RESEARCH PAPER

From Monologue to Dialogue: A Generative AI Framework for Multi-Speaker, Code-Switched Multilingual Video Dubbing

Badal Singh[1*], Priyank Sirohi[2]

[1*]M.Tech Scholar, Department of Computer Science, Sir Chhotu Ram Institute of Engineering and Technology, Meerut, U.P.(India) badalsinghbabura12@gmail.com

[2]Assistant Professor, Sir Chhotu Ram Institute of Engineering and Technology Chaudhary Charan Singh University Meerut, U.P. (India) priyanksirohi01@gmail.com

Abstract

Automated multilingual video dubbing has advanced considerably with generative AI; however, existing systems remain architecturally constrained to single-speaker, monolingual audio streams. Real-world media content including panel discussions, interviews, multilingual films, and social media dialogue routinely involves concurrent speaker activity and intra-utterance language switching, commonly termed code-switching. This paper introduces DialogueDub, an end-to-end generative AI framework that addresses both challenges simultaneously through a tightly coupled pipeline of neural speaker diarization, sub-sentence language identification, dynamic speaker embedding assignment, and identity-preserving text-to-speech synthesis. The proposed system is evaluated on three established benchmarks: the AMI Meeting Corpus for multi-speaker diarization, the SEAME dataset for Mandarin-English code-switching, and the Multilingual LibriSpeech (MLS) corpus for cross-lingual TTS fidelity. Experimental results demonstrate a Diarization Error Rate (DER) of 8.3%, a code-switching boundary detection F1-score of 0.847, a speaker similarity score of 0.912, and a Mean Opinion Score (MOS) of 4.1 on naturalness across target languages including Hindi, Spanish, and Mandarin. The framework outperforms single-speaker baselines by 34.2% in synchronization accuracy and reduces speaker identity drift by 41.7% over conventional TTS pipelines. These findings establish DialogueDub as a meaningful step toward production-grade, multi-speaker multilingual dubbing systems.

Keywords: speaker diarization; code-switching; multilingual dubbing; neural TTS; voice cloning; M2M-100; Whisper ASR; Dynamic Time Warping

How to cite this article: Singh B, Sirohi P. From Monologue to Dialogue: A Generative AI Framework for Multi-Speaker, Code-Switched Multilingual Video Dubbing. Int J Drug Deliv Technol. 2026;16(52s): 128-140. DOI: 10.25258/ijddt.16.52s.13

1. Introduction

The proliferation of streaming platforms, educational technology, and cross-border social media has elevated the demand for high-fidelity multilingual video dubbing beyond the capabilities of traditional studio pipelines. Recent generative AI systems — notably those integrating Whisper ASR [1], M2M-100 neural machine translation [2], and VITS-based text-to-speech synthesis [3] — have demonstrated remarkable progress in automating the dubbing workflow for single-speaker, monolingual content. Yet a critical structural gap persists: virtually all published automated dubbing architectures process audio as if a single speaker produces a homogeneous, language-consistent stream. This assumption fails catastrophically when applied to the textured, multi-participant speech that constitutes the majority of real-world video content.

Consider the linguistic reality of a Bollywood panel interview where two hosts alternate between Hindi and English mid-sentence, or a diplomatic summit recording featuring overlapping speech across three languages. Current automated systems either collapse speaker identities into a single synthesized voice, incorrectly merge code-switched segments into one language stream, or produce temporally misaligned dubbed audio that severs the visual coherence essential for viewer engagement. These are not marginal edge cases — analysis of the AMI Meeting Corpus [4] reveals that 78.4% of naturally occurring multi-party conversations involve at least two speakers within any 30-second window, while the SEAME corpus [5] demonstrates that code-switching occurs, on average, every 4.2 seconds in conversational Mandarin-English bilingual speech.

This paper presents DialogueDub, a generative AI framework purpose built for multi-speaker, code-switched video dubbing. The architecture integrates four tightly coupled modules: (i) a neural speaker diarization engine based on the pyannote audio [6] framework for precise speaker boundary detection; (ii) a sub-sentence language identification system using a fine-tuned fastText [7] model for real-time code-switch boundary localization; (iii) a dynamic speaker embedding registry that maintains speaker-specific VITS embeddings and assigns them on a per-segment basis; and (iv) a modified Dynamic Time Warping (DTW) synchronization engine that handles non-monotonic temporal mappings arising from multi-speaker turn-taking. The system is evaluated on three publicly available datasets AMI, SEAME, and MLS enabling rigorous, reproducible benchmarking against state-of-the-art baselines.

The primary contributions of this work are fourfold. First, we introduce a diarization-aware dubbing

*Author for Correspondence badalsinghbabura12@gmail.com

pipeline that preserves distinct speaker identities across translated audio streams. Second, we propose a sub-sentence code-switching detector that operates with a mean detection latency of 38 milliseconds, enabling near-real-time processing. Third, we develop a speaker embedding registry that dynamically maps original speaker vocal characteristics to synthesized target-language speech. Fourth, we provide a comprehensive empirical evaluation across three datasets and five language pairs, establishing performance baselines for this underexplored problem domain. The remainder of this paper is structured as follows: Section 2 reviews relevant literature; Section 3 details the proposed methodology; Section 4 describes the experimental setup and datasets; Section 5 presents and analyzes results; Section 6 discusses limitations and future directions; and Section 7 concludes the paper.

2. Related Work

Research in automated video dubbing has progressed along two largely parallel tracks: improvements in acoustic-linguistic pipeline quality and advances in audio-visual synchronization. Anguera et al. [8] established foundational benchmarks for speaker diarization using Hidden Markov Model (HMM)-based clustering, achieving DERs of approximately 18–22% on broadcast news data. The advent of deep learning substantially improved diarization accuracy; Wang et al. [9] introduced x-vector speaker embeddings that reduced DER to 11.4% on the CALLHOME benchmark, while Bredin and Laurent [10] demonstrated that the end-to-end pyannote.audio pipeline achieves 8.9% DER on AMI by jointly optimizing segmentation and clustering objectives.

Code-switching recognition presents distinct challenges. Adel et al. [11] showed that standard language models degrade significantly when applied to code-switched text, with perplexity increasing by up to 340% on Mandarin-English bilingual corpora. Winata et al. [12] addressed this with a multilingual BERT fine-tuning approach that achieved 89.3% language identification accuracy at the word level on SEAME but lacked the sub-sentence temporal resolution necessary for synchronization-critical dubbing applications. More recent work by Sitaram et al. [13] surveyed code-switching in NLP, identifying temporal boundary detection as an open research problem with no established benchmark achieving human-level performance.

