Cross-Lingual Transfer Learning for Low-Resource ASR via Sequential Fine-Tuning: A Study on Indian Languages

Anonymous ACL submission

Abstract

Low-resource languages remain poorly served by automatic speech recognition (ASR) systems despite the availability of large-scale multilingual models. We investigate whether sequential fine-tuning, that is, first adapting a pretrained model on a related high-resource language and then specialising on the low-resource target, can consistently improve recognition accuracy across Indian languages. Through a large-scale empirical study spanning seven Indian languages (Bengali, Assamese, Hindi, Urdu, Sanskrit, Tamil, Telugu), two models (Whisper-large-v3 and MMS-1b-all), two datasets (IndicVoices and Shrutilipi), and over 200 experimental configurations, we address three questions. First, we show that sequential transfer is broadly effective and model-independent for closely related pairs, with Bengali–Assamese improving WER by 1.6–5.8 percentage points across all configurations, while distant pairs can degrade Whisper but MMS remains robust due to its adapter architecture. Second, we demonstrate that closely related pairs benefit from transfer even at the smallest data volumes, whereas distant pairs require substantially more data and may never reliably help. Third, we find that phonetic overlap predicts transfer success better than script similarity: Hindi-to-Urdu transfer improves consistently despite different scripts, while Hindi-to-Sanskrit is inconsistent despite sharing the same Devanagari script. These results provide practitioners with concrete guidance on when sequential fine-tuning is a reliable strategy for low-resource Indic ASR and when target-only fine-tuning remains preferable.

1 Introduction

Modern multilingual ASR models such as Whisper (Radford et al., 2022) and MMS (Pratap et al., 2023) achieve strong performance on well-resourced languages but degrade substantially on low-resource languages with limited transcribed data. This disparity is stark for Indian languages, where most of the 22 officially recognised languages lack large-scale labeled speech corpora. These languages have different scripts, diverse phonological systems, and varying degrees of mutual similarity, from near-identical pairs sharing the same script (Bengali–Assamese) to phonetically close pairs with different scripts (Hindi–Urdu) to relatively dissimilar languages (Hindi–Tamil). This raises two natural questions: *When labeled data for a target language is scarce, can a pretrained ASR model benefit from first being adapted to a related language before being fine-tuned on the target?*, and *Do the choice of related language, its relatedness, and data availability affect whether such transfer helps or hurts?*

In this work, we investigate this through sequential fine-tuning—a two-stage pipeline where Stage 1 fine-tunes a pretrained model on a high-resource language, and Stage 2 further fine-tunes the resulting checkpoint on the low-resource target. While cross-lingual transfer through multilingual pre-training has been extensively studied (Babu et al., 2021), sequential fine-tuning at the adaptation stage has not been systematically evaluated for large speech models such as Whisper or MMS on Indian languages. Existing work focuses on prompt tuning (Ma et al., 2024; Tripathi et al., 2024), LoRA (Hu et al., 2021), data augmentation (Biswas et al., 2025), or single-language fine-tuning (Timmel et al., 2025), without comparing against a structured two-stage cross-lingual pipeline or examining how linguistic distance, data volume, and model architecture jointly determine transfer outcomes.

We address this gap through a large-scale empirical study spanning seven Indian languages (Bengali, Assamese, Hindi, Urdu, Sanskrit, Tamil, Telugu), two models (Whisper-large-v3 and MMS-1b-all), two datasets with contrasting acoustic conditions (IndicVoices (Javed et al., 2024) and Shrutilipi

(Bhogale et al., 2023)), and over 200 experimental configurations. We organise our investigation around three research questions.

RQ1: *Is sequential cross-lingual transfer effective across models, datasets, and language pairs, and for which pairs does it work or fail?*

RQ2: *How does data volume interact with linguistic relatedness? Do closely related pairs benefit from less high-resource data than distant pairs?*

RQ3: *When the low-resource target is fixed, how does varying the high-resource source across relatedness levels affect transfer? Does script similarity or phonetic overlap better predict success?*

Our main findings are as follows.

- Sequential fine-tuning with Bengali–Assamese, our most extensively tested pair, consistently improves WER by 1.6–5.8 percentage points over single-language baselines across all model-dataset combinations, and other closely related pairs show similar trends, while distant pairs can degrade Whisper but MMS remains robust due to its adapter architecture (RQ1).
- Closely related pairs benefit from transfer even at the smallest data volumes, whereas relatively distant pairs such as Hindi-Sanskrit show no reliable gain even when volumes are substantially larger (RQ2).
- Phonetic overlap predicts transfer success better than script similarity: Hindi-to-Urdu transfer improves consistently despite different scripts, while Hindi-to-Sanskrit is inconsistent despite sharing Devanagari script (RQ3).

2 Related Work

Self-Supervised and Multilingual ASR. Self-supervised models such as Wav2Vec 2.0 (Baevski et al., 2020), HuBERT (Hsu et al., 2021), and XLS-R (Babu et al., 2021) established that pre-training on unlabeled audio dramatically reduces labeled data requirements for ASR. Building on this, Whisper (Radford et al., 2022) uses supervised training on 680K hours of weakly labeled multilingual data with an encoder-decoder architecture, while MMS (Pratap et al., 2023) pairs a Wav2Vec 2.0 encoder with per-language connectionist temporal classification (CTC) heads and lightweight adapters for over 1,100 languages. This architectural difference is related to our study: Whisper has no per-language components, making it more receptible to cross-lingual interference during fine-tuning, whereas MMS's adapter separation provides robustness but limits multi-language mixing to sequential transfer only.

Fine-Tuning and Transfer for Low-Resource ASR. Adapting pre-trained models to low-resource languages has been explored through partial layer freezing (Timmel et al., 2025), LoRA (Hu et al., 2021), prompt tuning (Ma et al., 2024; Tripathi et al., 2024), and data augmentation (Biswas et al., 2025). Tripathi et al. (2024) leverage language-family information through soft prompts for Indian languages, while Liu et al. (2024) compare multiple strategies across seven languages, finding no single approach dominates universally. These methods focus on improving a single target language in isolation. Cross-lingual transfer at the fine-tuning stage, where a model is first adapted to a related high-resource language and then to the target, has been explored for self-supervised models in limited settings but not systematically evaluated for Whisper or MMS. For the Bengali–Assamese pair specifically, Kalwar (2025) demonstrates that phoneme-level weight mapping between a Bengali-fine-tuned Wav2Vec 2.0 model and an Assamese output layer consistently outperforms both direct fine-tuning and naive transfer with random re-initialization, confirming that linguistically informed initialization matters for closely related languages.

Our work extends this line of investigation to a broader set of language pairs, model architectures (including Whisper and MMS), and examines how linguistic distance interacts with transfer outcomes. Joint multilingual fine-tuning offers an alternative but risks degradation when the high-resource language dominates training. To our knowledge, no prior work systematically evaluates sequential fine-tuning across multiple Indic language pairs at varying degrees of linguistic relatedness, or examines how data volume, linguistic distance, and model architecture jointly determine transfer outcomes.

ASR for Indian Languages. India's 22 scheduled languages boast diverse scripts and phonological systems. ASR research for individual languages has historically relied on small corpora with limited diversity (Alam et al., 2022; Mandal et al., 2020; Deka et al., 2019). Two recent large-scale resources have improved data availability: IndicVoices (Javed et al., 2024), containing spontaneous crowd-sourced speech across all 22 languages, and Shrutilipi (Bhogale et al., 2023), providing broadcast audio from All India Radio

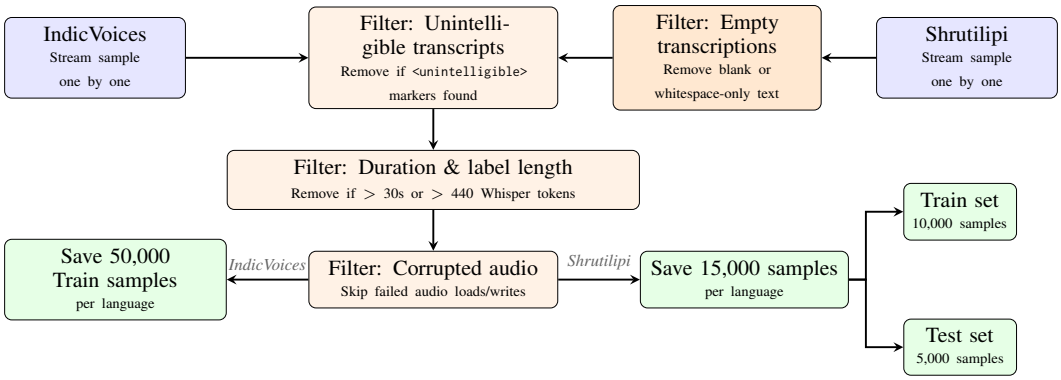

Figure 1: Data preparation pipeline. Both datasets pass through the same filtering stages, with Shrutilipi additionally requiring an empty-transcription filter. Final sample counts are listed in Table 1.

for twelve languages. We use both to test whether sequential fine-tuning generalises across acoustic conditions. Among the seven languages in our study, Bengali and Assamese form a very closely related language pair sharing the same script; Hindi and Urdu share extensive phonetic overlap despite different scripts; Sanskrit shares the Devanagari script with Hindi but differs in acoustic domain; Tamil and Telugu are again close while having different scripts. This range of relatedness enables systematic investigation of when cross-lingual transfer helps and when it breaks down.

3 Data

We use two publicly available Indian-language speech corpora, IndicVoices (Javed et al., 2024) and Shrutilipi (Bhogale et al., 2023). The two datasets differ substantially in acoustic conditions, allowing us to test whether the benefits of sequential fine-tuning generalise across recording settings rather than being an artifact of a particular data source.

