Indic-CodecFake meets SATYAM: Towards Detecting Neural Audio Codec Synthesized Speech Deepfakes in Indic Languages

Girish[1]* **Mohd Mujtaba Akhtar**[2]* **Orchid Chetia Phukan**[3]*† **Arun Balaji Buduru**[3]
[1]UPES, India [2]VBSPU, India [3]IIIT-Delhi, India
Correspondence: orchidp@iiitd.ac.in

Abstract

The rapid advancement of Audio Large Language Models (ALMs), driven by Neural Audio Codecs (NACs), has led to the emergence of highly realistic speech deepfakes, commonly referred to as CodecFakes (CFs). Consequently, CF detection has attracted increasing attention from the research community. However, existing studies predominantly focus on English or Chinese, leaving the vulnerability of Indic languages largely unexplored. To bridge this gap, we introduce Indic-CodecFake (ICF) dataset, the first large-scale benchmark comprising real and NAC-synthesized speech across multiple Indic languages, diverse speaker profiles, and multiple NAC types. We use IndicSUPERB as the real speech corpus for generation of ICF dataset. Our experiments demonstrate that state-of-the-art (SOTA) CF detectors trained on English-centric datasets fail to generalize to ICF, underscoring the challenges posed by phonetic diversity and prosodic variability in Indic speech. Further, we present systematic evaluation of SOTA ALMs in a zero-shot setting on ICF dataset. We evaluate these ALMs as they have shown effectiveness for different speech tasks. However, our findings reveal that current ALMs exhibit consistently poor performance. To address this, we propose **SATYAM**, a novel hyperbolic ALM tailored for CF detection in Indic languages. **SATYAM** integrates semantic representations from Whisper and prosodic representations from TRILLsson using through Bhattacharya distance in hyperbolic space, and subsequently performs the same alignment procedure between the fused speech representation and a input conditioning prompt. This dual-stage fusion framework enables **SATYAM** to effectively model hierarchical relationships both within speech (semantic–prosodic) and across modalities (speech–text). Extensive evaluations show that **SATYAM** consistently outperforms competitive end-to-end and ALM-based baselines on the ICF benchmark.

1 Introduction

Speech deepfakes have rapidly transitioned from proof-of-concept demonstrations to tools deployed in large-scale, real-world attacks. Recent years have witnessed a surge in financial frauds [1] [2]. With only a few seconds of recorded speech, attackers can now generate long, natural-sounding utterances that preserve a target speaker's accent, prosody, and speaking style (Khanjani et al., 2022). Beyond financial fraud, such synthesized speech has also been exploited for spreading disinformation and manipulating public opinion (Luong et al., 2025). These fake voices are predominantly generated by text-to-speech (TTS) models and voice conversion (VC) techniques. Modern TTS architectures such as WaveNet and VITS can synthesize expressive speech directly from text (van den Oord et al., 2016; Kim et al., 2021; Tan et al., 2021), while VC models such as AutoVC and StarGAN-VC2 enable many-to-many voice style transfer without requiring parallel data (Qian et al., 2019; Kaneko et al., 2019).

In response to these threats, the speech community has devoted substantial efforts towards building effective systems for detecting such fakes. Benchmark series such as ASVspoof (Wu et al., 2015; Wang et al., 2020; Liu et al., 2023; Wang et al., 2024) have driven progress on spotting synthetic and replayed speech, and a wide range of countermeasures have been explored. Early systems rely on handcrafted spectral features with classical machine learning techniques such as GMM, Random Forest and so on (Patel and Patil, 2015, 2016; Yu et al., 2017; Ji et al., 2017). Subsequent studies have leveraged deep learning algorithms for improved speech deepfake detection performance

*Equal contribution as first author.
†Core Ideation

[1]Red Goat
[2]Voice Cloning Heist

Findings of the Association for Computational Linguistics: ACL 2026, pages 43476–43489

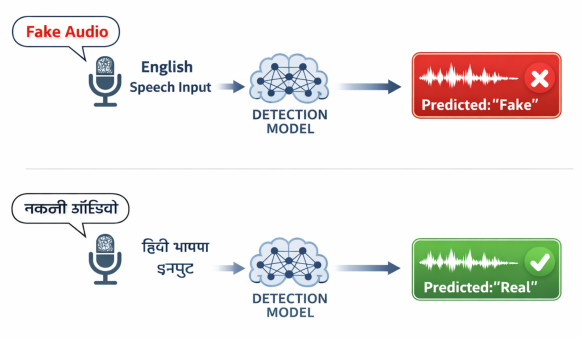

Figure 1: Existing CodecFake detectors perform well on English speech but frequently misclassify Hindi fake speech as real

(Lei et al., 2020; Dinkel et al., 2017; Jung et al., 2022; Huang et al., 2025). Building on this research landscape, the start of this decade marked a fundamental shift in speech deepfake detection with the widespread adoption of large-scale pre-trained models (PTMs) trained on diverse and extensive speech corpora. As a result, recent studies have evaluated a variety of state-of-the-art (SOTA) PTMs—including WavLM, Wav2Vec2, Whisper, and MMS—for speech deepfake detection (Kawa et al., 2023; Phukan et al., 2024a; Müller et al., 2024; El Kheir et al., 2025). More recently, audio large language models (ALMs), such as Qwen2-Audio, have also been explored for speech deepfake detection, motivated by their strong performance across a range of related speech processing tasks (Gu et al., 2025).

However, the majority of these studies primarily focus on speech deepfakes generated by TTS or VC systems. In recent years, driven by advances in ALMs, a new category of speech deepfakes—referred to as CodecFakes (CFs)—has emerged and attracted growing attention from the research community. These ALMs incorporate neural audio codecs (NACs) within their architectures for both speech encoding and synthesis. For instance, AudioLM relies on the SoundStream as the backbone NAC (Borsos et al., 2023). In response to this emerging threat, (Wu et al., 2024b) and (Lu et al., 2024a) initiated the first explorations of CF detection by releasing dedicated CF datasets. Their studies demonstrate that models trained on traditional speech deepfake datasets fail to generalize effectively to CF benchmarks, largely due to shifts in the underlying distributional characteristics of CFs compared to TTS- or VC-generated synthetic speech. As such various succeeding studies have been conducted in this direction for CF detection (Xie et al., 2025; Chen et al., 2025). Despite this progress, existing CF detection datasets are predominantly English-focused or atmost including Chinese. However, CF datasets for other language groups remains largely unaddressed.

In this work, we extend CF detection to Indic languages, one of the world's most linguistically diverse settings. India, now the most populous country globally encompasses speakers from major Indo-European and Dravidian families as well as Austro-Asiatic and Tibeto–Burman languages. This diversity has fueled rapid growth in Indic speech technologies, including large-scale benchmarks such as IndicSUPERB (Javed et al., 2023). At the same time, surveys report widespread exposure to AI-driven voice scams [3] . Despite this elevated risk, existing CF datasets remain largely English-centric. To address this gap, we introduce Indic-CodecFake (ICF) dataset, the first large-scale benchmark comprising real and NAC-synthesized speech across multiple Indic languages, diverse speaker profiles, and multiple NAC types. We show that models trained on previous CF detection benchmarks fails on ICF dataset (Figure 1). We further conduct a systematic zero-shot evaluation of SOTA ALMs on ICF dataset. We evaluate ALMs as previous research has explored them for speech deepfake detection but they haven't evaluated them for CF detection (Gu et al., 2025). We observe consistently poor performance on ICF dataset and thus showing the need for Indic-centric CF detection pipelines. To overcome these limitations, we propose **SATYAM**, a hyperbolic ALM tailored for Indic CF detection. **SATYAM** leverages Bhattacharya distance in hyperbolic space to align semantic representations from Whisper with prosodic representations from TRILLsson, followed by a second stage of the same alignment with a input conditioning prompt. This hierarchical modeling enables effective integration of speech–speech and speech–text relationships. Extensive experiments demonstrate that **SATYAM** consistently outperforms competitive ALM-based and end-to-end baselines on ICF, while also achieving strong performance on existing CF detection benchmarks.