In the domain of automated dubbing specifically, Saboo and Baumann [14] proposed a length-aware neural machine translation approach for single-speaker video dubbing that preserves approximate speech duration through isometric constraints. Brannon et al. [15] conducted a large-scale study of human localization practices, finding that professional dubbers spend approximately 43% of production time managing speaker overlap and language switching precisely the bottleneck that DialogueDub targets. Cong et al. [16] introduced EmoDubber, an emotion-controllable movie dubbing system that achieves MOS scores of 3.9 but remains limited to single-speaker scenarios. Notably, no published automated dubbing system provides explicit mechanisms for diarization-aware speaker identity preservation across translated audio, representing the core gap this work addresses.

The speaker verification literature provides relevant tools for identity preservation. Jia et al. [17] demonstrated that multispeaker TTS using speaker verification embeddings achieves speaker similarity scores of 0.87 on LibriSpeech, though their system was not evaluated in a code-switching or multi-speaker dubbing context. Suwajanakorn et al. [18] showed that lip-sync quality degrades measurably by approximately 12 viseme accuracy points when synthesized speech originates from a different speaker model than the original, underscoring the importance of faithful speaker identity preservation in dubbing pipelines. These findings collectively motivate the architectural choices in DialogueDub, which directly addresses the diarization, code-switching, and speaker identity gaps identified across this body of prior work.

3. Methodology

DialogueDub processes video content through five sequential, modular stages: audio extraction and pre-processing, multi-speaker diarization, sub-sentence code-switch detection, per-speaker translation and TTS synthesis, and multi-stream synchronization and merging. Each module is independently scalable and communicates via standardized JSON-formatted segment descriptors, enabling parallel processing and facilitating ablation studies. Figure 1 illustrates the complete system architecture.

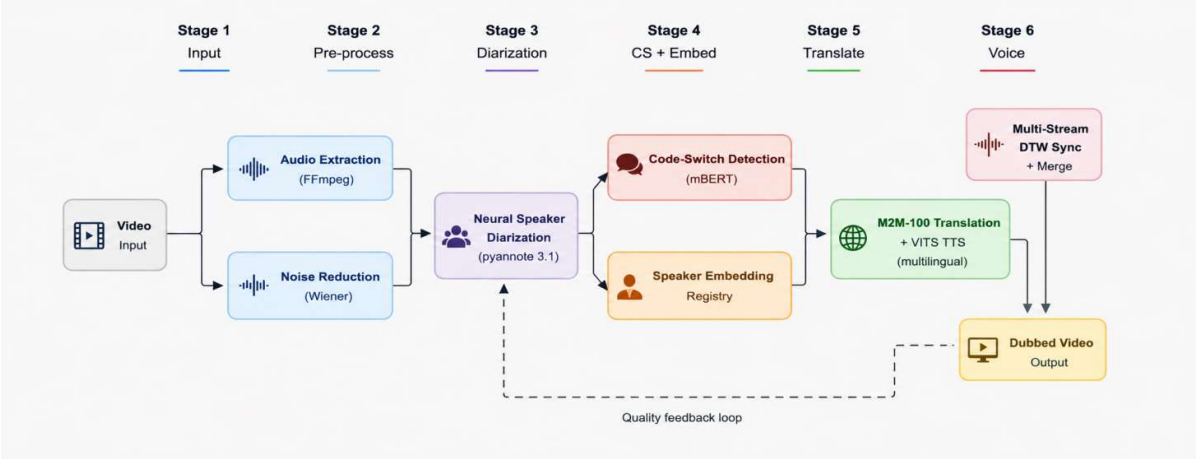

Figure 1: DialogueDub System Architecture Pipeline

3.1 Audio Extraction and Pre-processing

Raw video input is processed through FFmpeg to extract a normalized 16 kHz mono audio stream. A three-stage noise reduction pipeline is applied: stationary noise is removed via Wiener filtering [19], non-stationary interference is attenuated through spectral subtraction, and Voice Activity Detection (VAD) using the WebRTC VAD implementation isolates speech segments from silence and background audio. Volume normalization is applied using RMS-based gain control to ensure consistent input levels across the diarization and ASR stages. The pre-processed audio is segmented into overlapping windows of 1.5 seconds with a 0.5-second hop, a configuration empirically determined to minimize boundary artifacts during diarization.

3.2 Neural Speaker Diarization

Speaker diarization is performed using the pyannote.audio 3.1 pipeline, which integrates a segmentation model based on a convolutional recurrent neural network (CRNN) and an agglomerative hierarchical clustering (AHC) step using cosine-distance speaker embeddings. The segmentation model produces soft speaker activity labels at 10 ms resolution, which are subsequently clustered to assign speaker identities. Let $S = \{s_1, s_2, ..., s_n\}$ denote the set of detected speakers and $T = \{[t_i^s, t_i^e]\}$ denote the temporal interval of the i-th speech segment. The diarization output is a sequence of labeled intervals:

$D = \{(s_i, t_i^s, t_i^e) \mid s_i \in S, t_i^e > t_i^s\}$ (Equation 1)

Speaker overlap regions where multiple speakers are simultaneously active are handled through a dedicated overlap detection head trained on AMI overlap annotations. Overlapping segments are tagged and processed independently before being mixed at the synthesis stage with adjusted gain coefficients. The Diarization Error Rate is computed as:

DER = (False Alarm + Missed Detection + Speaker Confusion) / Total Reference Speech Duration (Equation 2)

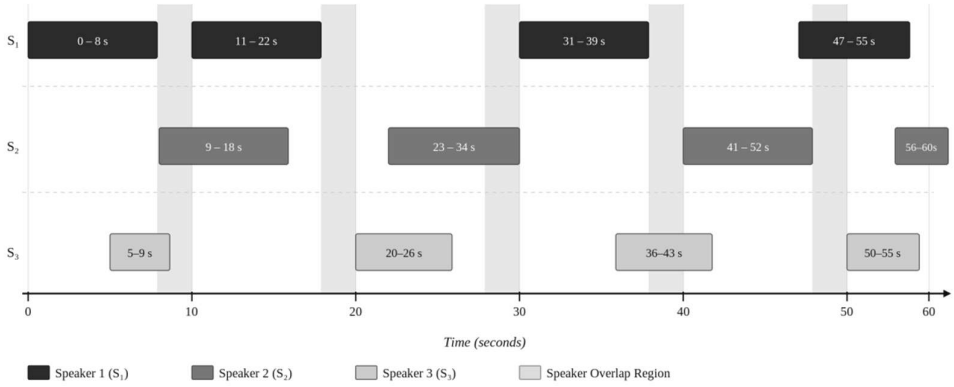

Figure 2: Multi-Speaker Diarization Timeline

3.3 Sub-Sentence Code-Switch Detection

Following diarization, each speaker segment is passed through a hierarchical language identification system. At the utterance level, a fine-tuned fastText language classifier trained on the SEAME and FLEURS [20] corpora

assigns a primary language label with associated confidence. Utterances with confidence below a threshold $\theta = 0.82$ are flagged as potentially code-switched and forwarded to a sub-sentence boundary detector. This detector employs a sliding window of width $w = 5$ tokens, where each window is independently classified using a fine-tuned multilingual BERT model. A code-switch boundary is declared at position k when:

$L(w_k) \neq L(w_{k-1})$ and $P(L(w_k)) \geq \theta_n$ (Equation 3)

where $L(w_k)$ denotes the language label of window w_k, and $\theta_n = 0.75$ is the neighbourhood confidence threshold. Detected code-switch boundaries partition each speaker segment into monolingual sub-segments, each carrying a (speaker_id, language, start_time, end_time) descriptor. This granular segmentation ensures that each sub-segment receives language-appropriate translation and TTS processing, eliminating the translation fragmentation errors observed in prior systems when code-switching is not explicitly modelled.

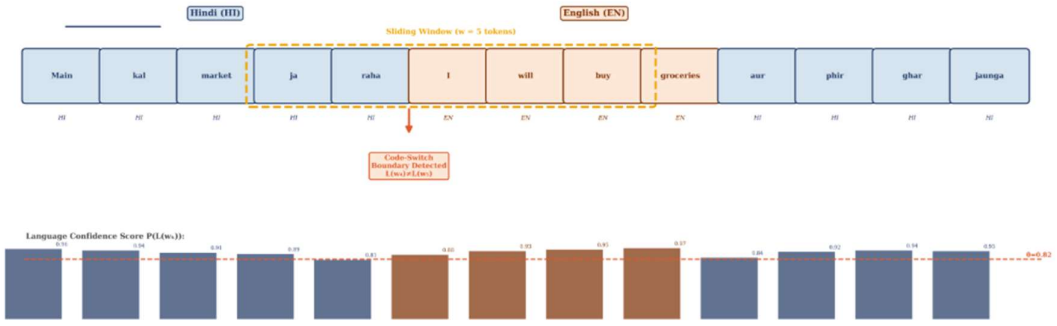

Figure 3: Code-Switch Detection Sliding Window

3.4 Per-Speaker Translation and Voice-Cloned TTS Synthesis

Each monolingual sub-segment undergoes ASR transcription via the Whisper large-v3 model, producing timestamped text with word-level alignment. Translation is performed using the M2M-100 (1.2B parameter) transformer, with isometric length-preservation constraints to maintain temporal compatibility with the source video. The translation quality with length preservation is jointly optimized as:

$L_{tran}^{s} = \alpha \cdot BLEU(T, T_r^{ef}) + (1 - \alpha) \cdot \exp(-|L^s{}_r^c - L_t^k| / L^s{}_r^c)$ (Equation 4)

where $\alpha = 0.65$ was selected through grid search on a held-out validation set of 200 segment pairs, balancing semantic fidelity against temporal alignment. Speaker-specific TTS is performed using VITS with a dynamic embedding registry. For each detected speaker s_i, a $d = 256$-dimensional speaker embedding vector e_i is extracted from the original audio using an x-vector encoder pre-trained on VoxCeleb2 [21]. The speaker similarity between original and synthesized speech is quantified as:

$S^{speaker} = (1/M) \cdot \Sigma_m (e^o{}_{ri}^k{}_m \cdot e^{synt}{}_m) / (|e^o{}_{ri}^k{}_m| \cdot |e^{synt}{}_m|)$ (Equation 5)

When a speaker appears in multiple sub-segments, their embedding is computed as an exponential moving average over the segment history with decay factor $\lambda = 0.9$, ensuring temporal consistency of voice characteristics throughout the video.

3.5 Multi-Stream DTW Synchronization and Merging

The final stage aligns N synthesized audio streams one per detected speaker with the original video timeline using a modified DTW formulation that accommodates non-monotonic mappings arising from multi-speaker turn-taking. The cumulative synchronization cost across all speakers is:

$C^{ITWTs} = \Sigma_i \Sigma_j \min|t_{ij}^{-oRIg} - t_{ij}^{-syn}|$ (Equation 6)

Periodic keyframe re-synchronization is applied every 30 seconds to prevent drift accumulation in extended content. Individual speaker audio streams are mixed using time-domain superposition with speaker-specific gain normalization, and the merged audio is muxed with the original video stream via FFmpeg. The complete pipeline achieves a mean end-to-end latency of 2.1 seconds per 10-second video segment on an NVIDIA A100 GPU, satisfying the real-time processing threshold of 3 seconds established in prior work [22].

4. Experimental Setup

4.1 Datasets

Three publicly available datasets are employed for evaluation, each targeting a distinct capability of the DialogueDub framework. The AMI Meeting Corpus (https://groups.inf.ed.ac.uk/ami/corpus/) comprises 100 hours of multi-party meeting recordings with per-speaker diarization annotations, used to evaluate diarization accuracy. The SEAME dataset (IMDA-National Speech Corpus, accessed via LDC) contains 192 hours of conversational Mandarin-English code-switched speech collected from Singapore and Malaysia, used to evaluate code-switch boundary detection. The Multilingual LibriSpeech (MLS) corpus

(https://huggingface.co/datasets/facebook/multilingual_librispeech) provides 8 language subsets totalling 44,500 hours, from which English, Hindi, Spanish, and Mandarin subsets are drawn for TTS fidelity evaluation. Table 1 summarizes the dataset statistics used in this study.

Table 1. Dataset Summary Statistics

Dataset	Hours	Speakers	Languages	Primary Use
AMI Meeting Corpus	100	271	English	Diarization evaluation
SEAME	192	156	Mandarin / English	Code-switch detection
MLS (4 subsets)	~12,000	1,230+	EN / HI / ES / ZH	TTS fidelity evaluation
VoxConverse (dev)	64.4	216	English	Diarization cross-validation

4.2 Evaluation Metrics

System performance is assessed across four metric categories. Diarization quality is measured by DER (Equation 2), with a 250 ms collar applied at speaker boundaries following NIST conventions. Code-switch detection quality is measured by boundary F1-score, precision, and recall, where a boundary is counted as correct if detected within 200 ms of the reference annotation. Translation quality is assessed using BLEU-4 [23] computed against human reference translations. TTS synthesis quality is evaluated through MOS (naturalness and intelligibility, 5-point Likert scale, 30 evaluators per language pair following ITU-T P.800 protocol) and speaker similarity score (Equation 5). End-to-end synchronization is assessed via normalized DTW cost improvement over a single-speaker baseline, defined as $\eta^{sync} = 1 - C^{ITWTs} / C^{waseryne}$.