3.1 Datasets

The **IndicVoices** dataset contains spontaneous, crowd-sourced speech recorded on personal mobile devices in everyday environments, covering all 22 scheduled Indian languages. We extracted local subsets of up to 50,000 training samples per language. All seven of our study languages are available in IndicVoices. Evaluation uses the official validation split hosted on HuggingFace.

The **Shrutilipi** dataset consists of professionally read news broadcasts from All India Radio with studio-quality audio, covering twelve Indian languages. We extracted 10,000 training samples and 5,000 held-out test samples per language after filtering, since Shrutilipi provides only a single train split with no official evaluation partition. The 15,000 total (10,000 train + 5,000 evaluation) was driven by data availability across languages. Sanskrit is the smallest subset with only 14,148 total samples, so we use 10,000 rows for training and the remainder for evaluation; all other languages support the full 15,000 size. Not all study languages are available in Shrutilipi, with Urdu being absent.

	IndicVoices		**Shrutilipi**	
Language	Trng./Eval.	Rej.	Trng./Eval.	Rej.
Bengali	50,000 / 3,906*	5,888	10,000 / 5,000	252
Assamese	50,000 / 5,062*	3471	10,000 / 5,000	910
Hindi	50,000 / 4,740*	2,494	10,000 / 5,000	6
Urdu	50,000 / 3,284*	741	–	–
Sanskrit	50,000 / 2,633*	383	10,000 / 4,148	29
Tamil	50,000 / 5,276*	1,030	10,000 / 5,000	61
Telugu	50,000 / 3,295*	2,389	10,000 / 5,000	801

Table 1: Sample counts after filtering. *Official evaluation split. – = unavailable. Rej. = rejected samples.

3.2 Data Preprocessing

We applied a uniform filtering pipeline to both datasets, removing samples with <unintelligible> transcription markers, audio exceeding 30 seconds (Whisper's input window limit), transcriptions exceeding 440 Whisper-large-v3 tokens (leaving headroom below the 448-token decoder maximum), and corrupted audio. For Shrutilipi, samples with empty transcriptions were additionally discarded. Although MMS does not share Whisper's constraints, we applied the stricter thresholds uniformly since both models use the same local subsets. All samples were saved as WAV files with JSONL metadata in HuggingFace AudioFolder format. Figure 1 illustrates the complete pipeline.

3.3 Dataset Statistics

Table 1 summarises the seven languages, dataset availability, and sample counts after filtering. These languages span a range of relatedness from

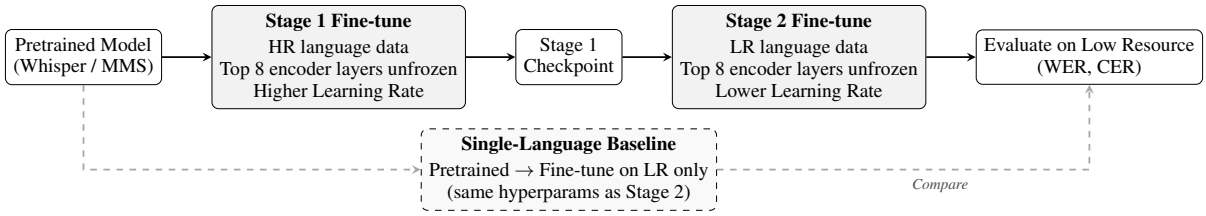

Figure 2: Two-stage sequential fine-tuning pipeline. Stage 1 adapts the pretrained model on high-resource (HR) data; Stage 2 fine-tunes the resulting checkpoint on low-resource (LR) data. The single-language baseline bypasses Stage 1 and uses identical Stage 2 hyperparameters.

	Whisper		MMS	
	Stage 1	Stage 2	Stage 1	Stage 2
Duration	2000 steps	1000 steps	5 epochs	10 epochs
Learning rate	1e-5	5e-6	3e-5	1e-5
Weight decay	–	–	0.005	0.01
Early stopping	–	–	2 epochs	3 epochs

Table 2: Key hyperparameters that vary between stages. Full configurations are in App. A.

near-identical pairs to relatively far-off languages.

Rejection rates differ a lot between the two corpora. IndicVoices, being crowd-sourced, exhibits higher rejection counts driven primarily by unintelligible transcription markers and label-length violations, with Bengali (5,888 rejected) and Assamese (3,471) being the most affected. Telugu also shows high rejection (2,389) largely due to label-length filtering. In contrast, Shrutilipi's professionally read broadcasts yield 0 unintelligible markers and overall much lower rejection counts, with Telugu (801, predominantly corrupt audio) and Assamese (910, predominantly label-length) being the only notable exceptions. Sanskrit in Shrutilipi contains only 14,148 samples in total, the smallest pool among our languages.

4 Methodology

4.1 Proposed Approach

The central idea is simple but effective. We first fine-tune a pretrained ASR model on a high-resource language (Stage 1), and then continue fine-tuning the resulting checkpoint on the low-resource target language (Stage 2). We hypothesize that exposure to a related language in Stage 1 provides a better starting point than fine-tuning on low-resource language directly. Training proceeds in two stages, as shown in Figure 2.

Stage 1 (High-resource adaptation). We fine-tune the pretrained model on high-resource language data. Following Pasad et al. (2022), we keep the bottom encoder layers frozen, as they capture universal acoustic features, and unfreeze only the top-8 encoder layers, which encode language-specific phonological patterns. An ablation comparing top-8 vs top-12 unfrozen layers for MMS showed negligible difference on both datasets (App. G) and, so, we retain top-8 layers by default. The decoder (Whisper) or the CTC temporal classification head (MMS) is fully trainable. Stage 1 uses a higher learning rate and longer training to accommodate the larger dataset.

Stage 2 (Low-resource transfer). We load the best Stage 1 checkpoint and continue fine-tuning on the low-resource target language. The same top-8 encoder layers remain trainable, but uses a lower learning rate and shorter training to refine rather than overwrite the Stage 1 representations, following standard practice to avoid catastrophic forgetting (Kirkpatrick et al., 2017).

A critical design principle is that the single-language baseline and Stage 2 of the sequential approach use identical hyperparameters, ensuring that any performance difference is attributable solely to the Stage 1 cross-lingual pre-adaptation. If they differ, any performance gap could be attributed to tuning rather than cross-lingual transfer. Table 2 summarises the key hyperparameters used across the experiments. App. C explains the decision to fix steps in Whisper and epochs in MMS.

4.2 Model Architectures

We evaluate on two widely deployed multilingual ASR models (Whisper-large-v3 and MMS-1b-all) with fundamentally different architectures (App. D). These models differ in nearly every design dimension: training objective, decoding strategy, input representation, and language handling. If sequential fine-tuning yields consistent improvements on both, the finding generalises across architectural paradigms rather than being specific to one modeling choice.

HR→LR	Relatedness	Settings	RQ
Bn→As	Close	All 4	1,2,3
As→Bn	Close	All 4	1,3
Hi→As	Mid	All 4	1,2,3
Hi→Bn	Mid	All 4	1,3
Hi→Ur	Cross-script	W+IV, M+IV	1,3
Hi→Sa	Same-script	W+IV,W+SL	1,2,3
Ta→As	Distant	All 4	1,3
Sa→As	Distant	W+IV, W+SL	1,3
Te→Bn	Distant	All 4	1,3
Bn→Te	Distant	M+SL	1,3
Hi→Te	Distant	M+SL	1,3
Ta→Te	Mid	M+SL	1,3

Table 3: Experiment inventory (W = Whisper, M = MMS, IV = IndicVoices, SL = Shrutilipi; All 4 = W+IV, W+SL, M+IV, M+SL). Languages are Bengali (Bn), Assamese (As), Hindi (Hi), Urdu (Ur), Sanskrit (Sa), Tamil (Ta), and Telugu (Te).

	W+IV		W+SL		M+IV		M+SL	
HR→LR	WER	CER	WER	CER	WER	CER	WER	CER
Bn→As	3.3	1.7	5.5	2.1	1.9	1.0	1.7	0.7
As→Bn	2.9	2.0	5.8	2.6	2.0	0.9	1.6	0.9
Hi→As	1.6	1.1	-0.3	-0.1	1.3	0.7	0.9	0.3
Hi→Bn	0.8	1.2	0.1	0.3	1.4	0.7	0.9	0.5
Hi→Ur	1.0	0.7	–	–	0.4	0.4	–	–
Hi→Sa	0.4	0.3	0.9	0.7	–	–	–	–
Ta→As	-0.4	-0.1	-1.2	-0.4	1.1	0.7	0.7	0.2
Sa→As	-0.9	-0.2	-1.3	-0.5	–	–	–	–
Te→Bn	0.6	0.7	-0.2	0.1	1.1	0.6	0.7	0.4
Bn→Te	–	–	–	–	–	–	0.7	0.4
Hi→Te	–	–	–	–	–	–	0.6	0.4
Ta→Te	–	–	–	–	–	–	0.6	0.4

Table 4: Average WER and CER improvement (percentage points) of sequential fine-tuning over single-language baseline, averaged over the five focused sweep configurations (10% LR ratio). Positive values indicate improvement. – = pair not tested on that setting.

4.3 Baselines

Single-language fine-tuning. We fine-tune the pretrained model directly on the low-resource language only, with no high-resource exposure. This baseline uses the same low-resource row count and identical hyperparameters as Stage 2 (Dodge et al., 2020) and, so, any improvement over it reflects Stage 1 cross-lingual adaptation only.

Naive combined fine-tuning. We concatenate high-resource and low-resource data and train in one stage with Stage 1 settings. This tests whether two stages are needed or whether mixed-language training suffices. This baseline is applicable to Whisper only, as MMS allows one active language adapter per run, making mixed-language single-stage training architecturally impossible.

