EAI Endorsed Transactions
on Intelligent Systems and Machine Learning Applications Research Article EAI.EU

Acoustic and Tonal Modeling of the *tpuri* Language through a Multi-Modular Hybrid Approach

Bayang Souloukna Jules Paulin[1,3], Nounamo Dabou Patrick[1,3], Dayang Paul[2,3], Kolyang[1,3]

[1]University of Maroua, Cameroon
[2]University of Ngaoundéré, Cameroon
[3]Laboratoire de recherche en Informatique (LARI)

Abstract

Automatic speech recognition (ASR) for tonal low-resource languages remains challenging due to the scarcity of labelled data and the need to model complex prosodic systems. This paper presents a hybrid multi-modular ASR architecture for *tpuri*, a Mboum-Day Niger-Congo language spoken in Cameroon and Chad that exhibits contrastive lexical tone, vowel length and nasalisation. The system combines a self-supervised Wav2Vec 2.0 acoustic encoder with a tonal processing module based on YIN pitch estimation and STFT-derived spectral features, and an adaptive fusion mechanism that integrates acoustic and tonal representations before decoding. We pretrain the acoustic encoder on 45 hours of read and spontaneous speech and fine-tune it on 19h35 of scripted speech. On the scripted test set, our best configuration reaches a word error rate (WER) of 10.4%, a phone error rate (PER) of 8.7% and a tone error rate (TER) of 6.1%. Ablation experiments show that removing the tonal module (+1.5 WER, +2.3 TER) or self-supervised pretraining (+3.4 WER) substantially degrades performance, while adaptive fusion and tone-aware data augmentation yield smaller but consistent gains. A fine-grained error analysis across tonal, grammatical, syllabic and morphological dimensions indicates that the architecture is particularly effective at modelling lexical tone and clause-level syntax, but still struggles with complex syllable structures and rich morphology. Overall, the results demonstrate that competitive ASR is attainable for under-resourced tonal languages such as *tpuri* by tightly coupling self-supervised acoustic modelling with explicit tonal representations, and provide a reusable blueprint for extending ASR to other Niger-Congo languages.

Received on 08 December 2025; accepted on 10 January 2026; published on 14 January

Keywords: tpuri, tonal language, Wav2Vec 2.0, YIN, STFT, adaptive fusion, speech recognition, low-resource

doi:10.4108/eetismla.11285

*Corresponding author. Email: paulinbayang@gmail.com

1. Introduction

Spoken language technologies such as automatic speech recognition (ASR), speech synthesis and spoken dialogue systems rely critically on the availability of high-quality linguistic resources, including text corpora, pronunciation lexicons and multimodal datasets[1, 2]. While languages such as English or Mandarin benefit from massive annotated corpora and mature toolchains, many of the world's languages remain largely absent from the digital sphere, particularly in Africa and parts of Asia[3, 4]. This imbalance severely limits access to speech technologies and contributes to a persistent "digital language divide".

Crucially, the degree to which a language is under-resourced does not depend solely on its number of speakers, but rather on the quantity, quality and internal consistency of its digital resources and tools [5, 6]. Large communities may still be underserved if there are few reliable corpora, no standard orthography or only rudimentary NLP and ASR tools. Conversely, some languages with relatively small speaker populations can be well supported when sustained investment has been made in corpus building and tool development. From this perspective, many African languages including those with several hundred thousand speakers remain clearly under-resourced.

This work focuses on *tpuri*, a Mboum-Day Niger-Congo language spoken in Cameroon and Chad.

Existing estimates place the number of speakers between roughly 500 000 and 750 000 [7], although recent fine-grained census data are lacking [8]. Tpuri is typologically challenging for ASR: it exhibits contrastive lexical tone, a rich vowel inventory with phonemic length and nasalisation, and complex morphophonological alternations. At the same time, it lacks large digital corpora and dedicated speech technologies, making it a prototypical example of an under-resourced, structurally complex African language. Bridging this gap calls for architectures that can exploit modern self-supervised learning while incorporating explicit linguistic knowledge about tone and prosody.

In this paper, we address these challenges by developing and empirically evaluating a multi-modular ASR system tailored to tpuri. Our approach combines a self-supervised acoustic encoder with an explicit tonal front-end and an adaptive fusion mechanism that integrates these complementary representations before decoding. Beyond obtaining a working system, our goals are to quantify how much explicit tonal modelling helps in a low-resource setting and to characterise which error patterns remain for this tonal, low-resource language.

Our main contributions are as follows:

- We document the acoustic and tonal characteristics of tpuri from an ASR perspective, highlighting those phonological properties lexical tone, vowel length and nasalisation, that are most critical for speech recognition.
- We propose a *hybrid multi-modular architecture* that combines a Wav2Vec 2.0-based acoustic encoder with a tonal processing module based on YIN pitch estimation and STFT-derived spectral features, together with an adaptive fusion mechanism that dynamically integrates acoustic and tonal information.
- We conduct an extensive empirical study on tpuri speech, including ablation experiments that quantify the relative contributions of explicit tonal modelling, adaptive fusion and self-supervised pretraining.
- We provide a fine-grained error analysis across tonal, grammatical, syllabic and morphological dimensions, showing that the proposed design substantially improves the modelling of lexical tone and clause-level syntax, while revealing remaining weaknesses in the treatment of syllable structure and complex morphology.

Taken together, these results demonstrate that competitive ASR systems can be built for a tonal, under-resourced language such as tpuri by tightly coupling self-supervised acoustic modelling with explicit tonal representations. More broadly, the methodology proposed here offers a reusable blueprint for extending ASR to other low-resource tonal languages beyond tpuri.

2. Related Work

Automatic speech recognition (ASR) for under-resourced languages remains challenging due to limited transcribed speech, sparse text resources for language modelling, and scarce linguistic tools. For tonal languages, these constraints are amplified because lexical meaning often depends on suprasegmental cues such as fundamental frequency (F_0) and voicing, which can be degraded by noise, channel effects, or speaker variability. This section positions our work at the intersection of low-resource ASR, tone-aware modelling, and self-supervised speech representation learning.

2.1. ASR for Low-Resource Languages

Traditional low-resource ASR strategies include data augmentation, pronunciation lexicon bootstrapping, and cross-lingual or multilingual transfer. Data augmentation has been shown to be particularly effective when labelled data is scarce, improving robustness and reducing overfitting [9]. More recently, transfer learning with pretrained acoustic encoders has become a dominant paradigm, as it enables strong performance with limited labelled speech by leveraging large-scale unlabelled pretraining. In African and other under-resourced settings, public corpora are still limited and language diversity is high; therefore, approaches that are effective under small-data regimes and that can incorporate language-specific cues are especially valuable. In this context, our work targets Tpuri, a low-resource tonal language, and explicitly models tone as a complementary information source rather than relying on acoustics alone.

2.2. Tonal and Pitch-Aware ASR

Tone-aware ASR has a long history, with two recurring design patterns. The first is *embedded* modelling, where pitch-related cues (e.g., smoothed F_0, ΔF_0, or voicing measures) are appended to short-term acoustic features and learned jointly by the acoustic model. The second is *explicit* modelling, where tone (or tone class) is predicted over longer spans (often syllable-level) and combined with phonetic decoding through rescoring, multi-stream fusion, or factorised representations.