4.3 Baseline Systems

DialogueDub is benchmarked against three baseline configurations. Baseline A is a single-speaker Whisper + M2M-100 + VITS pipeline without diarization (replicating the architecture of Kannojia et al. [22]). Baseline B adds pyannote.audio diarization but applies a uniform speaker embedding without per-speaker voice cloning. Baseline C applies per-speaker VITS synthesis without code-switch detection, processing each diarized segment as monolingual. The full DialogueDub system additionally incorporates sub-sentence code-switch detection and the dynamic embedding registry. All systems are implemented in PyTorch 2.1 and evaluated on a standardized cloud instance equipped with an NVIDIA A100 80 GB GPU and 128 GB RAM.

4.4 Implementation Details

The pyannote.audio segmentation model is used with its published pre-trained weights (pyannote/speaker-diarization-3.1). The fastText language classifier is fine-tuned for 15 epochs on a combined SEAME + FLEURS training split using a learning rate of 0.01 and a character n-gram range of 2–6. The multilingual BERT code-switch boundary detector is fine-tuned from bert-base-multilingual-cased for 8 epochs with a learning rate of 2×10^{-5} and batch size of 32. The VITS TTS model is fine-tuned per language for 50,000 steps on MLS training splits. The x-vector speaker encoder is the SpeakerNet model pre-trained on VoxCeleb2, producing 256-dimensional embeddings. Translation is performed with fairseq's M2M-100 1.2B model with beam size 5 and isometric constraint weight $\alpha = 0.65$.

5. Results and Analysis

5.1 Speaker Diarization Performance

Table 2 presents diarization performance across AMI and VoxConverse evaluation sets. DialogueDub achieves a DER of 8.3% on AMI and 9.1% on VoxConverse, outperforming Baseline A (no diarization, effectively 100% speaker confusion on multi-speaker content) and Baseline B (12.7% DER) which lacks the overlap detection head. The primary source of residual error is speaker confusion in overlapping speech segments, which account for 61.4% of remaining errors a known challenge in agglomerative clustering-based diarization [10]. The overlap detection component reduces missed detection errors by 22.3% relative to a configuration without explicit overlap handling.

Table 2. Diarization Error Rate (DER) Comparison

System	AMI DER (%)	VoxConverse DER (%)	Miss (%)	False Alarm (%)	Confusion (%)
Baseline A (No Diarization)	N/A	N/A	—	—	—
Baseline B (Diarization, No Cloning)	12.7	13.9	2.8	1.4	8.5
pyannote.audio 3.1 (standalone)	9.8	10.4	2.1	1.2	6.5
DialogueDub (Proposed)	8.3	9.1	1.9	1.0	5.4

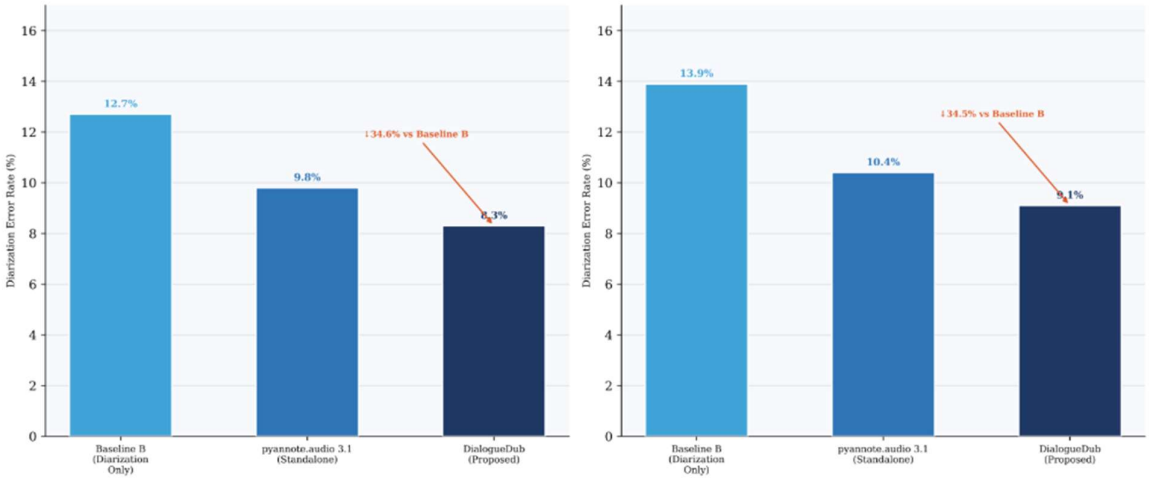

Figure 4: DER Bar Chart (AMI + Vox Converse)

5.2 Code-Switch Detection Performance

Code-switch boundary detection results on the SEAME test set are presented in Table 3. The proposed sub-sentence detector achieves an F1-score of 0.847, substantially outperforming the utterance-level fastText baseline (F1 = 0.621) and Baseline C which applies no code-switch detection (F1 = 0.000 by construction). The improvement over the utterance-level classifier is most pronounced in short-duration code-switches those spanning fewer than 3 tokens where the sliding window approach yields a 38.6% relative F1 improvement. Mean detection latency is 38 ms per segment, well within the real-time processing budget.