4.4 Evaluation Metrics

We evaluate with Word Error Rate (WER) and Character Error Rate (CER), computed via edit distance using `jiwer`. CER is particularly useful for Indic scripts, where a single word-boundary error can inflate WER despite largely correct character sequences. IndicVoices experiments use the official validation split; Shrutilipi experiments use our held-out test set.

4.5 Experimental Configurations

Each experiment is defined by four variables: the model (Whisper or MMS), the dataset (IndicVoices or Shrutilipi), the number of high-resource (HR) rows used in Stage 1, and the number of low-resource (LR) rows used in Stage 2, expressed as a percentage of the HR count. We use two sweep patterns across our experiments.

Extensive sweep. For three ordered language pairs, Bengali→Assamese, Hindi→Assamese, and Hindi→Sanskrit, we conduct thorough explorations on Whisper + IndicVoices, sweeping over ten HR row counts {500; 1,000; 2,000; 3,000; 4,000; 5,000; 10,000; 20,000; 30,000; 50,000} at three LR ratios {5%, 10%, 20%}, yielding 30 sequential configurations per pair, each paired with a single-language baseline. We restrict the extensive sweeps to IndicVoices because only this dataset provides a sufficiently large pool (50,000 samples per language) to support the full range of HR sizes. These sweeps span a close pair, a moderately related pair, and a same-script but acoustically mismatched pair, enabling the data volume analysis in RQ2 across different relatedness levels.

Focused sweep. For all other language pair and model-dataset combinations, we use a focused sweep with five high-resource (HR) row counts at a fixed 10% low-resource (LR) ratio. On IndicVoices, the HR counts are {1,000; 10,000; 20,000; 30,000; 50,000}, while on Shrutilipi they are {1,000; 2,000; 5,000; 8,000; 10,000}, reflecting the smaller data pool. The extensive sweeps already establish the relationship between transfer gains and data volume across multiple ratios; the focused sweeps validate generalisation across language pairs, models, and datasets without repeating the full grid.

Table 3 presents the complete inventory of language pairs, model–dataset settings, and the research questions each set of experiments addresses. Not all pairs are tested on every setting: Urdu is unavailable in Shrutilipi (Section 3), and MMS does not include a pretrained adapter for Sanskrit, restricting Sanskrit experiments to Whisper only.

Telugu as a low-resource target is tested only on MMS + Shrutilipi with Hindi, Bengali, and Tamil as sources, since RQ3 is already extensively covered using Assamese and Bengali as low-resource anchors and the Telugu experiments serve as additional validation. The script-versus-phonology analysis in RQ3, which compares Hindi as a source for Urdu, Sanskrit, Bengali, and Assamese, is conducted on Whisper + IndicVoices as this is the only setting where all four target languages are available under a single model–dataset combination. In total, our study covers 12 directed language pairs across over 200 individual configurations spanning both language families, both models, and both datasets.

5 Experimental Results

5.1 RQ1: Transfer Effectiveness Across Models, Datasets, and Language Pairs

Table 4 presents the WER and CER improvements of sequential fine-tuning over the single-language baseline across all language pairs and model-dataset combinations. Absolute values are provided in App. E. Several consistent patterns emerge that we next discuss.

Close pairs benefit reliably across all settings. Bn→As, the most closely related pair in our study, yields positive transfer in every model-dataset combination tested. On Whisper, WER improvements range from 3.3 pp (W+IV) to 5.5 pp (W+SL), while on MMS gains are smaller but consistent at 1.7–1.9 pp. Transfer is symmetric: As→Bn achieves comparable gains across all four settings (2.9 pp on W+IV, 5.8 pp on W+SL, 2.0 pp on M+IV, 1.6 pp on M+SL), confirming that the benefit is not direction-dependent. These results establish that sequential fine-tuning is a robust strategy when a closely related high-resource language is available, regardless of model architecture or data conditions.

Moderately related pairs show consistent but smaller gains. Hi→Bn and Hi→As produce positive transfer on MMS across both datasets, achieving 0.9–1.4 pp WER improvement. On Whisper, gains are more variable. Hi→As achieves 1.6 pp on W+IV, notably outperforming Hi→Bn (0.8 pp) on the same setting, but turns slightly negative on W+SL (-0.3 pp WER). Hi→Bn remains positive but modest across both Whisper settings (0.8 pp on W+IV, 0.1 pp on W+SL). This variability suggests that Whisper is more sensitive to linguistic distance than MMS when the source language is only moderately related.

Distant pairs degrade Whisper but not MMS. Ta→As, a fairly dissimilar pair, produces the most striking architectural contrast. On Whisper, it yields negative transfer on both datasets (-0.4 pp on W+IV, -1.2 pp on W+SL). On MMS, the same pair achieves consistent positive transfer (1.1 pp on M+IV, 0.7 pp on M+SL). Te→Bn partially fits this trend: positive on W+IV (0.6 pp) and both MMS settings (1.1 pp on M+IV, 0.7 pp on M+SL), but slightly negative on W+SL (-0.2 pp). Sa→As also shows degradation on both Whisper settings (-0.9 pp on W+IV, -1.3 pp on W+SL). This can be attributed to the architectural differences between the two models: Whisper's shared decoder is susceptible to cross-lingual interference when the source and target languages are phonologically distant, whereas MMS's per-language adapters preserve pretrained representations and buffer against such interference.

Summary. Sequential fine-tuning produces observable cross-lingual transfer regardless of model architecture (provided the model contains a shared encoder) or dataset choice, confirming that the phenomenon is not an artifact of a specific training paradigm or acoustic condition. For closely related pairs, this transfer is consistently beneficial across all settings. For distant pairs, the outcome depends on the architecture: MMS remains robust while Whisper is sensitive to language distance. The choice of model architecture, therefore, plays a critical role in determining whether cross-lingual transfer is beneficial. We also observe that Shrutilipi's clean broadcast audio amplifies both positive and negative transfer on Whisper; both findings are discussed in App. H. Additionally, for Bn→As on Whisper, sequential fine-tuning outperforms the naive combined baseline. On IndicVoices, the combined approach plateaus at higher data volumes due to data imbalance, while on Shrutilipi it remains competitive at lower volumes with sequential only marginally outperforming at higher volumes (App. F).

5.2 RQ2: Data Volume

Figure 3 in App. J presents the WER and CER improvement of sequential fine-tuning over the single-language baseline as a function of high-resource data volume, for three language pairs at three LR ratios on Whisper + IndicVoices; full per-configuration results across all 30 configura-

tions are also provided there. The three pairs represent different relatedness levels: Bn→As (close), Hi→As (moderate), and Hi→Sa (same-script but acoustically distant).

Close pairs benefit from small data volumes. Bn→As shows positive WER improvement from the earliest data points across all three LR ratios. At 5% ratio, improvement reaches 5.5 pp by 10K HR rows and remains above 4 pp thereafter. At 10% and 20%, the trend is similar though with slightly smaller peak values. The CER row corroborates this pattern: Bn→As CER improvement is positive at nearly every configuration, confirming that the transfer operates at the character level and is not solely a word-boundary effect.

Moderately related pairs require more data before transfer becomes beneficial. Hindi→Assamese starts negative at low data volumes across all ratios, particularly at 5% where it begins around -3 pp WER at 500 HR rows. It crosses zero around 3K–5K rows and gradually improves, reaching 2–3 pp at the largest volumes. A similar delayed pattern is visible in CER. This suggests that when the source language is only moderately related, the encoder initially learns features that are not well-aligned with the target, and a sufficient volume of target data is needed in Stage 2 to overcome this misalignment.

Same-script distant pairs show little to no benefit regardless of data volume. Hindi→Sanskrit remains near zero or negative across nearly all configurations. At 5% ratio, it starts at -4.5 pp WER and only approaches zero at the largest volumes. At 10%, it fluctuates between -2 and 0 for most of the range. CER tells a consistent story, remaining near zero or negative throughout. Despite sharing Devanagari script, increasing the amount of Hindi pre-training data does not translate into meaningful gains for Sanskrit recognition. Additional analysis of the effect of LR ratios on pair ordering and separation is provided in App. I.

Low-resource data is the dominant factor in absolute performance. The plots in App. K decompose the data volume effect by fixing one variable and varying the other. Across all three pairs, fixing the HR volume and increasing LR rows produces steep, consistent drops in WER and CER, with all HR-level lines converging at higher LR volumes. In contrast, fixing LR rows and increasing HR volume yields flatter curves, particularly for Hi→Sa where the lines are nearly horizontal. This asymmetry confirms that low-resource target data contributes more to final performance than high-resource source data, and that sequential fine-tuning provides an initialization advantage.

(a) Low-resource: Assamese

	W+IV		W+SL		M+IV		M+SL	
Source	WER	CER	WER	CER	WER	CER	WER	CER
Bengali	3.3	1.7	5.5	2.1	1.9	1.0	1.7	0.7
Hindi	1.6	1.1	-0.3	-0.1	1.3	0.7	0.9	0.3
Sanskrit	-0.9	-0.2	-1.3	-0.5	–	–	–	–
Tamil	-0.4	-0.1	-1.2	-0.4	1.1	0.7	0.7	0.2

(b) Low-resource: Bengali

	W+IV		W+SL		M+IV		M+SL	
Source	WER	CER	WER	CER	WER	CER	WER	CER
Assamese	2.9	2.0	5.8	2.6	2.0	0.9	1.6	0.9
Hindi	0.8	1.2	0.1	0.3	1.4	0.7	0.9	0.5
Telugu	0.6	0.7	-0.2	0.1	1.1	0.6	0.7	0.4

Table 5: WER/CER improvement (pp) with varying high-resource source while fixing low-resource target, averaged over the five focused sweep configurations (10% LR ratio). Positive = improvement. – = not tested.

