

VaaniCLIP: Cross-Lingual Speech-Text Retrieval for Indic Languages

Vamsi Adari

vamsiadari3@gmail.com

Abstract

Cross-lingual speech-text retrieval for Indic languages (retrieving English text using vernacular voice queries) remains an open problem despite recent progress in multilingual speech and text representation learning. We introduce **VaaniCLIP** and establish a systematic benchmark on BhasaAnuvaad (Sankar et al., 2025) across Telugu, Tamil, and Hindi, demonstrating that lightweight task-specific contrastive alignment achieves strong performance where established baselines fail. Cascade pipelines (Whisper ASR $\rightarrow$ NLLB translation $\rightarrow$ LaBSE retrieval) collapse on low-resource Indic languages: Telugu cascade retrieval reaches only 3.1% R@1, roughly $18\times$ below VaaniCLIP, and SeamlessM4T v2 (2.3B parameters) achieves near-random performance zero-shot. By contrast, VaaniCLIP aligns frozen Whisper and LaBSE encoders with partial Whisper fine-tuning (15M trainable parameters, against a 2.3B-parameter generalist model used zero-shot) and achieves R@1 of 86.4 on Hindi, 55.5 on Telugu, and 54.8 on Tamil at a normalized retrieval pool of $N = 2,869$, capturing 76–89% of the LaBSE text-text upper bound. A frozen Whisper-medium encoder with only 919K trainable parameters reaches 95–99% of per-language partial fine-tuning performance, confirming encoder representation quality as the primary bottleneck. We further show that the joint model transfers zero-shot to Kannada (an unseen Dravidian language), achieving R@1 of 25.1 ($\sim 700\times$ above random) with no Kannada training data, providing evidence that VaaniCLIP learns generalizable Dravidian audio-text alignment.

1 Introduction

Voice-driven applications such as smart assistants, e-commerce search, and multimedia content discovery depend on speech retrieval systems that can map spoken queries to relevant content. For

Indic languages, which collectively serve over a billion speakers, this capability remains substantially underdeveloped relative to English and other high-resource languages. The dominant production approach combines automatic speech recognition (ASR), machine translation, and monolingual text retrieval into a cascade pipeline that inherits the brittleness of each stage, with error propagation compounding through the pipeline. Training large multilingual speech-text models from scratch requires computational resources far beyond what most practitioners and researchers have access to. This paper investigates a middle path: **VaaniCLIP**, a lightweight contrastive alignment of pretrained speech and text encoders. We pair Whisper’s multilingual audio encoder (Radford et al., 2023) with LaBSE’s cross-lingual text encoder (Feng et al., 2022) and train only small projection heads (plus optionally the top audio encoder layers) via a symmetric NT-Xent contrastive loss (Chen et al., 2020; Radford et al., 2021). VaaniCLIP is conceptually simple, parameter-efficient, and substantially outperforms both cascade pipelines and large generalist multilingual models on low-resource Indic retrieval. Benchmarked on BhasaAnuvaad (Sankar et al., 2025), it captures 76–89% of the LaBSE text-text upper bound across Telugu, Tamil, and Hindi.

Contributions. Our specific contributions are:

- We introduce VaaniCLIP and establish what is, to our knowledge, the first systematic cross-lingual speech-text retrieval benchmark on BhasaAnuvaad across Telugu, Tamil, and Hindi, with normalized pool sizes for fair cross-language comparison.
- We demonstrate that cascade pipelines collapse on low-resource Indic languages, with the failure mode tracking ASR quality monotonically across Hindi (cascade R@1 of 46.2),

Tamil (21.6), and Telugu (3.1) at the normalized pool.

- We demonstrate that SeamlessM4T v2 used as a zero-shot retriever performs at near-random levels across all three Indic languages, despite its scale and Indic-language pretraining coverage, indicating that task-specific contrastive alignment remains necessary even in the era of large pretrained models.
- We show that encoder representation quality is the primary bottleneck for low-resource Indic retrieval: VaaniCLIP-PartialFT (15M trainable parameters) and a frozen Whisper-medium encoder (919K trainable parameters, $16\times$ fewer) reach comparable performance, with disproportionately large gains on languages where Whisper’s frozen representations are weakest (66.7% relative R@1 improvement on Telugu vs. 18.5% on Hindi).
- We show that joint multilingual training exhibits asymmetric cross-family transfer, degrading Hindi at both encoder scales while producing only a small benefit on both Dravidian languages that is within the range of single-seed variation, and that VaaniCLIP transfers zero-shot to Kannada (an unseen Dravidian language) at $\sim 700\times$ above random, providing evidence that it learns a Dravidian audio-text alignment that generalizes to an unseen language.

2 Related Work

Our work draws on three intersecting areas.

Speech representations. Self-supervised pretraining underpins modern speech systems. wav2vec 2.0 (Baevski et al., 2020) predicts masked quantized acoustic features via contrastive learning, and HuBERT (Hsu et al., 2021) replaces the quantization step with iterative k-means clustering. w2v-BERT (Chung et al., 2021) unifies these; w2v-BERT 2.0 (Barrault et al., 2023), the encoder underlying SeamlessM4T, scales to 4.5 million hours across 143 languages. Whisper (Radford et al., 2023) takes a supervised approach, training on 680,000 hours of weakly-supervised multilingual audio-text pairs; we use Whisper’s encoder because it covers all four languages we study, while noting that its supervision is distributed very

unevenly across them — a disparity our cascade results make visible (Section 5.2).

Text and multilingual models. Multilingual BERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020) produce cross-lingual representations via masked language modeling, but their sentence-level retrieval quality is weaker than purpose-built encoders. LaBSE (Feng et al., 2022) is designed for cross-lingual semantic similarity across 109 languages and produces sentence embeddings directly comparable across languages. NLLB-200 (NLLB Team et al., 2022) serves as the translation component in our cascade baseline. SeamlessM4T v2 (Barrault et al., 2023; Seamless Communication Team, 2023) unifies speech translation, recognition, and text translation in a single 2.3B-parameter model; our results show its encoder representations do not transfer to retrieval without task-specific alignment.

Contrastive alignment and Indic resources. CLIP (Radford et al., 2021) established contrastive alignment for vision-language pairs using the NT-Xent loss (Chen et al., 2020). Speech-CLIP (Shih et al., 2023) adapts this to speech-image alignment; CLAP (Elizalde et al., 2023) extends it to audio-text alignment. Closest to our setting, SAMU-XLSR (Khurana et al., 2022) distills LaBSE into an XLS-R speech encoder so that utterance embeddings are semantically aligned across languages, and evaluates cross-lingual speech-to-text translation retrieval; it shares our text encoder and task framing but distills a single encoder end-to-end rather than contrastively aligning frozen encoders, and does not target Indic languages. GLAP (Dinkel et al., 2025) extends CLAP with multilingual and multi-domain audio-text pretraining, but evaluates *same-language* audio-text retrieval rather than the cross-lingual speech-to-English-text setting we study. SONAR (Duquenne et al., 2023) is a purpose-built multilingual sentence-level speech-text embedding space with encoders covering all four languages we study; it is also the space in which BhasaAnuvaad’s mining scores are computed, so it is embedded in the data pipeline our filtering relies on (Section 4.1). We treat SONAR’s Indic speech encoders as the natural strong baseline for our task; we do not evaluate them here and leave a direct comparison to future work. Speech-text retrieval has also been benchmarked more broadly: FLEURS (Conneau et al., 2022) eval-