To summarize, the major contributions are as follows

- We introduce ICF dataset, the first large-scale CF dataset in Indic-languages.

[3] McAfee, *Beware the Artificial Impostor* (report).

- We evaluate previous SOTA CF detectors trained on previous CF benchmarks on ICF dataset and show its poor generalization. We also present a evaluation of SOTA ALMs on ICF dataset and observe poor performance. This highlights the need for Indic-centric CF detection modeling pipelines.

- We propose, **SATYAM**, a novel hyperbolic ALM for CF detection primarily in Indic Languages. To the best of our knowledge, we are the first study to explore extension of ALMs to hyperbolic space.

We release the dataset, data generation pipeline and code here[4].

2 Related Works

In this section, we briefly review prior work on CF detection. Early investigations into CF detection and its associated vulnerabilities were initiated by Wu et al. (Wu et al., 2024b) and Lu et al. (Lu et al., 2024b). These studies demonstrated that SOTA speech deepfake detectors trained on traditional deepfake datasets—primarily synthesized using TTS or VC systems—fail to generalize effectively to CF scenarios. Wu et al. (Wu et al., 2024b) constructed a CF dataset using the English VCTK corpus and a diverse set of NACs, and evaluated AASIST-based architectures for CF detection. In parallel, Lu et al. (Lu et al., 2024b) and Xie et al. (Xie et al., 2025) developed CF datasets spanning English and Chinese by leveraging VCTK and AISHELL-3 as real speech corpora, respectively, and similarly adopted AASIST-based modeling approaches. Building on these efforts, Du et al. (Xie et al., 2025) further expanded the CF detection landscape by incorporating a larger variety of NAC families while continuing to rely on AASIST-based architectures. In contrast, Xie et al. (Xie et al., 2025) proposed a novel CF detection strategy based on sharpness-aware minimization to improve robustness. Despite this progress, existing CF datasets are almost exclusively limited to high-resource languages, primarily English and Chinese, underscoring the need for multilingual CF benchmarks. In particular, no prior work has systematically focused on CF detection in Indic languages. While Cui et al. (Cui et al., 2025) explored multilingual CF detection by building upon the Common Voice corpus and included Tamil as an Indic language, their dataset is not publicly released and considers only a single Indic language. In addition, several Indic-language–focused datasets have been proposed for general speech deepfake detection mostly TTS, VC (Sharma et al., 2025; Ranjan et al., 2025); however, none specifically target CF detection. In this work, we address this gap by introducing ICF dataset, the first large-scale and comprehensive CF dataset focused on Indic languages. Furthermore, we propose **SATYAM**, a novel hyperbolic ALM tailored for CF detection in Indic languages.

3 Indic-CodecFake Dataset

In this section, we first describe the real-speech source datasets for Indic languages, followed by the SOTA NACs considered in our study. We then detail the data generation pipeline used to construct the ICF dataset.

Indic Speech Source: We use the IndicSUPERB[5] (Javed et al., 2023) dataset as the real speech corpus. It consists of 12 Indian-languages. IndicSUPERB is also used by IndicSynth (Sharma et al., 2025) as a base real speech corpus. IndicSynth is a SOTA large-scale speech deepfake benchmark on Indic-languages generated through TTS or VC methods. Additional details about IndicSUPERB is given in Appendix 7 IndicSUPERB.

Neural Audio Codecs: We follow prior work by Lu et al. (2024a) and Wu et al. (2024b) and adopt SOTA publicly released NACs that are widely available and easy to reproduce. **DAC** (Kumar et al., 2024): We use 16 kHz, 24 kHz, and 44 kHz variants. **Encodec** (Défossez et al., 2022): We use 24 kHz and 48 kHz models. **SoundStream** (Zeghidour et al., 2021): We use the 16 kHz configuration. **SpeechTokenizer** (Zhang et al., 2024): We use the default 16 kHz setup. **FunCodec** (Du et al., 2024): We use the official 16 kHz version. **AudioDec** (Wu et al., 2023): We use 28 kHz and 48 kHz variants. **SNAC** (Siuzdak et al., 2024): We use 24 kHz, 32 kHz, and 44 kHz models. **MIMI** (Défossez et al., 2024): It operates in 24 kHz.

Generation Pipeline of ICF: To construct ICF, we follow a controlled NAC-based resynthesis pipeline inspired by prior work on CF generation (Wu et al., 2024b). Specifically, we resynthesize real speech from IndicSUPERB using a diverse set of NACs.

[4] https://helixometry.github.io/IndicFake/

[5] https://github.com/AI4Bharat/IndicSUPERB?tab=readme-ov-file

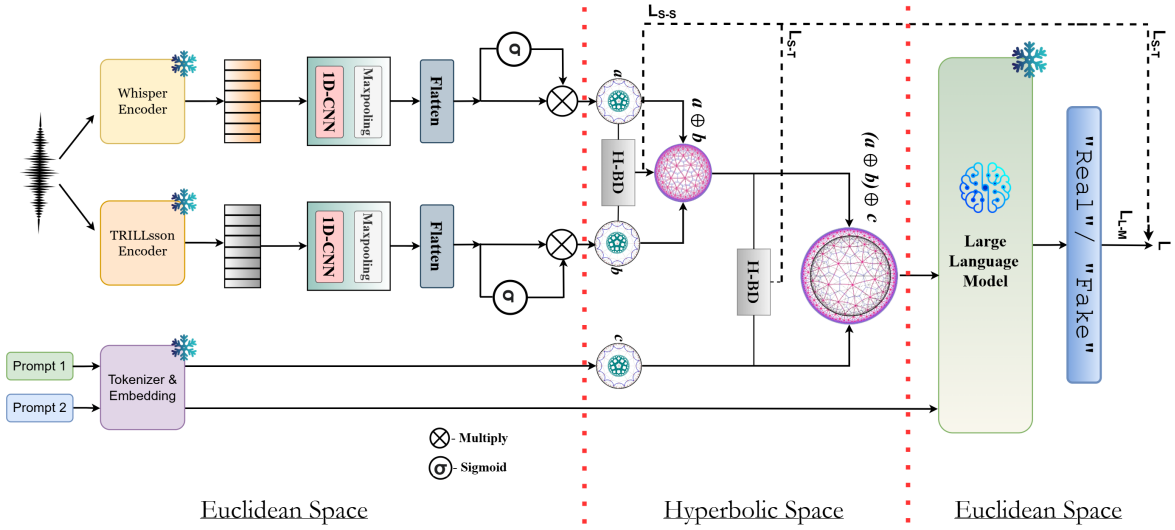

Figure 2: Proposed Framework: **SATYAM**; H-BD stands for Bhattacharya distance in Hyperbolic space

All source utterances are drawn directly from the official IndicSUPERB train/valid/test splits and treated as real references. Given a real speech waveform x, a NAC encoder $\mathcal{E}$ maps it to a discrete latent representation $z = \mathcal{E}(x)$, which is then reconstructed by a decoder $\mathcal{D}$ to obtain $\tilde{x} = \mathcal{D}(z)$. The reconstructed signal $\tilde{x}$ constitutes the corresponding CF sample. This process of encoding-decoding preserves linguistic content and speaker characteristics while introducing NAC-specific artifacts representative to each NAC. We apply this procedure to every utterance and each selected NAC, resulting in parallel corpora where each real speech sample has a one-to-one CF counterpart for every NAC configuration. We retain the original split assignment given in IndicSUPERB during the CF generation process. We define two evaluation settings: (i) Seen: Training and evaluation involve CFs synthesized using the same set of NACs (e.g., SNAC variants, DAC variants, Encodec, SoundStream, and SpeechTokenizer). As IndicSUPERB, contains two sets of test set, we use the test-known for this evaluation setting. (ii) Unseen: The test set contains CFs generated by NACs not observed during training (e.g., FunCodec variants, AudioDec variants, and MIMI), enabling evaluation of cross-codec generalization. We use the test-unkown test split for this evaluation setting. This pipeline is applied uniformly across all Indic languages in IndicSUPERB.