5.3 RQ3: Language Relatedness and the Role of Phonetic Overlap

Table 5 presents the transfer improvements when the low-resource target is fixed and the high-resource source is varied across relatedness levels. Table 6 isolates the effect of script similarity by fixing Hindi as the source. Several patterns emerge.

A clear relatedness trend exists, but its magnitude depends on the model. With Assamese as the target (Table 5a), Bengali consistently produces the largest gains across all four settings (1.7–5.5 pp WER), followed by Hindi (0.9–1.6 pp on MMS and W+IV), while Tamil and Sanskrit yield negative transfer on both Whisper settings. The same trend appears with Bengali as the target (Table 5b): Assamese provides the strongest gains (1.6–5.8 pp), Hindi is intermediate (0.1–1.4 pp), and Telugu is the weakest (0.6–1.1 pp on W+IV and MMS, slightly negative on W+SL). On MMS, however, the relatedness effect is substantially reduced. For the Assamese anchor, the gap between the best source (Bengali, 1.9 pp) and the worst (Tamil, 1.1 pp) is only 0.8 pp, compared to a 3.7 pp gap on W+IV (3.3 vs -0.4). MMS's per-language adapters appear to buffer against the interference that drives negative transfer on Whisper, narrowing the relatedness effect.

Phonetic overlap predicts transfer success better than script similarity. Table 6 provides the

Target	Script	WER	CER
Assamese	Different	1.6	1.1
Bengali	Different	0.8	1.2
Urdu	Different	1.0	0.7
Sanskrit	Same	0.4	0.3

Table 6: Script vs phonology: WER/CER improvement (pp) on Whisper + IndicVoices with Hindi as the fixed high-resource source. Same-script pairs share Devanagari; Urdu uses Arabic script. Averaged over the five focused sweep configurations (10% LR ratio).

most direct evidence for this claim. With Hindi as the fixed source on Whisper + IndicVoices, Urdu achieves 1.0 pp WER improvement despite using an entirely different script, while Sanskrit, which has the same script as Hindi, shows only 0.4 pp improvement. Hindi and Urdu are phonetically near-identical in spoken form despite their orthographic divergence, whereas Hindi and Sanskrit differ more substantially in their spoken acoustic characteristics despite their shared script. Since ASR operates on acoustic features rather than written text, phonetic similarity in the speech signal is what the encoder can exploit during sequential fine-tuning. Script similarity is not a reliable predictor of transfer success, though it may provide a secondary benefit under clean acoustic conditions as suggested by the improved Hindi→Sanskrit performance on Shrutilipi (0.9 pp vs 0.4 pp on IndicVoices).

The relatedness effect can vanish on MMS with clean data. As noted in RQ1, when Telugu is the low-resource target on MMS + Shrutilipi, Bengali (0.7 pp), Hindi (0.6 pp), and Tamil (0.6 pp) produce nearly identical improvements. Tamil offers no advantage over the other sources. Combined with the reduced relatedness effect observed for the Assamese and Bengali anchors on MMS, this suggests that MMS's pretrained encoder already captures sufficient cross-lingual acoustic structure that the identity of the source language becomes secondary to the act of sequential adaptation itself.

Whisper on Shrutilipi amplifies the relatedness effect. The contrast between close and distant sources is most extreme on Whisper + Shrutilipi. For Assamese, Bengali achieves 5.5 pp while Tamil degrades by -1.2 pp, a spread of 6.7 pp. For Bengali, Assamese achieves 5.8 pp while Telugu shows -0.2 pp, a spread of 6.0 pp. The corresponding spreads on Whisper + IndicVoices are 3.7 pp and 2.3 pp respectively. Clean broadcast audio, by reducing acoustic variability, appears to sharpen the model's sensitivity to phonological match between the source and target languages.

Summary. Language relatedness modulates transfer effectiveness in a graded fashion, with closely related sources providing the largest gains and distant sources potentially causing harm on Whisper. The primary factor is spoken phonetic overlap rather than script similarity, as demonstrated by the Hindi→Urdu versus Hindi→Sanskrit contrast, though script sharing may provide a secondary benefit on clean data. MMS's adapter architecture substantially reduces the relatedness effect, making it a more robust choice when the degree of source–target relatedness is uncertain.

6 Conclusions and Future Work

We evaluated sequential fine-tuning as a cross-lingual transfer strategy for low-resource Indic ASR across seven languages, two architectures, two datasets, and over 200 experimental configurations. For closely related pairs such as Bengali–Assamese, sequential fine-tuning consistently improves WER over single-language baselines across all tested model-dataset combinations, and the benefit holds in both transfer directions (RQ1). The data volume analysis shows that closely related pairs benefit from transfer even at the smallest data volumes, while moderately related pairs require substantially more data volume before transfer becomes beneficial; distant pairs such as Hindi–Sanskrit show little reliable improvement regardless of data volume (RQ2). Varying the high-resource source while fixing the target reveals a graded relatedness effect, with closer sources providing larger gains. The comparison of Hindi→Urdu (different scripts, positive transfer) and Hindi→Sanskrit (same script, negligible transfer) indicates that spoken phonetic overlap is a stronger predictor of transfer success than script similarity (RQ3).

Future work includes extending to additional Indian language pairs (e.g., Dogri–Hindi, Marathi–Konkani), exploring parameter-efficient transfer methods such as LoRA or adapter-based approaches to mitigate negative transfer with distant languages, employing formal linguistic distance metrics to quantify the relatedness–transfer relationship, and evaluating on noisier real-world conditions such as conversational or telephony speech.

Limitations

Our study has several limitations. First, we use fixed hyperparameters across all language pairs; pair-specific tuning could potentially improve results for mid-range and distant pairs. Second, we evaluate only two model architectures and two datasets; other multilingual ASR models based on different encoder designs, such as Nvidia's FastConformer-based Parakeet family (Sekoyan et al., 2025), and corpora may yield different patterns. Third, Urdu experiments are limited to IndicVoices since Shrutilipi does not include Urdu, and Sanskrit experiments are limited to Whisper since MMS lacks a Sanskrit adapter. Fourth, we do not employ a formal linguistic distance metric; our relatedness categories (close, mid, distant) are based on established typological classifications rather than quantitative measures such as lang2vec (Littell et al., 2017) feature distances. Finally, we do not explore parameter-efficient methods such as LoRA or adapter-based transfer, which could provide a complementary perspective on cross-lingual transfer mechanisms.

References

Samiul Alam, Asif Sushmit, Zaowad Abdullah, Shahrin Nakkhatra, MD. Nazmuddoha Ansary, Syed Mobassir Hossen, Sazia Morshed Mehnaz, Tahsin Reasat, and Ahmed Imtiaz Humayun. 2022. Bengali common voice speech dataset for automatic speech recognition. *Preprint*, arXiv:2206.14053.

Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2021. Xls-r: Self-supervised cross-lingual speech representation learning at scale. *Preprint*, arXiv:2111.09296.

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Preprint*, arXiv:2006.11477.

Kaushal Bhogale, Abhigyan Raman, Tahir Javed, Sumanth Doddapaneni, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M. Khapra. 2023. Effectiveness of mining audio and text pairs from public data for improving asr systems for low-resource languages. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Astik Biswas, Oleg Shevelev, Amine Abdaoui, Vivek Tyagi, and Abdelmoumene Boumadane. 2025. Adapting whisper for low-resource hindi-english code-mix speech with on-the-fly augmentation & llm-synthesised data. pages 4293–4297.

Barsha Deka, Priyankoo Sarmah, and Samudra Vijaya. 2019. Assamese database and speech recognition.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *Preprint*, arXiv:2002.06305.

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. *Preprint*, arXiv:2106.07447.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Tahir Javed, Janki Atul Nawale, Eldho Ittan George, Sakshi Joshi, Kaushal Santosh Bhogale, Deovrat Mehendale, Ishvinder Virender Sethi, Aparna Ananthanarayanan, Hafsah Faquih, Pratiti Palit, Sneha Ravishankar, Saranya Sukumaran, Tripura Panchagnula, Sunjay Murali, Kunal Sharad Gandhi, Ambujavalli R, Manickam K M, C Venkata Vaijayanthi, Krishnan Srinivasa Raghavan Karunganni, and 2 others. 2024. Indicvoices: Towards building an inclusive multilingual speech dataset for indian languages. *Preprint*, arXiv:2403.01926.

Punyabrat Kalwar. 2025. Leveraging linguistic similarity with Bengali for low-resource Assamese automatic speech recognition. MS(R) thesis, Indian Institute of Technology Kanpur, April.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain. Association for Computational Linguistics.

Yunpeng Liu, Xukui Yang, and Dan Qu. 2024. Exploration of whisper fine-tuning strategies for low-resource asr. *EURASIP Journal on Audio, Speech, and Music Processing*, 2024.

Hao Ma, Zhiyuan Peng, Mingjie Shao, Jing Li, and Ju Liu. 2024. Extending whisper with prompt tuning to target-speaker asr. *Preprint*, arXiv:2312.08079.

Sayan Mandal, Sarthak Yadav, and Atul Rai. 2020. End-to-end bengali speech recognition. *Preprint*, arXiv:2009.09615.

Ankita Pasad, Ju-Chieh Chou, and Karen Livescu. 2022. Layer-wise analysis of a self-supervised speech representation model. *Preprint*, arXiv:2107.04734.

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, Alexei Baevski, Yossi Adi, Xiaohui Zhang, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. 2023. Scaling speech technology to 1,000+ languages. *Preprint*, arXiv:2305.13516.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *Preprint*, arXiv:2212.04356.

Monica Sekoyan, Nithin Rao Koluguri, Nune Tadevosyan, Piotr Zelasko, Travis Bartley, Nikolay Karpov, Jagadeesh Balam, and Boris Ginsburg. 2025. Canary-1b-v2 parakeet-tdt-0.6b-v3: Efficient and high-performance models for multilingual asr and ast. *Preprint*, arXiv:2509.14128.