uates speech-text retrieval across 102 languages including ours; CLASP (Abootorabi and Asgari, 2025) performs multilingual audio-text retrieval against ASR-cascade baselines; and the SqCLR shared tasks (Dave et al., 2024) target spoken-query cross-lingual information retrieval for Indic languages. Against this backdrop, our contribution is not a first benchmark but a controlled empirical study of contrastive frozen-encoder alignment for Indic speech-to-English-text retrieval, isolating the audio-encoder bottleneck, cross-family transfer, and zero-shot behaviour. Recent work on Indic cross-lingual text retrieval reports steep degradation for low-resource and morphologically rich targets and proposes parameter-efficient adapter alignment (Arya, 2025); we extend this line from text-text to the audio-text modality. IndicSUPERB (Javed et al., 2023) provides speech corpora and benchmarks for Indic languages. BhasaAnuvaad (Sankar et al., 2025), our primary dataset, aligns Indic-language speech with English translations; the original paper evaluates machine translation but not cross-lingual audio-text retrieval, which we establish here. VaaniCLIP combines these components; the contribution lies in the systematic empirical study showing this combination produces retrieval performance unachievable by cascade pipelines or large generalist models on low-resource Indic languages.

3 Method

Our approach aligns pretrained speech and text encoders into a shared embedding space using contrastive learning. The architecture pairs Whisper (Radford et al., 2023), a 12-layer transformer encoder pretrained on 680,000 hours of multilingual speech, with LaBSE (Feng et al., 2022), a multilingual sentence encoder trained on 109 languages. Both encoders produce 768-dimensional representations: Whisper via mean pooling over its final-layer hidden states, and LaBSE via the [CLS] token. Two learned linear projection heads g_a, g_t map these to a shared 512-dimensional space, with L_2 -normalized outputs so cosine similarity reduces to a dot product. We train the projection heads, a learnable temperature parameter τ , and (optionally) the top 2 Whisper encoder layers via a symmetric NT-Xent contrastive loss (Chen et al., 2020; Radford et al., 2021); LaBSE remains frozen throughout. We refer to this model as **VaaniCLIP** throughout. Figure 1 illustrates the architecture. In

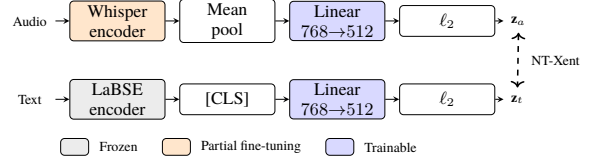


Figure 1: VaaniCLIP architecture. Audio (top track) is encoded by Whisper-small (mean-pooled); text (bottom track) by LaBSE ([CLS]). Both project to a shared 512-dim space, L_2 -normalized, and aligned via symmetric NT-Xent loss with learnable temperature τ . In partial fine-tuning, the top 2 Whisper encoder layers are additionally unfrozen with lower learning rate (10^{-5} vs. 10^{-3}); LaBSE remains frozen throughout.

preliminary experiments, varying the projection dimension from 256 to 512 produced negligible differences in R@1, indicating that the audio encoder rather than projection capacity is the bottleneck.

3.1 Training Objective

We train using a symmetric NT-Xent contrastive loss (Chen et al., 2020). For a batch of B audio-text pairs $\{(a_i, t_i)\}_{i=1}^B$, we compute embeddings $\{z_{a,i}, z_{t,i}\}$ and treat each matched pair as positive against the remaining $B - 1$ texts in the batch as in-batch negatives:

$$\mathcal{L}_{a \rightarrow t} = -\frac{1}{B} \sum_{i=1}^B \log \frac{\exp(z_{a,i} \cdot z_{t,i} / \tau)}{\sum_{j=1}^B \exp(z_{a,i} \cdot z_{t,j} / \tau)}. \quad (1)$$

The full training loss averages this with the symmetric text-to-audio loss $\mathcal{L}_{t \rightarrow a}$. The temperature τ is learnable, initialized at 0.07 following Radford et al. (2021), and clamped to $[0.01, 0.5]$ during optimization to prevent the softmax from becoming degenerate.

3.2 Training Configurations

We evaluate three training configurations. **VaaniCLIP-frozen** trains only the projection heads and temperature; both encoders remain fixed. **VaaniCLIP-PartialFT** additionally unfreezes the top 2 Whisper encoder layers (the 10 lower Whisper layers and all of LaBSE remain frozen), allowing limited adaptation of the audio representations to the retrieval task. We adopt this terminology rather than “end-to-end” for methodological precision. **VaaniCLIP-joint** uses the VaaniCLIP-PartialFT architecture trained on combined Telugu, Tamil, and Hindi data without language-specific weighting: batches are drawn from the pooled training set in proportion to

each language’s size, so lower-resource languages (Hindi) are correspondingly under-represented. This configuration additionally enables zero-shot evaluation on unseen Indic languages.

4 Experiments

We design our experiments to systematically compare cross-lingual speech retrieval approaches across Indic languages spanning the Dravidian and Indo-Aryan families. The following subsections describe our datasets and filtering criteria (Section 4.1), baselines and VaaniCLIP variants (Section 4.2), training setup (Section 4.3), and evaluation protocol with pool size normalization for fair cross-language comparison (Section 4.4).

4.1 Datasets and Splits

We use BhasaAnuvaad (Sankar et al., 2025), a large-scale parallel speech-text corpus covering Indic languages with aligned English text. Our experiments use the IndicVoices-ST subset, which provides high-quality audio recordings paired with English translations. In this subset the English side is not human-authored but generated by Llama-3.1-405B and then filtered by the alignment and mining scores described below (Sankar et al., 2025); our LaBSE text-text upper bound (Section 4.2) therefore measures retrieval against LLM-generated English rather than human reference translations. We evaluate on three primary languages (Telugu and Tamil are Dravidian; Hindi is Indo-Aryan), spanning two major Indic language families and a range of resource levels in pretrained speech models. We additionally include Kannada (Dravidian) as a held-out language for zero-shot transfer evaluation; no Kannada data is used in any training run. For a high-resource reference point we additionally evaluate the cascade on Spanish using mTEDx (Salesky et al., 2021), a multilingual corpus of TEDx talks with sentence-aligned English translations; Spanish is used only as a control in Section 5.2 and never for training.

Filtering. We apply three filtering criteria to the raw data. We retain samples with (i) audio duration between 1 and 10 seconds, removing very short utterances likely to be noise artifacts and very long utterances that introduce excessive padding overhead; (ii) Levenshtein-based alignment score $\tau \geq 0.8$, the value the BhasaAnuvaad authors apply to their own seed set (Sankar et al., 2025); and (iii) English mining score $\sigma \geq$

0.3, where σ is the cross-lingual similarity computed by SONAR (Duquenne et al., 2023) in the BhasaAnuvaad mining pipeline; our training data is therefore SONAR-filtered. The BhasaAnuvaad authors apply $\sigma \geq 0.6$ to their seed set but explicitly decline to prescribe a universal threshold, instead recommending language-specific thresholds (Sankar et al., 2025). Following that recommendation, we adopt a more permissive mining score to retain sufficient training data for low-resource Indic languages, particularly Telugu where the higher threshold would substantially reduce available training samples. We discuss the impact of this choice in the Limitations section. Filter retention differs markedly across languages: 77.6% of raw Telugu segments and 77.6% of Tamil segments survive, against only 52.6% for Hindi. The Hindi subset is therefore the most aggressively filtered of the three, a fact we return to in Section 5.1.

