 = 0.74. Notably, the streaming model outperforms its offline counterpart in the offline evaluation by a relative 6% (F1 0.74 vs. 0.70), suggesting that exposure to partial utterances during training acts as a form of regularisation that improves disfluency detection in full sequences. Adding the causal GRU belief accumulator improves streaming F1 from 0.63 to 0.66, with a modest increase in AWT (0.39 vs. 0.34) and TTD (1.53 vs. 1.47), while offline F1 remains stable at 0.74, thus improving streaming accuracy at a small latency cost without affecting full-sequence performance. In the offline evaluation setting, streaming models show higher precision (P = 0.78 vs. 0.65) but lower recall (R = 0.73 vs. 0.79) than their offline counterparts, indicating that the wait mechanism encourages more conservative and targeted predictions at the cost of missing some disfluent tokens.

Qwen3-30B-A3B performs substantially below all fine-tuned discriminative models despite its scale. In the offline setting, prompt informativeness drives a 100% relative improvement in F1 from zero-shot minimal (0.13) to few-shot with rules (0.26), confirming that in-context demonstrations provide meaningful signal for this task. In the streaming setting, CoT reasoning consistently outperforms its offline counterpart across all prompting conditions, indicating that forcing token-by-token deliberation over left context yields additional gains beyond what prompt engineering alone can achieve. However, streaming F1 remains flat at 0.30 regardless of prompt informativeness, suggesting that CoT deliberation reaches a ceiling that additional examples cannot overcome, and that the fundamental bottleneck is the difficulty of reducing fine-grained token classification to a generative prompting task without task-specific training. Notably, when streaming F1 ceils at 0.30, the zero-shot with rules condition achieves the lowest AWT (0.10) and TTD (1.13) across all Qwen streaming conditions suggesting that structural linguistic constraints in the prompt encourage faster and more decisive commitment while additional examples may confuse the model.

Per-category detection and latency Table 3 reports per-category F1 and AWT. Fluent and edit term tokens are consistently well detected by both BERT variants (F1 $\geq$ 0.90), while substitution and insertion remain the hardest categories across all models (BERT str.+GRU: F1 = 0.59 and 0.55 respectively; Qwen3: F1 = 0.00 and 0.05). Across most disfluency categories, the streaming model matches or outperforms its offline counterpart on full-sequence F1 despite never observing complete utterances during prefix training. Filled pause improves from 0.74 to 0.81, restart from 0.68 to 0.72,

Model (#Param.)	Training Scheme	Training Data	Streaming Detection					Offline Detection		
			F1↑	P↑	R↑	AWT↓	TTD↓	F1↑	P↑	R↑
Vanilla KR-BERT (95M)	Offline	Voxgnosis	0.36	0.36	0.50	**0.00**	**0.00**	0.45	0.39	0.66
		AIHub + Voxgnosis	0.47	0.48	0.57	**0.00**	**0.00**	0.67	0.61	0.76
Elderly-speech KR-BERT (95M)	Offline	Voxgnosis	0.38	0.42	0.49	**0.00**	**0.00**	0.53	0.46	0.71
		AIHub + Voxgnosis	0.52	0.49	0.59	**0.00**	**0.00**	0.70	0.65	**0.79**
	Streaming	AIHub + Voxgnosis	0.63	0.61	0.67	0.34	1.47	**0.74**	**0.78**	0.73
+ GRU (96.3M)	Streaming	AIHub + Voxgnosis	**0.66**	**0.63**	**0.70**	0.39	1.53	**0.74**	0.76	0.74
Qwen3-30B-A3B (30B)	–	Zero-shot minimal	0.15	0.16	0.16	0.49	2.72	0.13	0.15	0.17
	–	Zero-shot + rules	0.30	0.34	0.29	0.10	1.13	0.20	0.22	0.23
	–	Few-shot + rules	0.30	0.33	0.30	0.95	2.68	0.26	0.29	0.27

Table 2: Comparison of disfluency detection models under streaming and offline conditions. AWT = Average Wait Tokens, TTD = Tokens to Detection. Elderly-speech KR-BERT corresponds to our MLM-adapted model.