4 Methodology

In this section, we describe the methodology underlying the proposed hyperbolic ALM, **SATYAM**. An overview of the model architecture is illustrated in Figure 2. We propose **SATYAM**, a supervised hyperbolic ALM for CF detection that formulates detection as a conditional generation task, following prior ALM-based approaches for speech deepfake detection (Gu et al., 2025). They formulate speech deepfake detection as a audio question answering task and has shown its effectiveness. We begin by describing the audio encoders and LLM decoder employed in our approach. We then present the complete workflow of **SATYAM**, including hyperbolic fusion of semantic and paralinguistic speech representations and their alignment with prompt inputs. Our design is motivated by recent findings showing that audio encoders constitute the primary performance bottleneck in existing ALM pipelines, and that employing stronger, task-relevant speech encoders leads to substantial gains in downstream tasks (Li et al., 2025). While prior work has primarily focused on general speech deepfake detection, the potential of such encoder-centric ALM designs for CF detection remains largely unexplored. Furthermore, we adopt hyperbolic geometry for both speech–speech fusion and speech–prompt alignment based on the observation that semantic and paralinguistic cues might exhibit inherent hierarchical structure (Mary Zarate et al., 2015; Chen et al., 2023). Prior work by Phukan et al. (Phukan et al., 2025a) has shown evidence of hierarchical organization within speech representations, while hierarchical relationships across modalities such as speech, vision and text are a well-established principle in multimodal learning (Desai et al., 2023; Hong et al., 2023). Hyperbolic space, with its abil-

ity to naturally embed hierarchical structures, therefore provides a principled geometric framework for modeling these relationships in **SATYAM**.

Audio Encoders: We use Whisper (Radford et al., 2023) and TRILLsson (Shor and Venugopalan, 2022) as audio encoders as they have proven effective by previous research on synthetic speech detection (Phukan et al., 2024b; Das et al., 2025). More information about the audio encoders are given in Appendix 7 Additional Information about Audio Encoders.

LLM decoder: We employ Qwen2-7B[6] as the pre-trained decoder-only LLM backbone in our framework (Team et al., 2024). It is selected as it has demonstrated effectiveness across a range of speech and audio language modeling tasks, including speech deepfake detection (Li et al., 2025) and speech emotion recognition (Su et al., 2025). Furthermore, Qwen2-Audio, a SOTA ALM, is explicitly built upon the Qwen2 decoder architecture, further validating the suitability of Qwen2 as a strong LLM backbone for speech and audio tasks (Chu et al., 2024).

Workflow: Given an input speech utterance x, the objective of **SATYAM** is to determine whether the utterance is real or fake by generating a short natural-language decision. From x, we extract two complementary speech representations: a semantic representation using Whisper, $e_w \in \mathbb{R}^{d_w}$, and a paralinguistic representation using TRILLsson, $e_t \in \mathbb{R}^{d_t}$. Both representations are passed through a lightweight CNN block consisting of a 1D convolutional layer (filter size 3) followed by max pooling, as adopted in prior work (Phukan et al., 2024b). The resulting representations are then projected into a shared Euclidean space of dimension d: $e_w = W_w e_w$ and $e_t = W_t e_t$ where W_w and W_t are learnable projection matrices. We further introduce a sigmoid gating module that filters the input representations signal and forwards only salient information to the subsequent stage. We next map these representations into a d-dimensional hyperbolic space $\mathbb{H}^d_c$ with curvature $-c$, where $c > 0$. The Euclidean representations are mapped to the hyperbolic manifold using the exponential map at the origin:

$$\exp_0^c(u) = \tanh(\sqrt{c}\|u\|)\frac{u}{\sqrt{c}\|u\|}.$$

This yields hyperbolic speech representations $h_w = \exp_0^c(\tilde{e}_w)$ and $h_t = \exp_0^c(\tilde{e}_t)$ with $h_w, h_t \in \mathbb{H}^d_c$. We then proceed on to fusing the semantic and paralinguistic hyperbolic speech representations. To align semantic and paralinguistic speech representations, we minimize the Bhattacharyya distance (BD) (BD has shown effectiveness in aligning speech representations (Phukan et al., 2025b)), however, it was for euclidean space and we extend it to hyperbolic space in this work) between the corresponding hyperbolic distributions. Lower BD represents greater alignment and so we aim to optimize it to minima. For two feature distributions P and Q on $\mathbb{H}^d_c$, the BD is defined as

$$D_B(P, Q) = -\log \int_{\mathbb{H}^d_c} \sqrt{p(h)\, q(h)}\, d\mu_c(h).$$

The speech–speech alignment loss is given by $\mathcal{L}_{S\text{-}S} = D_B(h_w, h_t)$. After aligning the two speech views, we obtain a single fused speech representation using mobius addition, which preserves the geometry of hyperbolic space. For two points $x, y \in \mathbb{H}^d_c$, mobius addition is defined as

$$x \oplus_c y = \frac{(1 + 2c\langle x, y\rangle + c\|y\|^2)x + (1 - c\|x\|^2)y}{1 + 2c\langle x, y\rangle + c^2\|x\|^2\|y\|^2}.$$

Using this operation, we compute the fused speech embedding $h_f = h_w \oplus_c h_t$. We feed a conditioning prompt (*"Analyze the speech for unnatural artifacts"*) to emphasize on unnatural artifacts relevant for synthetic speech. Using Qwen-2, we extract hidden states from an intermediate transformer layer. A single prompt representation e_A is obtained via mean pooling over tokens and projected into the shared space. To inject task-level reasoning, we align the fused speech distribution with the prompt representation using the same BD after we transform the prompt representation to hyperbolic space using exponential map at the origin. $\mathcal{L}_{S\text{-}T} = D_B(h_f, h_A)$. Following this, we aggregate the aligned hyperbolic representations using mobius addition: $h_{\text{final}} = h_f \oplus_c h_A$. The aggregated representation is mapped back to Euclidean space using the logarithmic map, $u_{\text{final}} = \log_0^c(h_{\text{final}})$ and $g = W_g u_{\text{final}}$ where g is injected as prefix conditioning tokens into a Qwen-2 LLM decoder. Logarithmic map is given by:

$$\log_0^c(h) = \frac{1}{\sqrt{c}} \tanh^{-1}(\sqrt{c}\|h\|)\frac{h}{\|h\|}$$

We keep the LLM decoder frozen. A separate decision prompt, (*Determine whether the speech is real or fake. Answer only in one word: "Real" or*

[6] https://huggingface.co/Qwen/Qwen-7B

Table 1: Zero-shot evaluation of Qwen2-audio family on ICF and Codecfake (Wu et al., 2024b); ACC stands for Accuracy; All the scores are in %

ICF												
Method	**Prompt1**		**Prompt2**		**Prompt3**		**Prompt4**		**Prompt5**		**Prompt 6**	
	ACC ↑	**EER ↓**	**ACC ↑**	**EER ↓**	**ACC ↑**	**EER ↓**	**ACC ↑**	**EER ↓**	**ACC ↑**	**EER ↓**	**ACC ↑**	**EER ↓**
Qwen2 audio-chat	11.29	89.43	11.70	89.26	12.05	88.95	11.04	89.60	10.97	90.02	11.41	89.45
Qwen2 audio-base	11.20	88.99	13.35	88.66	**13.41**	**88.57**	12.68	89.13	12.93	89.74	12.71	89.02
CodecFake												
Qwen2 audio-chat	15.92	83.42	16.24	82.91	16.74	82.33	15.38	83.91	15.04	84.23	15.86	83.36
Qwen2 audio-base	15.43	83.79	16.43	82.28	**17.91**	**81.26**	16.87	81.59	16.29	81.86	16.59	82.16