Splits. For each language, we construct train, validation, and test splits using a fixed random seed of 42 to ensure reproducibility. Splits are drawn at the utterance level; because IndicVoices-ST is multi-speaker and we did not enforce speaker- or session-disjoint partitions, a speaker may appear in both training and test, which can inflate absolute retrieval numbers. Our conclusions rest on relative comparisons across models over identical pools, which this does not affect, but absolute values should be read with this caveat. Table 1 summarizes post-filtering dataset statistics. Validation is used exclusively for model selection and best-checkpoint identification during training; all reported retrieval results are computed on held-out test sets, evaluated exactly once per model.

4.2 Baselines

We compare VaaniCLIP against three baselines spanning the major paradigms for cross-lingual speech-text retrieval, along with a text-to-text upper bound for reference. Where we quote a multiple “above random”, the reference is the analytic expectation for random ranking, k/N .

Cascade pipeline. The standard approach to cross-lingual speech retrieval combines ASR, machine translation (MT), and monolingual text retrieval. For each audio query, we (i) transcribe the source-language speech using Whisper-small (Radford et al., 2023), (ii) translate the transcription to English using NLLB-200-distilled-

Language	Train	Validation	Test
Telugu (tel)	79,641	9,950	12,445
Tamil (tam)	146,350	18,294	18,294
Hindi (hin)	22,954	2,869	2,869
Kannada (kan)	—	—	67,023

Table 1: Dataset statistics after filtering across primary evaluation languages. Test column shows full held-out test set sizes; for cross-language retrieval comparisons (Section 5), we subsample test sets to a normalized pool of $N = 2,869$ as described in Section 4.4. Kannada is held out entirely for zero-shot evaluation. Telugu’s post-filter set (124,441 examples) is split 80/10/10; the reported test column is the 12,445-example outer test split. The 99,552-example training pool is further subdivided for early stopping, so the model trains on 79,641 examples and selects on a 9,950-example validation split. All splits are post-filter, and the test set is disjoint from all training data.

600M (NLLB Team et al., 2022), and (iii) retrieve the nearest English text using LaBSE (Feng et al., 2022) embeddings via cosine similarity. We use greedy decoding for both ASR and MT.

SeamlessM4T zero-shot retriever. We evaluate seamless-m4t-v2-large (Seamless Communication Team, 2023) (2.3B parameters) as a zero-shot retriever. We extract audio embeddings by mean-pooling the final hidden states of the speech encoder, and text embeddings analogously from the text encoder, projecting both into the model’s shared multimodal space. Retrieval is performed via cosine similarity, with no fine-tuning.

VaaniCLIP-frozen. To isolate the contribution of encoder adaptation, VaaniCLIP-frozen keeps both pretrained encoders fixed and trains only the projection heads and learnable temperature via NT-Xent loss. It shares the VaaniCLIP architecture and training data with VaaniCLIP-PartialFT, differing only in which parameters are updated.

LaBSE text-text upper bound. To establish the ceiling achievable by any speech retrieval method using LaBSE as the text encoder, we report a text-to-text baseline that retrieves using LaBSE on the source-language ground-truth text rather than audio. This isolates the audio understanding component: the gap between this upper bound and VaaniCLIP reflects the difficulty of recovering text-aligned semantic content from speech, while the upper bound itself reflects LaBSE’s cross-lingual text retrieval quality for each language pair. We use “upper bound” in the sense of an *oracle-text* ref-

erence — retrieval given the ground-truth source transcript — not a hard ceiling on achievable speech retrieval; a speech model could in principle exceed it, and it is itself computed against the Llama-3.1-405B-generated English side (Section 4.1).

4.3 Training Setup

Hardware and implementation. All experiments are conducted on a single NVIDIA A100 GPU with 80 GB of memory. Models are implemented in PyTorch (Paszke et al., 2019) using the HuggingFace Transformers library (Wolf et al., 2020) for Whisper and LaBSE.

Optimization. We train all VaaniCLIP variants using the Adam optimizer (Kingma and Ba, 2014). For VaaniCLIP-PartialFT, we use differential learning rates: $1e-5$ for the unfrozen top-2 Whisper encoder layers, and $1e-3$ for the projection heads and learnable temperature. The lower learning rate on pretrained Whisper layers preserves learned acoustic representations while allowing controlled adaptation, following standard practice in transformer fine-tuning (Howard and Ruder, 2018). Learning rates are reduced by a factor of 0.5 via ReduceLROnPlateau when validation R@1 fails to improve for 10 consecutive epochs. The learnable temperature is clamped to $[0.01, 0.5]$ during training to prevent the contrastive softmax from becoming degenerate.

Parameter efficiency. VaaniCLIP-PartialFT trains approximately 15M parameters in total: 14.2M from the top 2 Whisper encoder layers, 787K from the projection heads, and one scalar for the learnable temperature. VaaniCLIP-frozen trains only 787K parameters (projection heads and temperature). Throughout the paper, reported parameter counts are *trainable* parameters; the cascade and SeamlessM4T v2 baselines are used zero-shot and therefore train none. Total model sizes, where relevant, are given in prose.

Training duration and checkpointing. Per-language models are trained for up to 100 epochs. We retain the checkpoint with the best validation R@1 for test evaluation. Epoch count is not directly comparable across languages with different dataset sizes: one epoch over Tamil’s 146,350 training samples corresponds to $\approx 6.4\times$ more gradient updates than one epoch over Hindi’s 22,954 samples.

Joint multilingual training. In addition to per-language models, we train VaaniCLIP-joint on the combined Telugu, Tamil, and Hindi training sets using identical architecture and optimization protocol, with batches sampled from the combined data without language-specific weighting. Joint training tests whether shared encoder capacity benefits or harms individual language performance, and additionally enables zero-shot evaluation on unseen Indic languages (Section 5.6).

Reproducibility. All training runs use fixed random seeds for data splitting and retrieval-pool subsampling (seed 42); model weights use PyTorch’s default initialization and are not explicitly seeded, so exact weights are not bit-reproducible across runs. Code, checkpoints, and configuration files will be released upon publication.

4.4 Evaluation Protocol

We evaluate retrieval performance using Recall@ k ($k \in \{1, 5, 10\}$), defined as the fraction of audio queries for which the correct English text appears within the top- k retrieved candidates ranked by cosine similarity. All reported numbers are computed on held-out test sets, evaluated exactly once per model.

Retrieval pool normalization. Retrieval difficulty depends strongly on candidate pool size: a larger pool contains more potential distractors, lowering R@ k at fixed model quality. Since our test sets vary substantially in size across languages (Table 1), reporting raw R@ k would conflate model performance with pool size effects and prevent fair cross-language comparison. We therefore normalize the retrieval pool to $N = 2,869$ samples (the size of the smallest test set, Hindi) by randomly subsampling larger test sets to this fixed pool size, using a fixed random seed to ensure all models retrieve from an identical candidate pool. This isolates model quality differences from pool composition effects.