	BERT offline		BERT str.+GRU		Qwen3 CoT	
Category	F1	AWT	F1	AWT	F1	AWT
Fluent	0.98	0.00	0.98	0.35	0.91	0.82
Filled pause	0.74	0.00	0.81	0.33	0.44	2.20
Edit term	0.94	0.00	0.91	0.35	0.27	2.67
Restart	0.68	0.00	0.72	0.83	0.07	1.35
Repetition	0.70	0.00	0.74	0.63	0.12	2.34
Stutter	0.60	0.00	0.61	0.97	0.20	2.28
Substitution	0.43	0.00	0.59	0.52	0.00	1.71
Insertion	0.56	0.00	0.55	1.32	0.05	1.65
Macro avg	0.70	0.00	0.74	0.39	0.26	0.95

Table 3: Per-category F1 and AWT across model variants.

and most notably substitution from 0.43 to 0.59, while edit term shows a marginal drop (0.94 to 0.91). This per-category pattern confirms that prefix training acts as structured regularisation, reducing false positives on ambiguous tokens by teaching the model to delay commitment until sufficient context is available, which in turn sharpens its full-sequence predictions.

The per-category AWT of the streaming BERT model reveals that detection difficulty and commitment latency are aligned: insertion incurs the highest AWT (1.32 tokens) followed by stutter (0.97) and restart (0.83), indicating that the wait head correctly identifies these as categories requiring more right context before committing. For Qwen3, the CoT AWT pattern does not track F1: edit term and repetition show the highest deferral (2.67 and 2.34 tokens) despite low F1 (0.27 and 0.12), suggesting that extended deliberation does not compensate for the absence of task-specific training signal on rare disfluency categories.

Clinical insights Detection performance varies across diagnosis groups (Figure 2a). Discriminative models consistently detect AD disfluencies best (BERT streaming + GRU: F1 = 0.79) and struggle most with MCI (F1 = 0.67), reflecting the greater severity and structural consistency of disfluency in advanced pathology versus the ambiguous intermediate patterns of mild cognitive impairment. Qwen3 does not show this pattern, suggesting that generative prompting fails to capture the clinically grounded disfluency signal.

Figure 2b reveals that while misclassifications as fluent tokens dominate across all groups, MCI uniquely shows substantial confusion between disfluency categories: substitution is confused with repetition (53.8%), insertion with restart (20.5%), and stutter with restart (12.0%), suggesting that MCI disfluency patterns are not only harder to detect but structurally more ambiguous than those of AD or NC speakers. In AD, edit terms are frequently misclassified as filled pauses (40.0%), because AD speakers produce edit term-like fillers such as *uh* within restart sequences, which are surface-identical to the filled pauses that MCI and NC speakers produce in non-restart contexts. Further comparison between offline and streaming BERT models shows streaming reduces AD substitution misclassification from 100.0% to 33.3%, benefiting detection of salient word-finding failures in severe impairment, while offline detection better handles NC insertion (28.6% vs. 40.8%), as conservative commitment penalises detection of the subtle characteristics of cognitively intact speech.

5.2 Impact of Tag-Probability Conditioning

A key architectural contribution over (Chen et al., 2022) is conditioning the wait head on the tag head's output distribution, allowing the emit decision to be informed by the model's current disfluency belief. To isolate this contribution, we ablate the BERT+GRU model by removing the tag probability input to the wait head. Offline F1 is comparable across both variants (0.74 vs. 0.75), indi-

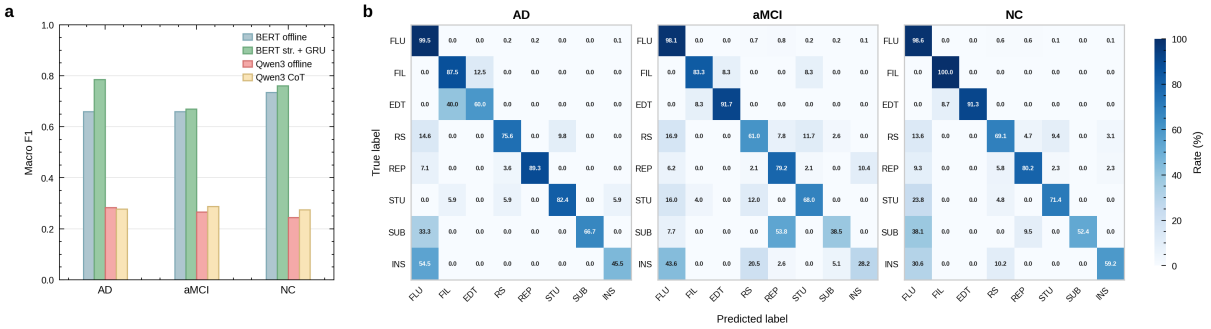

Figure 2: a. Offline F1 score comparison according to diagnosis and model type b. Disfluency detection confusion matrices from the streaming BERT model with GRU accumulator according to diagnosis