Mandarin has served as a primary testbed for tone modelling. Early work explored robust pitch tracking and tone feature extraction for Mandarin ASR [10], and subsequent studies examined integration strategies

that combine pitch cues with spectral features [11]. For Mandarin broadcast news, Lei et al. showed that embedded F_0 features and an explicit tone classifier can provide complementary gains when combined via lattice rescoring [12]. Beyond Mandarin, similar themes appear in other tonal languages. For Vietnamese LVCSR, Vu and Schultz compared tone-handling strategies supported by pitch extraction in a large-vocabulary setting [13]. For extremely low-resource tonal settings, Coto-Solano demonstrated that factorising tone in transcription (separating tone from segmental units) can yield consistent improvements in both HMM/GMM and CTC systems, even with under two hours of data [14]. Recent corpus-building work for Yorùbá also highlights that diacritics encode lexical tone and reports wav2vec 2.0 baselines for ASR [15].

These results collectively suggest that tonal evidence is complementary to segmental acoustics, and that the optimal integration depends on both the linguistic structure (tone-bearing units) and signal reliability (availability of stable F_0). Motivated by this literature, we adopt a multi-modular design: a strong SSL acoustic encoder coupled with a dedicated tonal module, combined through an adaptive fusion gate. This architecture follows the multi-stream intuition (acoustic vs. tonal evidence) while remaining compatible with end-to-end decoding; we further analyse the learned gate to quantify when tonal evidence is emphasised.

2.3. Self-Supervised Speech Representation Learning

Self-supervised representation learning has become a key enabler for low-resource ASR by providing pre-trained acoustic encoders that transfer effectively with limited labelled data. Wav2Vec 2.0 is a prominent example, demonstrating strong gains across languages and data regimes [15]. However, tone remains comparatively underexplored in the SSL literature beyond Mandarin-centric benchmarks. Recent analyses suggest that the temporal span and salience of tone cues captured by SSL models can vary across tonal systems and transfer conditions [16]. Our approach leverages SSL pretraining for robust acoustic modelling, while explicitly injecting tonal information through a dedicated module and an adaptive fusion mechanism, targeting reliable recognition in a low-resource tonal language.

3. Tpuri Language and Its Resources

3.1. Classification and Demographics

Tpuri (also written *Tupuri* in some sources) is primarily spoken in parts of Chad and Cameroon, where it functions as a local lingua franca in several urban centres and surrounding rural areas. In Chad, speakers are concentrated mainly in the Mayo-Kebbi Est region; in Cameroon, they are found in the Far North Region along the border with Chad. Existing estimates place the number of speakers between roughly 500 000 and 750 000[8, 17], although the absence of recent, fine-grained census data in both countries prevents a more precise demographic assessment.

From a genealogical perspective, Tpuri is commonly classified (following Boyd, as reported in descriptive accounts) among the Mbum languages (Group 6) of the Adamawa branch within the Niger-Congo family[8]. In both national contexts, Tpuri coexists with other local languages as well as regional and official languages. It is used in everyday communication within families and neighbourhoods, in markets and local administration, and in a range of ritual and ceremonial contexts. Oral traditions, narrative practices and songs play an important role in maintaining a shared cultural identity [7]. Written usage remains comparatively limited and orthographic conventions are not yet fully standardised, a factor that has direct implications for the construction of text corpora and pronunciation lexicons for ASR.

3.2. Phonological Inventory

Vowels. Descriptive work on Tpuri reports a relatively rich vowel system. Following Ruelland, the language distinguishes 24 vowel phonemes organised along three dimensions: vowel quality, orality versus nasality, and vowel length[18]. In practical orthography, long vowels are commonly indicated by doubling the vowel letter (aa, ee, ii, oo, , uu), while nasalisation is marked with a tilde (ã, , ĩ, õ, ũ) [18].

Length is phonemic and contributes to lexical contrasts; however, descriptive accounts note that the length contrast is most clearly distinctive in open syllables, whereas in closed syllables it may be neutralised and only slight lengthening can arise as a phonetic correlate of contour tones[18]. From an ASR perspective, this implies that the system must model durational variability jointly with spectral and tonal cues.

Consonants. The consonantal system of Tpuri further contributes to its segmental complexity. Descriptive analyses report around 25 consonants, including oral consonants as well as nasal and prenasalised segments[8, 18]. Prenasalised stops are particularly frequent and play an important role in syllable structure and morphophonology. From the standpoint of acoustic modelling, these segments give rise to characteristic transitions and coarticulatory patterns that must be captured reliably by the encoder.

In this work, we adopt the phoneme inventory observed in our corpus and lexicon (Table 1).

Table 1. Tpuri consonant inventory used in this work (IPA in TIPA notation).

Category	Phonemes
Stops	/p b t d k g ɓ/
Fricatives	/f s h/
Affricates	/tʃ dʒ/
Nasals	/m n ŋ/
Prenasalized	/mb nd ŋg/
Liquids	/l r/
Glides	/w j/

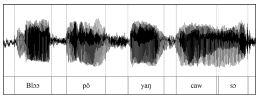
Tone. Tpuri is a tonal language: pitch patterns realised on syllables contribute to both lexical contrasts and grammatical distinctions. The system distinguishes six tones, four level (High, Mid-high, Mid-low, Low) and two contour tones (Rising and Falling). Tone interacts with morphology and syntax through phenomena such as tone spreading and tone sandhi, whereby the surface realisation of a tone depends on neighbouring tones and on prosodic phrasing.

For ASR, this rich tonal system is particularly challenging. First, tone is realised primarily through fundamental frequency (F_0) movements that are superimposed on segmental cues. Second, the six-way tonal contrast leads to a high density of minimal or near-minimal pairs, such that small errors in pitch tracking may result in changes of lexical meaning or grammatical function. Finally, sandhi and other contextual effects mean that the mapping from underlying tonal categories to surface F_0 contours is often many-to-many. These properties motivate the inclusion of a dedicated tonal module in our architecture (Sections 4 and 6) that can exploit fine-grained information about F_0 dynamics and their interaction with segmental context.

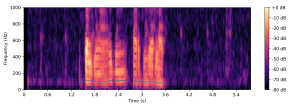
3.3. Audio Corpus

To support the development and evaluation of ASR systems for Tpuri, we collected a multi-genre speech corpus comprising scripted, read and spontaneous speech. Scripted speech consists primarily of carefully prepared sentences, elicited to cover a broad range of phonological and morphosyntactic patterns. Read speech involves controlled reading of texts such as narratives or dialogues, while spontaneous speech includes conversational and narrative material recorded in more naturalistic settings. This design aims to balance phonological coverage with the need to capture natural usage patterns, disfluencies and prosodic variation.

Table 2 summarises the composition of the corpus. All recordings were digitised at 16 kHz with 16-bit linear PCM encoding, which is standard for modern ASR and facilitates integration with existing toolchains.



(a) Waveform with orthographic segmentation.