Comparison to upper bound. To contextualize absolute R@ k values, we report each model’s performance as a percentage of the LaBSE text-text upper bound for each language. This headroom-relative measure isolates model quality from the inherent difficulty of each language pair under our text encoder choice.

5 Results and Analysis

We present cross-lingual speech-text retrieval results across Telugu, Tamil, and Hindi at a normalized retrieval pool size of $N = 2,869$ on held-out test sets, and analyse the mechanisms behind them. Table 2 reports R@1, R@5, and R@10 across all VaaniCLIP variants and baselines, alongside the LaBSE text-text upper bound of Section 4.2.

5.1 Main Retrieval Results

VaaniCLIP achieves R@1 of 55.5 (VaaniCLIP-PartialFT joint) on Telugu, 86.4 (VaaniCLIP-PartialFT per-lang) on Hindi, and 54.8 (VaaniCLIP-PartialFT joint) on Tamil. The LaBSE text-text upper bound varies substantially across languages (72.9 on Telugu, 97.6 on Hindi, 68.8 on Tamil), reflecting LaBSE’s inherent cross-lingual retrieval quality for each language pair. VaaniCLIP achieves 76.1%, 88.5%, and 79.7% of this ceiling respectively, indicating that absolute differences in R@1 across languages largely reflect the gap between Whisper’s audio representations and LaBSE’s text representations of the same content, rather than differences in alignment method effectiveness. Across all three languages, VaaniCLIP captures three-quarters to nine-tenths of the achievable retrieval headroom.

Why the Hindi ceiling is so high. The Hindi upper bound (97.6) sits roughly 25 points above Telugu (72.9) and 29 above Tamil (68.8), and Hindi attains this despite having by far the smallest training set. We attribute this to filter selectivity rather than to any property of the language: the alignment and mining thresholds of Section 4.1 retain only 52.6% of raw Hindi segments against 77.6% for both Telugu and Tamil, so the surviving Hindi pairs are drawn from a much more stringently selected pool and their English references are correspondingly more literal. A near-saturated text-text ceiling means Hindi offers little headroom in which retrieval methods can differentiate, which is worth bearing in mind when reading Hindi as evidence for or against any method. We did not measure reference literalness directly; reporting mean mining score per language after filtering would test this explanation, and we leave it to future work.

5.2 Cascade Pipelines Fail on Low-Resource Indic Languages

Cascade performance varies dramatically across languages and tracks the quality of the upstream

Model	Train.	Telugu			Hindi			Tamil		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
<i>LaBSE text-text (upper bound)</i>	—	72.9	82.2	85.3	97.6	99.2	99.4	68.8	80.0	83.7
SeamlessM4T v2 (zero-shot)	0	0.0	0.2	0.4	0.1	0.2	0.4	0.1	0.2	0.4
Cascade (Whisper→NLLB→LaBSE)	0	3.1	4.7	6.0	46.2	55.3	58.4	21.6	32.2	36.7
<i>Whisper-small encoder (88M)</i>										
VaaniCLIP-frozen (per-lang)	787K	32.4	51.0	58.8	72.9	89.8	93.8	36.7	53.9	60.7
VaaniCLIP-frozen (joint)	787K	36.2	53.2	60.2	63.1	83.0	88.3	36.5	52.8	59.2
VaaniCLIP-PartialFT (per-lang)	15M	54.0	70.1	75.2	86.4	95.3	97.3	52.7	68.1	73.8
VaaniCLIP-PartialFT (joint)	15M	55.5	70.8	75.8	80.6	93.3	95.5	54.8	69.0	74.2
<i>Whisper-medium encoder (307M), fully frozen</i>										
VaaniCLIP-medium-frozen (per-lang)	919K	51.5	68.6	74.4	85.1	95.2	97.3	50.9	67.0	72.1
VaaniCLIP-medium-frozen (joint)	919K	52.8	69.9	74.8	80.9	93.3	95.9	51.6	66.1	70.9

Table 2: Cross-lingual speech-text retrieval results on held-out test sets at normalized retrieval pool size $N = 2,869$. All values are percentages. *Train.* = trainable parameter count; the two zero-shot baselines train none (SeamlessM4T v2 has 2.3B total parameters, and the cascade combines Whisper-small, NLLB-600M and LaBSE). **Bold** indicates best non-upper-bound result per language across all VaaniCLIP variants. VaaniCLIP-medium-frozen (Whisper-medium held entirely frozen; only the 919K projection and temperature parameters are trained) reaches 95–99% of *per-language* VaaniCLIP-PartialFT R@1, confirming the audio encoder as the primary bottleneck. The joint-training asymmetry (Dravidian gain, Hindi loss) reproduces across encoder scales with smaller magnitude at larger scale.

Language	Family	ASR cov.	Casc. R@1
Hindi	Indo-Aryan	Med.-high	46.2
Tamil	Dravidian	Med.-low	21.6
Telugu	Dravidian	Low	3.1

Table 3: Cascade retrieval R@1 across Indic languages at the normalized pool $N = 2,869$, tracking Whisper-small pretraining coverage.

ASR component monotonically (Tables 2 and 3). On Hindi, cascade achieves R@1 of 46.2, reasonable but well below VaaniCLIP-PartialFT (86.4). On Tamil, cascade drops to 21.6. On Telugu, cascade collapses to 3.1% R@1, roughly $18\times$ below VaaniCLIP’s 55.5 and only about $89\times$ above the expectation for random ranking ($k/N = 0.035\%$ at $N = 2,869$). As a high-resource control, Spanish cascade retrieval reaches R@1 of 89.6 (on a smaller 690-sample mTEDx pool; Section 4.1), far exceeding any Indic language. This control uses a Helsinki-NLP es–en translation model rather than the NLLB-600M used in the Indic cascade, so it is an illustration of how upstream ASR coverage governs cascade quality rather than a strictly pipeline-matched comparison; the Indic collapse is not attributable to the MT stage, since the same NLLB model is used across all Indic languages. We do not train VaaniCLIP on Spanish.

Qualitative inspection of Whisper-small on Telugu reveals three failure modes: outputs

in incorrect scripts (Khmer, Kannada, other South/Southeast Asian scripts), repetition loops, and language confusion (most commonly Hindi). NLLB faithfully translates these into fluent but semantically unrelated English, and LaBSE retrieves candidates with no connection to the source audio. The cascade therefore inherits the ASR front-end’s coverage gaps: errors in early stages propagate and compound, so cascade quality tracks Whisper’s per-language coverage rather than the paradigm failing outright. A stronger or Indic-specific ASR model would likely narrow this gap; VaaniCLIP sidesteps it entirely by aligning audio and text directly without symbolic transcription.