"Fake"), is provided to the decoder, which generates the output sequence Y. This decision prompt is selected based on initial experiments (Section 5 Experimental Results: Zero-shot evaluation of ALMs) across different prompt templates. The decoder output is constrained to either *Real* or *Fake*. The language modeling loss is defined as

$$\mathcal{L}_{LM} = -\sum_{t=1}^{L} \log p(y_t \mid y_{<t}, g, P_B).$$

The complete training objective is

$$\mathcal{L} = \lambda_1 \mathcal{L}_{S\text{-}S} + \lambda_2 \mathcal{L}_{S\text{-}T} + \lambda_3 \mathcal{L}_{LM},$$

where λ_1, λ_2, and λ_3 control the contributions of speech fusion, prompt conditioning, and language generation, respectively. Also, the alignment losses are transformed back to Euclidean space and then added together with the training objective for joint optimization. As the BD losses are loss functions and don't generally add much overhead on parameters. So, the total trainable parameters is approximately 3.75M.

5 Experiments

5.1 Training Details and Hyperparameters

SATYAM is trained in a supervised manner on the ICF training split with only lightweight CNN layers, projection layers, and hyperbolic alignment modules are optimized. Training is performed using AdamW with a learning rate of 1×10^{-4}, batch size of 32, for 5 epochs. We keep the values of control parameters of the training objective λ_1, λ_2, λ_3 to be 1, 0.5, and 1 respectively after initial experimentation on the validation set of ICF. More details are given in Appendix 7 Additional Training Details.

5.2 Experimental Results

We use Accuracy (ACC) and EER as the evaluation metrics for our experiments as used by previous research on speech deepfake detection incudling CF detection (Phukan et al., 2024a; Wu et al., 2024b; Lu et al., 2024b). All the experiments with ICF has been carried out on CFs generated using test-known set of Indic-SUPERB. Only the experiments for **Unseen-codec evaluation (clean and noisy test-unknown)** below, we perform on CFs generated using test-unknown (Second evaluation setting as mentioned in Section 3 Generation Pipeline of ICF).

Training on previous benchmark CF dataset and testing on ICF: We train AASIST on CodecFake (Wu et al., 2024b) and then evaluate it zero-shot on ICF. While AASIST performs strongly in-domain on CodecFake (94.21% ACC / 10.13% EER, Table 2), its performance drops sharply on ICF to 48.0% ACC and 40.32% EER. This large degradation indicates a substantial distribution shift and poor generalization from English-centric CodecFake conditions to Indic scenarios.

Zero-shot Evaluation of ALMs on CF detection: Gu et al. (Gu et al., 2025) carried out the first study of evaluating ALMs for speech deepfake detection excluding CF detection. They showed the Qwen2-audio ALMs generally perform better than other ALMs. So, we also carried out zero-shot evaluation of Qwen2-audio family on ICF and Codecfake (Wu et al., 2024a) datasets. The results are presented in Table 1. Our results shows that Qwen2-audio-base generally performns better in both datasets as well under different prompt templates and the results with Prompt3 (Used for **SATYAM** Section 4) being the best. The prompt templates are given in Appendix 7 Table 4. Our results reciporcate with the results obtained by Gu et al. (Gu et al., 2025) with Qwen2-audio-base being the best in zero-shot manner. We also carried

out further zero-shot analysis of ALMs with different SOTA ALMs such Pengi (Deshmukh et al., 2023), Audio Flamingo 2 (Ghosh et al.), Audio Flamingo 3 (Ghosh et al.), Qwen-audio-chat, and lastly Qwen-audio-base (Chu et al., 2023). We carried out these experiments with Prompt 3. The results are presented in Table 2 Zero-shot evaluation of ALMs. We observe that Pengi performed the worst among all the ALMs and this can be due its lack in speech-specific pre-training done. While Qwen2-audio-base being the topmost, however, the results are the overall poor with very less accuracy and high EER. This calls for the need of CF specific training. Also, we believe we are the first study, to the best of our knowledge, to carry out evaluation of ALMs for CF detection.

Method	ICF		CF	
	ACC ↑	EER ↓	ACC ↑	EER ↓
Zero-shot evalution of ALMs				
Pengi	3.19	98.26	5.68	94.97
Audioflamingo 2	5.42	97.68	8.41	92.10
Audioflamingo 3	6.98	97.21	10.22	90.85
Qwen-audio-chat	10.63	89.71	13.00	86.61
Qwen-audio-base	11.17	89.23	15.82	85.74
Qwen2 audio-chat	12.05	88.95	16.74	82.33
Qwen2 audio-base	13.41	88.57	17.91	81.26
End-to-end method				
AASIST	90.60	12.47	94.21	10.13
Pre-Trained Backbone				
W-LCCN	91.98	11.89	93.38	7.92
Wav2vec2-AASIST	92.50	9.62	94.45	7.29
W + T (LF)	92.60	9.53	94.83	7.19
W + T (CA)	92.65	9.48	94.99	7.01
MiO	92.80	9.04	95.11	6.49
Fine-Tuning ALM				
Qwen2 audio-base	93.19	8.34	95.55	5.60
Our Approaches				
W + Qwen2-7B	92.98	8.61	94.64	6.02
T + Qwen2-7B	93.21	8.09	95.10	5.83
W + T + Qwen2-7B (C)	93.28	7.94	95.75	4.39
W + T + Qwen2-7B(MA)	94.01	7.02	95.31	4.07
W + T + Qwen2-7B (E-BD)	94.93	5.39	96.47	3.68
W + T + Qwen2-7B (H-BD-ST)	95.78	5.14	97.22	2.69
W + T + Qwen2-7B (H-BD-SS)	96.11	5.02	97.34	2.42
SATYAM	**98.32**	**3.27**	**99.11**	**1.94**
SATYAM with Qwen2-1.8B	97.14	4.53	98.32	2.11

Table 2: Evaluation Results on ICF and Codecfake (CF) (Wu et al., 2024b); ACC stands for accuracy. Blue cells indicate the best (ACC/EER), while yellow cells indicate the second highest results. Scores are in %; W, T stands for Whisper and TRILLsson respectively; CA, LF stands for Cross-attention and Late-Fusion respectively

In-domain Training and Evaluation: Table 2 reports the results obtained by training and evaluating on the ICF dataset. We consider AASIST (Wu et al., 2024b), Wav2vec2-AASIST (Lu et al., 2024b), Whisper-LCNN (Kawa et al., 2023), and MiO (Phukan et al., 2024b) as traditional classifier-based baselines. The corresponding training details are provided in the Appendix 7 Additional Training Details. We also add supervised fusion baselines with the audio encoders considered in our study for **SATYAM**, cross-attention between Whisper and TRILLsson representations (W+T (CA)) and late fusion (W+T (LF)). Both approaches use the same CNN projection module as in **SATYAM**, whose downstream effectiveness has been shown in prior work (Phukan et al., 2024a). In late fusion, encoder representations are first processed through the CNN projection module to produce class-level logits, followed by a projection layer and final classification layer as in Sharma et al. (Sharma et al., 2023). For all supervised baselines, the audio encoders were kept frozen and the models were trained using the same configuration as MiO (Phukan et al., 2024a), a representative encoder fusion framework for speech deepfake detection. Among these approaches, MiO achieves the best performance, highlighting the effectiveness of multi-encoder fusion for CF detection. This observation is consistent with prior findings, where MiO demonstrated SOTA performance for vocoder-based synthetic speech detection using multi-encoder fusion. Further, we add a fine-tuning of ALM baseline (Fine-Tuning ALM). We fine-tune the projection layer of Qwen2-Audio-base (Qwen2-Audio-base uses Whisper audio encoder + Qwen2-7B LLM decoder), selected because it achieved the strongest zero-shot performance in our experiments and has demonstrated effectiveness in prior work (Gu et al., 2025). For a fair comparison with **SATYAM**, we followed the same training protocol: both the audio encoders and the LLM decoder were kept frozen, and only the projection layers were trained using identical training settings.