(b) Spectrogram of the same utterance (optionally with F_0 overlay).

Figure 1. Tpuri utterance illustrating sandhi-induced phoneme fusion without pause. (a) Waveform and orthographic segmentation; (b) spectrogram showing continuous energy across word boundaries.

Table 2. Tpuri audio corpus summary.

Speech Type	Duration	Speakers
Scripted Speech	19 h 35 m 24 s	35 (18 F / 17 M)
Read Speech	7 h 32 m 17 s	12 (6 F / 6 M)
Spontaneous	36 h 34 m 28 s	28 (15 F / 13 M)

Dataset Availability and Reproducibility. To facilitate reproducibility, we will release the pronunciation lexicon (with tone-marked entries), the metadata for train/dev/test splits, and the complete training/decoding recipes (configuration files and scripts)[19]. Public redistribution of raw audio depends on speaker consent and local ethics requirements. When full public release is not possible, we will provide access to the audio under a research-only data sharing agreement, while still making all non-audio resources openly available via `https://github.com/bayang89/asr_tpuri`.

4. Preliminary Acoustic Analysis

4.1. Spectrogram and Formants

Figure 1 illustrates a Tpuri utterance in which sandhi processes lead to phoneme fusion without an audible pause between words. The signal is segmented into orthographic units, shown below the waveform. From left to right, the segments βlɔ: pö jaŋ caw sɔ form a prosodic unit with continuous energy, as speakers often avoid inserting clear silence at word boundaries.

In the spectrogram (Figure 1b), frequency is plotted on the vertical axis and time on the horizontal axis, with darker regions indicating higher energy. Formant trajectories and consonant bursts reflect the segmental structure, while the F_0 contour (when available) carries lexical and grammatical tone. The absence of clear acoustic silence between words makes automatic segmentation difficult and motivates a robust acoustic encoder together with an explicit tonal analysis module capable of leveraging pitch-related cues in tightly connected sequences.

4.2. Fundamental Frequency (F_0) Statistics

We estimated fundamental frequency with the YIN algorithm, constraining candidate pitch periods to 2-20 ms (50-500 Hz). We use a 25 ms analysis window and a 10 ms hop size, matching the STFT framing in Table 4 to maintain temporal alignment between acoustic and tonal streams. We also retain YIN's voicing probability as a pitch-confidence measure.

5. Signal Preprocessing

The primary goals of signal preprocessing are to enhance the signal-to-noise ratio, to normalise loudness across recordings and to segment the audio stream into speech chunks that can be processed efficiently by the acoustic and tonal encoders. This section summarises the main steps of the pipeline.

5.1. Preprocessing Parameters

Table 3 lists the hyperparameters used throughout the preprocessing stage. They were selected empirically to maximise speech/noise separation and spectral stability while remaining compatible with standard ASR practices.

Table 3. Audio preprocessing hyperparameters.

Parameter	Value
NMF rank r	50
Convergence threshold ϵ	10^{-4}
Regularisation term η	10^{-6}
Target RMS power	−25 dBFS
STFT window	25 ms (400 samples)
STFT hop	10 ms (160 samples)
Window function	Hann
Silence threshold θ	$\mu_{\text{noise}} + 1.5\,\sigma_{\text{noise}}$

5.2. Spectral Representation and NMF-Based Denoising

We first compute a complex short-time Fourier transform (STFT) of the input signal using a 25 ms Hann window and a 10 ms hop. The magnitude spectrogram is denoted by $\mathbf{X} \in \mathbb{R}_+^{F\times T}$, where F is the number of frequency bins and T the number of time frames.

To reduce stationary background noise, we apply non-negative matrix factorisation (NMF), approximating the spectrogram as

$$\mathbf{X} \approx \mathbf{W}\mathbf{H},$$

where $\mathbf{W} \in \mathbb{R}_+^{F\times r}$ contains spectral basis vectors and $\mathbf{H} \in \mathbb{R}_+^{r\times T}$ contains their time-varying activations. Optimisation is performed with multiplicative updates under a regularised divergence criterion, using the rank r, convergence threshold ϵ and regularisation term η given in Table 3.

After convergence, we identify basis vectors associated with noise (typically broad and slowly varying) and those associated with speech. Retaining only the latter yields a denoised spectrogram

$$\mathbf{X}_{\text{speech}} = \mathbf{W}_{\text{speech}}\mathbf{H}_{\text{speech}},$$

which is then inverted with the original phase to obtain a cleaned time-domain signal. This step improves robustness of both the acoustic and tonal encoders, especially in spontaneous recordings collected in noisy environments.

5.3. Loudness Normalisation

For each speech segment $S(t)$ of duration T, we compute its root-mean-square (RMS) power as

$$P = \sqrt{\frac{1}{T}\int_0^T S(t)^2\,dt}.$$

All segments are then rescaled to a fixed target power level:

$$F = \frac{P_{\text{target}}}{P}, \quad P_{\text{target}} = -25\ \text{dBFS}, \quad S_{\text{norm}}(t) = F \cdot S(t).$$

This simple normalisation reduces loudness variability across speakers, recording sessions and speech types, which in turn stabilises training and improves convergence of the downstream neural models.

5.4. Temporal Segmentation and VAD

Finally, we perform temporal segmentation using an energy-based voice activity detection (VAD) scheme. For each recording, noise statistics are estimated from low-energy regions, yielding a mean μ_{noise} and standard deviation σ_{noise}. Frames whose energy exceeds the threshold $\theta = \mu_{\text{noise}} + 1.5\,\sigma_{\text{noise}}$ are classified as speech; others are treated as silence or background noise.

Contiguous stretches of detected speech are grouped into segments, with short pauses optionally merged when they fall below a minimum duration. These segments are then passed to the acoustic encoder and tonal module described in Section 6. This segmentation strategy strikes a balance between preserving prosodic continuity and producing units of manageable length for training and decoding.

6. Multi-Modular ASR Architecture

Our ASR system combines three main components (Figure 2): a self-supervised acoustic encoder, a tonal

analysis module and an adaptive fusion layer integrates these complementary representations before feeding a Transformer-based CTC prediction head. Final word hypotheses are obtained using lexicon-constrained beam search with an external language model. This section describes each component and its role in the overall architecture.

Given a raw waveform $X = (x_1, \ldots, x_T)$, the system first computes a sequence of contextualised acoustic representations with a Wav2Vec 2.0-style encoder [15]. In parallel, a tonal front-end extracts pitch-related and spectral features using YIN-based F_0 estimation and short-time Fourier analysis. The resulting tonal representations are passed through a small convolutional network. An adaptive fusion mechanism then combines acoustic and tonal features at each time step before being mapped to token posteriors by a Transformer-based prediction head optimized with a CTC objective; final word hypotheses are obtained via lexicon-constrained beam search with an external language model.

Formally, the architecture maps an input waveform X to a sequence of fused hidden states

$$H^{\text{fusion}} = \mathcal{F}\big(H^{\text{acoustic}}, H^{\text{tonal}}\big),$$

where H^{acoustic} denotes the sequence of acoustic encoder outputs, H^{tonal} the tonal representations and $\mathcal{F}$ the adaptive fusion function. The CTC prediction head produces frame-level posterior distributions over the output label inventory. A lexicon-constrained beam search combined with the external language model yields the final token/word sequence.