5.3 Large Generalist Models Do Not Transfer Zero-Shot to Retrieval

SeamlessM4T v2 (Seamless Communication Team, 2023) is a 2.3-billion-parameter unified model trained on 4.5 million hours of audio across 100+ languages, including substantial Indic data. If pretraining scale and architectural unification were sufficient for cross-lingual retrieval, SeamlessM4T should perform well without task-specific training. In fact, its zero-shot retrieval performance is near-random across all three Indic languages ($R@1 < 0.1$; Table 2). The failure reflects a representation-objective mismatch: SeamlessM4T’s encoder is trained under a sequence-to-sequence objective and need

only produce hidden states useful to the decoder’s cross-attention, not standalone semantic representations. Retrieval instead requires semantically similar utterances to lie close under cosine similarity, a property a sequence-to-sequence encoder has no training signal to enforce. This does not imply that large pretrained models cannot support retrieval — purpose-built aligned spaces such as SONAR (Duquenne et al., 2023) are designed for exactly this — but that the alignment must be present, whether by architectural design or by task-specific training. It is the mean-pooled SeamlessM4T encoder specifically, not scale in general, that lacks it.

5.4 Partial Fine-tuning Yields Disproportionate Gains for Low-Resource Languages

The relative R@1 improvements from unfreezing the top 2 Whisper encoder layers are 66.7% on Telugu (32.4 $\rightarrow$ 54.0), 18.5% on Hindi (72.9 $\rightarrow$ 86.4), and 43.6% on Tamil (36.7 $\rightarrow$ 52.7) for per-language models. The gain is largest where the frozen baseline is weakest. Where Whisper already produces useful representations (Hindi), the projection head can adapt them effectively; where Whisper’s representations are weaker (Telugu, Tamil), a linear projection cannot create discriminative structure that is absent from the encoder, and unfreezing the top layers is necessary to recover it. Section 5.7 confirms this reading directly.

5.5 Joint Multilingual Training Shows Asymmetric Cross-Family Transfer

VaaniCLIP-joint, trained on the combined Telugu, Tamil, and Hindi data using identical architecture and training protocol, reveals an asymmetric transfer pattern (Table 2). Hindi, the only Indo-Aryan language in our training set, underperforms its per-language model substantially (−6.7% relative, −5.8 points). Both Dravidian languages show small positive deltas (Telugu +2.8%, Tamil +4.0% relative; $\Delta \leq 2.1$ points) whose magnitude is within the range we would expect from single-seed variation, so we do not read them as a reliable effect. The asymmetry is therefore between a large Indo-Aryan cost and Dravidian changes too small to interpret from single-seed runs. Telugu and Tamil share Dravidian phonological structure (agglutinative morphology, retroflex consonants, similar vowel systems), which offers one plausi-

ble account of why joint training does not degrade them; Hindi, the single Indo-Aryan language and also the smallest and most heavily filtered training set, is the one that degrades. We cannot separate a cross-family interference explanation from these confounds (data proportion, filtering, a single Indo-Aryan language, and the truncated joint schedule) with the present setup, and report the pattern rather than a definitive cause. One caveat applies to all joint results: the run was stopped at epoch 17, when validation retrieval began to degrade while the training loss was still falling, which pre-empted the learning-rate scheduler (Appendix C). The joint numbers are therefore a lower bound. The Hindi gap is not attributable to a shorter budget, since both models are reported at their validation optima.

5.6 Zero-Shot Transfer to an Unseen Dravidian Language

To test whether VaaniCLIP-joint has learned transferable Dravidian representations rather than per-language alignment, we evaluate it on Kannada, a Dravidian language fully held out from training, without any Kannada-specific fine-tuning. VaaniCLIP-joint achieves R@1 of 25.1, R@5 of 40.3, and R@10 of 46.6 on Kannada at $N = 2,869$, approximately $700\times$ above random. This corresponds to 77% of the trained VaaniCLIP-frozen per-lang baseline for Telugu (32.4) and 68% for Tamil (36.7), both achieved with zero target-language training data. Zero-shot Kannada performance also substantially exceeds Telugu cascade (3.1% R@1) and SeamlessM4T zero-shot retrieval on any of our trained languages (R@1 at most 0.1).

This transfer is not achieved through a language-invariant space. A linear language-identity probe (Appendix E) shows that projection leaves source-language decodability essentially unchanged (95.0% vs. 95.2% raw; chance 33.3%): VaaniCLIP learns a cross-lingual alignment geometry that generalises across Dravidian languages while preserving language-specific structure. The residual gap to trained performance reflects Kannada-specific phonological features (consonant clusters, vowel realizations, prosodic patterns) not fully shared with Telugu and Tamil.

5.7 Encoder Scale: Frozen Whisper-medium Matches Partial Fine-tuning

Section 5.4 attributes partial fine-tuning gains to inadequate audio representations from frozen

Whisper-small. If correct, upgrading the encoder while keeping the rest of the pipeline unchanged should close most of the same gap. We test this by substituting Whisper-medium (307M encoder parameters, 24 layers vs. Whisper-small’s 12) and training only the projection heads and temperature ($\sim 919\text{K}$ trainable parameters, comparable to VaaniCLIP-frozen’s 787K; the wider Whisper-medium embedding enlarges the audio projection, and both counts are dominated by the projection heads).

VaaniCLIP-medium-frozen reaches 95.4%, 98.5%, and 96.6% of VaaniCLIP-PartialFT R@1 on Telugu, Hindi, and Tamil respectively for per-language models; on Telugu it brings R@1 from 32.4 to 51.5, within 5% of the 54.0 that partial fine-tuning achieves with $16\times$ more trainable parameters. On Hindi, VaaniCLIP-medium-frozen per-lang (R@1 = 85.1) even exceeds VaaniCLIP-PartialFT joint (R@1 = 80.6). Two qualitatively distinct interventions therefore converge to near-identical retrieval quality, which is direct empirical confirmation that the audio encoder, not projection capacity, is the primary bottleneck.

Joint training is partly compensatory. The cross-family asymmetry from Section 5.5 persists with Whisper-medium but at smaller magnitudes: the Telugu joint delta shrinks from +11.7% (Whisper-small frozen) to +2.5% (Whisper-medium frozen), Tamil moves from -0.5% to $+1.4\%$, and the Hindi degradation shrinks from -13.4% to -4.9% . The Hindi effect is large and negative at both scales; the Dravidian deltas stay small at both scales (Tamil even flips sign, $-0.5\% \rightarrow +1.4\%$), consistent with their being within single-seed variation rather than a reliable effect. We read the residual interference as a property of how the shared projection accommodates typologically distinct languages rather than an artifact of Whisper-small specifically: a weaker encoder leaves more representation gaps for cross-lingual co-training to fill, and a stronger encoder reduces the need for it.

Two deployment paths. These results identify two practical approaches: (a) upgrade the encoder and train only projections (919K trainable, higher inference cost), or (b) fine-tune a smaller encoder (15M trainable, lower inference cost). Both reach comparable retrieval quality; the choice depends on deployment constraints rather than achievable

performance.

6 Conclusion

We introduced VaaniCLIP and presented a systematic benchmark for cross-lingual speech-text retrieval on Indic languages using BhasaAnuvaad. Two natural baselines fail on low-resource Indic languages: cascade pipelines collapse due to ASR brittleness, and SeamlessM4T v2 zero-shot produces near-random retrieval despite its 2.3B parameters. VaaniCLIP achieves strong retrieval across Telugu, Tamil, and Hindi — 76–89% of the LaBSE text-text upper bound with partial fine-tuning of Whisper-small, and 71–87% with a fully frozen Whisper-medium encoder trained on only 919K parameters — and transfers zero-shot to Kannada at $\sim 700\times$ above random. The two configurations reach comparable R@1 while exposing different inference and training cost tradeoffs, with the encoder governing achievable retrieval quality in both cases.