cating that tag conditioning does not meaningfully affect full-sequence classification. The difference emerges in the streaming setting: conditioning reduces AWT fourfold (from 1.94 to 0.39 tokens) and TTD from 2.75 to 1.53, while maintaining streaming F1 (0.66 vs. 0.67). Tag conditioning therefore allows the wait head to commit earlier and more confidently without sacrificing accuracy.

5.3 Generalization to Spontaneous Speech

We evaluate our discriminative models on a Korean equivalent of the "Trip to New-York" task (Harris et al., 2008), where the participant produces spontaneous speech on how they would prepare for, and execute a trip to Jeju island (from our Voxgnosis in-house corpus). Disfluencies were tagged in the same way as the word-ordering task on a total of 820 utterances.

Detailed per-category results are reported in Table 5 in Appendix A.5. Our best models transfer reasonably to this setting: the streaming model with GRU accumulator yields macro F1 = 0.62 (vs. 0.74 on word-ordering). This indicates that our model captures generalizable disfluency representations that transfer beyond the structured word-ordering task to spontaneous clinical speech. While the offline model performs similarly, the streaming model without GRU drops substantially to F1 = 0.47, likely due to the longer utterance lengths in the Trip to Jeju corpus (mean length = 12 vs 6 for word-ordering), for which full-sequence training and the GRU accumulator provide better generalisation. Per-diagnosis results reveal an interesting pattern: while AD is reliably detected group on the word-ordering task, it shows a 50% F1 drop relative to MCI and NC on the spontaneous task across all models, suggesting that disfluency patterns are task-dependent. The word-ordering task appears to elicit more discriminative disfluency patterns for MCI, while AD spontaneous speech presents more atypical patterns compared to the other groups.

6 Conclusion and Future Work

We presented a streaming BERT-based model for fine-grained disfluency detection in Korean elderly speech that simultaneously classifies tokens into eight clinical categories and dynamically decides when to commit to a prediction or wait for further context. Evaluated on a clinical corpus spanning AD, MCI, and cognitively normal speakers and two different speech tasks, our system achieves competitive detection accuracy at low latency, with streaming training further improving full-sequence detection over offline fine-tuning. We additionally show that large language model prompting does not match task-specific fine-tuning for fine-grained clinical disfluency classification, though chain-of-thought streaming simulation consistently outperforms offline prompting.

Future work will focus on integrating a streaming ASR model upstream, or a joint ASR-disfluency model, requiring careful handling of disfluency-relevant phenomena such as repetitions and stutters, which CTC-based decoding is known to suppress. Furthermore, the disfluency features extracted by our system could serve as interpretable inputs to a broader NLP pipeline for cognitive impairment screening, including multi-task classification of AD, MCI, and NC and regression of continuous cognitive scores such as MMSE, providing a pathway to automated neuropsychological assessment. Additionally, future work on generative models should explore fine-tuning disfluency-annotated text or multimodal speech-text data, as prompt-based approaches alone have consistently fallen short of acceptable detection performance.

7 Limitations

Our in-house Voxgnosis corpus is meaningfully limited in size compared to the augmented AIHub dataset which could lead to the model fitting predominantly to synthetic data rather than real clinical speech. We try to mitigate this concern by creating empirically derived rules from the clinical dataset for augmentation, meaning the synthetic data is grounded in the same distribution as our clinical corpus. This is reflected in our results as BERT model training on both AIHub and Voxgnosis datasets yields better performance than training on Voxgnosis alone, pointing to genuine generalization benefits rather than noise from the augmented dataset. Furthermore, though stratified by diagnosis and speaker, our Voxgnosis 8 :2 train-test split leads to a validation sample size below standard practice when involving clinical data. More robust validation methods, such as Leave-One-Patient-Out (LOPO) cross-validation remains a target of future works as our dataset grows.

