

Urdu Sukhan ASR: Accurate Urdu Speech Recognition at Half the Size of Whisper large-v3

Asad Shehzad
Loudness.ai

Technical report LA-RB-2026-004, version 1.0, 21 October 2026
doi:10.5281/zenodo.23518703

Abstract

We describe Urdu Sukhan ASR v1, an Urdu speech recognition model that also handles English and Urdu-English code-switching, and how it was trained and tested, including the rounds that failed. The model is a full fine-tune of Whisper large-v3-turbo on about 110 hours of transcribed audio per pass: public Urdu corpora, two audiobooks read by two narrators, with transcripts proofread by the author, Urdu-English clips we assembled, and no-speech audio. Against the base model with the same decoding, it makes fewer word errors on all seven held-out Urdu test sets, on average 39% fewer: from 30.9% to 12.7% on IndicVoices, from 39.8% to 28.1% on Common Voice speakers never seen in training and from 21.0% to 16.2% on FLEURS. On one- or two-word replies the error falls from 88.9% to 23.6%, on clips that switch from Urdu to English from 53.6% to 19.7%, and English stays at 4.9%. Two of the six training rounds learned to hear words in faint hiss; we trace this to training speech that was fainter than the hiss in our silence test, and we show that the single word Whisper models write on silence by default comes from a decoding rule that forbids an empty answer. With that rule off, our model writes nothing on 100 noise, music and silence clips and no speech score changes. We report speed on four machines in whisper.cpp, faster-whisper and Core ML, and a compression experiment whose 16-layer student was 1.9 times faster but not accurate enough to release.

1 Introduction

Urdu is spoken by well over 200 million people, but open speech recognition for it lags far behind English. General multilingual models such as Whisper [1] saw much less Urdu than English in training, and a recent benchmark of Urdu recognition [2] found high error rates for Whisper, MMS and Seamless models, especially on conversational speech. Applications that transcribe Urdu (dictation, audiobooks, interviews, voice notes) also meet audio that public test sets rarely cover: quiet and phone-quality recordings, one-word replies, English words and sentences inside Urdu speech, and silence, on which Whisper-style models are known to invent text [3].

This report describes how we adapted Whisper large-v3-turbo to this setting with a laptop and one rented 16 GB GPU. The result keeps turbo’s speed and its size, 809 million parameters, about half the 1.55 billion of Whisper large-v3. Our contributions are:

1. **Urdu Sukhan ASR v1**, in whisper.cpp (q8_0, q5_0, q4_0) and CTranslate2 formats, with a large improvement over the base model on every Urdu test set we have (Section 6.1).
2. A **proofreading loop** in which each model drafts transcripts of two audiobooks and the author corrects them, with an Urdu house style applied to every training and test sentence (Section 3.2).

3. The **complete training history**: six rounds, two of which collapsed on quiet audio, five weight averages that were rejected, and the stress tests that chose the released checkpoint (Sections 4 and 6.3).
4. An **analysis of invented text on silence**: three sources of faint labelled audio in training, and a decoding default that forces every Whisper model to write at least one word (Section 6.5).
5. **Speed measurements** on four machines and three runtimes, including the cost of language auto-detection, of each quantisation format and of thread counts (Section 7).
6. A **compression experiment** by encoder pruning and distillation, reported as a partial result (Section 8), and **lessons on evaluating speech models** that changed our own numbers along the way (Section 9).

2 Background

Whisper. Whisper [1] is an encoder–decoder transformer trained on 680,000 hours of weakly labelled audio. The encoder reads a log-mel spectrogram of a fixed 30-second window; the decoder writes text tokens after a short prefix that names the language and the task. Whisper large-v3-turbo keeps the 32-layer encoder of large-v3 (128 mel bins) and reduces the decoder from 32 layers to 4, for 809 million parameters. Its decoder is fast; its encoder dominates the cost on a CPU.

Urdu speech data. Public transcribed Urdu is small and uneven. FLEURS [4] has about seven hours of read Wikipedia sentences for Urdu, Common Voice [5] has crowd-read sentences from a limited set of speakers, and IndicVoices [6] adds read and conversational speech from many speakers in India. AI4Bharat’s Rasa corpus has an Urdu part. Several community datasets on Hugging Face are the work of single authors.

Adaptation and distillation. LoRA [9] trains small low-rank updates instead of all weights. Weight averaging of fine-tuned models (“model soups”) [10] can combine their strengths. Knowledge distillation [11] trains a small model to match a large one; Distil-Whisper [12] keeps Whisper’s whole encoder and shrinks the decoder. Because turbo’s decoder is already small, a faster turbo on a CPU must remove encoder layers instead (Section 8).

Runtimes. whisper.cpp [13] runs Whisper on CPUs and Apple GPUs; its q8_0, q5_0 and q4_0 formats store weights in 8, 5 and 4 bits with a scale per block of 32. faster-whisper [14] runs Whisper on CTranslate2. Silero VAD [15] is a small voice activity detector that whisper.cpp can run before the model.

3 Data

3.1 Sources and terms

Table 1 lists the training sources of the released model. Every source except the two audiobooks is public. The UAT dataset carries no public licence and its transcripts were made by a machine, so we used only its audio, with transcripts made by our own earlier model (Section 3.4), by agreement with its developers. The nzs73 dataset’s own licence allows training speech models with credit

to its author and forbids misuse such as deepfakes. We reviewed and left out four further sets: a Mozilla Urdu text-to-speech set (non-commercial licence), a synthetic text-to-speech corpus, a podcast collection without a licence and an emotion corpus without transcripts.

Source	Terms	Clips per pass
IndicVoices Urdu [6] (AI4Bharat)	CC BY 4.0	20,000
Common Voice Urdu [5] (Mozilla; Kaggle repack)	CC0	12,000
UAT, audio only, transcribed by our round-4 model	no public licence; by agreement	10,000
UrduSpeech (Urdu part of AI4Bharat’s Rasa)	CC BY 4.0	4,999
nzg73 Urdu_Audio-Realistic (one speaker)	its own licence, credit required	3,000
FLEURS [4] Urdu train and dev; English train	CC BY 4.0	2,370; 2,000
Two audiobooks read by two narrators; transcripts proofread by the author	not redistributed	1,214 (521 $\times$ 2 or $\times$ 4)
Urdu–English clips joined from FLEURS and Common Voice	as their sources	782
No-speech: MUSAN [7] noise and music, generated hiss and hum	CC BY 4.0	2,800
Total		59,165

Table 1: Training data of Urdu Sukhan ASR v1 (round 6): about 110 hours in one pass, sampled from 87,571 cleaned clips (about 151 hours).

3.2 The audiobooks and the proofreading loop

Two audiobooks, each read by a different narrator, were used: an Urdu translation of a self-help book (3.97 hours) and an Urdu novel (about 1.2 hours). They were cut at pauses with Silero VAD into clips of at most 25 seconds (1,037 clips, 305 minutes of speech) and drafted by the base model in whisper.cpp on the laptop’s GPU in 59 minutes, with a confidence for every word. The author corrected the drafts in a browser tool built for the purpose: uncertain words are shown in red or yellow, any word plays on click, words fixed automatically are shown in blue, and a header shows the drafts’ error rate on the clips finished so far, the most direct measure of a model on these recordings. After every five finished clips, the corrections were reviewed and turned into rules for the remaining drafts: 765 rules by the end, each checked against every draft it would touch, and written as whole phrases when the wrong form is itself a real word.

Each round’s model redrafted the unfinished clips. On the first book, the author changed 6.1% of the words of the base model’s drafts (295 clips) and 3.4% of the round-1 drafts (188 clips). In all, 579 clips (2.85 hours) were proofread: 521 for training and 58 held out by recording position (48 from the first book, 10 from the second); 318 clips of the first book were skipped and 140 of the second keep round-3 drafts. The review policy changed four times at the author’s request, ending with “transcribe what the speaker says”: spelling, names and house style are corrected, but a speaker’s wording is never “improved”, so that the model does not learn to correct speakers.

3.3 House style

Urdu has several accepted spellings for common words, and training on all of them teaches a model to be inconsistent. We applied one house style, the author’s own, to every training and test sentence and to both sides before scoring: for example لے (liye) rather than لئے (chahiye) rather than چاہئے (unhen) rather than انھیں, and compound verbs written as two words (*kar diya*, not *kardiya*). Punctuation was removed from all training text (298 of the 299 FLEURS test sentences had some), Western digits were written as Urdu digits and spoken years as digits. This alone lowered the base model’s FLEURS error from 22.5% to 21.0%, so spelling variation was worth about 1.5 points.