6.1. Self-Supervised Acoustic Encoder

The acoustic encoder follows the Wav2Vec 2.0 design [15]. A stack of temporal convolutions acts as a feature encoder, mapping the input waveform X to a sequence of latent representations

$$C = (c_1, \ldots, c_L) = \text{ConvEnc}(X),$$

where each c_ℓ summarises a short window of the signal. A multi-layer Transformer then produces contextualised representations

$$H^{\text{acoustic}} = (h_1^{\text{acoustic}}, \ldots, h_L^{\text{acoustic}}) = \text{Transformer}(C),$$

which integrate longer-range phonetic and prosodic dependencies.

We first pretrain the encoder on 45 hours of unlabelled tpuri speech (read and spontaneous) using the contrastive self-supervised objective of [15], and then fine-tune it on 19h35 of scripted speech with a CTC loss. Pretraining allows the model to learn robust acoustic patterns from a larger pool of data, which is crucial in our low-resource setting.

6.2. Tonal Analysis Module

In parallel with the acoustic encoder, we compute tonal features that explicitly encode fundamental frequency and local spectral shape. Fundamental frequency $F_0(t)$ is estimated with the YIN algorithm [20] within a linguistically motivated range of candidate periods. We use the same framing scheme as for the STFT described in Section 5, thereby maintaining temporal alignment between acoustic and tonal streams.

For each frame, we extract:

- the estimated F_0 value and voicing probability;
- log-scaled energy;
- a low-dimensional projection of the magnitude spectrum (e.g. via mel filterbanks or principal components).

These features are concatenated and passed through a small convolutional network with K layers:

$$H^{\text{tonal}} = (h_1^{\text{tonal}}, \ldots, h_L^{\text{tonal}}) = \text{CNN}_{\text{tone}}(F),$$

where F denotes the frame-level tonal feature sequence. The convolutional network captures local contour patterns and short-range dependencies in the tonal domain while keeping the dimensionality of H^{tonal} compatible with that of H^{acoustic}.

6.3. Adaptive Fusion Mechanism

The two representation streams are combined by an adaptive fusion mechanism operating at each time step. Given an acoustic vector h_ℓ^{acoustic} and a tonal vector h_ℓ^{tonal} for frame ℓ, we compute a 2-way gate and a fused representation:

$$\begin{aligned} g_\ell &= \text{softmax}\big(W[h_\ell^{\text{acoustic}}; h_\ell^{\text{tonal}}] + b\big) \in \mathbb{R}^2, \\ h_\ell^{\text{fusion}} &= g_{\ell,1}\, h_\ell^{\text{acoustic}} + g_{\ell,2}\, h_\ell^{\text{tonal}}, \end{aligned} \tag{1}$$

where W and b are trainable parameters and $[\cdot;\cdot]$ denotes concatenation. By construction, $g_{\ell,1} + g_{\ell,2} = 1$, so the model can smoothly trade off between acoustic and tonal evidence at each frame. This simple mechanism allows the network to emphasise tonal cues when they are informative for instance in lexical minimal pairs while relying more heavily on the acoustic encoder when tonal information is less reliable or less discriminative.

The fusion parameters are learned jointly with the rest of the model during fine-tuning. In practice, we observe that the fusion gate assigns higher weights to the tonal stream in segments dominated by vowels and sonorants, and higher weights to the acoustic stream in consonant clusters and noisy regions.

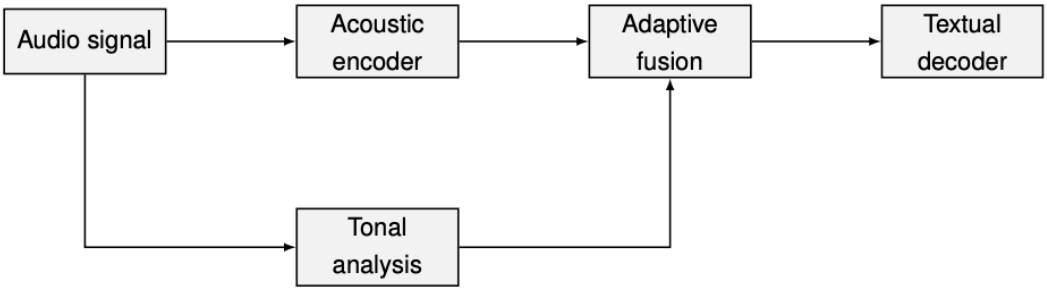


Figure 2. Overview of the multi-modular ASR architecture for tpuri.

6.4. Decoder and Language Model

The fused sequence H_{fusion} is fed to a Transformer based CTC prediction head (non-autoregressive), followed by a linear projection over the label inventory. Note that decoding is performed externally via lexicon constrained CTC beam search (Eq. 2) rather than by an autoregressive seq2seq decoder. We use lexicon-constrained CTC decoding with beam search and an external language model trained on Tpuri text.

During decoding, each candidate hypothesis is scored by interpolating the CTC acoustic score with the language-model score and a word insertion penalty:

$$\hat{Y} = \arg\max_{Y} \left(\log p_{\text{CTC}}(Y \mid H_{\text{fusion}}) + \lambda \log p_{\text{LM}}(Y) + \gamma |Y| \right), \tag{2}$$

where λ controls the influence of the language model and γ acts as a word insertion penalty that encourages or discourages longer hypotheses.

The external LM is a 4-gram word-level model trained with modified Kneser-Ney smoothing using the KenLM toolkit [21]. The training corpus consists of the normalized transcripts of the scripted training split (80% of the scripted subset: 12,013 sentences, 162,927 running word tokens). We apply the same text normalization used for WER/PER/TER scoring: tones and diacritics are preserved in the orthography, punctuation is standardized, and casing and spacing are made consistent. The LM vocabulary is derived from the training transcripts and the pronunciation lexicon; out-of-vocabulary words are mapped to a dedicated <unk> symbol. Singleton n-grams (those appearing only once in the training data) are pruned for $n \geq 3$ to reduce model size and improve generalization. The resulting vocabulary contains 6,381 unique word types, with an OOV rate of 4.7% on the test set.

Decoding is performed with a beam width of $B = 100$. The fusion weight λ and insertion penalty γ are tuned on the development split via grid search over the ranges $\lambda \in [0.5, 2.0]$ (step 0.1) and $\gamma \in [-1.0, 0.0]$ (step 0.1), and then kept fixed for all ablations. The optimal values used in all reported results are $\lambda = 1.2$ and $\gamma = -0.3$. All decoding settings, including the complete grid-search space and beam-search options, are released with the training and decoding recipes.

This modular design makes it possible to upgrade the language model or pronunciation lexicon without retraining the acoustic and tonal encoders. It also isolates the contributions of acoustic-tonal modelling from purely textual improvements, which is important for interpreting the ablation results in Section 7.