Conceptually, our disfluency detection model was intended for live deployment settings in an end-to-end pipeline from streaming ASR to disfluency detection. However, here we use gold transcripts, meaning the model was not evaluated under realistic ASR conditions, limiting its deployment potential. Future work will address a disfluency-aware streaming ASR model up-front in the pipeline.

Our BERT vs. LLM comparison is asymmetric, as BERT models receive full supervised training while Qwen3-30B-A3B is evaluated exclusively under zero and few-shot prompting conditions. This reflects our proposition that LLMs should be leveraged for their ability to generalize without task-specific training, given the scarcity and cost of clinical annotations. However, our results suggest the bottleneck is not prompt informativeness but a fundamental difficulty in reducing fine-grained token classification to autoregressive generation without task-specific supervision. Parameter-efficient fine-tuning of LLMs on disfluency-annotated data and more sophisticated reasoning strategies such as Tree-of-Thought remain avenues for future work, though the latter may introduce unwanted delays at odds with real-time deployment.

References

Nguyen Bach and Fei Huang. 2019. Noisy BiLSTM-based models for disfluency detection. In *Proceedings of Interspeech 2019*, pages 4230–4234.

Timo Baumann, Okko Buß, and David Schlangen. 2011. Evaluation and optimisation of incremental processors. *Dialogue & Discourse*, 2(1):113–141.

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.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Angelica Chen, Vicky Zayats, Daniel D. Walker, and Dirk Padfield. 2022. Teaching BERT to wait: Balancing accuracy and latency for streaming disfluency detection. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 827–838, Seattle, United States. Association for Computational Linguistics.

Qian Chen, Mengzhe Chen, Bo Li, and Wen Wang. 2020. Controllable time-delay transformer for real-time punctuation prediction and disfluency detection. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8069–8073.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Pedro L. S. de Lima and Cláudio E. C. Campelo. 2024. Disfluency detection and removal in speech transcriptions via large language models. In *Proceedings of the 15th Brazilian Symposium in Information and Human Language Technology (STIL)*, pages 127–136. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Shahla Farzana, Ashwin Deshpande, and Natalie Parde. 2022. How you say it matters: Measuring the impact of verbal disfluency tags on automated dementia detection. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 37–48, Dublin, Ireland. Association for Computational Linguistics.

Hayato Futami, Emiru Tsunoo, Kentaro Shibata, Yosuke Kashiwagi, Takao Okuda, Siddhant Arora, and Shinji Watanabe. 2023. Streaming joint speech recognition and disfluency detection. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

John J. Godfrey, Edward C. Holliman, and Jane McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 517–520.

Jana Lia Harris, Swathi Kiran, Thomas P. Marquardt, and Valerie B. Fleming. 2008. Communication Wellness Check-Up©: Age-related changes in communicative abilities. *Aphasiology*, 22(7-8):813–825.

Julian Hough and David Schlangen. 2015. Recurrent neural networks for incremental disfluency detection. In *Proceedings of Interspeech 2015*, pages 849–853.

Julian Hough and David Schlangen. 2017. Joint, incremental disfluency detection and utterance segmentation from speech. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 326–336, Valencia, Spain. Association for Computational Linguistics.

Tedd Kourkounakis, Amirhossein Hajavi, and Ali Etemad. 2021. FluentNet: End-to-end detection of stuttered speech disfluencies with deep learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:2986–2999.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, pages 611–626. ACM.

Victoria M. Leavitt, Leila Simani, Afshin Borhani Haghighi, Marcus Koch, Alexandra König, and Jan Rusz. 2025. Language dysfunction as a primary feature of cognitive decline in neurological populations. *Journal of Neural Transmission*. Epub ahead of print 6 September 2025.

Sangah Lee, Hansol Jang, Yunmee Baik, Suzi Park, and Hyopil Shin. 2020. KR-BERT: A small-scale Korean-specific language model. *arXiv preprint arXiv:2008.03979*.

Jiachen Lian, Xuanru Zhou, Zoe Ezzes, Jet Vonk, Brittany Morin, David Baquirin, Zachary Miller, Maria Luisa Gorno-Tempini, and Gopala Krishna Anumanchipalli. 2024. SSDM: Scalable speech dysfluency modeling. In *Advances in Neural Information Processing Systems*, volume 37.

Paria Jamshid Lou and Mark Johnson. 2020. Improving disfluency detection by self-training a self-attentive model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3754–3763, Online. Association for Computational Linguistics.