We next evaluate our proposed framework, **SATYAM**, which achieves the best overall performance on the ICF dataset, demonstrating its effectiveness. Fine-tuning Qwen2-Audio-base improves performance over the W+Qwen2-7B setup (which shares the same architecture), likely due to prior exposure to large-scale audio during pretraining; however, **SATYAM** still achieves the best performance. To better understand the contribution of each component, we further conduct a series of ab-

lation studies with **SATYAM**, as reported under *Our Approaches* in Table 2. All ALM-based variants employ the same conditioning prompt and follow the same training protocol as the full **SATYAM** model. We first experiment with single-encoder configurations (*W + Qwen2-7B* and *T + Qwen2-7B*). Among these, TRILLsson-based model perform better, reflecting the predominantly paralinguistic nature of speech deepfake detection. We then explore simple concatenation in Euclidean space for both speech–speech and speech–prompt fusion (*W + T + Qwen2-7B (C)*). This is followed by geometry-aware fusion using mobius addition after mapping representations to hyperbolic space in both fusion stages (*W + T + Qwen2-7B (MA)*). Next, we evaluate BD–based alignment in euclidean space for both fusion stages (*W + T + Qwen2-7B (E-BD)*), following prior work (Phukan et al., 2025b). Finally, we disentangle the role of hyperbolic alignment by applying BD only to speech–speech fusion (*H-BD-SS*) and, in contrast, only to speech–prompt fusion (*H-BD-ST*). These ablations collectively highlight the complementary benefits of hyperbolic geometry and dual-stage alignment in **SATYAM**. Furthermore, we replace the original decoder with a lightweight LLM, Qwen2-1.8B[7], and observe a slight decrease in performance compared to the full **SATYAM** configuration using Qwen2-7B. Nevertheless, this lightweight variant still substantially outperforms the single-encoder configurations with Qwen2-7B, indicating that the quality of the audio encoders constitutes the primary performance bottleneck, consistent with the findings of Li et al. (Li et al., 2025). Importantly, these results demonstrate that competitive performance can be achieved even without fine-tuning the LLM decoder, further validating the effectiveness and efficiency of the proposed framework. We also present the results of **SATYAM** across individual Indic-languages in Appendix 7 Table 3 showing consistent performance across all the languages. Furthermore, we carried out statistical significance test on our results in Appendix 7 Statistical Significance and thus supporting the statistical validation of our obtained results. Similar performance is observed acorss Codecfake (Wu et al., 2024b) too, with **SATYAM** performing the best. Further, we conduct a prompt analysis of **SATYAM**, including its lightweight variant **SATYAM** with Qwen2-1.8B, which achieves the second-best overall performance. The results are presented in Appendix 7 Prompt Analysis of **SATYAM**. We observe with varied prompts also through usage of conditioning prompt with **SATYAM**, we are getting better performance than without usage of conditioning prompt.

Comparison of SOTA models and SATYAM on previous CF benchmark: We evaluate performance on prior CF detection benchmark, CodecFake by Wu et al. (Wu et al., 2024b). We perform two types of evaluation: (i) in-domain training and testing on CodecFake, and (ii) cross-evaluation between ICF and CodecFake. As a baseline, we consider AASIST as it shown its effectiveness in CF detection (Wu et al., 2024b; Lu et al., 2024a). In the in-domain setup, **SATYAM** outperforms the standard AASIST baseline by a large margin (Table 2). For cross-benchmark transfer, **SATYAM** remains robust in both directions, achieving low EER when trained on ICF and tested on CodecFake (3.79% EER), and when trained on CodecFake and tested on ICF (7.43% EER). In contrast, AASIST exhibits substantial degradation under the same distribution shifts, reaching 29.81% EER for ICF→CodecFake transfer and 40.32% EER for CodecFake→ICF transfer.

Language Family: We further analyze generalization across languages by evaluating **SATYAM** under (i) random cross-lingual splits and (ii) language-family transfer between the two dominant families in our dataset, namely Dravidian and Indo-European (Appendix 7 IndicSUPERB shows the Dravidian and Indo-European languages). Under the random cross-lingual setting, where the model is trained on six randomly selected languages and evaluated on the held-out remaining language, **SATYAM** maintains low error in both directions (6.34% and 7.09% EER). In contrast, AASIST showed much higher EER (26.74% and 31.11%). For language-family transfer, we train on Dravidian languages and evaluate on Indo-European languages, and vice versa. **SATYAM** remains stable under this structured distribution shift, obtaining 7.78% EER for Dravidian→Indo-European transfer and 8.48% EER for Indo-European→Dravidian transfer. For AASIST, for Indo-European (Train)-> Dravidian (Test), we got EER of 33.45% and vice-versa, we got EER of 38.73%. Overall, these results indicate that **SATYAM** generalizes effectively across unseen languages and even across language families.

Unseen-codec evaluation (clean and noisy test-unknown): To assess robustness under unseen

[7] https://huggingface.co/Qwen/Qwen-1_8B

generation conditions, we evaluate on two held-out `test-unknown` splits: a clean split and a noisy split. The clean split consists of CFs generated using NACs that are unseen during training, as defined in the second evaluation setting in Section 3 Generation Pipeline of ICF. The noisy split is constructed by generating CFs from the `test-unknown (noisy)` portion of IndicSUPERB using the same set of unseen NACs as in the clean split. Across both unseen splits, **SATYAM** remains reliable, with only a moderate performance degradation when moving from clean to noisy conditions, achieving an EER of 5.23% on the clean unseen split and 7.41% on the noisy unseen split. In contrast, AASIST performs substantially worse under the same conditions, with EERs of 14.38% and 16.29% on the clean and noisy unseen splits, respectively. These results highlight the robustness of **SATYAM** to both codec mismatch and adverse acoustic conditions.

Inference: **SATYAM** introduces only lightweight alignment operations (3.75M parameters) while keeping both audio encoders and the LLM frozen, so inference is dominated by a single backbone forward pass and the hyperbolic mappings add negligible overhead. In practice, W + Qwen2-7B takes 8.00 s on a single-core A100, **SATYAM** takes 8.18 s, and **SATYAM** with the lighter decoder takes 6.53 s (averaged over the ICF test set). Notably, **SATYAM** with the lighter decoder (Qwen2-1.8B) achieves better performance than W + Qwen2-7B, despite sharing a similar architectural (language model decoder) backbone (Qwen2-Audio), which has shown strong performance for speech deepfake detection (Gu et al., 2025).

6 Conclusion

In this work, we introduced ICF, the first large-scale benchmark comprising real and NAC-synthesized speech across multiple Indic languages, diverse speaker profiles, and multiple NAC types. Our experiments show that SOTA CF detectors trained on English-centric datasets perform poorly on ICF underscoring the effect of changes in linguistic distribution. We further conducted a systematic zero-shot evaluation of SOTA ALMs on ICF, revealing consistent performance degradation despite their effectiveness on other speech tasks. To overcome these limitations, we proposed **SATYAM**, a hyperbolic ALM tailored for CF detection in Indic languages. By leveraging dual-stage hyperbolic Bhattacharya distance to align semantic and prosodic speech representations and subsequently integrate speech and textual prompts, **SATYAM** effectively models hierarchical relationships within and across modalities. Extensive evaluations demonstrate that **SATYAM** consistently outperforms competitive end-to-end and ALLM-based baselines on the ICF benchmark.