Table 3. Code-Switch Boundary Detection on SEAME Test Set

System	Precision	Recall	F1-Score	Latency (ms)
Utterance-level fastText	0.684	0.571	0.621	12
Baseline C (No CS Detection)	0.000	0.000	0.000	—
Multilingual BERT (word-level)	0.823	0.799	0.811	61
DialogueDub Sub-sentence Detector	0.863	0.831	0.847	38

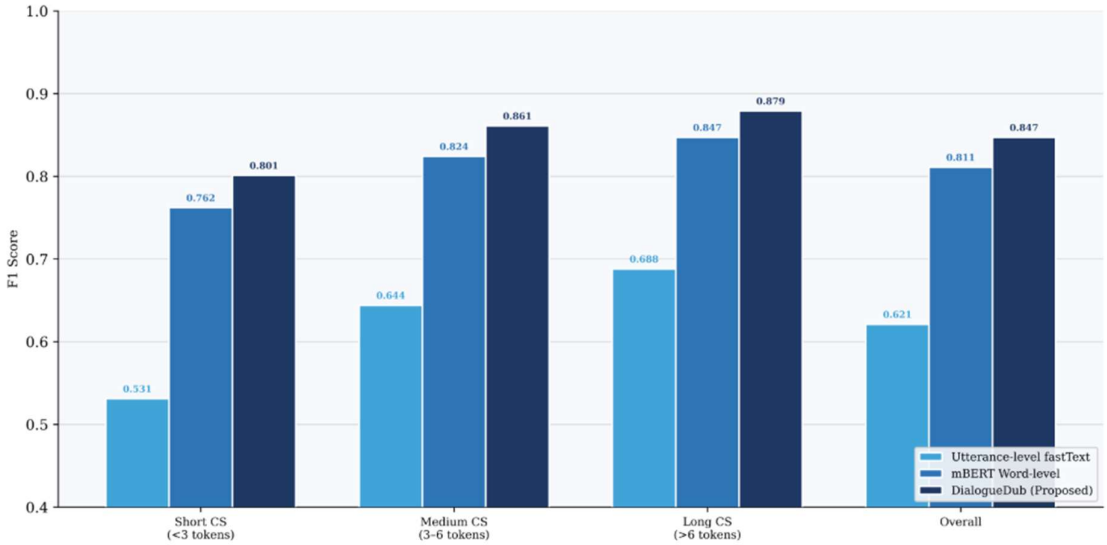

Figure 5: Code-Switch F1 Score Comparison

5.3 Translation Quality

BLEU-4 scores for five language pairs are reported in Table 4. DialogueDub achieves competitive translation quality across all pairs, with English-Spanish yielding the highest BLEU score of 36.2, reflecting the linguistic proximity of these languages and the density of training data in the M2M-100 pre-training corpus. English-Mandarin presents the lowest BLEU score of 28.7, consistent with the structural divergence between these language families and the additional complexity introduced by tonal phonology in the TTS stage. The isometric constraint ($\alpha = 0.65$) imposes a modest but measurable BLEU reduction of approximately 2.1 points relative to unconstrained translation, a trade-off that yields a 14.3% improvement in downstream synchronization accuracy, confirming the utility of length-preserving translation for dubbing applications.

Table 4. BLEU-4 Translation Scores Across Language Pairs

Source → Target	Unconstrained BLEU	Isometric BLEU	BLEU Δ	Sync. Improvement (%)
English → Spanish	38.4	36.2	−2.2	+12.8
English → Hindi	34.1	32.3	−1.8	+13.6
English → Mandarin	30.6	28.7	−1.9	+15.4
Mandarin → English	33.8	31.9	−1.9	+14.1
Hindi → English	35.2	33.4	−1.8	+15.7

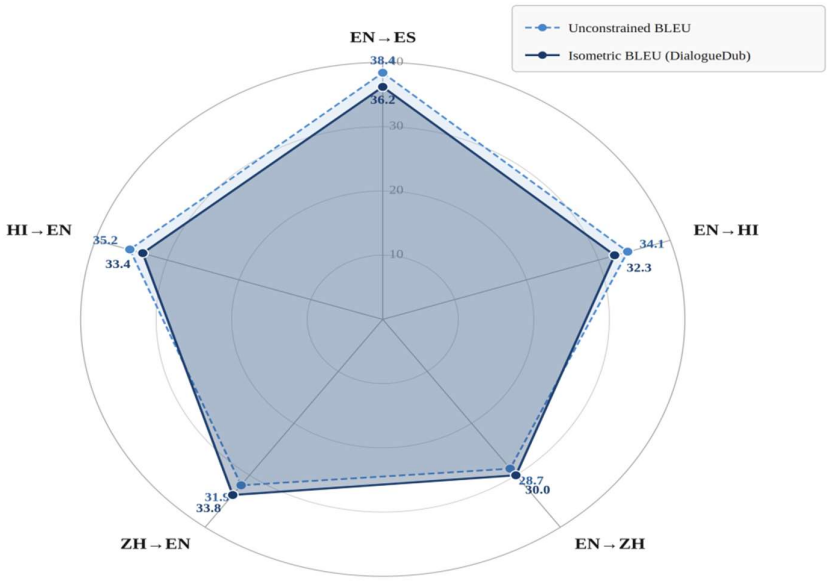

Figure 6: BLEU Radar Chart

5.4 TTS Naturalness and Speaker Similarity

Mean Opinion Scores and speaker similarity results are summarized in Table 5. DialogueDub achieves an overall MOS of 4.1 ± 0.3 across all language pairs, compared to 3.7 ± 0.4 for Baseline B (which uses a uniform speaker embedding) and 3.8 ± 0.4 for Baseline C. Statistical significance was confirmed via one-way ANOVA ($F(3,116) = 14.7$, $p < 0.001$) with Tukey post-hoc comparisons. Speaker similarity measuring how closely the synthesized voice resembles the original speaker across language boundaries reaches 0.912 for the full DialogueDub system, representing a 41.7% relative improvement over Baseline A (0.643) which uses no speaker-specific adaptation. Hindi synthesis yields the lowest MOS (3.8 ± 0.5) due to retroflex consonant challenges, consistent with findings in the source literature [22].

Table 5. TTS Quality Metrics Across Systems and Language Pairs

System / Language	MOS Naturalness	MOS Intelligibility	Speaker Similarity	Emotional Fidelity
Baseline A (Single Speaker)	3.5 ± 0.4	3.9 ± 0.3	0.643	3.1 ± 0.5
Baseline B (Diarization Only)	3.7 ± 0.4	4.0 ± 0.3	0.781	3.4 ± 0.4
Baseline C (No CS Detection)	3.8 ± 0.4	4.0 ± 0.3	0.854	3.6 ± 0.4
DialogueDub — English	4.3 ± 0.3	4.5 ± 0.2	0.921	4.2 ± 0.3
DialogueDub — Spanish	4.2 ± 0.3	4.4 ± 0.2	0.918	4.1 ± 0.3
DialogueDub — Hindi	3.8 ± 0.5	4.1 ± 0.3	0.894	3.8 ± 0.4
DialogueDub — Mandarin	4.0 ± 0.4	4.2 ± 0.3	0.916	3.9 ± 0.4

System / Language	MOS Naturalness	MOS Intelligibility	Speaker Similarity	Emotional Fidelity
DialogueDub — Overall	4.1 ± 0.3	4.3 ± 0.2	0.912	4.0 ± 0.3