Brielen Madureira and David Schlangen. 2020. Incremental processing in the age of non-incremental encoders: An empirical assessment of bidirectional models for incremental NLU. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 357–374, Online. Association for Computational Linguistics.

Marie Meteer and Ann Taylor. 1995. Dysfluency annotation stylebook for the Switchboard corpus. Technical report, University of Pennsylvania, Philadelphia, PA. Revised June 1995.

Kimberly D. Mueller, Bruce Hermann, Jonilda Mecollari, and Lyn S. Turkstra. 2018. Connected speech and language in mild cognitive impairment and Alzheimer's disease: A review of picture description tasks. *Journal of Clinical and Experimental Neuropsychology*, 40(9):917–939.

Shamila Nasreen, Morteza Rohanian, Julian Hough, and Matthew Purver. 2021. Alzheimer's dementia recognition from spontaneous speech using disfluency and interactional features. *Frontiers in Computer Science*, 3:640669.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Xiaoke Qi, Qing Zhou, Jian Dong, and Wei Bao. 2023. Noninvasive automatic detection of Alzheimer's disease from spontaneous speech: A review. *Frontiers in Aging Neuroscience*, 15:1224723.

Johann C. Rocholl, Vicky Zayats, Daniel D. Walker, Noah B. Murad, Aaron Schneider, and Daniel J. Liebling. 2021. Disfluency detection with unlabeled data and small BERT models. In *Proceedings of Interspeech 2021*, pages 766–770.

Morteza Rohanian and Julian Hough. 2021. Best of both worlds: Making high accuracy non-incremental transformer-based disfluency detection incremental. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3693–3703, Online. Association for Computational Linguistics.

Morteza Rohanian, Julian Hough, and Matthew Purver. 2020. Multi-modal fusion with gating using audio, lexical and disfluency features for Alzheimer's dementia recognition from spontaneous speech. In *Proceedings of Interspeech 2020*, pages 2187–2191.

Morteza Rohanian, Julian Hough, and Matthew Purver. 2021. Alzheimer's dementia recognition using acoustic, lexical, disfluency and speech pause features robust to noisy inputs. In *Proceedings of Interspeech 2021*, pages 3820–3824.

Amrit Romana, Kazuhito Koishida, and Emily Mower Provost. 2024. Automatic disfluency detection from untranscribed speech. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Amrit Romana, Minxue Niu, Matthew Perez, Angela Roberts, and Emily Mower Provost. 2022. Enabling off-the-shelf disfluency detection and categorization for pathological speech. In *Proceedings of Interspeech 2022*, pages 1916–1920.

Maria Teleki, Sai Janjur, Haoran Liu, Oliver Grabner, Ketan Verma, Thomas Docog, Xiangjue Dong, Lingfeng Shi, Cong Wang, Stephanie Birkelbach, Jason Kim, Yin Zhang, and James Caverlee. 2025. DRES: Benchmarking LLMs for disfluency removal. *arXiv preprint arXiv:2509.20321*.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. HuggingFace's Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Vicky Zayats, Mari Ostendorf, and Hannaneh Hajishirzi. 2016. Disfluency detection using a bidirectional LSTM. In *Proceedings of Interspeech 2016*, pages 2523–2527.

Xuanru Zhou, Anshul Kashyap, Steve Li, Ayati Sharma, Brittany Morin, David Baquirin, Jet Vonk, Zoe Ezzes, Zachary Miller, Maria Luisa Gorno-Tempini, Jiachen Lian, and Gopala Krishna Anumanchipalli. 2024a. YOLO-Stutter: End-to-end region-wise speech dysfluency detection. In *Proceedings of Interspeech 2024*, pages 937–941.

Xuanru Zhou, Jiachen Lian, Cheol Jun Cho, Jingwen Liu, Zongli Ye, Jinming Zhang, Brittany Morin, David Baquirin, Jet Vonk, Zoe Ezzes, Zachary Miller, Maria Luisa Gorno-Tempini, and Gopala Anumanchipalli. 2024b. Time and tokens: Benchmarking end-to-end speech dysfluency detection. *arXiv preprint arXiv:2409.13582*.