3.4 Transcribing UAT with our own model

UAT’s transcripts matched the base Whisper model’s output word for word on 21 of 24 sampled clips, including its errors, so they were machine-made. We transcribed its 20,444 clips with our round-4 model and kept a clip when the mean of the per-token maximum log-probability gave a confidence of at least 0.6, the text was not repetitive and its length fitted the audio: 20,235 clips. Round 5 used those with confidence ≥ 0.8 and round 6 those with confidence ≥ 0.85 (19,721).

3.5 Cleaning

Before round 6, a clip was removed when it was shorter than 0.5 or longer than 30 seconds, had no text, held more than 22 letters per second (25 for clips under 1.5 seconds) or fewer than 1.5 per second when longer than 4 seconds (audio that cannot hold its text), or was an Urdu clip whose text was less than 60% Urdu script. This removed 27 Common Voice clips whose audio ended before their text, 81 IndicVoices clips in another script, 488 UAT clips with unsure transcripts and a few dozen others. No-speech and code-switch clips were exempt.

3.6 No-speech and code-switch clips

Round 6 added 2,800 clips with empty transcripts: 2,500 segments of MUSAN noise and instrumental music and 300 generated clips of digital silence, microphone hiss (−80 to −45 dBFS) and mains hum. It also added 782 clips that join an Urdu clip (FLEURS or Common Voice) and an English FLEURS clip with a short pause, Urdu first in 70% of them, transcribed with the English part in English letters under the Urdu language tag.

3.7 Test and stress sets

Seven test sets were never used in training: the IndicVoices Urdu validation set (300 clips, at most three per speaker), Common Voice Urdu clips from 15 speakers absent from our training data (260), the FLEURS Urdu test set (299), the UrduSpeech test split (100), held-out nsg73 clips (300; the same single speaker as its training part), and held-out chapters of the two audiobooks (48 and 10 clips). Nine stress sets probe robustness: noise (40) and instrumental music (30) from MUSAN files that we hold out (every file whose name’s MD5 is divisible by ten); silence, hiss and mains hum (30); quiet speech, test clips turned down by 30 dB (40); the same clips through a phone line (40); one- or two-word replies such as جی (ji) and ہاں (haan) (40); English (30); an Urdu test clip followed by an English one (20); and 25–30 seconds of joined test speech (15).

4 Training

4.1 Settings shared by all rounds

All rounds trained with AdamW (weight decay 0.01; the 8-bit version from bitsandbytes for full fine-tunes), 10% linear warm-up and linear decay to zero, gradient clipping at 1.0 and seed 0, with labels `<|ur|><|transcribe|><|notimestamps|> text <|endoftext|>`. LoRA runs used rank 32, alpha 64 and dropout 0.05 on the attention projections and feed-forward layers, in bf16; full fine-tunes kept fp32 master weights with bf16 computation and gradient checkpointing. From round 3 on, 30% of clips with known word timings were trained in timestamp form, so that decoding with timestamps keeps working.

4.2 Six rounds

Table 2 summarises the rounds. Rounds 1–3 ran on the laptop (Apple M4, 16 GB) as LoRA on the decoder only, with the encoder’s outputs computed once and cached, as Distil-Whisper does; they taught the model the two audiobook voices. Rounds 4–6 ran on one rented RTX A4000 (16 GB). Round 4 trained LoRA on the encoder and the decoder with audio augmentation and added nzg73 and all of FLEURS. Round 5 was the first full fine-tune of all 809 million weights and added Common Voice and transcribed UAT audio. Round 6, the released model, added IndicVoices, UrduSpeech, FLEURS English, the code-switch and no-speech clips and MUSAN background noise, used half of round 5’s learning rate, and ran a single pass with a checkpoint every 925 steps.

Round	Method	Data per pass	Steps $\times$ batch	LR	Time
R1	LoRA, decoder (laptop)	218 book clips $\times$ 4, 600 FLEURS	368×8	2e-4	40 min
R2	LoRA, decoder (laptop)	365 book clips $\times$ 3, 600 FLEURS	422×8	2e-4	73 min
R3	LoRA, decoder, timestamps (laptop)	435 + 85 book clips ($\times$ 2, $\times$ 5), 600 FLEURS	472×8	2e-4	67 min
R4	LoRA, encoder and decoder	books, nzg73 3,000, FLEURS 2,109; 6,409 clips	800×16	1e-4	99 min
R5	full fine-tune	R4 data, Common Voice 8,000, UAT 3,000; 17,409	$2,176 \times 16$	1e-5	164 min
R6	full fine-tune	Table 1; 59,165 clips	$3,697 \times 16$	5e-6	268 min

Table 2: Training rounds. R1–R3 ran on the laptop’s GPU, R4–R6 on an RTX A4000. Round 6 trained for 4.5 GPU-hours (4.4 s per step).

4.3 Augmentation

From round 4, half of the clips were augmented: speed 0.9–1.1, a gain change, room echo (an exponentially decaying noise impulse response) on 30% and background noise on 60% at 5–30 dB below the speech. Rounds 5 and 6 added SpecAugment [8] (5% time and frequency masks). Round 6 changed three things: one augmented clip in five was made “very quiet” (12–30 dB down), 15% went through a phone line (8 kHz with clipping), and 70% of the background noise came from 1,590 MUSAN recordings. Section 6.5 shows that two of these settings were mistakes.

4.4 Choosing the checkpoint

Before round 6 we fixed a pass mark: no loops, invented text on at most a few no-speech clips, better than the R4+R5 average on the quiet, phone, short, mixed and long stress sets, and at least as good on the seven test sets. Each checkpoint was scored on all sixteen sets; the stress sets ruled out checkpoints that looked good on the test sets (Section 6.4).

5 Experimental setup

Hardware. Laptop: Apple MacBook Air with an M4 chip and 16 GB of memory, used for drafting, proofreading, rounds 1–3 and whisper.cpp tests with Metal and Core ML. GPU server (rented): NVIDIA RTX A4000 with 16 GB, whose CPU is an Intel Xeon E3-12xx v2 from 2012 (Ivy Bridge, AVX but no AVX2). Cloud server: 12 virtual AMD EPYC cores at 2.0 GHz (AVX2) with other services running, the kind of machine a product would use.

Software. PyTorch 2.14.1 with Transformers and bitsandbytes 0.50.2; whisper.cpp v1.9.5 (CPU, Metal and Core ML builds) with Silero VAD v5.1.2; CTranslate2 4.5.0 for conversion and faster-whisper 1.2.1; coremltools 9.0; Chartist 2.1.0 [16] for the figures.

Metrics. We report words wrong (word error rate, WER) and letters wrong (character error rate, CER, spaces ignored) after the same normalisation of reference and output: punctuation, diacritics, Arabic and Urdu letter forms, digit systems and hamza variants are made consistent and the house style is applied to both sides. A *loop* is an output more than twice as long as the reference (and at least seven words), or a phrase of one to six words repeated four or more times in a row more often than the speaker said it; tables computed with an earlier, looser rule over-count loops on speech (11 instead of 1 on IndicVoices for the released model), so we quote loops only for no-speech clips and where recounted. *Invented text* is two or more words on a no-speech clip; *any word* counts no-speech clips with at least one word. On the second audiobook’s test chapter we also count how many of its 23 name occurrences are spelled exactly right.

Decoding. GPU evaluations use Transformers’ `generate` with greedy decoding, the language set to Urdu (`auto` for the English and mixed stress sets) and no timestamps. whisper.cpp evaluations run as an application would: beam search of width 5 with temperature fallback, no timestamps, no carried-over context, automatic language detection and the `q8_0` format, with three variants: default settings, the Silero voice detector first (threshold 0.5, minimum silence 2 s, padding 300 ms), and blank suppression turned off (Section 6.5).

6 Results

6.1 Against the base model

Table 3 and Figure 1 compare the base model and ours with identical decoding. Our model is better on every Urdu set. The largest relative gains are on conversational speech from many speakers (IndicVoices, 59% fewer word errors), the single speaker of nzg73 (67%), short replies (73%) and Urdu–English clips (63%). FLEURS, read Wikipedia text with many names and technical words, improves least among the public sets (23%), as does the second audiobook (23%), whose narrator reads many names (Section 6.6). English is unchanged at 4.9%.