Vincenzo Timmel, Claudio Paonessa, Reza Kakooee, Manfred Vogel, and Daniel Perruchoud. 2025. Fine-tuning whisper on low-resource languages for real-world applications. *Preprint*, arXiv:2412.15726.

Kumud Tripathi, Raj Gothi, and Pankaj Wasnik. 2024. Enhancing whisper's accuracy and speed for indian languages through prompt-tuning and tokenization. *Preprint*, arXiv:2412.19785.

	IndicVoices				Shrutilipi			
	Whisper		MMS		Whisper		MMS	
Language	WER	CER	WER	CER	WER	CER	WER	CER
Bengali	85.2	51.6	46.9	19.2	94.5	52.0	53.8	16.5
Assamese	115.5	84.1	56.1	24.2	106.1	61.7	47.8	13.7
Hindi	45.7	29.6	43.9	20.5	55.8	34.5	33.8	11.8
Urdu	70.2	57.2	41.4	18.0	–	–	–	–
Sanskrit	147.0	105.7	–	–	141.9	73.1	–	–
Tamil	90.0	48.9	84.4	34.3	74.0	19.6	61.8	11.9
Telugu	149.1	105.1	68.1	27.0	161.2	115.1	65.9	18.3

Table 9: Zero-shot WER/CER (%) for Whisper-large-v3 and MMS-1b-all on both datasets. – = language unavailable in that dataset.

A Training Hyperparameters

Tables 7 and 8 list the complete training configurations for both models.

Hyperparameter	**Stage 1 (HR)**	**Stage 2 / Single**
Max steps	2,000	1,000
Learning rate	1×10^{-5}	5×10^{-6}
Warmup steps	200	100
Per-device batch size	8	8
Gradient accum. steps	2	2
Effective batch size	16	16
Eval interval	500 steps	200 steps
Encoder layers unfrozen	Top 8 / 32	Top 8 / 32
Decoder	Fully trainable	Fully trainable
Precision	FP16	FP16
Speed perturbation	{0.9, 1.0, 1.1}	{0.9, 1.0, 1.1}

Table 7: Training hyperparameters for Whisper-large-v3.

Hyperparameter	**Stage 1 (HR)**	**Stage 2 / Single**
Training epochs	5	10
Learning rate	3×10^{-5}	1×10^{-5}
Warmup ratio	0.1	0.1
Per-device batch size	4	4
Gradient accum. steps	4	4
Effective batch size	16	16
Weight decay	0.005	0.01
Early stopping patience	2 epochs	3 epochs
Encoder layers unfrozen	Top 8 / 48	Top 8 / 48
CTC head	Fully trainable	Fully trainable
Precision	FP16	FP16
Speed perturbation	{0.9, 1.0, 1.1}	{0.9, 1.0, 1.1}

Table 8: Training hyperparameters for MMS-1b-all.

Hardware and software details are provided in Appendix L.

B Zero-Shot Baselines

Table 9 reports zero-shot WER and CER for both models on both datasets before any fine-tuning, providing context for the magnitude of improvements reported in the main text.

C Important Design Choices

Stage 1 uses a higher learning rate and more training steps, providing sufficient capacity to learn from the larger high-resource dataset. Stage 2 uses a lower learning rate to refine rather than overwrite the Stage 1 representations. For Whisper, Stage 2 also uses fewer steps, reflecting the smaller low-resource dataset. For MMS, Stage 2 uses more epochs than Stage 1 to ensure sufficient passes over the smaller dataset, with early stopping to prevent overfitting. Whisper training is specified in fixed steps because its cross-entropy loss degrades gradually under overtraining and periodic evaluation with best-checkpoint selection acts as implicit early stopping. MMS training is specified in epochs with explicit early stopping because its CTC loss converges more sharply and is more sensitive to the ratio of training steps to dataset size; in preliminary experiments, using fixed steps for MMS produced noticeably worse results. Both models use FP16 mixed precision and speed perturbation with factors {0.9, 1.0, 1.1}, effectively tripling the training data.

D Model Architecture Details

Table 10 summarises the key architectural differences between the two models used in this study.

	Whisper	**MMS**
Architecture	Enc–dec	Enc + CTC
Decoding	Autoregressive	Frame-level CTC
Input	128-bin log-Mel	Raw waveform
Enc. layers	32 Transformer	48 Transformer
Parameters	∼1.55B	∼965M
Pretraining	680K hrs sup.	491K hrs self-sup.
Lang. adapt.	None (shared dec.)	Per-lang. adapters

Table 10: Architectural comparison of the two models

E Detailed Results

Tables 11–14 present the absolute average WER and CER values for both sequential fine-tuning (Seq.) and single-language baseline (Sngl.), averaged across data volume configurations. The improvements reported in Table 4 are derived from these values.

	WER		CER	
HR→LR	Seq.	Sngl.	Seq.	Sngl.
Bn→As	52.7	56.0	24.7	26.4
As→Bn	37.4	40.3	16.6	18.6
Hi→As	54.4	56.0	25.3	26.4
Hi→Bn	39.5	40.3	17.4	18.6
Hi→Ur	16.2	17.2	7.4	8.1
Hi→Sa	46.5	46.9	13.9	14.2
Ta→As	56.4	56.0	26.5	26.4
Sa→As	56.9	56.0	26.6	26.4
Te→Bn	39.7	40.3	17.9	18.6

Table 11: Whisper + IndicVoices: absolute WER/CER (%), averaged over the five focused sweep configurations (10% LR ratio).

	WER		CER	
HR→LR	Seq.	Sngl.	Seq.	Sngl.
Bn→As	46.2	51.7	14.9	17.0
As→Bn	51.7	57.5	17.2	19.8
Hi→As	52.0	51.7	17.1	17.0
Hi→Bn	57.4	57.5	19.5	19.8
Hi→Sa	62.3	63.2	12.7	13.4
Ta→As	52.9	51.7	17.4	17.0
Sa→As	53.0	51.7	17.5	17.0
Te→Bn	57.7	57.5	19.7	19.8

Table 12: Whisper + Shrutilipi: absolute WER/CER (%), averaged over the five focused sweep configurations (10% LR ratio).

	WER		CER	
HR→LR	Seq.	Sngl.	Seq.	Sngl.
Bn→As	46.7	48.6	17.3	18.3
As→Bn	37.6	39.6	13.1	14.0
Hi→As	47.3	48.6	17.6	18.3
Hi→Bn	38.2	39.6	13.3	14.0
Hi→Ur	33.9	34.3	13.3	13.7
Ta→As	47.5	48.6	17.6	18.3
Te→Bn	38.5	39.6	13.4	14.0

Table 13: MMS + IndicVoices: absolute WER/CER (%), averaged over the five focused sweep configurations (10% LR ratio).

	WER		CER	
HR→LR	Seq.	Sngl.	Seq.	Sngl.
Bn→As	40.5	42.2	10.4	11.1
As→Bn	49.4	51.0	13.8	14.7
Hi→As	41.3	42.2	10.8	11.1
Hi→Bn	50.1	51.0	14.2	14.7
Ta→As	41.5	42.2	10.9	11.1
Te→Bn	50.3	51.0	14.3	14.7
Bn→Te	59.5	60.2	13.6	14.0
Hi→Te	59.6	60.2	13.6	14.0
Ta→Te	59.6	60.2	13.6	14.0

Table 14: MMS + Shrutilipi: absolute WER/CER (%), averaged over the five focused sweep configurations (10% LR ratio).

F Naive Combined Baseline Comparison

		Sequential		Naive	
HR	**LR**	WER	CER	WER	CER
		IndicVoices			
1,000	100	73.6	34.0	73.6	33.4
10,000	1,000	53.5	24.8	56.6	26.5
20,000	2,000	48.4	23.0	55.7	26.1
30,000	3,000	44.2	21.0	56.0	26.1
50,000	5,000	43.7	21.0	54.9	25.3
		Shrutilipi			
1,000	100	62.0	21.2	59.7	19.7
2,000	200	54.2	17.7	51.5	16.5
5,000	500	42.3	13.2	42.4	13.2
8,000	800	37.2	11.4	37.9	11.7
10,000	1,000	35.3	10.7	36.3	11.1

Table 15: Sequential vs naive combined fine-tuning for Bengali→Assamese on Whisper. Per-configuration WER/CER (%) at 10% LR ratio. Shrutilipi HR/LR configs differ due to smaller data pool.

Table 15 compares sequential fine-tuning against the naive combined baseline for Bengali→Assamese on Whisper. The naive combined approach concatenates both high-resource and low-resource data and trains in a single stage(with Stage 1 settings). This baseline is applicable to Whisper only, as MMS's per-language adapter design does not support mixed-language single-stage training.

IndicVoices. At the smallest configuration (1,000/100), sequential and naive perform identically on WER (73.6%) with naive marginally better on CER (33.4 vs 34.0%). However, as data volume increases, sequential improves steadily from 73.6 to 43.7% WER, while naive plateaus around 54–56% WER from the 10,000-row configuration onward. By the largest configuration (50,000/5,000), sequential outperforms naive by 11.2 pp WER and

4.3 pp CER. The naive approach's plateau can be attributed to data imbalance: as the high-resource portion grows, Bengali increasingly dominates the mixed training set, preventing the model from adequately specialising on Assamese.

Shrutilipi. The pattern differs substantially on clean broadcast data. At lower volumes (1,000/100 and 2,000/200), the naive approach actually outperforms sequential (59.7 vs 62.0% and 51.5 vs 54.2% WER respectively), suggesting that when total data is scarce, the model benefits from seeing both languages jointly rather than sequentially. The two approaches converge at the 5,000/500 configuration (42.3 vs 42.4% WER), and sequential gains only a marginal advantage at higher volumes (35.3 vs 36.3% WER at 10,000/1,000). Unlike IndicVoices, naive does not plateau on Shrutilipi, improving steadily across all configurations. The reduced acoustic variability in broadcast audio may allow the model to extract useful cross-lingual signal even from mixed training, diminishing the advantage of the two-stage approach.