A Appendix

A.1 Disfluency Category Definitions

The following definitions describe the eight disfluency categories used for annotation in this work, derived from the Switchboard corpus annotation manual (Meteer and Taylor, 1995) and adapted to the linguistic properties of our cognitive impaired elderly speech dataset.

- **Fluent:** Fluent tokens are words that do not belong to any of the disfluency categories defined below and constitute normal, uninterrupted speech.

Standalone disfluencies

- **Filled pause:** Filled pauses are uttered speech blocks that carry no semantic content. Some examples include: uh (Korean 어), um (Korean 음)

- **Repetition:** Repetitions are correctly produced words that are repeated a second time without communicative necessity. Unlike stutters, they carry semantic meaning.

- **Stutter:** Stutters are incomplete word fragments that carry no semantic meaning in isolation and must be followed or preceeded by the complete root word to be interpretable.

Restart sequences Restart sequences adopt the following form:

```
Show flights from Boston on uh from Denver on Monday
           |-----RM-----|-IM-|----RR----|
                        "IP"
```

RM = reparandum
IP = interruption point
IM = interregnum (filled pause or edit term)
RR = repair

The interregnum consists of a filled pause or an edit term. While filled pauses may occur outside restart sequences, edit terms are exclusive to the interregnum position and always follow a reparandum. The restart sequence concludes with a repair, which takes the form of a repetition, a substitution, or an insertion.

- **Restart:** The restart marks the onset word of the repair sequence, corresponding to the reparandum. Here, the term restart is used in place of reparandum for clarity.

- **Edit term:** Edit terms are interregnum elements that carry partial semantic content and occur between the restart onset and the repair. Some examples include: no (Korean 아니다, 아니야 and variants), wait (Korean 잠깐).

- **Repetition:** Within a restart sequence, a repetition repair is one in which a word from before the interruption point is carried over as the first word of the repair, before the utterance continues normally.

- **Substitution:** A substitution repair occurs when a word from before the interruption point is replaced by a different word from the same part of speech (POS). Any repair word that does not appear in the left context and belongs to a different POS from the replaced word is labelled as an insertion.

- **Insertion:** Insertions are repair words that introduce new content not present in the left context and do not belong to the same POS as any replaced word.

Complex restart sequences Some restart sequences exhibit additional structural complexity:

Chaining:

- *Repetition + Insertion:* one word from before the interruption point is repeated and followed by a new repair word not present in the left context.

- *Repetition + Substitution:* one word from before the interruption point is repeated and followed by a substituted word from a different part of speech.

Nesting: a complete restart sequence is embedded within another restart sequence, resulting in a hierarchically structured repair.

A.2 Synthetic Disfluency Augmentation

Algorithm 1 Rule-based disfluency injection

Require: $\mathbf{w} = (w_1, \ldots, w_N)$, rates τ, FastText model F
Ensure: augmented sequence with disfluency tags

result $\leftarrow$ [], produced $\leftarrow$ []
for each $w_i \in \mathbf{w}$ **do**
  if w_i is pause **then** append (w_i, FLU); **continue**
  end if
  rem $\leftarrow$ non-pause tokens after i
  if Bernoulli(τ_{STT}) **then**
    $n \sim \{1, 2, 3\}$, weights $(0.70, 0.20, 0.10)$
    for $k = 1, \ldots, n$ **do** append $(w_i[:\ell], \text{STT})$, $\ell \sim \mathcal{U}(1, |w_i| - 1)$
    end for
    append (w_i, FLU); **continue**
  end if
  if Bernoulli(τ_{FIL}) **then**
    append (FP(), FIL); with prob. 0.15 append second (FP(), FIL)
  end if
  if Bernoulli(τ_{RS}) **then**
    $d \sim \{1, 2, 3\}$, weights $(0.50, 0.25, 0.25)$
    for $k = 1, \ldots, d$ **do**
      append (w_i, RS); $r \sim \mathcal{U}(0, 1)$
      if $r < p_{\text{EDT}}$ **then** append ET(); optionally (REP) or (INS)
      else if $r < p_{\text{EDT}} + p_{\text{REP}}$ **then** optionally append (STT) then (REP)
      else if $r < p_{\text{EDT}} + p_{\text{REP}} + p_{\text{SUB}}$ **then** append (NN(w_i, F), SUB)
      else if rem $\neq \emptyset$ append (rem[0], INS) **else** append ET()
      end if
    end for
  else if Bernoulli(τ_{REP}) **and** produced $\neq \emptyset$ **then**
    append (w_i, FLU), (w_i, REP); **continue**
  end if
  append (w_i, FLU); produced $\leftarrow$ produced $\cup \{w_i\}$
end for
return result

A.3 Evaluation Metrics

Let $x = (x_1, \ldots, x_T)$ be an input utterance of length T, $y = (y_1, \ldots, y_T)$ the corresponding gold label sequence, and $c(t)$ the prefix step at which the wait head commits for token t. Let $\hat{y}_{i,t}$ denote the tag head prediction for token t after consuming prefix $x_{1:i}$.