Test set (clips)	WER (%)		CER (%)	
	Base	Ours	Base	Ours
IndicVoices Urdu validation (300)	30.9	12.7	21.9	7.5
Common Voice, unseen speakers (260)	39.8	28.1	25.3	16.4
FLEURS Urdu test (299)	21.0	16.2	8.3	6.3
UrduSpeech test split (100)	14.5	8.8	5.8	2.1
nzg73, held-out clips (300)	4.8	1.6	1.7	0.6
Audiobook 1, held-out chapters (48)	5.9	4.0	2.6	1.6
Audiobook 2, held-out chapter (10)	17.2	13.2	6.0	4.8
Quiet speech (40)	35.6	23.5	24.8	15.9
Phone-band speech (40)	36.2	25.0	21.8	14.8
One- or two-word replies (40)	88.9	23.6	66.7	27.2
Urdu then English in one clip (20)	53.6	19.7	49.8	14.2
Continuous speech, 25–30 s (15)	35.0	29.8	26.0	20.7
English, FLEURS (30)	4.9	4.9	2.0	1.7
Invented text: noise / music / silence	15 / 16 / 1	6 / 4 / 3		

Table 3: Base model (Whisper large-v3-turbo) and Urdu Sukhan ASR v1 on the held-out test sets (top) and the stress sets (bottom); Transformers, greedy decoding, lower is better, the better value in bold. The last row counts no-speech clips with two or more words.

6.2 Examples

Error rates hide what the errors look like. We ran 47 public test clips with speech, and two without, through both models as an application would (whisper.cpp, q8_0, beam 5, temperature fallback). Over these clips, our model was better on 27, worse on 4 and the same on 16 (19.1% against 37.8% words wrong). Table 4 shows short clips that we picked by hand to show where the fine-tune helps most, from the top of a ranking of all clips by words fixed, and one of the 4 clips where it did worse; the model card has them with their audio. The base model turns short replies into other words, even “music” for a spoken “hello” and “Allah, Allah” for “separately”. The largest gains were on clips that switch from Urdu to English, too long for the table: on one, the base model wrote four English words for a 28-word clip; on another, it translated the Urdu half into English instead of writing it down. Ours transcribed both, with two wrong words in all. It still loses on some word spacing, as the last example shows. On silence, allowing an empty result only helps our model: the base model still writes a word.

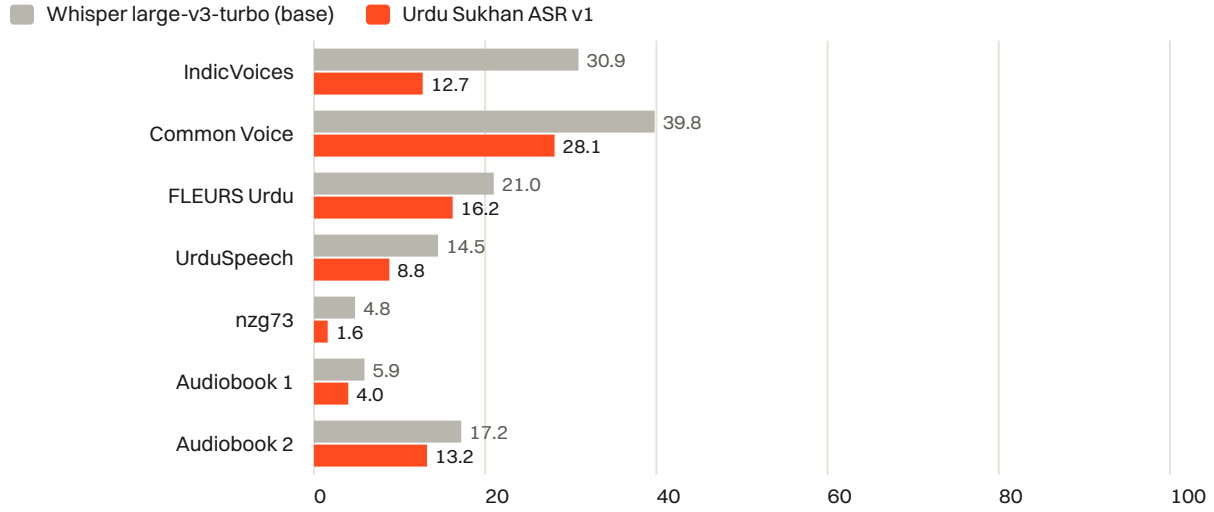
6.3 The training rounds

Rounds 1–3: the audiobook voices. The first three rounds trained on a few hundred book clips and 600 FLEURS clips. They lowered the error on the books (Table 5) but not on other voices: on Common Voice speakers, round 3 scored 37.0% against the base model’s 37.1% (on the earlier test file with duplicate rows). Round 2 also showed that a model trained without timestamps and decoded with them (whisper.cpp’s default) can stop at a long pause and drop the rest of a clip: 3 of 40 test clips were cut short, one by 59 words. Decoding without timestamps fixed it (4.8% to 3.2%), and from round 3 on, 30% of the training clips were in timestamp form. These early test sets changed from round to round and were proofread from round-1 drafts, which flatters round 1; they are not comparable with the later tables.

Clip and meaning	System	Transcript
Two-word reply, IndicVoices “separately”	Reference	الگ الگ
	large-v3-turbo	اللہ اللہ
	Urdu Sukhan	الگ الگ
One-word reply, IndicVoices “hello”	Reference	ہیلو
	large-v3-turbo	موسیقی
	Urdu Sukhan	ہیلو
IndicVoices “yes, yes, yes, yes”	Reference	جی جی جی ہاں
	large-v3-turbo	دی دی دی ہاں
	Urdu Sukhan	جی جی جی ہاں
Quiet speech “Tom survived the car accident.”	Reference	ٹام کار کے حادثے میں بچ گیا تھا
	large-v3-turbo	ٹوم کار کی ہاتھ سے میں بچ گیا تھا
	Urdu Sukhan	ٹوم کار کے حادثے میں بچ گیا تھا
Phone line “the freedom that one sees in this atmosphere”	Reference	اس ماحول میں جو آزادی نظر آتی ہے
	large-v3-turbo	اس محول میں جو ازادی نظر آتی ہے
	Urdu Sukhan	اس ماحول میں جو آزادی نظر آتی ہے
UrduSpeech, ours worse “Gaya–Anand Vihar Garib Rath Express (a train)”	Reference	گیا آند وہار غریب رتھ ایکسپریس
	large-v3-turbo	گیا آند وہار غریب رتھ ایکسپریس
	Urdu Sukhan	گیا آند وہار [-] غریب رتھ ایکسپریس
Faint hiss, no speech	large-v3-turbo	موسیقی
	Urdu Sukhan	اچھا
	Urdu Sukhan, blank allowed	(nothing)

Table 4: Examples from public test sets (whisper.cpp, q8_0, beam 5). Wrong words are in red, a missing word is shown as [-]; spelling and word spacing are compared with the test set’s reference.

Held-out test sets



Stress sets

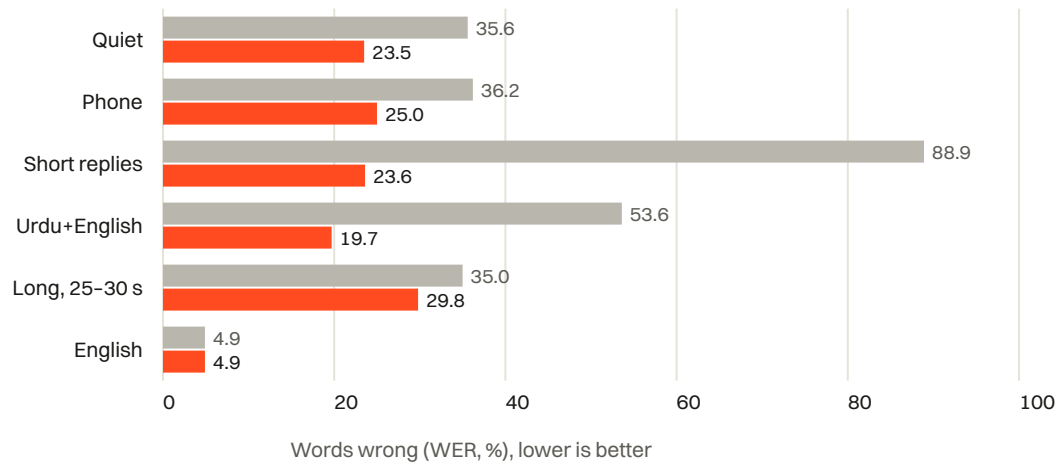


Figure 1: Words wrong for the base model (grey) and Urdu Sukhan ASR v1 (orange) on every speech test and stress set.

WER (%)	Base	R1	R2	R3	R4
Book 1, 40 clips (rounds 1-2)	5.5	2.7	3.2	–	–
Book 1, 48 clips	5.9	3.9	3.9	4.3	3.8
Book 2, 10 clips	17.0	15.1	15.7	13.2	11.7

Table 5: Rounds 1–4 on the audiobook test clips (whisper.cpp q8_0, beam 5, no timestamps, laptop GPU).