Limitations

One limitation of our work is that we consider a single LLM decoder family. However, prior studies have shown that the choice of LLM decoder has a relatively limited impact on performance in audio LLM pipelines (Li et al., 2025), a trend that is also reflected in our experimental results. In addition, our framework employs two audio encoders; while both are SOTA and well-suited for CF detection, alternative encoder choices may lead to minor performance variations. In future work, we plan to explore a broader range of LLM decoder architectures as well as additional audio encoders to further assess the generality of the proposed framework.

Ethical considerations

This work is motivated by the need to improve the robustness of CF detection in low-resource and multilingual settings. We do not collect any new human-subject recordings; the ICF dataset is constructed by applying NACs to the openly available IndicSUPERB corpus, in accordance with its original licenses and usage terms. Although our contributions are defensive in nature, we acknowledge that insights into codec artifacts and detector behavior could potentially be misused. We strongly discourage any unethical or malicious use of the ICF dataset. The benchmark and models introduced in this work are intended strictly for research purposes.

References

Arnesh Batra, Dev Sharma, Krish Thukral, Ruhani Bhatia, Naman Batra, and Aditya Gautam. 2025. Melody or machine: Detecting synthetic music with dual-stream contrastive learning. *arXiv preprint arXiv:2512.00621*.

Zalán Borsos, Raphaël Marinier, Damien Vincent, and 1 others. 2023. Audiolm: A language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2523–2537.

Weidong Chen, Xiaofen Xing, Xiangmin Xu, Jianxin Pang, and Lan Du. 2023. Speechformer++: A hierarchical efficient framework for paralinguistic speech processing. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:775–788.

Xuanjun Chen, Jiawei Du, Haibin Wu, Lin Zhang, I Lin, I Chiu, Wenze Ren, Yuan Tseng, Yu Tsao, Jyh-Shing Roger Jang, and 1 others. 2025. Codecfake+: A large-scale neural audio codec-based deepfake speech dataset. *arXiv preprint arXiv:2501.08238*.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, and 1 others. 2024. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*.

Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. 2023. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. *arXiv preprint arXiv:2311.07919*.

Jianqiao Cui, Bingyao Yu, Qihao Wang, Fei Meng, and Jiwen Lu. 2025. Whiadd: Semantic-acoustic fusion for robust audio deepfake detection. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 11610–11618.

Arnab Das, Yassine El Kheir, Carlos Franzreb, Tim Herzig, Tim Polzehl, and Sebastian Möller. 2025. Generalizable Audio Spoofing Detection using Non-Semantic Representations. In *Interspeech 2025*, pages 4553–4557.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. 2022. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. 2024. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*.

Karan Desai, Maximilian Nickel, Tanmay Rajpurohit, Justin Johnson, and Shanmukha Ramakrishna Vedantam. 2023. Hyperbolic image-text representations. In *International Conference on Machine Learning*, pages 7694–7731. PMLR.

Soham Deshmukh, Benjamin Elizalde, Rita Singh, and Huaming Wang. 2023. Pengi: An audio language model for audio tasks. *Advances in Neural Information Processing Systems*, 36:18090–18108.

Heinrich Dinkel, Nanxin Chen, Yanmin Qian, and Kai Yu. 2017. End-to-end spoofing detection with raw waveform cldnns. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4860–4864. IEEE.

Zhihao Du, Shiliang Zhang, Kai Hu, and Siqi Zheng. 2024. Funcodec: A fundamental, reproducible and integrable open-source toolkit for neural speech codec. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 591–595. IEEE.

Yassine El Kheir, Younes Samih, Suraj Maharjan, Tim Polzehl, and Sebastian Möller. 2025. Comprehensive layer-wise analysis of ssl models for audio deepfake detection. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4070–4082.

Sreyan Ghosh, Zhifeng Kong, Sonal Kumar, S Sakshi, Jaehyeon Kim, Wei Ping, Rafael Valle, Dinesh Manocha, and Bryan Catanzaro. Audio flamingo 2: An audio-language model with long-audio understanding and expert reasoning abilities. In *Forty-second International Conference on Machine Learning*.

Hao Gu, Jiangyan Yi, Chenglong Wang, Jianhua Tao, Zheng Lian, Jiayi He, Yong Ren, Yujie Chen, and Zhengqi Wen. 2025. Allm4add: Unlocking the capabilities of audio large language models for audio deepfake detection. In *Proceedings of the 33rd ACM International Conference on Multimedia*, MM '25, page 11736–11745, New York, NY, USA. Association for Computing Machinery.

Jie Hong, Zeeshan Hayder, Junlin Han, Pengfei Fang, Mehrtash Harandi, and Lars Petersson. 2023. Hyperbolic audio-visual zero-shot learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7873–7883.

Wen Huang, Yanmei Gu, Zhiming Wang, Huijia Zhu, and Yanmin Qian. 2025. SpeechFake: A large-scale multilingual speech deepfake dataset incorporating cutting-edge generation methods. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9985–9998, Vienna, Austria. Association for Computational Linguistics.

Tahir Javed, Kaushal Santosh Bhogale, Abhigyan Raman, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M. Khapra. 2023. IndicSUPERB: A speech processing universal performance benchmark for Indian languages. In *Proc. AAAI*, volume 37, pages 13043–13051.

Zhe Ji, Zhi-Yi Li, Peng Li, Maobo An, Shengxiang Gao, Dan Wu, and Faru Zhao. 2017. Ensemble learning for countermeasure of audio replay spoofing attack in asvspoof2017. In *Interspeech*, pages 87–91.

Jee-weon Jung, Hee-Soo Heo, Hemlata Tak, Hye-jin Shim, Joon Son Chung, Bong-Jin Lee, Ha-Jin Yu, and Nicholas Evans. 2022. Aasist: Audio anti-spoofing using integrated spectro-temporal graph attention networks. In *ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 6367–6371. IEEE.

Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo. 2019. StarGAN-VC2: Rethinking conditional methods for StarGAN-based voice conversion. In *Proc. Interspeech*, pages 679–683.

Piotr Kawa, Marcin Plata, Michał Czuba, Piotr Szymański, and Piotr Syga. 2023. Improved deepfake detection using whisper features. In *Interspeech 2023*, pages 4009–4013.

Zahra Khanjani, Gabrielle Watson, and Vandana P. Janeja. 2022. Audio deepfakes: A survey. *Frontiers in Big Data*, 5:1001063.

Jaehyeon Kim, Jungil Kong, and Juhee Son. 2021. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *Proc. ICML*, pages 5530–5540.

Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. 2024. High-fidelity audio compression with improved rvqgan. *Advances in Neural Information Processing Systems*, 36.

Zhenchun Lei, Yingen Yang, Changhong Liu, and Jihua Ye. 2020. Siamese convolutional neural network using gaussian probability feature for spoofing speech detection. In *Interspeech*, pages 1116–1120.

Yupei Li, Li Wang, Yuxiang Wang, Lei Wang, Rizhao Cai, Jie Shi, Björn W Schuller, and Zhizheng Wu. 2025. Dfallm: Achieving generalizable multitask deepfake detection by optimizing audio llm components. *arXiv preprint arXiv:2512.08403*.

Xuechen Liu, Nicholas Evans, Massimiliano Todisco, and 1 others. 2023. Asvspoof 2021: Towards spoofed and deepfake speech detection in the wild. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Yi Lu, Yuankun Xie, Ruibo Fu, Zhengqi Wen, Jianhua Tao, Zhiyong Wang, Xin Qi, Xuefei Liu, Yongwei Li, Yukun Liu, Xiaopeng Wang, and Shuchen Shi. 2024a. Codecfake: An initial dataset for detecting llm-based deepfake audio. In *Interspeech 2024*, pages 1390–1394.