7. Experimental Results

7.1. Evaluation Protocol

To contextualise the performance of our multi-modular architecture, we compare it against two strong baseline systems trained on the same 19.5 hours of scripted tpuri speech.

Baseline A: Wav2Vec 2.0 without explicit tonal module. The first baseline uses the same Wav2Vec 2.0 acoustic encoder as our full system, initialised from the same self-supervised pretraining. The encoder directly followed by the same Transformer-based CTC prediction head described in Section 6, without any parallel tonal CNN or adaptive fusion. Instead, the CTC prediction head operates solely on the acoustic representations H^{acoustic}. Training and decoding hyperparameters, including the external language model and beam search settings, are kept identical to those of the proposed system. This baseline isolates the contribution of explicit tonal modelling and adaptive fusion.

Baseline B: CNN-BLSTM-CTC. The second baseline is a conventional fully supervised acoustic model trained from scratch. The architecture consists of a stack of 5 two-dimensional convolutional layers with batch normalisation and ReLU activations, followed by 3 bidirectional LSTM layers with 512 units in each direction and a final linear projection to the label inventory. The input is a sequence of 80-dimensional log-mel filterbank features with 25 ms window and

10 ms hop, computed on the denoised waveform. We train this model with the CTC loss using the same mixed phoneme-tone inventory as in our main system, but without any self-supervised pretraining. No external language model is used at decoding time; instead, we perform simple beam search with beam width 20. This baseline reflects the performance of a strong traditional architecture under the same data conditions, but without self-supervised learning or explicit tonal fusion.

7.2. Computational Environment, Training Setup and Reproducibility

All experiments were conducted on a single NVIDIA Tesla T4 GPU (16 GB memory) with a multi-core CPU environment, using PyTorch with CUDA acceleration. We use mixed-precision training (FP16/AMP).

Audio is resampled to 16 kHz. We fine-tune a pretrained Wav2Vec 2.0 acoustic encoder with a CTC objective on the scripted corpus (19 h 35 m of transcribed speech, split into 80%/10%/10% train/dev/test with disjoint speakers). This corresponds to approximately 15.7 h of training audio and about 2.0 h each for development and test.

Optimisation uses AdamW (weight decay 0.01) with a peak learning rate of 3×10^{-5}, scheduled with linear warmup (1000 steps) followed by linear decay. Due to GPU memory constraints, we train with mini-batches of 4-8 utterances per GPU and gradient accumulation (4 steps). We clip gradients to 1.0 for stability and freeze the convolutional feature extractor during the first 10k updates.

We apply tone-aware augmentation consisting of speed perturbation (0.9/1.0/1.1) and additive noise (10-20 dB SNR), while avoiding pitch shifting to preserve lexical tone patterns.

We fix random seeds 42 for all runs and report results using predefined train/dev/test splits. We release training/decoding configuration files, including beam size and shallow-fusion parameters (λ, γ), together with evaluation scripts and text normalization for WER/PER/TER computation.

Evaluation Metrics. We evaluate recognition performance at three complementary levels: words, phones and tones. For all metrics, we use standard minimum-edit-distance alignment between reference and hypothesis sequences, and report error rates as percentages.

Word error rate is computed as

$$\mathrm{WER} = 100 \times \frac{S_w + D_w + I_w}{N_w}, \tag{3}$$

where S_w, D_w and I_w denote the number of word substitutions, deletions and insertions, respectively, and N_w is the total number of reference words.

Phone error rate is computed analogously on phoneme sequences obtained from the pronunciation lexicon:

$$\mathrm{PER} = 100 \times \frac{S_p + D_p + I_p}{N_p}, \tag{4}$$

where S_p, D_p, I_p and N_p are defined in terms of phoneme tokens.

Tone error rate measures the accuracy of lexical tone recognition independently of segmental correctness. We define *tone-bearing units* (TBUs) as vowel nuclei in the phonemic transcription. Each TBU is annotated with one of six tonal categories: High (H), Mid-High (MH), Mid-Low (ML), Low (L), Rising (R) and Falling (F). Given a reference transcription, we extract the sequence of tonal labels $(\tau_1, \ldots, \tau_{N_t})$ associated with its TBUs. For each hypothesis transcription, we obtain the corresponding tonal sequence by applying the same lexicon-based mapping from phones to tonal labels. We then perform edit-distance alignment between reference and hypothesis tonal sequences, and compute

$$\mathrm{TER} = 100 \times \frac{S_t + D_t + I_t}{N_t}, \tag{5}$$

where S_t, D_t and I_t are the numbers of tonal substitutions, deletions and insertions, and N_t is the number of reference TBUs. TBUs without underlying lexical tone (e.g. toneless clitics, if any) are excluded from the computation.

Confidence Intervals and Statistical Significance. To assess the robustness of our results, we complement point estimates with confidence intervals and significance testing.

Bootstrap Confidence Intervals. For each system and each metric (WER, PER, TER), we estimate 95% confidence intervals using non-parametric bootstrap resampling at the utterance level. Concretely, we generate $B = 1{,}000$ bootstrap samples of the test set by sampling utterances with replacement. For each bootstrap replicate b, we recompute the metric of interest, yielding a sample $\{\hat{m}^{(b)}\}_{b=1}^{B}$. The 2.5th and 97.5th percentiles of this sample define the lower and upper bounds of the 95% confidence interval.

Significance Testing. When comparing two systems A and B, we use the same bootstrap samples to test whether the difference in error rates is statistically significant. For each bootstrap replicate b, we compute the difference $\Delta^{(b)} = \hat{m}_A^{(b)} - \hat{m}_B^{(b)}$. The empirical p-value is given by

$$p = 2 \min\left(\frac{1}{B} \sum_{b=1}^{B} \mathbf{1}\{\Delta^{(b)} > 0\}, \frac{1}{B} \sum_{b=1}^{B} \mathbf{1}\{\Delta^{(b)} < 0\} \right). \tag{6}$$

where $\mathbb{I}[\cdot]$ is the indicator function. We consider differences with $p < 0.05$ to be statistically significant. This procedure allows us to identify, for example, whether the gains of the proposed tonal fusion architecture over the Wav2Vec 2.0 baseline are unlikely to be due to random variation in the test set.

We evaluate the proposed multi-modular architecture on the scripted portion of the annotated *tpuri* corpus described in Section 3 (see Table 2). The scripted data amount to 19h35 of transcribed speech (approximately 19.5 hours), corresponding to 162 927 tokens and 12 013 sentences. We split this corpus into 80% training, 10% development and 10% test, with disjoint speakers and a balanced gender distribution across splits. Unless otherwise stated, all results reported in this section are computed on the scripted test set.

Recognition performance is measured in terms of word error rate (WER), phone error rate (PER) and tone error rate (TER). On the scripted test set, the full multi-modular system achieves a WER of 10.4%, a PER of 8.7% and a TER of 6.1%. Given the relatively small amount of labelled data and the tonal and morphophonological complexity of tpuri, these figures indicate that the proposed architecture provides a competitive entry point for ASR in this language.