G MMS Encoder Layer Ablation

Table 16 compares unfreezing the top 8 versus top 12 encoder layers in MMS-1b-all for Bengali→Assamese, averaged across all focused sweep configurations at 10% LR ratio.

		Top-8		Top-12	
Dataset	**Method**	WER	CER	WER	CER
IndicVoices	Sequential	46.7	17.3	45.6	16.8
IndicVoices	Single-lang.	48.6	18.3	47.3	17.7
Shrutilipi	Sequential	40.5	10.4	40.4	10.4
Shrutilipi	Single-lang.	42.2	11.1	41.0	10.7

Table 16: MMS ablation on encoder unfreezing depth for Bengali→Assamese, averaged across all data configurations (10% LR ratio).

On IndicVoices, unfreezing 12 layers yields a modest 1.1 pp WER improvement over 8 layers for sequential (46.7→45.6%) and 1.3 pp for the single-language baseline (48.6→47.3%). On Shrutilipi, the difference is negligible (40.5 vs 40.4% WER for sequential, identical CER at 10.4%). Crucially, the sequential advantage over the single-language baseline persists at both layer counts on both datasets, confirming that the benefit of cross-lingual transfer is independent of the unfreezing depth. Since the top-8 configuration is more computationally efficient and the additional gains from 12 layers are marginal, we retain top-8 as the default for all MMS experiments.

H Additional RQ1 Observations

Shrutilipi amplifies transfer effects on Whisper. An unexpected finding is that Shrutilipi's clean broadcast audio magnifies both positive and negative transfer on Whisper relative to IndicVoices. Close pairs improve more on Shrutilipi (Bn→As: 5.5 vs 3.3 pp; As→Bn: 5.8 vs 2.9 pp), while distant pairs degrade more (Ta→As: -1.2 vs -0.4 pp; Te→Bn: -0.2 vs 0.6 pp; Sa→As: -1.3 vs -0.9 pp). Hindi→Sanskrit is a notable exception: its modest gain on W+IV (0.4 pp) increases to 0.9 pp on W+SL. More broadly, the reduced acoustic variability in broadcast recordings appears to amplify whatever transfer signal exists, whether positive or negative. This effect is not observed on MMS, where improvements remain comparatively stable across datasets.

Telugu as target shows a flat relatedness effect on MMS. When Telugu is the low-resource target on MMS+Shrutilipi, all three source languages (Bengali, Hindi, Tamil) produce nearly identical improvements of 0.6–0.7 pp WER, despite spanning different relatedness levels. Tamil, a fairly close language, provides no measurable advantage over Bengali or Hindi, both of which are not so close. This suggests that on clean broadcast data, MMS's pretrained encoder already captures sufficient cross-lingual acoustic structure that the choice of source language matters less than the act of sequential adaptation itself.

I Additional RQ2 Observations

The pair ordering is consistent across LR ratios. In all six panels, the ordering Bn→As > Hi→As > Hi→Sa holds. Changing the LR ratio affects the magnitude of improvements but not which pairs benefit. This consistency across ratios indicates that the relationship between linguistic relatedness and transfer effectiveness is stable and not dependent on a specific data configuration.

Higher LR ratios reduce the differences between pairs. At 20% ratio, the three pairs Bn→As, Hi→As, and Hi→Sa are closer together than at 5%, in both WER and CER. With more LR data available in Stage 2, the model can better recover from a suboptimal Stage 1 initialization, reducing the gap between close and distant pairs. Conversely,

	(a) 5% LR Ratio					(b) 10% LR Ratio					(c) 20% LR Ratio			
	Seq.		Single			Seq.		Single			Seq.		Single	
HR / LR	WER	CER	WER	CER	**HR / LR**	WER	CER	WER	CER	**HR / LR**	WER	CER	WER	CER
500 / 25	83.3	39.6	84.3	42.5	500 / 50	78.8	36.8	78.7	37.2	500 / 100	73.8	33.7	74.6	34.6
1k / 50	77.1	35.8	78.7	37.2	1k / 100	73.6	34.0	74.6	34.6	1k / 200	68.2	31.0	70.2	32.0
2k / 100	72.9	33.9	74.6	34.6	2k / 200	67.8	30.8	70.2	32.0	2k / 400	62.8	28.6	62.7	29.0
3k / 150	69.8	31.7	70.5	32.5	3k / 300	64.6	29.5	67.9	31.0	3k / 600	59.2	26.9	62.0	28.4
4k / 200	67.4	30.9	70.2	32.0	4k / 400	62.9	28.7	62.7	29.0	4k / 800	56.0	25.8	59.4	27.5
5k / 250	64.7	29.6	69.8	32.3	5k / 500	60.5	27.3	63.2	28.8	5k / 1k	54.2	25.1	57.1	26.5
10k / 500	57.7	26.3	63.2	28.8	10k / 1k	53.5	24.8	57.1	26.5	10k / 2k	49.8	23.6	52.4	24.9
20k / 1k	51.4	23.6	57.1	26.5	20k / 2k	48.4	23.0	52.4	24.9	20k / 4k	44.3	21.1	48.6	23.1
30k / 1.5k	47.7	22.2	52.8	24.5	30k / 3k	44.2	21.0	47.5	22.5	30k / 6k	41.4	19.8	44.7	21.0
50k / 2.5k	46.6	22.4	50.6	24.2	50k / 5k	43.7	21.0	48.6	23.4	50k / 10k	40.6	19.2	44.7	21.2

Table 17: Bengali→Assamese extensive sweep on Whisper + IndicVoices. WER/CER (%).

	(a) 5% LR Ratio					(b) 10% LR Ratio					(c) 20% LR Ratio			
	Seq.		Single			Seq.		Single			Seq.		Single	
HR / LR	WER	CER	WER	CER	**HR / LR**	WER	CER	WER	CER	**HR / LR**	WER	CER	WER	CER
500 / 25	87.1	42.8	84.3	42.5	500 / 50	79.8	38.3	78.7	37.2	500 / 100	75.9	35.4	74.6	34.6
1k / 50	79.1	36.7	78.7	37.2	1k / 100	76.5	35.0	74.6	34.6	1k / 200	69.4	31.8	70.2	32.0
2k / 100	74.6	34.1	74.6	34.6	2k / 200	70.0	32.1	70.2	32.0	2k / 400	63.4	29.2	62.7	29.0
3k / 150	72.9	33.3	70.5	32.5	3k / 300	65.8	30.0	67.9	31.0	3k / 600	60.0	27.5	62.0	28.4
4k / 200	69.9	32.2	70.2	32.0	4k / 400	63.0	28.9	62.7	29.0	4k / 800	57.4	26.3	59.4	27.5
5k / 250	67.9	31.0	69.8	32.3	5k / 500	61.9	28.3	63.2	28.8	5k / 1k	55.6	25.6	57.1	26.5
10k / 500	60.7	27.9	63.2	28.8	10k / 1k	54.9	25.1	57.1	26.5	10k / 2k	49.8	23.4	52.4	24.9
20k / 1k	54.2	25.0	57.1	26.5	20k / 2k	49.4	23.2	52.4	24.9	20k / 4k	45.1	21.3	48.6	23.1
30k / 1.5k	51.2	23.8	52.8	24.5	30k / 3k	47.1	22.2	47.5	22.5	30k / 6k	43.4	20.4	44.7	21.0
50k / 2.5k	48.2	22.9	50.6	24.2	50k / 5k	44.1	20.9	48.6	23.4	50k / 10k	42.7	20.0	44.7	21.2

Table 18: Hindi→Assamese extensive sweep on Whisper + IndicVoices. WER/CER (%).

at 5% ratio the separation is widest, as the limited LR data leaves the model more dependent on the quality of the Stage 1 pre-adaptation.

J Extensive Sweep Results and Plot

Tables 17–19 and Figure 3 present the full extensive sweep results for the three language pairs used in the RQ2 data volume analysis, all on Whisper + IndicVoices. Each table covers 10 high-resource sizes at three LR ratios (5%, 10%, 20%), yielding 30 configurations per pair. Figure 3 visualizes the WER and CER improvement over the single-language baseline across all three ratios.

K Data Volume Analysis Plots

Figures 4–9 decompose the data volume effect for the three extensive sweep pairs on Whisper + IndicVoices. For each pair, we present two views: fixing the high-resource volume and varying the low-resource volume, and fixing the low-resource volume and varying the high-resource volume. Together, these plots show that low-resource data is the dominant factor in absolute performance across all three pairs, while high-resource data provides diminishing returns.

L Reproducibility Details

L.1 Hardware and Software

All experiments were conducted on a server running Ubuntu 24.04.2 LTS, equipped with an AMD EPYC 9534 64-core processor and 1.2 TB of system memory. The server has 8× NVIDIA RTX A6000 GPUs (48 GB VRAM each), though each experiment used a single GPU. Training was performed with FP16 mixed precision using PyTorch 2.8.0 (CUDA 12.8) and Hugging Face Transformers 4.57.0. All experiments used a fixed random seed of 42.

L.2 Implementation Notes

For Whisper experiments, gradient checkpointing was disabled on the encoder because partially frozen layers cause runtime errors when gradient checkpointing is enabled. Gradient checkpointing was enabled on the decoder to reduce memory usage.