Future work includes scaling to larger audio encoders (Whisper-large-v3, w2v-BERT 2.0) to further close the gap on Telugu and Tamil where the encoder bottleneck dominates. Our Whisper-medium ablation suggests this direction is promising: whether Whisper-large-v3 saturates against the LaBSE text-text upper bound or continues to close the gap is an open question. Additional directions include systematic evaluation across the long tail of Indic languages and parameter-efficient fine-tuning methods (adapters, LoRA) for further efficiency gains. We release our code, checkpoints, and evaluation protocol upon publication.

Limitations

Our work has several limitations. First, all experiments use BhasaAnuvaad (IndicVoices-ST subset), and results may not generalize to other domains, speaking styles, or code-switched speech; our primary evaluation covers only three Indic languages plus one zero-shot language, so the cross-family transfer asymmetry we observe is based on a single Indo-Aryan training language and may not fully characterize Indo-Aryan dynamics. Second, we adopt a more permissive English mining score threshold ($\sigma \geq 0.3$) than the $\sigma \geq 0.6$ the BhasaAnuvaad authors apply to their seed set to retain training data for low-resource languages. We did not run a controlled comparison between thresholds, and Tamil partial fine-tuning exhibits

early convergence followed by validation degradation, suggesting the permissive threshold may admit noisier pairs that overfit with sufficient gradient updates. Relatedly, the joint multilingual run was stopped at epoch 17, when validation retrieval began to degrade while the training loss continued to fall (Appendix C). This pre-empted the learning-rate scheduler’s patience criterion, so the joint configuration was never trained through a learning-rate reduction and its results should be read as a lower bound on that configuration. We additionally report single-seed results due to compute constraints; while the large-magnitude effects we report are unlikely to reverse with different seeds, absolute $R@k$ values should be interpreted as point estimates. Third, we evaluate Whisper-small (88M encoder) and Whisper-medium (307M encoder); Whisper-large-v3 and other encoder families (w2v-BERT 2.0, IndicWhisper) remain unexplored. Our retrieval pool ($N = 2,869$) is far smaller than production catalogs (10^5 to 10^7 items); relative method ordering should persist at scale but absolute deployment performance requires evaluation under realistic catalog sizes. Finally, our train/validation/test splits are random at the utterance level and are not speaker- or session-disjoint; since the corpus is multi-speaker, speaker overlap between train and test may inflate absolute retrieval scores, though it applies equally to all models compared on the same pools.

Ethics Statement

Our experiments use BhasaAnuvaad (IndicVoices-ST), a publicly released corpus. We collect no new data and conduct no human-subject research. The corpus consists of recordings of human speech and therefore carries speaker-identifiable characteristics; we use only the released data, make no attempt at speaker identification, and redistribute no audio.

Our motivation is to improve speech retrieval for Indic languages that current systems serve poorly. Retrieval quality nonetheless varies substantially across the languages we evaluate ($R@1$ from 54.8 to 86.4 at $N = 2,869$), and our evaluation pools are far smaller than production catalogues. A system built on these results would work materially better for some language communities than others, and aggregate figures should not be read as uniform capability. Our cross-family finding bears directly on this: because a single joint multilingual

model measurably degrades Indo-Aryan performance, choosing one joint model on cost grounds distributes that cost unevenly across speaker populations.

Our filters also retain different proportions of each language (52.6% of raw Hindi segments against 80.0% of Telugu). Selective filtering raises measured quality but narrows the speech being represented, and may exclude accents, dialects and recording conditions that are harder to align. Results should not be taken to cover the full variation within any of these languages.

All training used a single NVIDIA A100. The joint configuration, our largest training set, required approximately 45 GPU-hours; per-epoch cost is reported in Appendix C so that the compute footprint of reproduction is explicit.

References

- Mohammad Mahdi Abootorabi and Ehsaneddin Asgari. 2025. CLASP: Contrastive language-speech pretraining for multilingual multimodal information retrieval. In *Advances in Information Retrieval (ECIR)*.
- Pulkit Arya. 2025. [Monolingual adapter networks for efficient cross-lingual alignment](#). In *Proceedings of the 5th Workshop on Multilingual Representation Learning (MRL 2025)*, pages 360–368, Suzhou, China. Association for Computational Linguistics.
- Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Loïc Barrault, Yu-An Chung, Mariano Cora Meglioli, and 1 others. 2023. SeamlessM4T: Massively multilingual & multimodal machine translation. *arXiv preprint arXiv:2308.11596*.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*.
- Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. 2021. w2v-BERT: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *ACL*.

- Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. 2022. FLEURS: Few-shot learning evaluation of universal representations of speech. In *IEEE Spoken Language Technology Workshop (SLT)*.
- Bhargav Dave, Prasenjit Majumder, and 1 others. 2024. Findings of shared task on spoken query cross-lingual information retrieval for the Indic languages at FIRE 2024. In *Proceedings of the 16th Annual Meeting of the Forum for Information Retrieval Evaluation (FIRE)*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.
- Heinrich Dinkel, Zhiyong Yan, Tianzi Wang, Yongqing Wang, Xingwei Sun, Yadong Niu, Jizhong Liu, Gang Li, Junbo Zhang, and Jian Luan. 2025. GLAP: General contrastive audio-text pretraining across domains and languages. *arXiv preprint arXiv:2506.11350*.
- Paul-Ambroise Duquenne, Holger Schwenk, and Benoît Sagot. 2023. SONAR: Sentence-level multimodal and language-agnostic representations. *arXiv preprint arXiv:2308.11466*.
- Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. 2023. CLAP: Learning audio concepts from natural language supervision. In *ICASSP*.
- Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Ariavazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *ACL*.
- Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *ACL*.
- Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. HuBERT: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.
- 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 *AAAI*.
- Sameer Khurana, Antoine Laurent, and James Glass. 2022. SAMU-XLSR: Semantically-aligned multimodal utterance-level cross-lingual speech representation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.
- Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- NLLB Team, Marta R. Costa-jussà, James Cross, and 1 others. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.
- Adam Paszke, Sam Gross, Francisco Massa, and 1 others. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *ICML*.
- 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 (ICML)*.
- Elizabeth Salesky, Matthew Wiesner, Jacob Bremerman, Roldano Cattoni, Matteo Negri, Marco Turchi, Douglas W. Oard, and Matt Post. 2021. The multilingual TEDx corpus for speech recognition and translation. In *Interspeech*, pages 3655–3659.
- Ashwin Sankar, Sparsh Jain, Nikhil Narasimhan, Devilal Choudhary, Dhairya Suman, Mohammed Safi Ur Rahman Khan, Anoop Kunchukuttan, Mitesh M. Khapra, and Raj Dabre. 2025. Towards building large scale datasets and state-of-the-art automatic speech translation systems for 14 Indian languages. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Seamless Communication Team. 2023. Seamless: Multilingual expressive and streaming speech translation. *arXiv preprint arXiv:2312.05187*.
- Yi-Jen Shih, Hsuan-Fu Wang, Heng-Jui Chang, Layne Berry, Hung-yi Lee, and David Harwath. 2023. SpeechCLIP: Integrating speech with pre-trained vision and language model. In *IEEE Spoken Language Technology Workshop (SLT)*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, and 1 others. 2020. HuggingFace’s Transformers: State-of-the-art natural language processing. In *EMNLP: System Demonstrations*.