Rounds 4 and 5: general Urdu, then a collapse. Round 4 was the first to improve general Urdu: FLEURS from 21.0% to 17.6% and unseen Common Voice speakers from 39.8% to 31.7% (Table 6). Round 5, the first full fine-tune, was the most accurate model so far on read speech, but on unseen Common Voice clips it wrote runaway repeats on 21 of 260 clips, 17 of them from one quiet speaker with very short clips, and scored 149.3% words wrong. Precision and loudness were ruled out by re-scoring in fp16 and after peak normalisation. On the stress sets it invented text on 30 of 30 silence and 38 of 40 noise clips (Figure 2). The likely causes were the lack of any no-speech training data, Common Voice clips whose audio ends before their text, and a full fine-tune over-fitting on unusual audio; we did not test them one at a time. With temperature fallback in whisper.cpp, round 5 did not loop and scored 27.1% on the same clips, which shows how much the decoder settings hide.

The R4+R5 average. A 50/50 average of the round-4 and round-5 weights kept most of round 5’s gains and looped on only 3 of 260 Common Voice clips (a 30/70 average looped on 9). It was our best model until round 6 and is still the best on the second audiobook (9.7% against 13.2%), but it invented text on 35 of 100 no-speech clips and failed the mixed and short stress sets.

Round 6. Round 6 trained on five times as much data, with no-speech and code-switch clips, and is best or near-best on every set at step 2,775. It is the first round that writes English words in English letters (Section 6.7).

WER (%) / clips	Base	R4	R5	R4+R5	R6/925	R6/1850	R6/2775	R6/3697
IndicVoices	30.9	19.6	19.7	22.3	14.5	15.0	12.7	13.0
Common Voice	39.8	31.7	149.3	56.0	38.9	37.9	28.1	92.6
FLEURS Urdu	21.0	17.6	15.8	15.7	20.9	17.2	16.2	15.4
UrduSpeech	14.5	12.2	12.0	11.2	11.8	9.8	8.8	8.8
Audiobook 1	5.9	4.1	3.5	3.7	5.6	5.4	4.0	3.9
Audiobook 2	17.2	11.6	11.1	9.7	17.4	13.6	13.2	13.5
Quiet	35.6	27.8	93.5	26.1	28.1	24.5	23.5	271.6
Phone	36.2	29.3	169.0	26.1	27.3	23.3	25.0	135.3
Short replies	88.9	69.4	73.6	65.3	31.9	25.0	23.6	26.4
Urdu+English	53.6	65.6	103.4	72.8	22.7	45.8	19.7	19.7
Long (25–30 s)	35.0	29.6	29.6	27.9	33.9	30.4	29.8	29.3
English	4.9	4.8	5.6	5.1	5.2	5.1	4.9	4.9
Noise: invented	15	12	38	14	2	8	6	21
Music: invented	16	14	8	10	1	2	4	2
Silence: invented	1	9	30	11	0	0	3	27
Loops on no-speech clips	3	3	35	4	3	5	5	36

Table 6: Every round on the shared test and stress sets (WER %, Transformers, greedy). R4: LoRA; R5: full fine-tune; R4+R5: weight average of the two; R6/*n*: round 6 after *n* steps. The best value per row is bold. nzg73 is left out because rounds 4 and 5 used a smaller held-out part. Loops counted with the first rule.

6.4 Choosing the checkpoint

Accuracy on speech improved up to step 2,775 (Figure 3). Over the last 922 steps, while the learning rate decayed to zero, round 6 learned to hear words in faint audio: invented text on silence rose

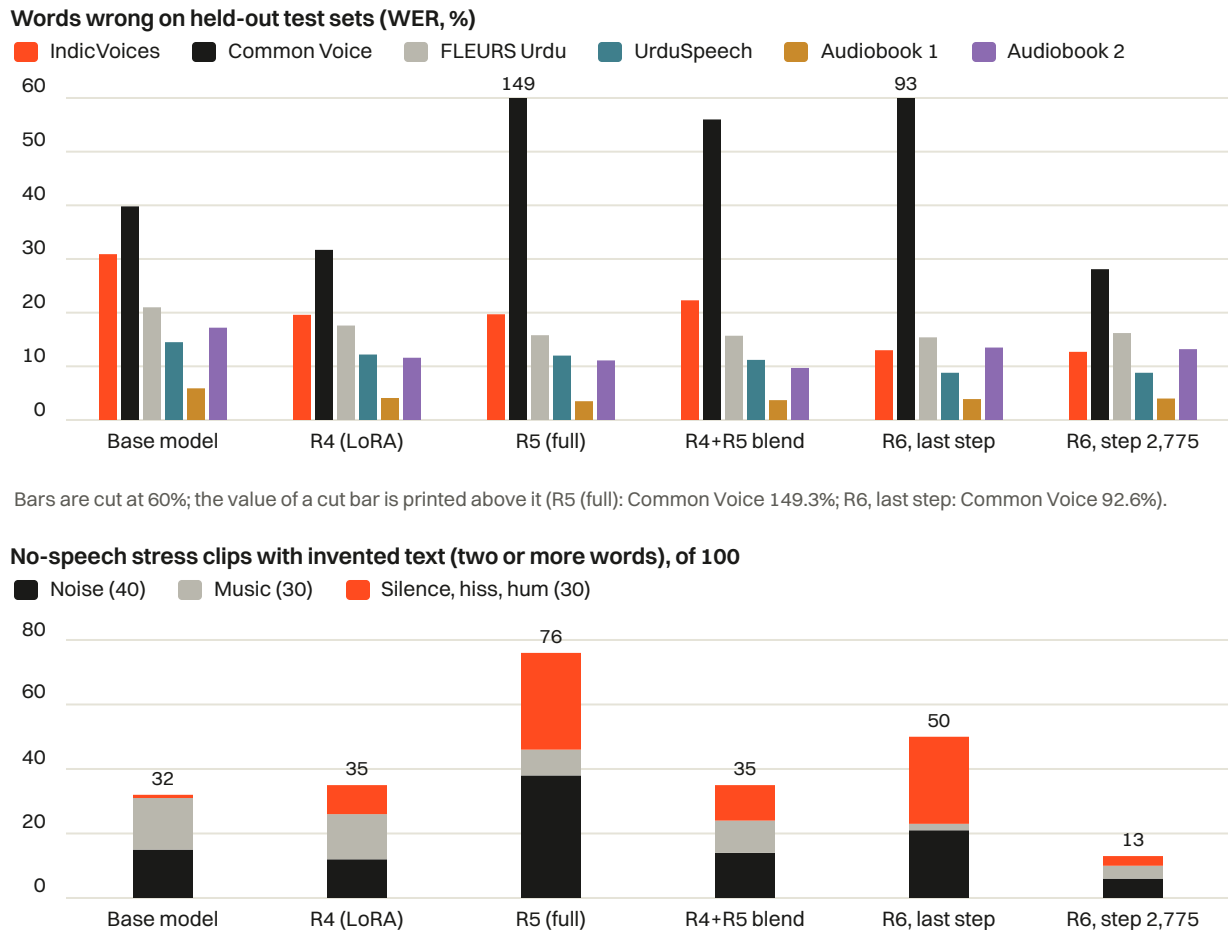


Figure 2: Words wrong on six test sets per round (top) and no-speech stress clips on which each round wrote two or more words (bottom).

from 3 to 27 of 30 clips and on noise from 6 to 21 of 40, quiet speech rose from 23.5% to 271.6% words wrong, and Common Voice from 28.1% to 92.6% with 7 loops. We release step 2,775. The earlier checkpoints were less accurate: step 925 scored 20.9% on FLEURS and step 1,850 45.8% on the Urdu–English set.

Weight averaging did not help. We averaged step 2,775 with the final step, and with the R4+R5 average at 25% and 50% (Table 7). Each gained up to 2.5 points on the second audiobook and lost elsewhere: the averages invented text on 12–17 silence clips and doubled or tripled the errors on phone speech. None was kept.