For the MMS encoder layer ablation described in App. G, the top-8 configuration unfreezes layers 40–47 and the top-12 configuration unfreezes layers 36–47, with all other hyperparameters held constant as

	(a) 5% LR Ratio					(b) 10% LR Ratio					(c) 20% LR Ratio			
	Seq.		Single			Seq.		Single			Seq.		Single	
HR / LR	WER	CER	WER	CER	**HR / LR**	WER	CER	WER	CER	**HR / LR**	WER	CER	WER	CER
500 / 25	78.9	24.8	74.1	25.0	500 / 50	73.0	23.0	70.8	22.2	500 / 100	67.3	19.9	67.5	21.4
1k / 50	72.0	21.9	70.8	22.2	1k / 100	67.0	20.4	67.5	21.4	1k / 200	62.2	20.3	60.2	17.5
2k / 100	67.4	20.0	67.5	21.4	2k / 200	61.8	18.2	60.2	17.5	2k / 400	55.3	16.1	53.3	15.9
3k / 150	63.5	18.7	62.8	19.7	3k / 300	58.8	17.4	57.6	17.1	3k / 600	52.1	15.1	49.9	14.1
4k / 200	61.0	17.8	60.2	17.5	4k / 400	54.8	16.1	53.3	15.9	4k / 800	49.0	14.1	47.7	14.0
5k / 250	58.9	17.0	56.8	16.4	5k / 500	53.8	15.6	51.5	14.6	5k / 1k	48.2	14.0	46.7	13.3
10k / 500	52.4	15.3	51.5	14.6	10k / 1k	47.0	13.3	46.7	13.3	10k / 2k	41.8	12.2	42.0	12.3
20k / 1k	46.6	13.1	46.7	13.3	20k / 2k	41.5	12.2	42.0	12.3	20k / 4k	38.5	11.8	37.8	11.2
30k / 1.5k	45.3	13.6	43.6	12.6	30k / 3k	39.3	11.8	41.3	12.8	30k / 6k	35.5	10.6	36.2	10.9
50k / 2.5k	40.6	12.0	41.8	12.3	50k / 5k	37.5	11.8	37.1	11.0	50k / 10k	36.0	11.0	35.6	10.6

Table 19: Hindi→Sanskrit extensive sweep on Whisper + IndicVoices. WER/CER (%).

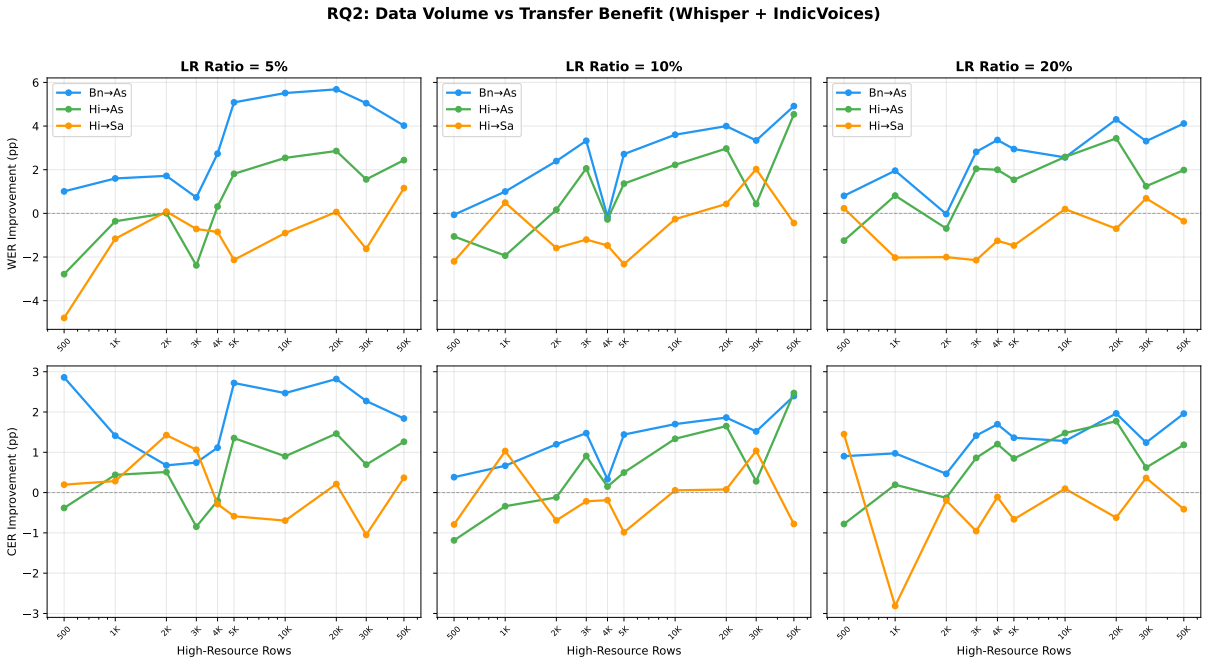

Figure 3: WER (top) and CER (bottom) improvement over the single-language baseline as a function of high-resource data volume on Whisper + IndicVoices. Columns correspond to LR ratios of 5%, 10%, and 20%. The dashed line indicates zero improvement.

specified in Table 8.

Whisper evaluation used forced decoder IDs to set the low-resource language, while MMS evaluation loaded the corresponding language adapter. All WER and CER values were computed using the `jiwer` library integrated with HuggingFace's `evaluate` framework.

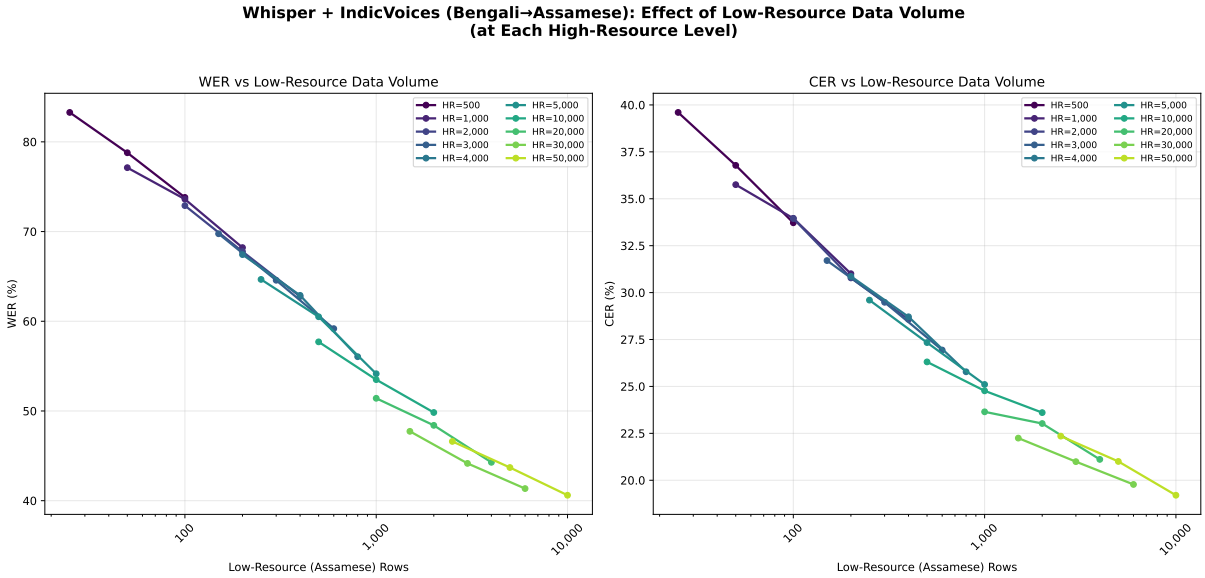

Figure 4: Bengali→Assamese: WER/CER vs low-resource data volume at each high-resource level (Whisper + IndicVoices).

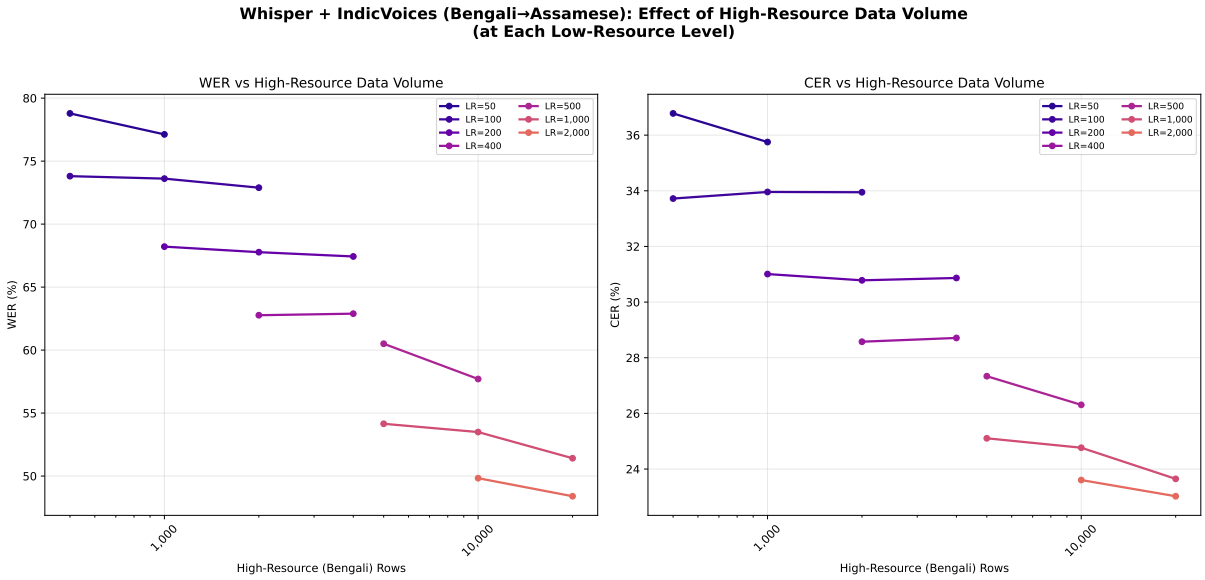

Figure 5: Bengali→Assamese: WER/CER vs high-resource data volume at each low-resource level (Whisper + IndicVoices).