Appendix

A Full-Pool Retrieval Results

Table 4 reports retrieval results on the full held-out test sets, without subsampling to the normalized pool size of $N = 2,869$. Telugu evaluates against 12,445 candidates, Tamil against 18,294, and Hindi against 2,869 (Hindi’s natural test set already matches the normalized pool size, so Hindi numbers are identical to Table 2).

At larger pool sizes, $R@k$ values are substantially lower because the candidate pool contains more potential distractors. Comparing Telugu $R@1$ between the normalized pool ($N = 2,869$) and full pool ($N = 12,445$) for the partial fine-tuning joint model shows a drop from 55.5% to 43.9%, reflecting roughly $4.3\times$ more distractors in the full pool. Despite these absolute differences, the relative ordering of methods remains consistent: partial fine-tuning outperforms frozen variants across all languages, and zero-shot baselines remain near random.

Key observations from full-pool evaluation.

Three patterns are worth noting. First, the partial fine-tuning joint and per-language models are roughly comparable on Telugu at full pool (43.9 vs 44.0 $R@1$), unlike at $N = 2,869$ where joint shows a small advantage (55.5 vs 54.0). This suggests the joint model’s slight edge at normalized pool size may not fully persist at larger pools, where harder distractors expose remaining limitations. Second, the gap between partial fine-tuning and frozen variants is preserved across pool sizes for all languages, reinforcing the central finding that encoder adaptation is necessary for low-resource Indic retrieval. Third, the SeamlessM4T zero-shot baseline reaches even smaller $R@k$ at full pool (0.01 $R@1$ for Telugu and Tamil), confirming that its near-random performance is not an artifact of pool size.

B Whisper-Small Failure Modes on Telugu

Section 5.2 describes the three dominant failure modes of Whisper-small on Telugu audio: outputs in incorrect scripts, repetition loops, and language-confusion errors. Table 5 shows representative examples drawn from cascade retrieval errors. The transcriptions are semantically unrelated to the source audio, and downstream NLLB translation faithfully translates these incorrect transcriptions into fluent but unrelated English. This compounds the error: cascade retrieval then ranks candidates that match the incorrect English translation, with no semantic connection to the original audio.

Whisper output	NLLB translation
ស្កុំ... (Khmer repetition)	“The city of Sunder-land...”
మారిల్లు మారిల్లు మారిల్లు (Telugu repetition)	“The Marbles The Marbles”
ಸಾರಿಲಿಲಿ... (Kannada script)	“For the word of the Lord...”
वाटि तो ने न समयान (Hindi)	“I thought you were not a little late.”

Table 5: Representative Whisper-small failure modes on Telugu audio input. The transcriptions are semantically unrelated to the source audio, and downstream NLLB translation faithfully translates these incorrect transcriptions into fluent but unrelated English.

C Training Dynamics Across Languages

Training configurations differ substantially in convergence behavior and, in the case of the joint model, in training budget. Table 6 reports the best-validation epoch and the epoch at which each run was stopped. We report these explicitly because the budgets are *not* equal across configurations, and this bears directly on how the joint-training results in Section 5.5 should be read.

Configuration	Best epoch	Stopped epoch
Telugu, partial FT	91	100
Hindi, partial FT	95	100
Tamil, partial FT	10	33
Joint, partial FT	12	17

Table 6: Best-validation epoch and stopping epoch per training configuration. Budgets are not equal in epochs: Telugu and Hindi ran the full 100 epochs, while Tamil stopped at 33 and the joint run at 17, both after validation retrieval began to degrade. All models are reported at their best-validation- $R@1$ checkpoint. Tamil exhibits early convergence despite having the largest training set (146,350 samples vs. 79,641 for Telugu and 22,954 for Hindi).

Despite having the largest training set, Tamil partial fine-tuning converges in the fewest epochs and exhibits validation degradation thereafter. We hypothesize this reflects a higher noise rate in BhasaAnuvaad’s Tamil subset: with more gradient updates per epoch ($146,350/320 \approx 457$ updates per epoch for Tamil vs. 71 for Hindi), the model has more opportunities to memorize noisy alignments. This observation reinforces the importance of validation-based early stopping in low-resource Indic training and suggests that more aggressive filtering ($\sigma \geq 0.6$, the value the BhasaAnuvaad authors apply to their seed set) may be appropriate

Model	Telugu ($N = 12,445$)			Hindi ($N = 2,869$)			Tamil ($N = 18,294$)		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
<i>LaBSE text-text (upper bound)</i>	66.8	76.9	80.0	97.6	99.2	99.4	58.5	70.8	74.8
SeamlessM4T v2 (zero-shot)	0.01	0.05	0.09	0.1	0.2	0.4	0.01	0.04	0.07
VaaniCLIP-frozen (per-lang)	20.0	36.3	44.0	72.9	89.8	93.8	22.5	37.7	44.4
VaaniCLIP-frozen (joint)	24.3	40.5	47.1	63.1	83.0	88.3	22.8	37.5	43.7
VaaniCLIP-PartialFT (per-lang)	44.0	60.0	66.0	86.4	95.3	97.3	38.2	55.2	61.1
VaaniCLIP-PartialFT (joint)	43.9	60.6	66.4	80.6	93.3	95.5	40.6	56.9	62.6

Table 4: Full-pool retrieval results on held-out test sets. All values are percentages. **Bold** indicates best non-upper-bound result per language at the full-pool size. Hindi numbers match Table 2 because the Hindi test set is already at $N = 2,869$. Model names match Table 2. Cascade full-pool numbers are omitted here; the main paper reports cascade results at the normalized pool $N = 2,869$ (Table 2). The Whisper-medium encoder-scale ablation was run only at the normalized pool and therefore has no full-pool rows.

for Tamil specifically, even at the cost of reduced training data.

Why the joint run was stopped at epoch 17.

The joint model was stopped because validation retrieval had begun to degrade while the training loss was still falling — the signature of overfitting, not of an exhausted compute budget. Table 7 reports the trajectory. Validation R@1 peaks at epoch 12 and R@5 and R@10 at epoch 10; between epochs 12 and 17 the training loss falls by 38% ($0.561 \rightarrow 0.349$) while validation R@1 falls 0.90 points ($0.4027 \rightarrow 0.3937$). Continuing would have improved the training objective while degrading the retrieval metric of interest. We therefore selected the epoch-12 checkpoint under the same best-validation-R@1 criterion applied to every per-language model. Each epoch of this configuration costs approximately 2.65 GPU-hours on a single A100, so the run as reported represents roughly 45 GPU-hours.

Epoch	R@1	R@5	R@10	Train loss
1	0.2461	0.3997	0.4631	2.4541
7	0.3923	0.5562	0.6130	0.8487
10	0.4015	0.5656	0.6213	0.6654
12	0.4027	0.5639	0.6208	0.5612
17	0.3937	0.5579	0.6126	0.3485

Table 7: Validation trajectory of the joint partial fine-tuning run, evaluated against the combined multilingual validation pool ($N = 31,113$). **Bold** marks the peak of each retrieval metric. Training loss continues to fall monotonically after validation retrieval turns, which is why the run was stopped; the epoch-12 checkpoint is the one reported throughout the paper.