6.5 Invented text on silence

Faint labelled audio. Round 5 and the end of round 6 learned that faint hiss means speech. Measuring the level of the audio explains why (Figure 4). FLEURS Urdu recordings are very quiet: median -45 dBFS, with one clip in twenty below -60 dBFS, which overlaps the hiss in our silence test (-66 to -52 dBFS). Round 6’s augmentation added two more sources. The “very quiet” setting

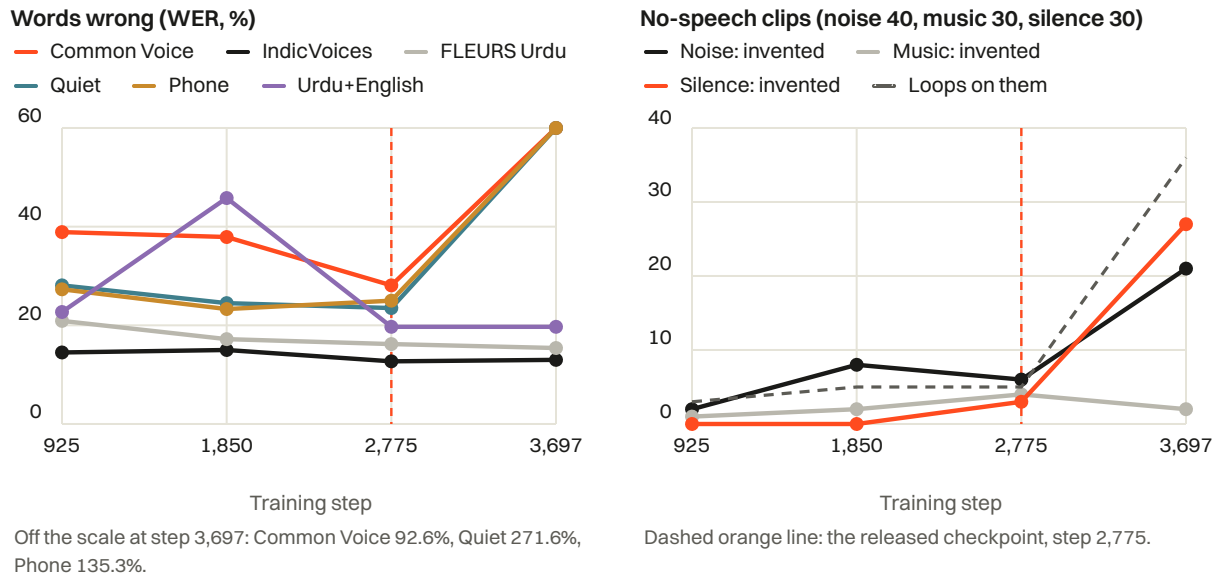


Figure 3: Round-6 checkpoints: words wrong on three test and three stress sets (left) and no-speech clips with invented text, with the loops on them (right).

WER (%) / clips	R6/2775	late	+25% blend	+50% blend
Common Voice	28.1	55.1	34.1	130.3
IndicVoices	12.7	13.5	12.0	13.2
Audiobook 2	13.2	12.3	11.4	10.7
Quiet	23.5	149.7	24.2	83.7
Phone	25.0	80.2	61.8	79.9
Urdu+English	19.7	19.7	62.2	70.2
Noise: invented	6	17	20	28
Music: invented	4	4	1	1
Silence: invented	3	12	17	14

Table 7: Weight averages of round 6 (WER % and invented-text clips; Transformers, greedy). “Late”: step 2,775 and the final step; “blend”: step 2,775 with the R4+R5 average.

lowered a fifth of the augmented clips by 12–30 dB, with background noise up to as loud as the speech. And the room-echo filter normalised its impulse response by its sum, which lowers the level of an echoed clip by a further 20–25 dB. Together they put speech *with full transcripts* at -40 to -75 dBFS, below the hiss of the silence test, so a model rewarded for transcribing it learns to transcribe hiss. The fix is to set every clip to a normal level first, apply the final level after the echo filter, never go below -40 dBFS, and keep noise 15–35 dB below quiet speech. We used it in the compression experiment (Section 8) and will use it in future rounds.

The one-word answer is a decoding rule. By default, whisper.cpp (`suppress_blank`) and Transformers (`begin_suppress_tokens`) forbid the end-of-text token and a blank as the *first* generated token, so a Whisper model cannot return an empty transcript. Measured right after the prefix on no-speech clips, our model puts nearly all its probability on a space, because the training code encoded an empty transcript as a single space token; the compressed model of Section 8 puts

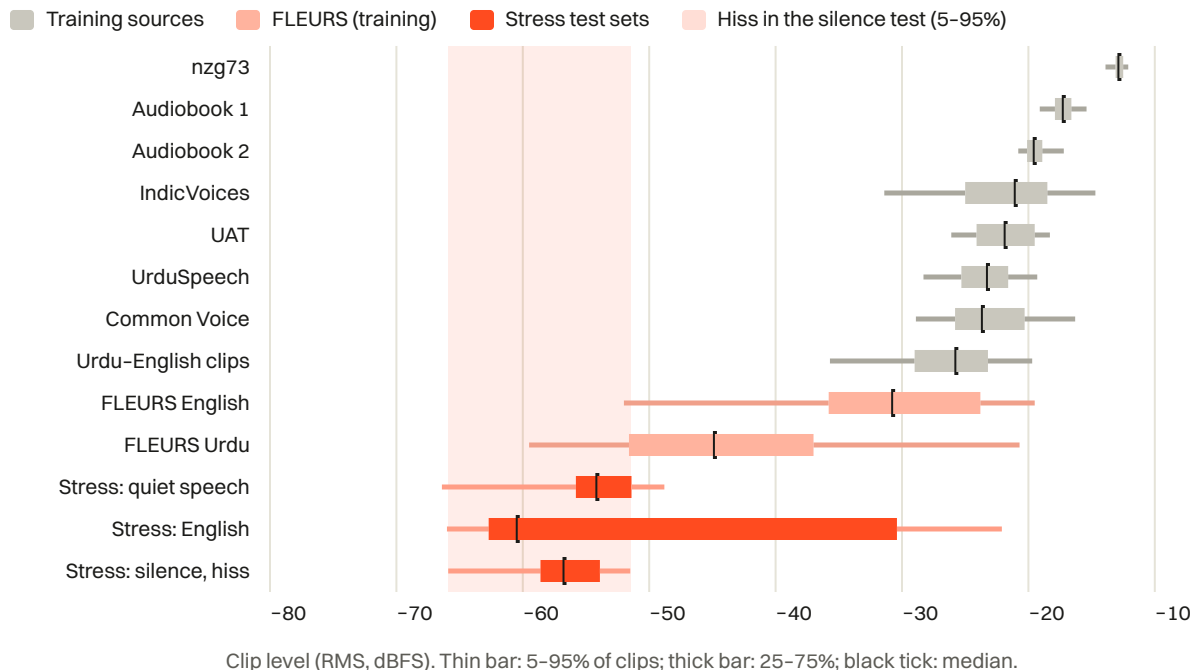


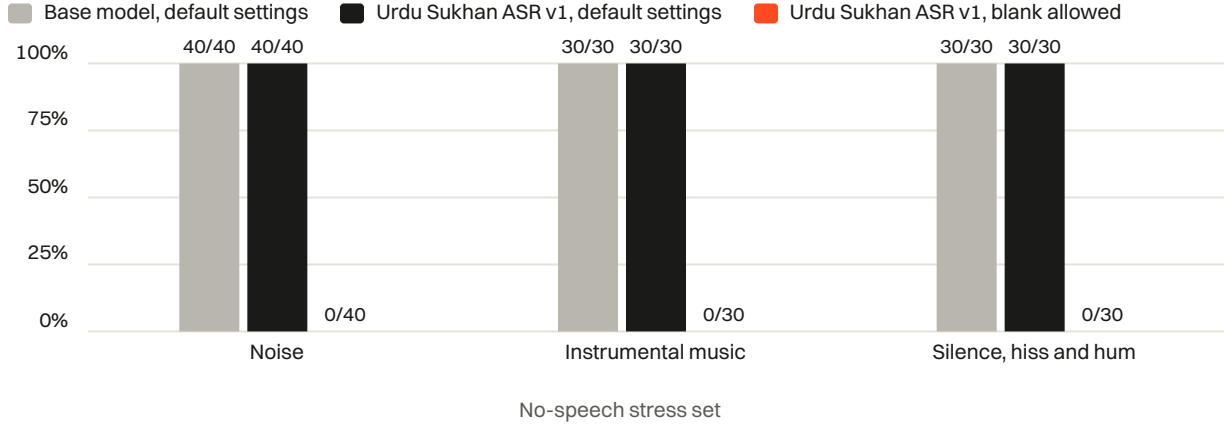
Figure 4: Clip levels (RMS) of up to 300 random clips per training source and of three stress sets. FLEURS is the faintest training source; the band marks the hiss of the silence test. 10 clips of pure digital silence are left out. The English stress clips are also faint, which matters for the voice detector.

probability 1.00 on end-of-text. Both answers are exactly the forbidden tokens, so by default the model must write a word, typically *اچھا* (*achha*, “all right”) or *ہاں* (*haan*, “yes”), and on music *موسیقی* (*mosiqi*, “music”). The usual two-word measure hides this: by default both the base model and ours write at least one word on every noise, music and silence clip (Figure 5). With blank suppression off (one line in an application: `params.suppress_blank = false`), our model writes nothing on all 100 clips and every speech score is unchanged (Table 8).