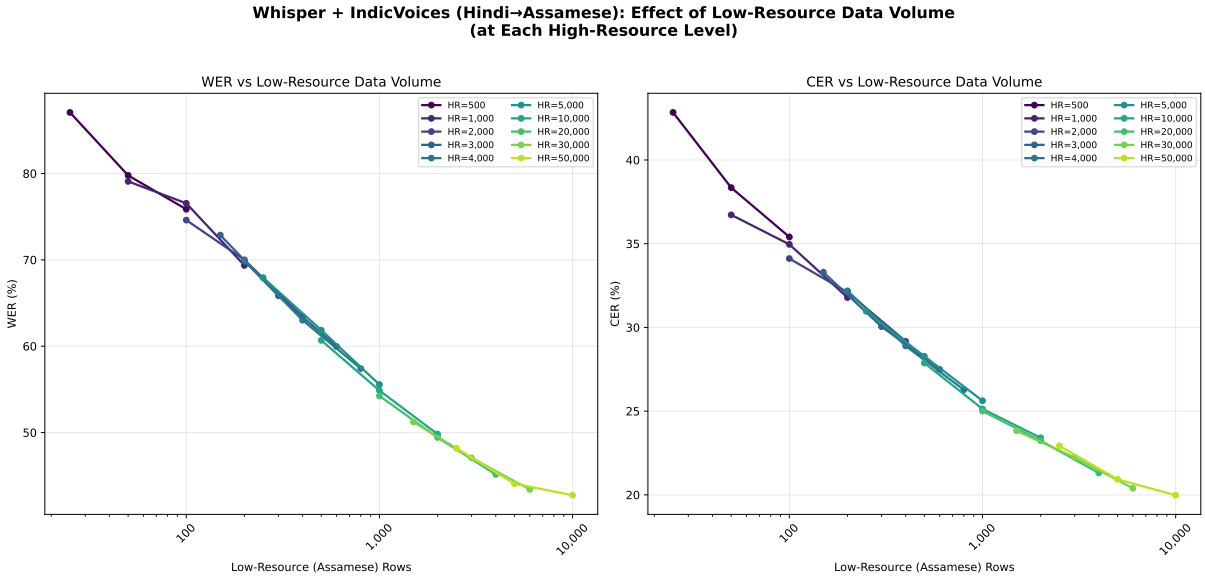

Figure 6: Hindi→Assamese: WER/CER vs low-resource data volume at each high-resource level (Whisper + IndicVoices).

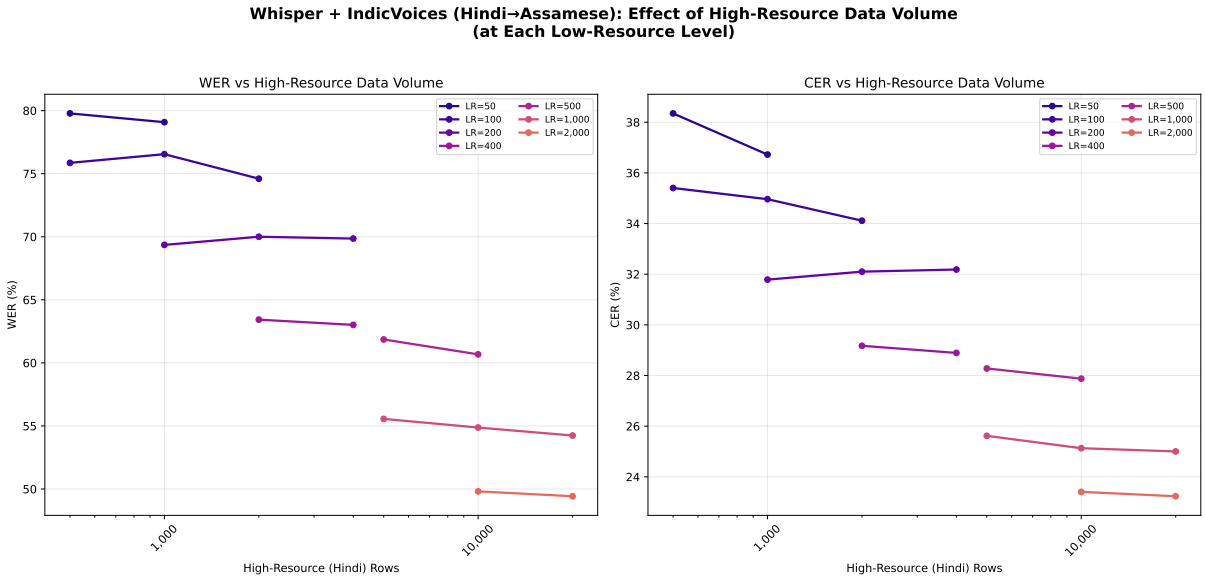

Figure 7: Hindi→Assamese: WER/CER vs high-resource data volume at each low-resource level (Whisper + IndicVoices).

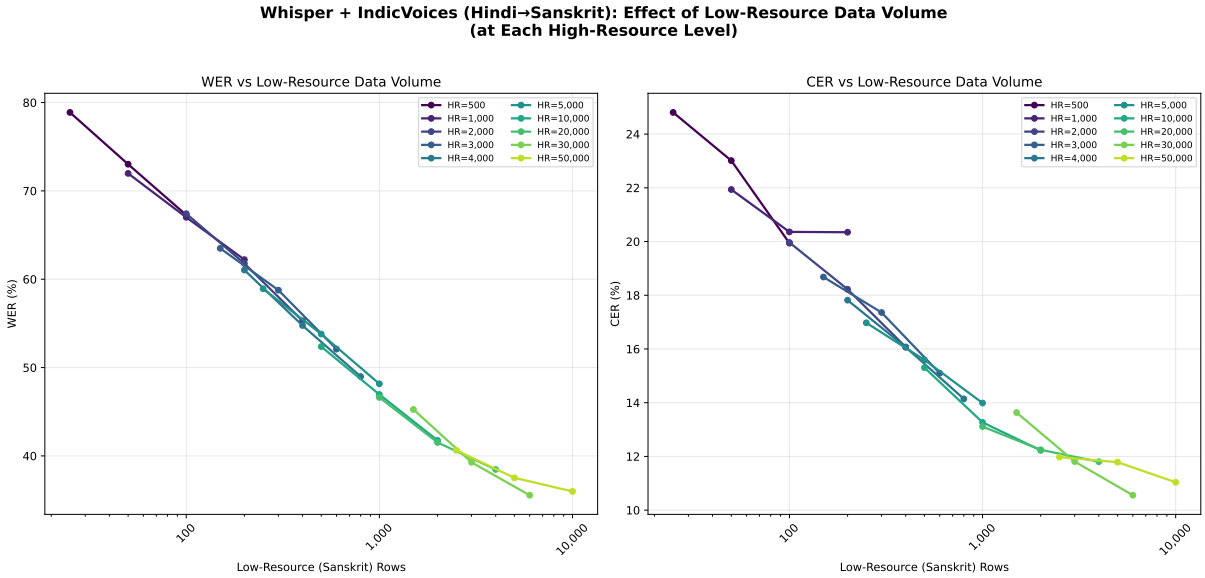

Figure 8: Hindi→Sanskrit: WER/CER vs low-resource data volume at each high-resource level (Whisper + IndicVoices).

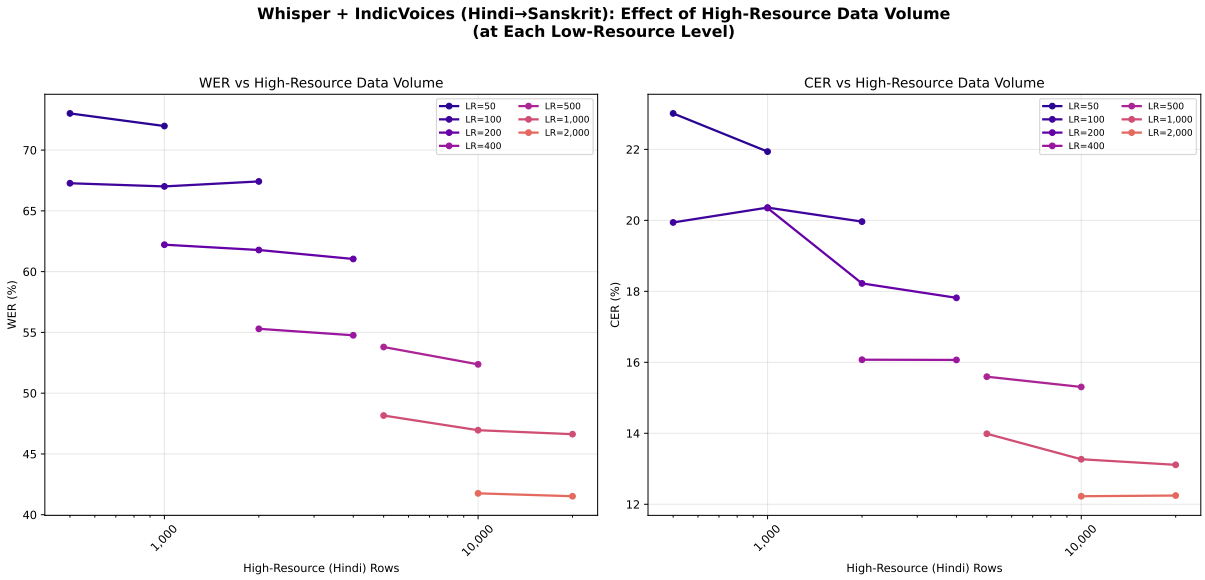

Figure 9: Hindi→Sanskrit: WER/CER vs high-resource data volume at each low-resource level (Whisper + IndicVoices).