Yi Lu, Yuankun Xie, Ruibo Fu, Zhengqi Wen, and 1 others. 2024b. Codecfake: An initial dataset for detecting LLM-based deepfake audio. In *Proc. Interspeech*.

Hieu-Thi Luong, Haoyang Li, Lin Zhang, Kong Aik Lee, and Eng Siong Chng. 2025. Llamapartialspoof: An llm-driven fake speech dataset simulating disinformation generation. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Jean Mary Zarate, Xing Tian, Kevin JP Woods, and David Poeppel. 2015. Multiple levels of linguistic and paralinguistic features contribute to voice recognition. *Scientific reports*, 5(1):11475.

Nicolas M Müller, Piotr Kawa, Wei Herng Choong, Edresson Casanova, Eren Gölge, Thorsten Müller, Piotr Syga, Philip Sperl, and Konstantin Böttinger. 2024. Mlaad: The multi-language audio anti-spoofing dataset. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE.

Tanvina B Patel and Hemant A Patil. 2015. Combining evidences from mel cepstral, cochlear filter cepstral and instantaneous frequency features for detection of natural vs. spoofed speech. In *Interspeech*, pages 2062–2066.

Tanvina B Patel and Hemant A Patil. 2016. Cochlear filter and instantaneous frequency based features for spoofed speech detection. *IEEE Journal of Selected Topics in Signal Processing*, 11(4):618–631.

Orchid Chetia Phukan, Girish, Mohd Mujtaba Akhtar, Swarup Ranjan Behera, Pailla Balakrishna Reddy, Arun Balaji Buduru, and Rajesh Sharma. 2025a. HYFuse: Aligning Heterogeneous Speech Pre-Trained Representations in Hyperbolic Space for Speech Emotion Recognition. In *Interspeech 2025*, pages 131–135.

Orchid Chetia Phukan, Girish, Mohd Mujtaba Akhtar, Swarup Ranjan Behera, Pailla Balakrishna Reddy, Arun Balaji Buduru, and Rajesh Sharma. 2025b. Investigating the Reasonable Effectiveness of Speaker Pre-Trained Models and their Synergistic Power for SingMOS Prediction. In *Interspeech 2025*, pages 3090–3094.

Orchid Chetia Phukan, Gautam Kashyap, Arun Balaji Buduru, and Rajesh Sharma. 2024a. Heterogeneity over homogeneity: Investigating multilingual speech pre-trained models for detecting audio deepfake. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2496–2506.

Orchid Chetia Phukan, Gautam Siddharth Kashyap, Arun Balaji Buduru, and Rajesh Sharma. 2024b. Heterogeneity over homogeneity: Investigating multilingual speech pre-trained models for detecting audio deepfake. *arXiv preprint arXiv:2404.00809*.

Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, and Mark Hasegawa-Johnson. 2019. Autovc: Zero-shot voice style transfer with only autoencoder loss. In *Proc. ICML*, pages 5210–5219.

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

Rishabh Ranjan, Mayank Vatsa, and Richa Singh. 2025. Indicfake meets SAFARI-LLM: Unifying semantic and acoustic intelligence for multilingual deepfake detection. *Transactions on Machine Learning Research*.

Divya V Sharma, Vijval Ekbote, and Anubha Gupta. 2025. Indicsynth: A large-scale multilingual synthetic speech dataset for low-resource indian languages. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22037–22060.

Gagan Sharma, R Chinmay, and Raksha Sharma. 2023. Late fusion of transformers for sentiment analysis of

code-switched data. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6485–6490.

Joel Shor and Subhashini Venugopalan. 2022. TRILLsson: Distilled universal paralinguistic speech representations. In *Proc. Interspeech*, pages 356–360.

Hubert Siuzdak, Florian Grötschla, and Luca A Lanzendörfer. 2024. Snac: Multi-scale neural audio codec. In *Audio Imagination: NeurIPS 2024 Workshop AI-Driven Speech, Music, and Sound Generation*.

Bo-Hao Su, Hui-Ying Shih, Jinchuan Tian, Jiatong Shi, Chi-Chun Lee, Carlos Busso, and Shinji Watanabe. 2025. Reasoning beyond majority vote: An explainable speechlm framework for speech emotion recognition. *arXiv preprint arXiv:2509.24187*.

Xu Tan, Tao Qin, Frank Soong, and Tie-Yan Liu. 2021. A survey on neural speech synthesis. *arXiv preprint arXiv:2106.15561*.

Qwen Team and 1 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2(3).

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. In *9th ISCA Workshop on Speech Synthesis Workshop (SSW 9)*, page 125.

Xin Wang, Héctor Delgado, Hemlata Tak, Jee-weon Jung, Hye-jin Shim, Massimiliano Todisco, Ivan Kukanov, Xuechen Liu, Md Sahidullah, Tomi Kinnunen, and 1 others. 2024. Asvspoof 5: Crowdsourced speech data, deepfakes, and adversarial attacks at scale. *arXiv preprint arXiv:2408.08739*.

Xin Wang, Junichi Yamagishi, and 1 others. 2020. Asvspoof 2019: A large-scale public database of spoofed and fake audio. *Computer Speech & Language*, 64:101114.

Haibin Wu, Yuan Tseng, and Hung-yi Lee. 2024a. Codecfake: Enhancing anti-spoofing models against deepfake audios from codec-based speech synthesis systems. In *Proc. Interspeech*.

Haibin Wu, Yuan Tseng, and Hung yi Lee. 2024b. Codecfake: Enhancing anti-spoofing models against deepfake audios from codec-based speech synthesis systems. In *Interspeech 2024*, pages 1770–1774.

Yi-Chiao Wu, Israel D Gebru, Dejan Marković, and Alexander Richard. 2023. Audiodec: An open-source streaming high-fidelity neural audio codec. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Table 3: Language-wise EER (%)

#	Language	EER
1	Bengali	2.95
2	Gujarati	2.98
3	Kannada	2.69
4	Hindi	2.34
5	Malayalam	3.64
6	Marathi	2.52
7	Odia	2.85
8	Punjabi	3.16
9	Sanskrit	3.41
10	Tamil	4.11
11	Telugu	4.01
12	Urdu	3.38

Zhizheng Wu, Tomi Kinnunen, Nicholas Evans, Junichi Yamagishi, Cemal Hanilçi, Md Sahidullah, and Aleksandr Sizov. 2015. Asvspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge. In *INTERSPEECH 2015 16th Annual Conference of the International Speech Communication Association*, pages 2037–2041. International Speech Communication Association.

Yuankun Xie, Yi Lu, Ruibo Fu, Zhengqi Wen, Zhiyong Wang, Jianhua Tao, Xin Qi, Xiaopeng Wang, Yukun Liu, Haonan Cheng, and 1 others. 2025. The codecfake dataset and countermeasures for the universally detection of deepfake audio. *IEEE Transactions on Audio, Speech and Language Processing*.

Hong Yu, Zheng-Hua Tan, Zhanyu Ma, Rainer Martin, and Jun Guo. 2017. Spoofing detection in automatic speaker verification systems using dnn classifiers and dynamic acoustic features. *IEEE transactions on neural networks and learning systems*, 29(10):4633–4644.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. 2021. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507.

Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. 2024. Speechtokenizer: Unified speech tokenizer for speech language models. In *The Twelfth International Conference on Learning Representations*.