The voice detector. Running Silero VAD first also removes most invented text, but it lowers the score on the Urdu-English set from 20.4% to 44.6%. The English halves of that set are FLEURS English recordings at their natural level (one at -69 dBFS against -23 dBFS for its Urdu half), and the detector drops them as silence: a flaw of the test set rather than of the detector.

6.6 Names

Names are the weakest point. The second audiobook’s test chapter has 23 name occurrences; the base model spelled 4 right and ours 11 with greedy decoding (the R4+R5 average 15). Its 86 training clips contain the names *عمیر* (*Umail*) and *عزیر* (*Uzair*) once each, and the model writes them as the common word *امیر* (*ameer*, “rich”) and as *اوزیر*; it also writes *دائیا* for *تائیا* (*Tania*). A whisper.cpp prompt that lists the chapter’s names raised the count from 12 to 17 of 23 without loops (Table 9).



Share of no-speech clips on which whisper.cpp wrote at least one word (q8_0, beam 5, temperature fallback). "Blank allowed": suppress_blank turned off.

Figure 5: No-speech clips on which whisper.cpp wrote at least one word: the base model and ours with default settings, and ours with blank suppression off.

Stress set (clips)	large-v3-turbo	Urdu Sukhan ASR v1		
	default	default	voice detector	blank allowed
Noise (40)	16 (40)	10 (40)	4	0 (0)
Instrumental music (30)	21 (30)	1 (30)	0	0 (0)
Silence, hiss and hum (30)	11 (30)	4 (30)	0	0 (0)
Quiet speech (40)	35.3	24.2	23.5	24.2
Phone-band speech (40)	34.8	25.0	25.6	25.0
One- or two-word replies (40)	90.3	22.2	23.6	22.2
Urdu then English in one clip (20)	–	20.4	44.6	20.4
Continuous speech, 25–30 s (15)	–	31.4	32.7	31.4
English, FLEURS (30)	–	5.2	5.8	5.2

Table 8: whisper.cpp as an application runs it (q8_0, beam 5, temperature fallback). No-speech rows: clips with invented text (in brackets: with any word at all); speech rows: WER %. "Blank allowed": suppress_blank off. The base model’s run stopped after six sets when our GPU server was shut down.

Model	Prompt	WER (%)	Loops	Names right
Urdu Sukhan ASR v1	no prompt	11.7	0	12 of 23
Urdu Sukhan ASR v1	name prompt	11.3	0	17 of 23
R4+R5 blend	no prompt	10.0	0	14 of 23
R4+R5 blend	name prompt	9.5	0	17 of 23
Student S1 (not released)	no prompt	17.7	0	10 of 23
Student S1 (not released)	name prompt	32.3	1	9 of 23

Table 9: Second audiobook’s test chapter in whisper.cpp with and without a prompt listing its names.

6.7 English inside Urdu

Rounds 1–5 were trained only on Urdu-script transcripts and wrote English words in Urdu letters. Round 6 was the first trained with Urdu–English clips whose English part is written in English letters under the Urdu language tag, and with 2,000 English FLEURS clips, so it keeps English words in English letters inside Urdu text. The author prefers this for reading. It differs from the house style of the audiobook transcripts, which write English loanwords in Urdu script; the two have not yet been reconciled in the training text.

6.8 Two decoders, two answers

The same model scores differently in the two decoders. Round 5 scored 149.3% on unseen Common Voice speakers with greedy decoding in Transformers and 27.1% in whisper.cpp, whose temperature fallback re-decodes a looping output. Released numbers should name the decoder; we report both.

7 Speed and deployment

Where the time goes. The released model has exactly the layout of Whisper large-v3-turbo, so its speed is the base model’s. In whisper.cpp the encoder dominates CPU time: it always processes a full 30-second window, and for a typical clip it is about 95% of the work. A ten-second clip at beam 5 needs about 200 beam-token steps, about 0.8 s of decoding on the old Xeon against 22 s of encoding. Every clip pays for a full window, so a one-second reply costs as much encoder time as 30 seconds of speech.

Machines. Figure 6 gives the encoder time for one window on four machines. The laptop’s GPU encodes a window in 1.3 s; including model loading, test sets of book clips (about 17 seconds each) ran at 4–6 times real time on it and sets of 4-second Common Voice clips at 0.8–1.7 times, and drafting the 305 minutes of audiobook speech took 59 minutes. CPUs are much slower: 5.4 s on the laptop’s CPU, 11.2 s on the cloud server and 21.1 s on the 2012 Xeon, which lacks AVX2 and is a poor stand-in for a modern server.

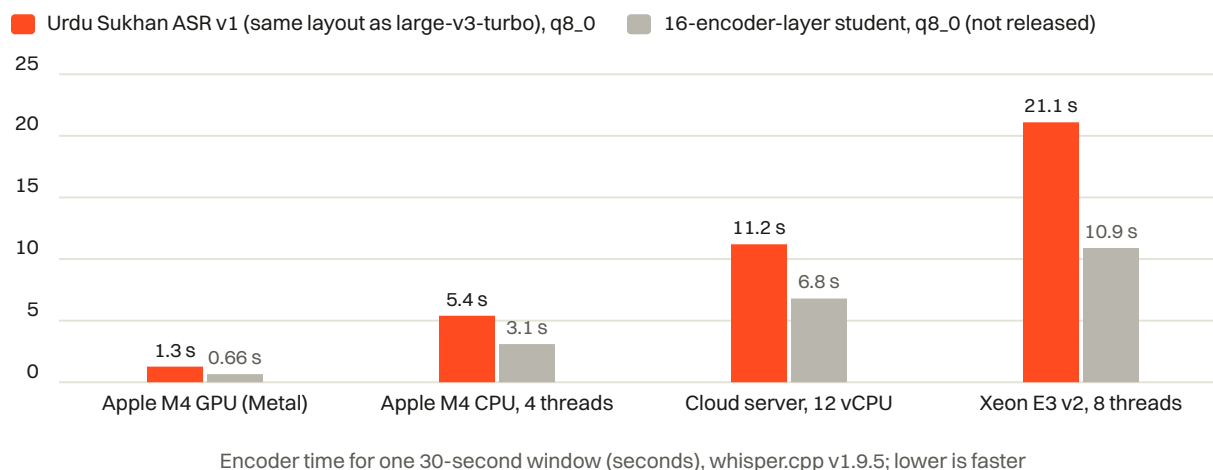


Figure 6: Encoder time for one 30-second window, the released model and the compressed student (Section 8), whisper.cpp v1.9.5.

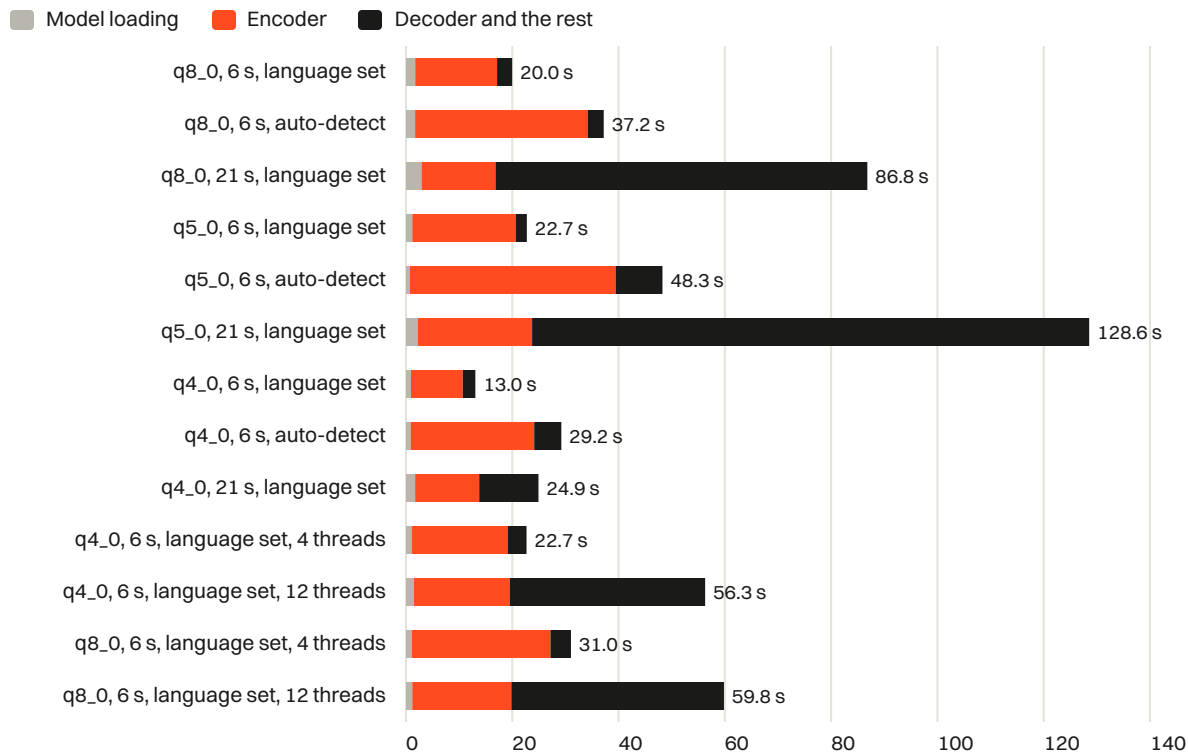
One clip on the cloud server. Table 10 and Figure 7 time the released model on a 6-second and a 21-second held-out clip on the cloud server, fastest of three runs, while its other services were running (load average 10–15 on 12 cores). Four findings matter for applications. First, *language auto-detection runs the encoder twice*: q8_0 took 37.2s instead of 20.0s on the short clip. Second, *q4_0 is the fastest format* on this AVX2 processor (encoder 9.7s against 15.4s for q8_0) and q5_0 the slowest (19.5s); on the laptop’s ARM processor q5_0 is slightly faster than q8_0. Third, *threads must leave room for the rest of the machine*: with all 12 threads the short clip took 59.8s instead of 20.0s, nearly all of it in the decoder, and with 4 threads 31.0s; on the same server with less load, 8 and 12 threads had been equally fast. Fourth, *temperature fallback can multiply the decoding*: on the long clip q8_0 and q5_0 rejected their first result and decoded again at higher temperatures (6,600 token steps against 1,096 for q4_0) before keeping the same text, taking 86.8s and 128.6s against 24.9s. Earlier, with the server quieter, the same q8_0 file took 23.2s on a 5.8-second clip with auto-detection (two encoder passes of 11.2s), so with the other services busy each encoder pass took about 45% longer.