Offline metrics For offline models, the complete utterance is available before any prediction is made. Precision, recall, and macro-F1 are computed over all token positions using standard token-level formulas:

$$P = \frac{TP}{TP + FP},$$

$$R = \frac{TP}{TP + FN},$$

$$F1 = \frac{2 \cdot P \cdot R}{P + R}$$

where TP, FP, and FN are aggregated at the token level and macro-averaged over the eight disfluency categories.

Streaming metrics At inference time, the model processes the utterance token by token. For each token t, the wait head emits at step $c(t) \geq t$, locking in the tag head prediction $\hat{y}_{c(t),t}$ as the final committed prediction. For each category k, streaming true positives, false positives, and false negatives are defined as:

$$TP_{\text{streaming}} = \sum_{t=1}^{T} \mathbf{1}[\hat{y}_{c(t),t} = k \wedge y_t = k],$$

$$FP_{\text{streaming}} = \sum_{t=1}^{T} \mathbf{1}[\hat{y}_{c(t),t} = k \wedge y_t \neq k],$$

$$FN_{\text{streaming}} = \sum_{t=1}^{T} \mathbf{1}[\hat{y}_{c(t),t} \neq k \wedge y_t = k]$$

Streaming precision, recall, and F1 are then:

$$P_{\text{streaming}} = \frac{TP_{\text{streaming}}}{TP_{\text{streaming}} + FP_{\text{streaming}}},$$

$$R_{\text{streaming}} = \frac{TP_{\text{streaming}}}{TP_{\text{streaming}} + FN_{\text{streaming}}},$$

$$F1_{\text{streaming}} = \frac{2 \cdot P_{\text{streaming}} \cdot R_{\text{streaming}}}{P_{\text{streaming}} + R_{\text{streaming}}}$$

all macro-averaged over the eight disfluency categories. This differs from Chen et al. (2022), who score over all (i, t) prefix pairs where $t \leq i$, crediting any prefix step at which the correct label appears. Our formulation scores only the single committed prediction per token, more directly reflecting deployment conditions where predictions cannot be revised once emitted.

Average Wait Tokens (AWT) AWT (Chen et al., 2022) measures the average number of additional tokens consumed between first seeing token t and emitting a prediction for it:

$$\text{AWT} = \frac{1}{T} \sum_{t=1}^{T} (c(t) - t)$$

A value of AWT $= 0$ indicates that the model commits immediately upon first seeing each token, with no lookahead.

Tokens To Detection (TTD) TTD (Hough and Schlangen, 2017) is computed exclusively over correctly detected disfluent tokens and measures the average number of tokens consumed after the disfluent position before the first correct emission:

$$\text{TTD} = \frac{1}{|D|} \sum_{t \in D} (c(t) - t)$$

where $D = \{t : y_t \neq \text{fluent} \wedge \hat{y}_{c(t),t} = y_t\}$ is the set of correctly detected disfluent tokens.

A.4 AIHub Test Set Results

Model	Streaming					Offline		
	F1↑	P↑	R↑	AWT↓	TTD↓	F1↑	P↑	R↑
Vanilla KR-BERT, Offline	0.69	0.75	0.70	**0.00**	**0.00**	0.83	0.82	0.86
Elderly KR-BERT, Offline	0.75	0.77	0.77	**0.00**	**0.00**	0.95	0.92	0.98
Elderly KR-BERT, Streaming	0.88	0.84	0.95	0.42	1.04	**0.96**	**0.94**	**0.99**
Elderly KR-BERT + GRU, Streaming	**0.91**	**0.87**	**0.97**	0.47	1.25	**0.96**	0.93	0.98

Table 4: Results on the AIHub test set (AIHub + Voxgnosis training). AWT = Average Wait Tokens, TTD = Tokens to Detection.