Figure 3 shows the evolution of the training and validation loss as a function of the number of training iterations. Both curves decrease steadily before plateauing, and the gap between them remains limited throughout training. No strong signs of overfitting are observed in the final stages, suggesting that the model capacity is reasonably well matched to the size of the corpus. Early stopping based on the development loss typically selects models in the 40-50 epoch range.

To better understand the strengths and weaknesses of the system, we analyse performance across four linguistic dimensions: tonal phenomena, grammatical constructions, syllable structure and morphology. For each dimension, we evaluate WER and PER on a subset of the test corpus where the corresponding phenomena are particularly salient (Figure 4). The system performs best on tonal and grammatical contexts, where WER remains below 10.2% and PER below 8.0%. Syllable structure and morphology remain more challenging, with WER of up to 15.3% and PER of up to 11.2%. This pattern confirms that explicit tonal modelling is beneficial, while highlighting the need for richer modelling of complex syllable types and morphological alternations.

Finally, Table 4 presents an ablation study quantifying the contribution of each module. Removing the tonal branch increases WER by 1.5 points and TER by 2.3 points, demonstrating the importance of explicit tonal features even in the presence of a powerful self-supervised encoder. Disabling self-supervised pretraining degrades WER by 3.4 points, confirming the benefit of pretraining on unlabelled tpuri speech. Adaptive fusion and tone-aware data augmentation yield smaller but consistent improvements. Together, these results support our central claim: competitive ASR performance on a low-resource tonal language can be achieved by tightly integrating self-supervised acoustic representations with dedicated tonal modelling.

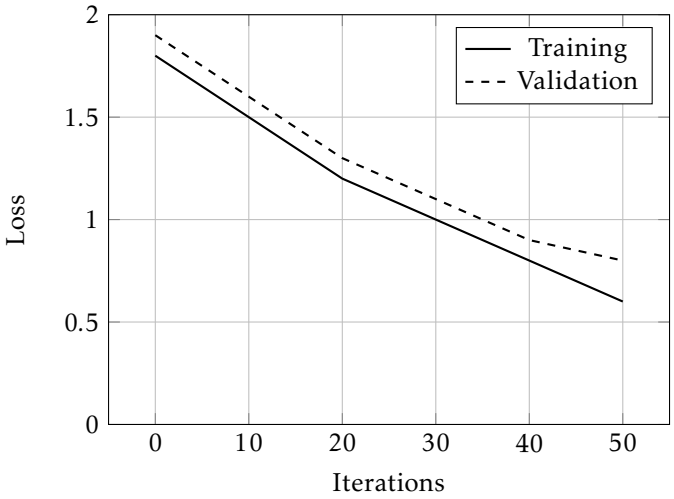


Figure 3. Training and validation loss as a function of training iterations.

The joint convergence of training and validation losses, without a marked divergence in validation, indicates a globally stable optimization regime and a model capacity that is well matched to the data size. The moderate gap between training and validation curves suggests good generalization, and an early stopping point around 40–50 iterations would likely prevent late-stage overfitting while preserving peak performance.

The highest error rates are observed for syllable structure, indicating difficulties in modeling consonant clusters and phonotactic constraints. Strengthening decoding constraints (e.g., via a more structured language model or explicit penalties on illicit sequences), together with additional targeted training data, is likely to improve performance in this area. In contrast, tonal error rates are the lowest among all tasks, suggesting that the acoustic features leveraged, in particular F_0 contours, effectively capture relevant tonal contrasts. The gaps between WER and PER also highlight lexical and segmentation errors beyond the phoneme level (e.g., function words, agreement, morphological markers), pointing to potential gains from more powerful language-model rescoring and a more finely normalized

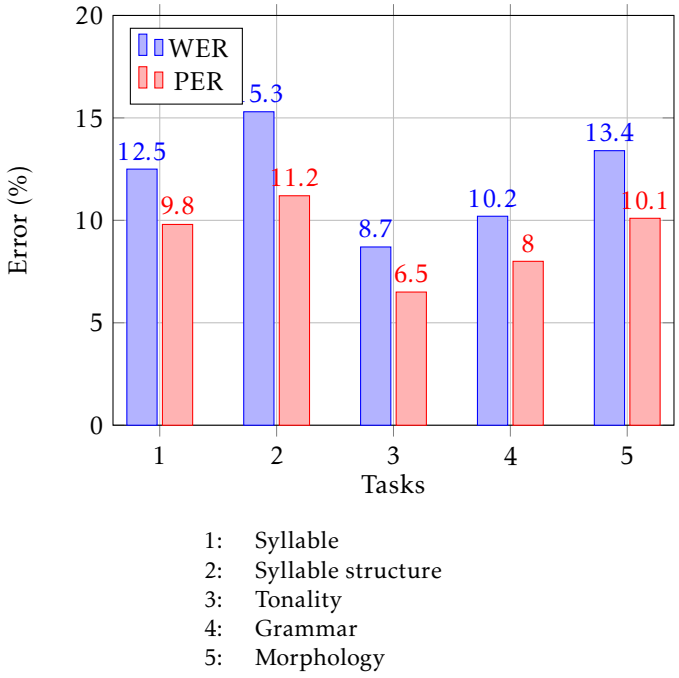


Figure 4. Comparison of WER and PER across different evaluation tasks.

lexicon. Overall, these results indicate that the proposed multi-modular architecture is particularly effective for tonal and grammatical recognition in *tpuri*, while also revealing clear avenues for future improvements in syllable structure and morphology.

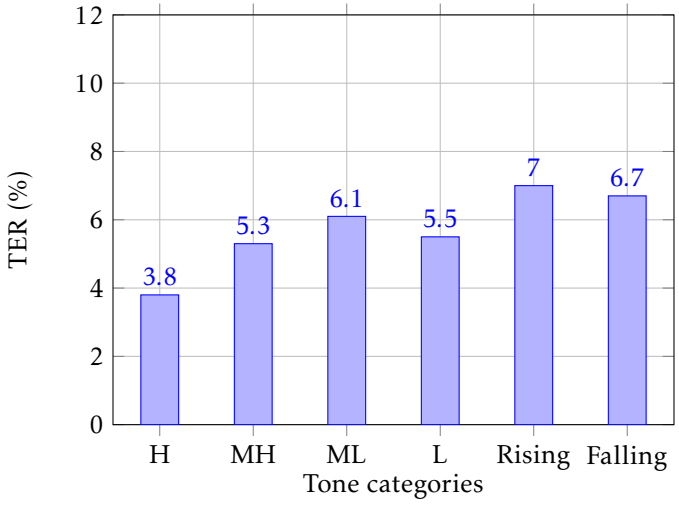


Figure 5. Tone Error Rate (TER) by tone category for the *tpuri* ASR system.

Figure 5 reports the tone error rate (TER) per tone category. Overall, the system achieves relatively low error rates across all tones (all TERs below 8%), which confirms that the explicit tonal modeling and the dedicated tonal module are effective for *tpuri*. However, the distribution of errors is not uniform and reveals systematic differences between level and contour tones.