Format	Clip (s)	Language	Threads	Model load	Encode (runs)	Decode	Total	× real time
q8_0	6	set (ur)	8	1.7	15.4 (1)	2.6	20.0	0.30
q8_0	6	auto-detect	8	1.7	32.5 (2)	2.8	37.2	0.16
q8_0	21	set (ur)	8	2.9	13.9 (1)	69.6	86.8	0.24
q5_0	6	set (ur)	8	1.2	19.5 (1)	1.9	22.7	0.26
q5_0	6	auto-detect	8	0.7	38.7 (2)	8.6	48.3	0.12
q5_0	21	set (ur)	8	2.2	21.5 (1)	104.4	128.6	0.16
q4_0	6	set (ur)	8	0.9	9.7 (1)	2.1	13.0	0.46
q4_0	6	auto-detect	8	0.9	23.2 (2)	4.9	29.2	0.21
q4_0	21	set (ur)	8	1.7	12.1 (1)	10.9	24.9	0.85
q4_0	6	set (ur)	4	1.1	18.1 (1)	3.2	22.7	0.27
q4_0	6	set (ur)	12	1.5	18.0 (1)	36.4	56.3	0.11
q8_0	6	set (ur)	4	1.1	26.1 (1)	3.6	31.0	0.19
q8_0	6	set (ur)	12	1.2	18.6 (1)	39.4	59.8	0.10

Table 10: The released model in whisper.cpp v1.9.5 on the cloud server (12 shared AMD EPYC cores; beam 5, temperature fallback, no timestamps); seconds, fastest of three runs. “Decode” includes sampling; encode runs in brackets.

Long recordings. On the cloud server we transcribed a 27.4-minute recording made of 118 held-out clips with faster-whisper (CTranslate2 int8 weights, voice-detected pieces in batches of eight, eight threads). It took 15.6 minutes with greedy decoding (1.76 times real time, 14.2% WER) and 18.1 minutes with beam 5 (1.52 times, 13.3%). We had expected CTranslate2 to be two to four times faster than whisper.cpp, as it often is on Intel processors, whose 8-bit instructions its kernels are tuned for; this server’s shared 2.0 GHz AMD cores are the limit. We did not run whisper.cpp on this file. From the encoder time per window on the quiet server we estimated 12–13 minutes in q8_0 and 9–10 in q4_0, about the same as faster-whisper; under the load of Table 10, whisper.cpp q4_0 reached only 0.85 times real time on a single 21-second clip. Either way, a two-hour recording takes an hour or more on this server.

Sizes and formats. The whisper.cpp files take 874MB (q8_0), 574MB (q5_0) and 474MB (q4_0), and the CTranslate2 int8 model 824MB. A Core ML encoder for Apple’s Neural Engine converted cleanly for the 16-layer student (0.22s per window) but the full 32-layer encoder agreed



Seconds per clip, whisper.cpp v1.9.5, beam 5, 8 threads unless stated; lower is faster

6 s and 21 s held-out clips on a cloud server with 12 shared AMD EPYC cores; other services ran at the same time (load average 10–15). Fastest of three runs.

Figure 7: Where the time goes for one clip on the cloud server.

with PyTorch only to a cosine similarity of 0.984 in fp16, so it is not released. We did not measure memory use, and the model has not yet run on an iPhone.

Recommended settings. Set the language when it is known, or detect it once per conversation or file; turn off blank suppression; run a voice detector first except where faint speech matters; decode without timestamps and without carried-over context; on x86 servers use q4_0 and about eight threads per job, with several jobs side by side.

8 A compression experiment

To make the model faster on CPUs we tried a student with fewer layers, distilled from the released model. Speed depends only on the layout, so we first timed untrained layer subsets (Table 11): encoder time is linear in the number of layers, while decoder layers cost little (2 instead of 4 saved about 1.4 ms per beam-token against 11 s of encoding). We kept all 4 decoder layers, and with them whisper.cpp’s word-timing heads. whisper-small was not an option: its encoder is only about 30% faster than an 8-layer turbo student, its 12-layer decoder is slower and its Urdu is far weaker.

Two 600-step pilots compared 16 and 8 encoder layers: the 8-layer student was still above 100% WER on every held-out source, so we trained the 16-layer one (every other layer, keeping the first and the last). The loss combined cross-entropy on transcripts, a KL divergence to the teacher’s

Layout	Layers	Encoder (s)		Decoder ms/token	M4 GPU enc. (s)	Size (MB)	
		enc / dec	8 thr. 4 thr.			q8_0	q5_0
Full model (32/4)	32 / 4		22.1 41.1	3.94	1.3–1.8	874	574
16/4	16 / 4		11.1 20.2	3.89	0.85	539	357
16/2	16 / 2		10.7 21.9	2.5	–	483	321
12/2	12 / 2		8.2 16.0	2.41	0.58	399	266
8/4	8 / 4		5.7 11.6	3.79	–	371	248
8/2	8 / 2		5.5 10.8	2.44	0.38	315	212
whisper-small	12 / 12		4.0 7.9	4.77	–	264	175
whisper-base	6 / 6		1.2 2.3	1.63	–	82	55

Table 11: Speed and size of untrained layer subsets of the released model and of two stock Whisper models (whisper-bench, Xeon E3 v2; one 30-second window; decoder cost per beam-token).

next-token distribution at temperature 2, the mean squared error to the teacher’s encoder output and to the layer-normalised outputs of the teacher layers each student layer was copied from, with SpecAugment on the student only. Book clips and no-speech clips were trained on transcripts only, so the student would not copy the teacher’s misspelt names. Every clip was set to a normal level before augmentation (Section 6.5), and two to four short clips were packed into one 30-second window.

Two problems needed restarts (Figure 8). The student wrote one word on every silence clip, so we weighted the end-of-text token of no-speech windows 40 times; its answers on no-speech clips then drifted between checkpoints, so we added “a word was forced first, now stop”: on half the no-speech windows a common word is placed in the decoder input without being scored, and only the end-of-text after it is. Each restart briefly broke the no-speech behaviour, which recovered by the next checkpoint. After 7,200 steps (about 11 GPU-hours) the student kept at step 6,600, S1, was 1.9 times faster in the encoder on every machine and invented text on 0, 1 and 0 noise, music and silence clips in whisper.cpp against the released model’s 10, 1 and 4. But it stayed 0.5–5.4 points behind on the test sets and was weak in English (Table 12), and the last 3,000 steps gained about one point. Removing half of a 32-layer encoder discards more than 110 hours of audio can restore; more teacher-transcribed audio, not more steps, is the way forward. The author tested both models on the cloud server and ranked the released model first; S1 is not released.