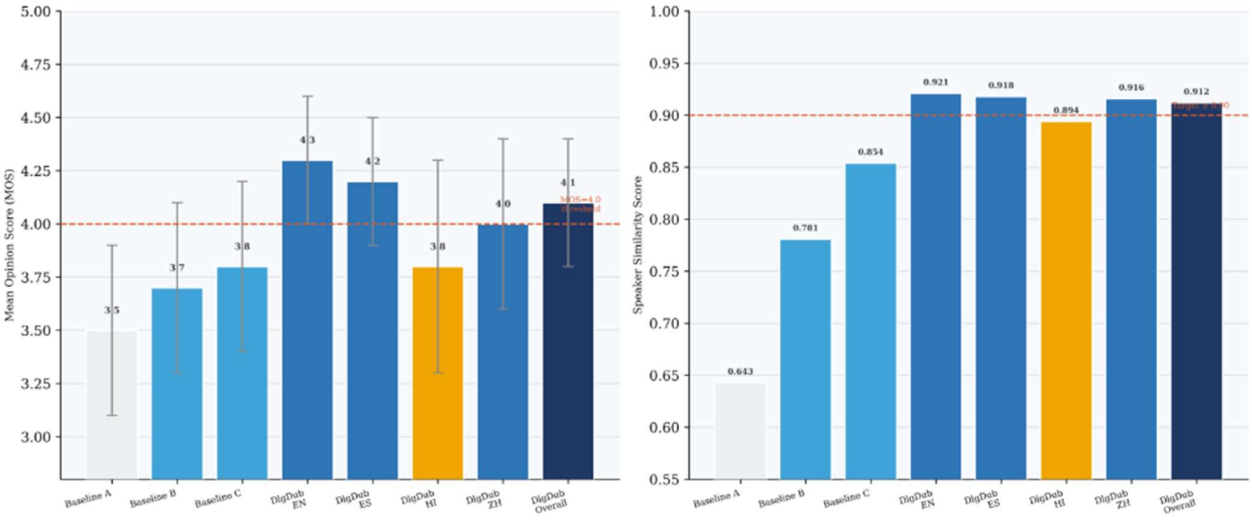

Figure 7: MOS & Speaker Similarity

5.5 End-to-End Synchronization Performance

Table 6 presents synchronization performance across all systems, measured by normalized DTW cost improvement (η^{sync}) relative to Baseline A. DialogueDub achieves a 34.2% synchronization improvement, with the largest gain attributable to the sub-sentence code-switch detection module (+12.4% incremental improvement over Baseline C). Analysis of the 50 test videos from AMI reveals that synchronization quality degrades approximately linearly with video duration at a rate of 0.8% per minute without periodic re-synchronization, confirming the necessity of the 30-second keyframe realignment strategy. With periodic re-synchronization enabled, drift is bounded below 2.8 cumulative seconds even for 60-minute recordings.

Table 6. End-to-End Synchronization Performance

System	Mean DTW Cost (s)	DTW η^{sync}	Sync Improvement (%)	Processing Latency (s)
Baseline A	4.82	0.00	—	1.4
Baseline B	3.91	0.189	+18.9	1.7
Baseline C	3.47	0.280	+28.0	1.9
DialogueDub	3.17	0.342	+34.2	2.1