The lowest TER is obtained for high level tones (H, 3.8%), suggesting that these are acoustically the most salient and the most consistently realized in the corpus. Mid-high (MH, 5.3%) and mid-low (ML, 6.1%) tones exhibit slightly higher error rates, which is consistent with the fact that their F_0 ranges partially overlap and are therefore more prone to confusion, both for human annotators and for the model. Low tones (L, 5.5%) remain relatively well recognized, although their lower intensity and potential interaction with voicing and segmental context make them somewhat more challenging than H tones.

As expected, contour tones (Rising and Falling) are the most difficult to model, with TERs of 7.0% and 6.7%, respectively. These tones require the system to track not only absolute F_0 values but also their temporal dynamics over the syllable, and they are more sensitive to local prosodic variation, coarticulation, and phrase-level intonation. The fact that contour tones still remain below 8% TER indicates that the combination of F_0 trajectories, spectral features, and contextual embeddings captures a substantial portion of the tonal structure, but also points to a clear avenue for improvement. In particular, increasing the amount of annotated data for contour tones and incorporating more fine-grained modeling of phrase-level prosody are likely to further reduce these residual errors.

As shown in Table 4, both the tonal module and the self-supervised pretraining on tpuri play a central role in the overall performance of the system. Removing the tonal branch leads to a +1.5 absolute increase in WER and a +2.3 increase in TER, while disabling self-supervised pretraining degrades WER by +3.4 points. Adaptive fusion and tonal-aware data augmentation bring additional but more moderate gains.

Configuration	WER (%)	PER (%)	TER (%)
Full multi-modular system (ours)	**10.4**	**8.7**	**6.1**
w/o tonal module	11.9	9.8	8.4
w/o self-supervised pretraining on tpuri	13.8	11.4	9.7
w/o adaptive fusion (simple late concat.)	11.3	9.1	7.0
w/o tone-preserving augmentation (speed+noise)	11.0	9.0	6.9

Table 4. Ablation study of the proposed tpuri ASR system on the scripted speech test set.

7.3. Analysis of the Adaptive Fusion Gate

Logging and aggregation. During inference on the scripted test set, we log g_ℓ for every frame and aggregate these weights globally, by broad phonetic class (vowel/sonorant vs. obstruent), and by local signal conditions (high-SNR vs. low-SNR frames, approximated by frame-level energy). Broad phonetic classes are obtained by force-aligning the reference

transcription with the acoustic stream and mapping phones to classes.

Figure 6 shows the distribution of tonal weights $g_{\ell,2}$, and Table 5 reports the mean and standard deviation by class. Overall, the model assigns a higher tonal weight on vowel nuclei and sonorant segments (tone-bearing regions), while down-weighting tonal features in consonant clusters and low-energy/noisy frames. This behaviour is consistent with the linguistic status of tone in *tpuri* (primarily realised on vowel nuclei) and supports the design choice of an explicit tonal branch.

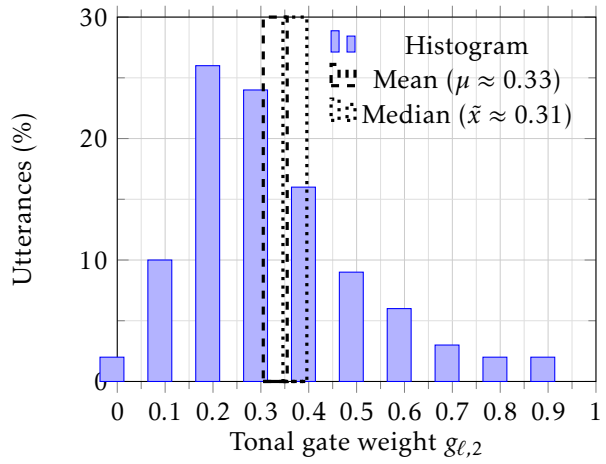


Figure 6. Distribution of tonal gate weights $g_{\ell,2}$ on the scripted test set. Lower values indicate acoustic-dominated decisions ($g_{\ell,1} = 1 - g_{\ell,2}$), while higher values indicate stronger reliance on tonal cues.

For the signal classes, we compute a frame-level SNR proxy in dB by comparing speech-frame energy to the estimated noise-floor energy (from non-speech regions used in VAD), and bin frames into < 10 dB, 10-20 dB and ≥ 20 dB. Pitch confidence is derived from YIN's voicing probability (high if > 0.6, low otherwise).

Across phonetic classes, $g_{\ell,2}$ is highest on vowels and sonorants, i.e., tone-bearing and F0-stable regions, and lowest on unvoiced obstruents and pauses where pitch cues are absent. Across signal conditions, tonal reliance decreases under low SNR or low pitch-confidence, indicating that the fusion gate adapts to the reliability of tonal evidence rather than enforcing tone uniformly.

(a) Phonetic classes (tone-bearing segments show higher $g_{\ell,2}$).

Class	Mean	Std	Med.	P_{10}	P_{90}
Vowels (tone-bearing)	0.44	0.17	0.42	0.21	0.69
Sonorants (N/L/G)	0.37	0.15	0.35	0.18	0.58
Voiced obstruents	0.28	0.13	0.27	0.12	0.47
Unvoiced obstruents	0.16	0.10	0.15	0.05	0.32
Silence / pause	0.05	0.04	0.04	0.01	0.12

(b) Signal classes (tonal reliance decreases when F_0 becomes less reliable).

Class	Mean	Std	Med.	P_{10}	P_{90}
High SNR (≥20 dB)	0.36	0.14	0.35	0.17	0.58
Mid SNR (10-20 dB)	0.33	0.15	0.31	0.15	0.56
Low SNR (<10 dB)	0.24	0.14	0.22	0.08	0.48
High pitch conf.	0.40	0.15	0.39	0.20	0.63
Low pitch conf.	0.22	0.13	0.20	0.06	0.44

Table 5. Summary statistics of tonal gate weights $g_{\ell,2}$ by phonetic and signal classes on the scripted test set. Higher values indicate stronger reliance on tonal cues.

8. Limitations

Despite the strong results obtained on scripted *tpuri* speech, the present study has several limitations that should be considered when interpreting the findings and when reusing the proposed architecture.

Data size and domain coverage. Our labelled training set is limited (19h35 scripted speech) and primarily read/scripted. Performance on spontaneous speech, code-switching, and noisy field recordings may be lower. While we include self-supervised pretraining on 45 hours of read and spontaneous speech, this remains small compared to high-resource benchmarks, and additional unlabelled audio could further improve robustness.

Generalisability to other tonal systems. Tpuri exhibits a six-way lexical tone system with both level and contour tones. While the modular design is meant to transfer to other tonal languages, different tone inventories (e.g. register vs. contour, downstep, tone sandhi) may require modifications to the tonal feature extraction and the tone-aware lexicon.

Tonal feature extraction. We use YIN-based F_0 estimation, which can be sensitive to breathy voice, creaky voice, and low SNR conditions. Although the adaptive fusion layer can down-weight unreliable tonal cues, a more robust neural pitch estimator or joint pitch-ASR training could reduce failure cases.

Resource availability constraints. Full release of raw audio may be constrained by speaker consent and institutional policies. To support reproducibility, we will at minimum release the pronunciation lexicon, tone annotations, train/dev/test splits, and training/decoding recipes, and we will provide access to audio via an appropriate data-sharing mechanism where public redistribution is not possible.