Relation to the stated schedule. Stopping at epoch 17 pre-empts the ReduceLROnPlateau criterion of Section 4.3, which reduces the learning

rate only after 10 consecutive epochs without validation improvement; the run was stopped 5 epochs after its best checkpoint. The consequence is narrow but real: this configuration was never trained through a learning-rate reduction, and continued training at a reduced rate might recover further performance, so we treat the reported joint results as a lower bound. The model is not, however, under-trained in the ordinary sense. Validation retrieval had already turned, and the reported checkpoint is the configuration’s validation optimum, selected on the same criterion as every other model in the paper.

Epochs are the wrong unit of comparison. The apparent spread in convergence time across configurations is largely an artifact of counting epochs. In optimizer steps, Hindi converges after $\approx 6,800$ updates, Tamil after $\approx 4,600$, the joint model after $\approx 9,300$, and Telugu after $\approx 22,600$: a $5\times$ spread in updates against a $9.5\times$ spread in epochs. The joint model performs the most updates per epoch of any configuration (778, against 72 for Hindi), and it overfits earliest in epoch terms as a direct consequence. This is the same pattern Tamil exhibits, and it supports the interpretation offered above — that the permissive $\sigma \geq 0.3$ threshold admits noisy pairs which are memorised once enough gradient updates are taken.

What this does and does not license. Two consequences follow, and we state both. First, the joint numbers are a lower bound: the configuration never saw a learning-rate reduction, so its reported performance on all three languages may understate what it can achieve. Second, and by contrast, the Hindi degradation cannot be explained by a shorter training budget. Both the joint and the

per-language models are reported at their respective validation optima, and the joint model’s optimum arrives earlier in epochs only because each epoch contains an order of magnitude more updates. The remaining explanation we can offer for the Hindi result is cross-family interference, as argued in Section 5.5; rerunning the joint configuration through a full learning-rate schedule would nonetheless tighten the lower bound.

D Detailed Hyperparameters

Table 8 lists the complete configuration used for every experiment reported in the paper. Settings are shared across all languages and all VaaniCLIP variants except where a variant is defined by the difference — VaaniCLIP-frozen trains only the projection heads and temperature, and VaaniCLIP-medium-frozen substitutes the Whisper-medium encoder while training 919K parameters (the wider audio embedding enlarges the audio projection).

E Language-Identity Probing

To test whether VaaniCLIP’s projection removes language identity—the property underlying its cross-lingual transfer—we train a linear probe (logistic regression) to predict the source language from embeddings before and after projection, with classes balanced across the three languages so chance is 33.3% (Table 9). The probe recovers the source language from raw Whisper audio embeddings at 95.2% accuracy; after projection into the shared retrieval space, accuracy is essentially unchanged at 95.0%. The projection therefore does *not* remove language identity—the shared space is not language-invariant. Zero-shot transfer (Section 5.6) cannot be attributed to language invariance; instead, VaaniCLIP learns a cross-lingual alignment geometry that generalizes across Dravidian languages while preserving language-specific structure. For text, source-language decodability falls to 54.1%; because all text is English, this residual reflects cross-corpus content differences rather than language identity per se.

Probe input	Lang. acc. (%)
Raw Whisper audio	95.2
Projected audio (shared)	95.0
Projected text (shared)	54.1
Chance (3 languages)	33.3

Table 9: Linear language-probe accuracy (balanced classes, chance 33.3%). Projection leaves audio language-identity intact; the shared space is not language-invariant.

F Implications for Production Deployment

Our findings have practical implications for low-resource Indic speech retrieval that go beyond the main paper’s empirical claims. Many Indic languages have insufficient parallel speech-text data for direct training but belong to language families with better-resourced relatives. Our results suggest that joint training on resource-available family members can produce usable retrieval models for resource-scarce members without any target-language data, a substantially lower bar than fine-tuning approaches that require labeled examples.

The asymmetric cross-family transfer pattern also raises practical questions about multilingual Indic system design. A single joint model across all Indic languages may not be optimal: for production deployment, the cost-quality tradeoff between (i) one joint model serving all languages and (ii) family-grouped models (Dravidian-only, Indo-Aryan-only) should be evaluated empirically. Our results suggest family-grouped training is likely preferable when retrieval quality is the primary objective, while joint training may be preferable when deployment cost dominates.

G Extended Analysis of SeamlessM4T Failure

Section 5.3 argues that SeamlessM4T’s zero-shot retrieval failure reflects a representation-objective mismatch. Two specific consequences of this mismatch are worth elaborating.

First, semantic information may be encoded in distributed and decoder-specific ways that do not align with a meaningful geometric structure. A retrieval system requires that semantically similar utterances produce embeddings close under cosine similarity. SeamlessM4T’s encoder has no training signal to enforce this property; its hidden states need only be useful when consumed by the decoder.

Hyperparameter	Value
<i>Architecture</i>	
Audio encoder	Whisper-small (88M)
Text encoder	LaBSE (frozen)
Whisper encoder hidden dim	768
LaBSE hidden dim	768
Projection dim (d)	512
Audio pooling	Mean pooling
Text pooling	[CLS] token
<i>Data filtering</i>	
Audio duration range	1–10 seconds
Alignment score threshold	$\tau \geq 0.8$
Mining score threshold	$\sigma \geq 0.3$
<i>Hardware</i>	
GPU	NVIDIA A100 80GB
Framework	PyTorch + HF

Hyperparameter	Value
<i>Optimization</i>	
Optimizer	Adam (no wt. decay)
Whisper layer LR	$1e-5$
Projection head LR	$1e-3$
Temperature LR	$1e-3$
Temperature init	0.07
Temperature range	[0.01, 0.5]
LR scheduler	ReduceLROnPlateau
LR reduction factor	0.5
Patience	10 epochs
Batch size	320
Max epochs	100
<i>Trainable parameters</i>	
Frozen variant (small)	787K
Frozen variant (medium)	919K
Partial fine-tuning	15.0M total
Top 2 Whisper layers	14.2M
Projection heads	787K
Temperature	1

Table 8: Complete hyperparameter configuration for all reported experiments.

Second, encoder representations may encode language-specific or modality-specific features that the decoder learns to interpret but that prevent direct cross-modal comparison. Mean-pooling the encoder’s hidden states, a reasonable approach for retrieval, likely averages over these features in a way that loses discriminative signal.

The broader implication is that scale alone does not produce universal representations: large multilingual pretraining produces usable representations for downstream tasks only when those representations are aligned with the downstream objective. Lightweight contrastive fine-tuning achieves this alignment far more efficiently than re-training a large generalist model.

H Reproducibility

All training runs use fixed random seeds. Seed 42 is used for data splitting (train/val/test partition), seed 42 for the retrieval pool subsampling to $N = 2,869$, and PyTorch default seed initialization for model weights. The same Hindi test subset (2,869 samples) is used across all models evaluated on Hindi; analogous fixed subsets are used for Telugu and Tamil normalized evaluations.

Code, model checkpoints, evaluation scripts, and the specific subsample indices used for normalized retrieval pool evaluation will be released upon publication.