A.5 Generalization to Spontaneous Speech

	BERT Offline		BERT Streaming		BERT Str. + GRU	
Category	*Word Ord.*	*Trip*	*Word Ord.*	*Trip*	*Word Ord.*	*Trip*
Macro F1	0.70	0.60	0.74	0.47	0.74	0.62
Fluent	0.98	0.97	0.99	0.97	0.98	0.97
Filled Pause	0.74	0.84	0.79	0.72	0.81	0.78
Edit Term	0.94	1.00	0.92	0.57	0.91	1.00
Restart	0.68	0.26	0.76	0.12	0.72	0.27
Repetition	0.70	0.64	0.81	0.45	0.74	0.56
Stutter	0.60	0.46	0.63	0.64	0.61	0.75
Substitution	0.43	0.35	0.46	0.07	0.59	0.37
Insertion	0.56	0.25	0.57	0.26	0.55	0.22
Per-Diagnosis						
AD	0.69	0.28	0.70	0.31	0.79	0.32
MCI	0.66	0.59	0.68	0.54	0.67	0.58
NC	0.73	0.60	0.79	0.36	0.76	0.64

Table 5: Per-category offline F1 scores across model variants and evaluation task. *Word Ord.* = Word Ordering Task; *Trip* = Trip to Jeju Task.

A.6 Model Training and Hyperparameter Tuning

All experiments are conducted on an NVIDIA GB10 Spark with 128GB of memory. For the discriminative BERT-based models, we train several variants to ablate the contributions of domain adaptation, streaming training, tag conditioning, and the GRU accumulator (256 hidden). Offline models are evaluated under two training data conditions: Voxgnosis only, and AIHub combined with Voxgnosis while streaming models are all trained with the combined set. All variants share the same backbone and optimisation setup: AdamW with weight decay $= 0.01$, learning rate 2×10^{-5} with linear warmup over 10% of total steps followed by linear decay, and 5 training epochs, implemented with PyTorch (Paszke et al., 2019) and the `transformers` library (Wolf et al., 2020). Offline variants use a batch

size of 256 while streaming variants are reduced to batch size 16 due to the memory cost of the vectorised prefix expansion, which processes $BT \times T$ instances per forward pass. To compensate for this reduction, Voxgnosis samples are upweighted by a factor of 16 in streaming training conditions, ensuring that the effective exposure to Voxgnosis data remains comparable. For the full objective training $\mathcal{L}_{\text{train}} = \mathcal{L}_{\text{full}} + \gamma\mathcal{L}_{\text{prefix}} + \lambda\mathcal{L}_{\text{latency}}$, we follow the recommendations of Chen et al. (2022) with $\gamma = 1.5$ and $\lambda = 10^{-5}$. To counteract severe label imbalance across the eight disfluency categories, class weights are set to the inverse of per-class corpus frequency and applied to all cross-entropy terms.

For Qwen3-30B-A3B, no gradient-based training is performed. The model is served locally via a vLLM (Kwon et al., 2023) inference server. All conditions use greedy decoding at temperature $= 0$.

A.7 Voxgnosis Corpus: Participant Recruitment and Ethics

Participants were recruited from Boramae Medical Center as part of a clinical research protocol. Eligibility criteria required participants to speak standard Korean as determined by clinical interview. Exclusion criteria included current smoking status, alcohol consumption four or more times per month, clinically significant hearing impairment, and a history of major neurological or psychiatric disorders affecting speech production, including stroke, parkinsonism, or other neurodegenerative or motor speech conditions. Participants were not recruited through crowdsourcing platforms and received modest compensation in accordance with the IRB-approved study protocol, considered appropriate for the local clinical research context. Speech task stimuli were presented on a computer screen using a PsychoPy-based experimental protocol; for the word-ordering task, participants were instructed to arrange a set of words or phrases displayed on screen into a grammatically and semantically appropriate sentence aloud, with the following instruction given: *"Please look carefully at the words shown on the screen and say them aloud in an order that forms a correct sentence."* Written informed consent was obtained from all participants prior to data collection, explaining that speech recordings and relevant clinical information would be used for research purposes, with all data de-identified and analyzed in accordance with the IRB-approved protocol of Boramae Medical Center (approval number: 20-2022-63).