Evaluation scope. We report WER/PER/TER on a held-out test set with disjoint speakers, but we do

not yet provide extensive out-of-domain evaluations (new speakers, new domains) or human perceptual evaluations of tone errors. These are important directions for follow-up work.

9. Conclusion

This paper has presented a hybrid multi-modular architecture for automatic speech recognition in *tpuri*, a low-resource tonal language of the Mboum-Day branch of the Niger-Congo family. The system combines a self-supervised Wav2Vec 2.0 acoustic encoder with a dedicated tonal analysis module based on YIN pitch estimation and STFT-derived spectral features, integrated through an adaptive fusion mechanism and a Transformer-based CTC prediction head.

On a scripted corpus of 19h35 of *tpuri* speech, the proposed architecture achieves a word error rate of 10.4%, a phone error rate of 8.7% and a tone error rate of 6.1%. Ablation experiments show that both self-supervised pretraining and explicit tonal modelling are crucial to this performance: removing the tonal branch or disabling pretraining leads to substantial degradations in WER and TER. A fine-grained error analysis across tonal, grammatical, syllabic and morphological dimensions further indicates that the system models lexical tone and clause-level syntax relatively well, while complex syllable structure and rich morphology remain challenging.

These findings support our central claim that competitive ASR systems can be built for under-resourced tonal languages by tightly coupling self-supervised acoustic representations with linguistically informed tonal features. Beyond *tpuri*, the methodological choices made here corpus design, tonal feature extraction, adaptive fusion and error analysis offer a reusable blueprint for extending ASR to other Niger-Congo and tonal languages facing similar data constraints.

Future work will focus on several directions. First, we plan to enrich the language model and pronunciation lexicon in order to better handle morphological variation and rare lexical items. Second, we aim to incorporate more detailed phrase-level prosodic information, including boundary tones and intonational patterns, to further reduce confusions among contour tones. Third, we will explore cross-lingual transfer from related Mboum-Day and Niger-Congo languages, as well as multilingual self-supervised pretraining, to investigate how far carefully chosen auxiliary languages can compensate for the limited amount of labelled *tpuri* data. More broadly, we hope that the resources and results reported in this work will encourage further research on speech and language technologies for under-resourced African languages.

References

[1] Bird, S. (2009) Natural language processing and linguistic fieldwork. *Computational linguistics* **35**(3): 469–474.

[2] Michailovsky, B., Mazaudon, M., Michaud, A., Guillaume, S., François, A. and Adamou, E. (2014) Documenting and researching endangered languages: the pangloss collection .

[3] De Vries, N.J., Davel, M.H., Badenhorst, J., Basson, W.D., De Wet, F., Barnard, E. and De Waal, A. (2014) A smartphone-based ASR data collection tool for under-resourced languages. *Speech Communication* **56**(1): 119–131. doi:10.1016/j.specom.2013.07.001, URL `http://dx.doi.org/10.1016/j.specom.2013.07.001`.

[4] Bhaskararao, P. (2004) Phonetic documentation of endangered languages: Creating a knowledge-base containing sound recording, transcription and analysis. *Acoustical Science and Technology* **25**(4): 219–226. doi:10.1250/ast.25.219, URL `https://www.jstage.jst.go.jp/article/ast/25/4/25_4_219/_pdf/-char/ja`.

[5] Benahmed, Y. (2018) *Analyse Sémantique pour Systèmes de Dialogue Verbaux*. Ph.D. thesis, Institut National de la Recherche Scientifique (Canada).

[6] Mehta, D., Diddee, H., Saxena, A., Shukla, A., Santy, S., Kommiya, R., Sharma, A. *et al.* (2022) Learnings from technological interventions in a low resource language. *arXiv preprint arXiv:2211.16172* .

[7] Seignobos, C. and Tourneux, H. (2002) *Le Nord-Cameroun à travers ses mots: dictionnaire de termes anciens et modernes: province de l'extrême-nord* (KARTHALA Editions).

[8] Kolyang, D.T. (2010) *PARLONS TPURI-Cameroun et Tchad* (Harmattan).

[9] Ragni, A., Knill, K.M., Madhavi Mallidi, S.R., Gales, M.J.F. and Woodland, P.C. (2014) Data augmentation for low resource languages. In *Proceedings of Interspeech*: 810–814.

[10] Huang, H. and Seide, F. (2000) Pitch tracking and tone features for mandarin speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, **3**: 1523–1526. doi:10.1109/ICASSP.2000.861942.

[11] Wang, Y. and Lee, L. (2010) Mandarin tone recognition using affine-invariant prosodic features and tone posteriorgram. In *Proceedings of Interspeech*: 2850–2853. doi:10.21437/Interspeech.2010-305.

[12] Lei, X. and Ostendorf, M. (2007) Word-level tone modeling for mandarin speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*: 665–668.

[13] Schultz, T., Vu, N.T. and Schlippe, T. (2013) Globalphone: A multilingual text & speech database in 20 languages. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (IEEE): 8126–8130.

[14] Coto-Solano, R. (2021) Explicit tone transcription improves ASR performance in extremely low-resource languages: A case study in Bribri. In Mager, M., Oncevay, A., Rios, A., Ruiz, I.V.M., Palmer, A., Neubig, G. and Kann, K. [eds.] *Proceedings of*

the First Workshop on Natural Language Processing for Indigenous Languages of the Americas (Online: Association for Computational Linguistics): 173–184. doi:10.18653/v1/2021.americasnlp-1.20, URL https://aclanthology.org/2021.americasnlp-1.20/.

[15] Baevski, A., Zhou, Y., Mohamed, A. and Auli, M. (2020) wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, **33**: 12449–12460.

[16] Pratap, V., Sriram, A., Tomasello, P., Hannun, A., Liptchinsky, V., Synnaeve, G. and Collobert, R. (2020) Massively multilingual asr: 50 languages, 1 model, 1 billion parameters. In *Proceedings of Interspeech*: 4751–4755.

[17] Reitmaier, T. and colleagues (2022) Opportunities and challenges of automatic speech recognition systems for low-resource language speakers. *Proceedings of the ACM on Human-Computer Interaction* **6**(CSCW1): 1–28. doi:10.1145/3517639.

[18] Ruelland, S. (1992) Description du parler tupuri de Mindaore (Tchad). In *Phonologie, morphologie, syntaxe* (Université de la Sorbonne Nouvelle Paris III).

[19] Wittenburg, P., Mosel, U. and Dwyer, A. (2002) Methods of language documentation in the DOBES project. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)* (Las Palmas, Canary Islands - Spain: European Language Resources Association (ELRA)). URL https://aclanthology.org/L02-1221/.

[20] De Cheveigné, A. and Kawahara, H. (2002) Yin, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America* **111**(4): 1917–1930.

[21] Heafield, K. (2011) Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation* (Edinburgh, Scotland: Association for Computational Linguistics): 187–197. doi:10.18653/v1/W11-2123, URL https://aclanthology.org/W11-2123.