WER (%)	Full model	Student S1	Difference
IndicVoices Urdu validation	12.7	17.2	+4.5
Common Voice, unseen speakers	28.1	30.9	+2.8
FLEURS Urdu test	16.2	21.3	+5.1
UrduSpeech test split	8.8	10.8	+2.0
nzg73, held-out clips	1.6	2.1	+0.5
Audiobook 1, held-out chapters	4.0	6.6	+2.6
Audiobook 2, held-out chapter	13.2	18.6	+5.4
Quiet speech (whisper.cpp)	24.2	28.8	+4.6
Phone-band speech (whisper.cpp)	25.0	25.9	+0.9
One- or two-word replies (whisper.cpp)	22.2	23.6	+1.4
Urdu then English in one clip (whisper.cpp)	20.4	38.8	+18.4
Continuous speech, 25–30 s (whisper.cpp)	31.4	34.8	+3.4
English, FLEURS (whisper.cpp)	5.2	23.0	+17.8

Table 12: The released model and the 16-layer student S1. Test sets: Transformers, greedy; stress sets: whisper.cpp q8_0, blank suppression off.

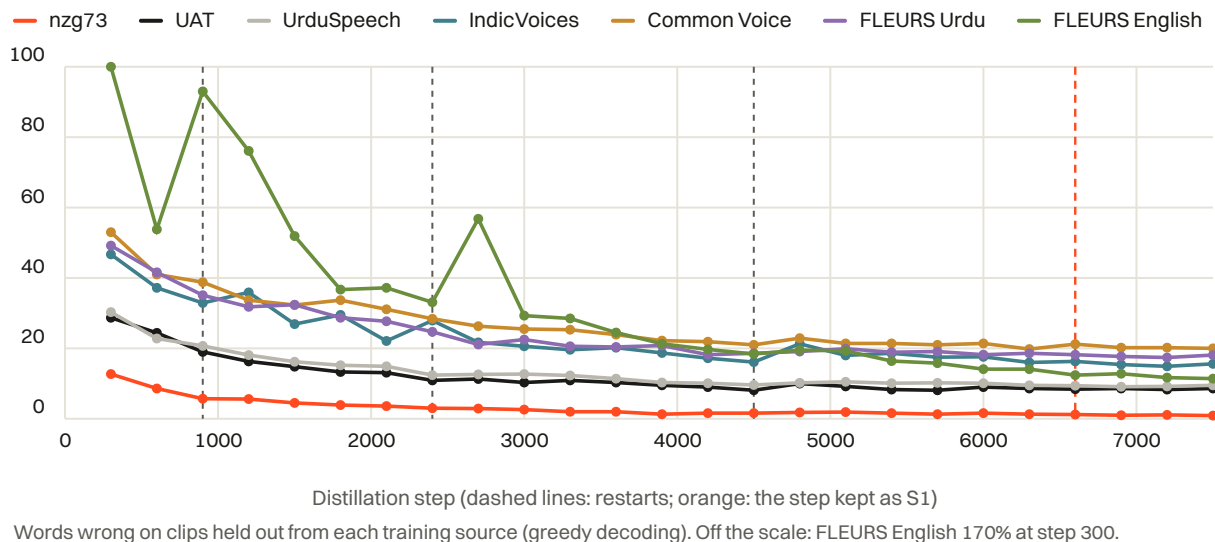


Figure 8: The student’s words wrong on clips held out from each training source during distillation.

9 Lessons

Four lessons from this work apply beyond Urdu. First, *evaluation code needs as much care as training code*: a test file with duplicate rows, a loop rule that changed three times, a two-word rule that hid one-word answers, and two scoring jobs that deleted each other’s outputs and showed a false collapse all changed numbers we had already reported. Second, *audio level is part of the label*: faint speech with full transcripts taught two rounds to transcribe hiss. Third, *decoder defaults are part of the model*: timestamps, blank suppression, language detection and temperature fallback each changed results more than some training decisions did. Fourth, *test speed on the target machine*: the GPU server’s 2012 processor was several times slower than the cloud server and gave the wrong picture, and on the cloud server the load of other services changed which thread count was fastest.

10 Limitations and threats to validity

The audiobook test sets are small (58 clips, the second only 2.9 minutes) and come from the same two narrators whose other chapters were used in training; the short-reply and mixed sets have 40 and 20 clips. We report no confidence intervals. IndicVoices is mostly Indian Urdu, women are under-represented in the Common Voice test set, and we have no test of spontaneous conversation recorded on a phone by Pakistani speakers. Round 6 heard the MUSAN files of the noise and music stress sets during augmentation, so its scores there are slightly flattering. The causes we give for the collapses of round 5 and of round 6’s final step fit the measurements but were not tested one at a time. Rounds 1–4 were measured on test sets that changed between rounds, and some early Common Voice numbers come from a file with duplicates. We never scored Whisper large-v3 itself, measured memory, or ran the model on a phone. The cloud server’s timings vary with the other services running on it. The base model’s whisper.cpp stress run is incomplete.

11 Reproducibility and availability

The code (proofreading tool, house style, data preparation, training, distillation, evaluation, exports and the test page), the result table of every evaluation and the scripts that build this paper’s tables and figures from them are in the Urdu Sukhan ASR repository, <https://github.com/Loudness-ai/urdu-sukhan-asr>. The model is published on Hugging Face as Loudness-ai/urdu-sukhan-asr-v1 (<https://huggingface.co/Loudness-ai/urdu-sukhan-asr-v1>) in Transformers, whisper.cpp and CTranslate2 formats under the CC BY-NC 4.0 licence for non-commercial use; access is granted on request. The original full-precision checkpoint was lost when our training server was shut down at the end of the work; the Transformers weights were rebuilt from the q8_0 file and checked against the results reported here. Installation and settings are at <https://loudness.ai/developers/urdu-sukhan-asr/>. The audiobook recordings and the transcribed UAT audio are not redistributed. Our Common Voice, FLEURS and stress test sets are available on request as Loudness-ai/urdu-sukhan-asr-test-sets (<https://huggingface.co/datasets/Loudness-ai/urdu-sukhan-asr-test-sets>). Public datasets are available from their publishers under their own terms. This report is archived on Zenodo, <https://doi.org/10.5281/zenodo.23518703>, under the CC BY 4.0 licence.

12 Competing interests

The author develops Loudness.ai, whose products will use the model described here. No external funding was received.

Version history. Version 1.0 (21 October 2026): first publication.

References

- [1] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey and I. Sutskever. Robust speech recognition via large-scale weak supervision. In *Proc. ICML*, 2023.
- [2] S. Arif, A. J. Khan, M. Abbas, A. A. Raza and A. Athar. WER we stand: benchmarking Urdu ASR models. arXiv:2409.11252, 2024.

- [3] A. Koenecke, A. S. G. Choi, K. X. Mei, H. Schellmann and M. Sloane. Careless Whisper: speech-to-text hallucination harms. In *Proc. ACM FAccT*, 2024.
- [4] A. Conneau, M. Ma, S. Khanuja, Y. Zhang, V. Axelrod, S. Dalmia, J. Riesa, C. Rivera and A. Bapna. FLEURS: few-shot learning evaluation of universal representations of speech. In *Proc. IEEE SLT*, 2023.
- [5] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers and G. Weber. Common Voice: a massively-multilingual speech corpus. In *Proc. LREC*, 2020.
- [6] T. Javed et al. IndicVoices: towards building an inclusive multilingual speech dataset for Indian languages. In *Findings of ACL*, 2024.
- [7] D. Snyder, G. Chen and D. Povey. MUSAN: a music, speech, and noise corpus. arXiv:1510.08484, 2015.
- [8] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk and Q. V. Le. SpecAugment: a simple data augmentation method for automatic speech recognition. In *Proc. Interspeech*, 2019.
- [9] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang and W. Chen. LoRA: low-rank adaptation of large language models. In *Proc. ICLR*, 2022.
- [10] M. Wortsman et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proc. ICML*, 2022.
- [11] G. Hinton, O. Vinyals and J. Dean. Distilling the knowledge in a neural network. arXiv:1503.02531, 2015.
- [12] S. Gandhi, P. von Platen and A. M. Rush. Distil-Whisper: robust knowledge distillation via large-scale pseudo labelling. arXiv:2311.00430, 2023.
- [13] G. Gerganov and contributors. whisper.cpp, version 1.9.5. <https://github.com/ggml-org/whisper.cpp>.
- [14] SYSTRAN. faster-whisper, version 1.2.1. <https://github.com/SYSTRAN/faster-whisper>.
- [15] Silero Team. Silero VAD: pre-trained enterprise-grade voice activity detector. <https://github.com/snakers4/silero-vad>.
- [16] G. Kunz and contributors. Chartist, version 2.1.0. <https://chartist.dev>.