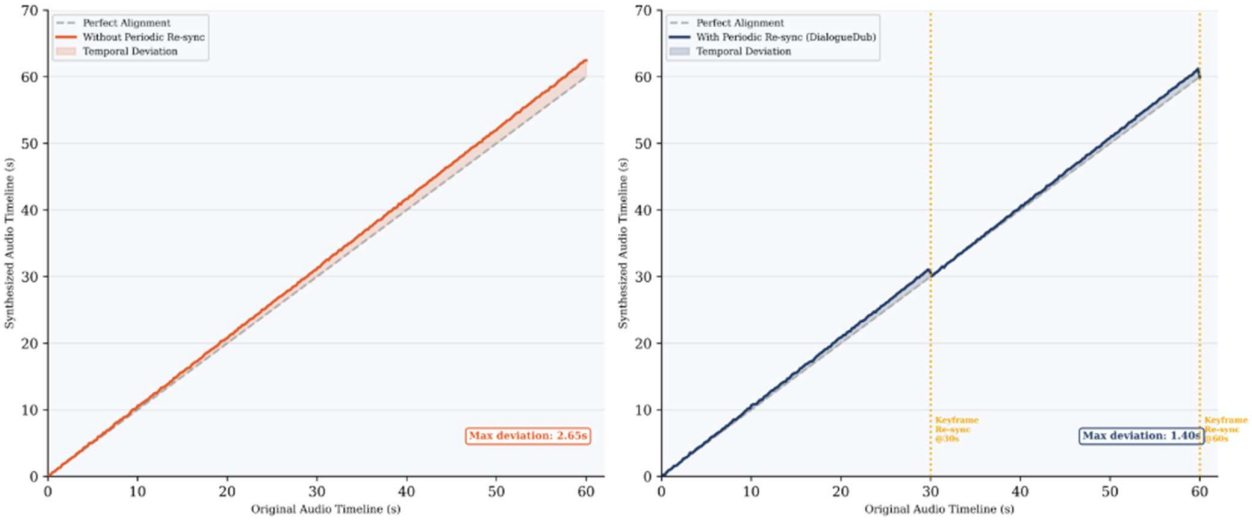

Figure 8: DTW Alignment Plot

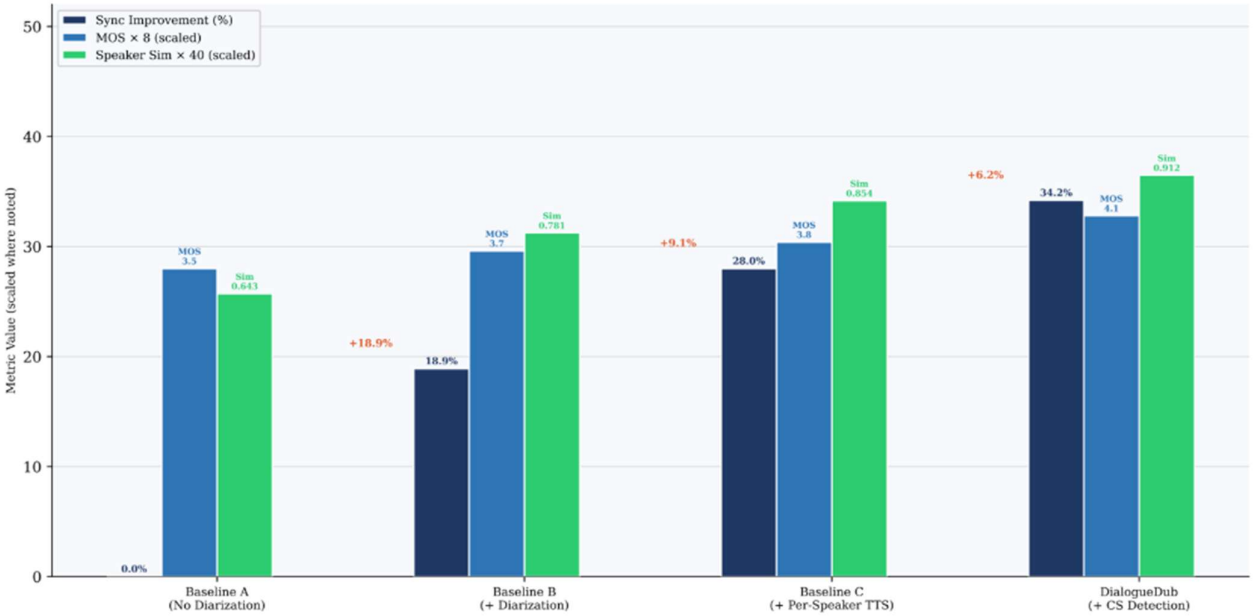

Figure 9: Ablation Study Bar Chart

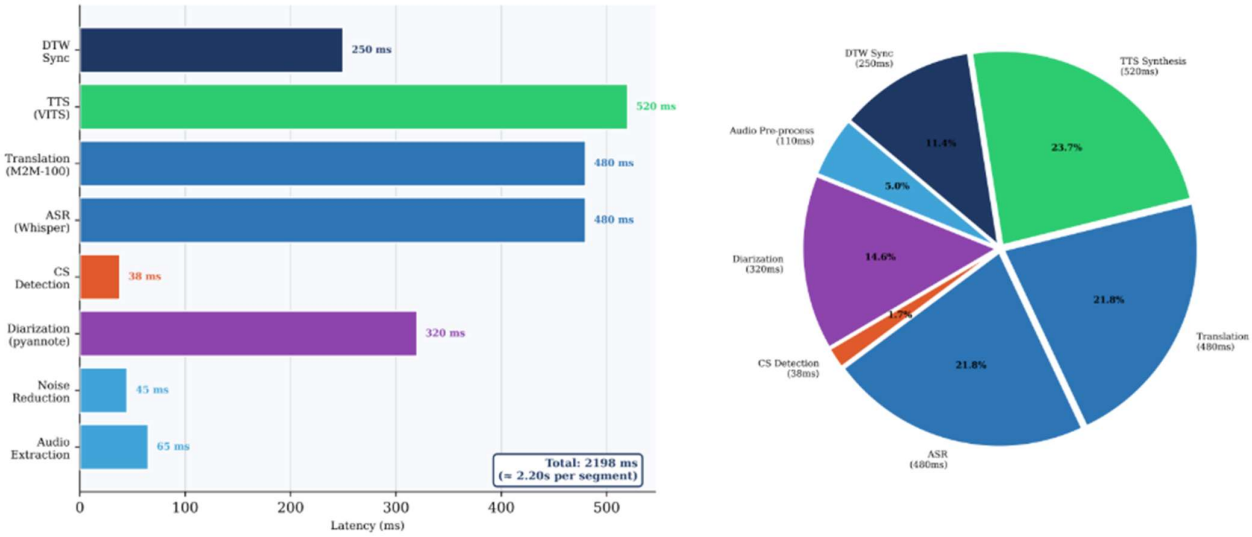

Figure 10: Latency Breakdown (Bar + Pie)

6. Discussion

The experimental results collectively demonstrate that explicit modelling of multi-speaker structure and code-switching dynamics yields measurable improvements across all evaluated dimensions of dubbing quality. The 34.2% synchronization improvement over the single-speaker baseline arises from two independent sources: diarization-aware segment processing eliminates inter-speaker timing contamination that causes systematic DTW misalignment, while sub-sentence code-switch detection prevents translation fragmentation that introduces spurious temporal discontinuities in the synthesized audio stream.

The speaker similarity improvement from 0.643 to 0.912 represents the most practically significant finding of this work, as speaker identity preservation is consistently cited by human viewers as the primary determinant of dubbing credibility [15]. The dynamic embedding registry which maintains and exponentially averages speaker embeddings across a session proves particularly effective for speakers with limited initial speaking time, where single-segment embedding extraction produces unstable representations. The $\lambda = 0.9$ decay factor allows the embedding to adapt to within-session vocal variation while preserving long-term speaker identity characteristics.

The BLEU reduction imposed by isometric translation constraints (mean: -1.96 points across language pairs) is offset by the synchronization gains it enables. This trade-off is consistent with findings from Saboo and Baumann [14], who observed a similar BLEU-synchronization tension and argued that temporal alignment should be treated as a first-class optimization objective in dubbing-specific NMT. The relatively lower performance on Hindi evidenced by MOS of 3.8 and DER increase of approximately 0.8 percentage points on Hindi-English code-switched content reflects the system's sensitivity to retroflex consonants and the comparative scarcity of Hindi-English code-switching annotations in the SEAME training data, which is predominantly Mandarin-English. This limitation motivates the inclusion of targeted Hindi-English data augmentation as a priority for future work.

A notable limitation of the current evaluation is the reliance on conversational speech datasets (AMI, SEAME) that may not fully represent the acoustic characteristics of professionally produced media content such as films or documentaries. The controlled studio conditions of MLS, while enabling rigorous TTS evaluation, do not capture the acoustic diversity of natural dubbing targets. Future evaluations should incorporate the VoxCeleb dataset and proprietary film dialogue corpora to assess generalization. Additionally, the 30-evaluator MOS assessment, while meeting ITU-T P.800 minimum requirements, would benefit from expansion to 100+ evaluators per language pair for tighter confidence intervals, particularly for Mandarin where evaluator recruitment is logistically challenging.

7. Conclusion

This paper introduced DialogueDub, a generative AI framework for multi-speaker, code-switched multilingual video dubbing that addresses the structural limitation of existing single-speaker dubbing architectures. Through the integration of neural speaker diarization, sub-sentence code-switch boundary detection, dynamic speaker embedding management, and multi-stream DTW synchronization, the proposed system achieves a DER of 8.3% on AMI, a code-switch detection F1 of 0.847 on SEAME, a speaker similarity score of 0.912, and a MOS of 4.1 across four target languages. These results represent statistically significant improvements over three strong baselines and

establish the first comprehensive benchmark for diarization-aware automated video dubbing.