7 Appendix

7.1 IndicSUPERB

IndicSUPERB consists of languages: Bengali, Gujarati, Hindi, Kannada, Malayalam, Marathi, Odia, Punjabi, Sanskrit, Tamil, Telugu, and Urdu. Kannada, Malayalam, Tamil, and Telugu falls under

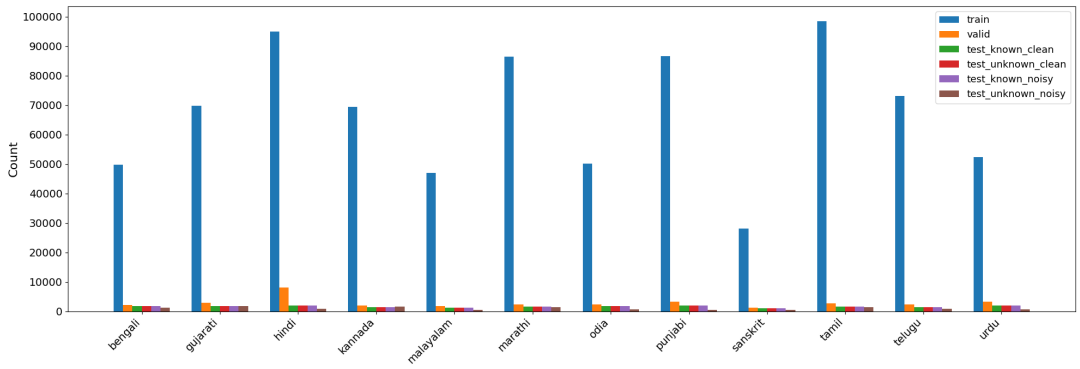

Figure 3: IndicSUPERB data distribution across Train, Val, and Test sets for different Indic languages

- Prompt 1: *Is this speech real or fake? Reply with one word only: "Real" or "Fake".*
- Prompt 2: *What is the authenticity of this speech? Answer "Fake" or "Real".*
- Prompt 4:*Can you determine if this speech is fake or real? Answer "Fake" or "Real".*
- Prompt 5:*Is this a real speech recording? Answer "Fake" or "Real".*
- Prompt 6:*Is this a AI-generated speech sample? Answer "Fake" or "Real".*

Table 4: Decision Prompt Templates

Dravidian language family whereas the others languages falls under Indo-European language family. The distribution plot of IndicSUPERB across train, val, and test splits is given in Figure 3.

7.2 More Information about NACs

DAC[8]: It is high-fidelity VQ-GAN codec with RVQ and a mix of adversarial and multi-scale spectral losses to improve reconstruction.

Encodec[9]: It is a real-time, high-fidelity NAC with a convolutional encoder–decoder and RVQ. It combines time- and frequency-domain losses with spectrogram-based adversarial training.

SoundStream[10]: A low-bitrate RVQ codec with multi-scale STFT discriminators, designed to balance fidelity and compression. It supports 3–18 kbps.

SpeechTokenizer[11]: It is a unified tokenizer that bridges semantic and acoustic cues via hierarchical RVQ layers.

FunCodec[12]: It is RVQ-based NAC enhanced with semantic augmentation and adversarial strategies.

AudioDec[13]: It high-fidelity NAC trained in two stages: metric losses for stability, followed by decoder-only adversarial finetuning for realism.

SNAC[14]: It is an RVQ extension with hierarchical quantizers at multiple time scales.

MIMI[15]: It is a encoder-decoder based high fidelity NAC with quantization trained in a end-to-end manner.

7.3 More Information about the Audio Encoders

Whisper [16] is a transformer-based encoder–decoder architecture trained on 96 languages in a multi-task setting. We use only the encoder to extract representations, obtaining a 512-dimensional embedding after average pooling. Since Whisper is primarily trained for automatic speech recognition, it effectively captures semantic and linguistic information in speech. Following this, we employ TRILLsson[17], a distilled self-supervised model pretrained for paralinguistic speech processing. TRILLsson has demonstrated strong performance in tasks such as speech emotion recognition, speaker identification, and speech deepfake detection. We extract a 1024-dimensional representation from TRILLsson. Both Whisper and TRILLsson are kept frozen dur-

[8] https://huggingface.co/descript/dac_16khz
[9] https://huggingface.co/facebook/encodec_24khz
[10] https://github.com/haydenshively/SoundStream
[11] https://github.com/ZhangXInFD/SpeechTokenizer.git
[12] https://github.com/modelscope/FunCodec
[13] https://github.com/facebookresearch/AudioDec
[14] https://huggingface.co/hubertsiuzdak/snac_44khz
[15] https://huggingface.co/kyutai/mimi
[16] https://huggingface.co/openai/whisper-base
[17] https://www.kaggle.com/models/google/trillsson

ICF												
Prompt	**Prompt1**		**Prompt2**		**Prompt3**		**Prompt4**		**Prompt5**		**Prompt6**	
	ACC ↑	**EER ↓**	**ACC ↑**	**EER ↓**	**ACC ↑**	**EER ↓**	**ACC ↑**	**EER ↓**	**ACC ↑**	**EER ↓**	**ACC ↑**	**EER ↓**
SATYAM with Qwen2-1.8B	88.13	9.63	88.29	9.36	88.87	8.18	88.85	9.04	87.52	9.49	88.33	9.14
SATYAM	92.00	7.10	90.08	7.37	91.35	7.08	92.46	6.39	89.75	7.95	91.13	7.18
CodecFake (Wu et al., 2024b)												
SATYAM with Qwen2-1.8B	90.54	6.42	90.42	5.93	92.01	5.61	90.06	5.72	89.75	6.32	90.56	6.08
SATYAM	92.00	5.67	92.05	5.38	93.83	5.27	92.64	5.18	92.49	5.87	92.60	5.47

Table 5: Evaluation scores with different prompt templates and without the conditioning prompt; ACC stands for Accuracy; All the scores are in %

ICF												
Prompt	**Prompt1**		**Prompt2**		**Prompt3**		**Prompt4**		**Prompt5**		**Prompt6**	
	ACC ↑	**EER ↓**	**ACC ↑**	**EER ↓**	**ACC ↑**	**EER ↓**	**ACC ↑**	**EER ↓**	**ACC ↑**	**EER ↓**	**ACC ↑**	**EER ↓**
SATYAM with Qwen2-1.8B	94.87	5.87	94.63	6.16	97.14	4.53	95.38	5.29	94.46	5.96	95.02	5.68
SATYAM	97.35	3.62	96.78	3.89	98.32	3.27	97.64	3.35	96.45	3.97	97.31	3.62
CodecFake (Wu et al., 2024b)												
SATYAM with Qwen2-1.8B	97.47	2.54	96.32	2.54	98.32	2.11	96.89	2.64	96.22	2.72	96.62	2.63
SATYAM	97.26	2.21	97.49	2.08	99.11	1.94	98.43	1.99	97.80	2.00	98.02	2.04

Table 6: Evaluation scores with different prompt templates and without the conditioning prompt; The conditioning prompt is kept same for all the different versions of the decision prompt; ACC stands for Accuracy; All the scores are in %

ing training, and all input utterances are resampled to 16 kHz before being passed into the encoders.

7.4 Additional Training Details

We use four-core A100 for training our models. Also, the end-to-end baselines and the baselines with pre-trained backbone are trained for 20 epochs with same learning rate and batch size as of **SATYAM**.

7.5 Prompt analysis of SATYAM

We use the same set of decision prompts employed in the zero-shot evaluation of AMs (prompt templates are provided in the Table 4).Table 5 reports results obtained using only the decision prompts, while Table 6 reports results when both the conditioning prompt and the decision prompt are used. Across both model variants, we observe that Prompt3, which is adopted in the final **SATYAM** configuration, consistently yields the best performance. Moreover, incorporating the conditioning prompt leads to systematically improved results across all prompt templates, highlighting the effectiveness of prompt-conditioned alignment in the proposed framework.

7.6 Statistical Significance

We assess statistical significance using a two-sided McNemar's test on paired predictions from the same test set. McNemar's test is a preferred statistical signifance used in previous deepfake detection work (Batra et al., 2025). SATYAM shows statistically significant improvements over AASIST, the strongest SSL baseline (MiO), and SATYAM with Qwen2-1.8B on ICF ($p < 0.001$ for all comparisons), confirming that the observed gains are statistically significant.