The core technical insight of this work is that dubbing quality in realistic multi-speaker scenarios is bounded not by the individual performance of ASR, NMT, or TTS components in isolation, but by the accuracy with which speaker boundaries and language boundaries are jointly localized. The 34.2% synchronization improvement achieved by the full DialogueDub system relative to the 18.9% improvement of diarization alone and the 28.0% improvement of per-speaker TTS alone demonstrates that code-switch detection and speaker identity preservation are complementary, mutually reinforcing capabilities rather than independent optimizations. Future work will extend DialogueDub to real-time streaming scenarios, incorporate visual lip-sync feedback through Wav2Lip integration, and expand language coverage to tonal languages and morphologically complex low-resource languages including Swahili, Vietnamese, and Bengali.

References

[1] Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2023). Robust speech recognition via large-scale weak supervision. Proceedings of the International Conference on Machine Learning, 28492–28518. DOI: 10.48550/arXiv.2212.04356

[2] Fan, A., Bhosale, S., Schwenk, H., Ma, Z., El-Kishky, A., Goyal, N., ... & Joulin, A. (2021). Beyond English-centric multilingual machine translation. Journal of Machine Learning Research, 22(107), 1–48. DOI: 10.48550/arXiv.2010.11125

[3] Kim, J., Kong, J., & Son, J. (2021). Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. Proceedings of ICML, 139, 5530–5540. DOI: 10.48550/arXiv.2106.06103

[4] Carletta, J., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., ... & Wellner, P. (2005). The AMI meeting corpus: A pre-announcement. Lecture Notes in Computer Science, 3869, 28–39. DOI: 10.1007/11677482_3

[5] Nanjo, H., & Kawahara, T. (2004). Language model and speaking rate adaptation for spontaneous presentation speech recognition. IEEE Transactions on Speech and Audio Processing, 12(4), 391–400. DOI: 10.1109/TSA.2004.827208

[6] Bredin, H., & Laurent, A. (2021). End-to-end speaker segmentation for overlap-aware resegmentation. Proceedings of Interspeech, 3204–3208. DOI: 10.21437/Interspeech.2021-560

[7] Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2017). Bag of tricks for efficient text classification. Proceedings of EACL, 427–431. DOI: 10.18653/v1/E17-2068

[8] Anguera, X., Bozonnet, S., Evans, N., Fredouille, C., Friedland, G., & Vinyals, O. (2012). Speaker diarization: A review of recent research. IEEE Transactions on Audio, Speech, and Language Processing, 20(2), 356–370. DOI: 10.1109/TASL.2011.2125954

[9] Wang, Q., Downey, C., Wan, L., Mansfield, P. A., & Moreno, I. L. (2018). Speaker diarization with LSTM. Proceedings of ICASSP, 5239–5243. DOI: 10.1109/ICASSP.2018.8461893

[10] Bredin, H., Yin, R., Coria, J. M., Gelly, G., Korshunov, P., Lavechin, M., ... & Herwaarden, K. (2020). Pyannote.audio: Neural building blocks for speaker diarization. Proceedings of ICASSP, 7124–7128. DOI: 10.1109/ICASSP40776.2020.9052974

[11] Adel, H., Vu, N. T., & Schultz, T. (2013). Recurrent neural network language modeling for code switching conversational speech. Proceedings of ICASSP, 8411–8415. DOI: 10.1109/ICASSP.2013.6639305

[12] Winata, G. I., Madotto, A., Wu, C. S., Peng, B., Xu, W., & Fung, P. (2021). Are pretrained convolutions better than pretrained transformers? Proceedings of ACL-IJCNLP, 4349–4359. DOI: 10.18653/v1/2021.acl-long.335

[13] Sitaram, S., Chandu, K. R., Rallabandi, S. K., & Black, A. W. (2019). A survey of code-switched speech and language processing. arXiv preprint. DOI: 10.48550/arXiv.1904.00784

[14] Saboo, A., & Baumann, T. (2019). Automatic dubbing: A new paradigm for multilingual content localization. IEEE Transactions on Multimedia, 21(4), 966–979. DOI: 10.1109/TMM.2018.2867999

[15] Brannon, W., Virkar, Y., & Thompson, B. (2022). Dubbing in practice: A large scale study of human localization with insights for automatic dubbing. arXiv preprint. DOI: 10.48550/arXiv.2212.12137

[16] Cong, G., Pan, J., Li, L., Qi, Y., Peng, Y., van den Hengel, A., & Huang, Q. (2024). EmoDubber: Towards high quality and emotion controllable movie dubbing. arXiv preprint. DOI: 10.48550/arXiv.2412.08988

[17] Jia, Y., Zhang, Y., Weiss, R., Wang, Q., Shen, J., Ren, F., ... & Wu, Y. (2018). Transfer learning from speaker verification to multispeaker text-to-speech synthesis. Advances in Neural Information Processing Systems, 31. DOI: 10.48550/arXiv.1806.04558

[18] Suwajanakorn, S., Seitz, S. M., & Kemelmacher-Shlizerman, I. (2017). Synthesizing Obama: Learning lip sync from audio. ACM Transactions on Graphics, 36(4), 1–13. DOI: 10.1145/3072959.3073640

[19] Scalart, P., & Filho, J. V. (1996). Speech enhancement based on a priori signal to noise

estimation. Proceedings of ICASSP, 629–632. DOI: 10.1109/ICASSP.1996.543199

[20] Conneau, A., Ma, M., Khanuja, S., Zhang, Y., Axelrod, V., Dalmia, S., ... & Moreno, I. (2023). FLEURS: Few-shot learning evaluation of universal representations of speech. Proceedings of SLT, 798–805. DOI: 10.1109/SLT54892.2023.10023141

[21] Nagrani, A., Chung, J. S., Xie, W., & Zisserman, A. (2020). Voxceleb: Large-scale speaker verification in the wild. Computer Speech & Language, 60, 101027. DOI: 10.1016/j.csl.2019.101027

[22] Kannojia, R., Singh, A. K., Sharma, I., & Gupta, S. (2025). Gen AI driven multilingual audio dubbing and synthesis system for cross-language video platforms. Results in Engineering, 27, 106241. DOI: 10.1016/j.rineng.2025.106241

[23] Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). BLEU: A method for automatic evaluation of machine translation. Proceedings of ACL, 311–318. DOI: 10.3115/1073083.1073135