

SANKET: ON-DEVICE, OPEN-VOCABULARY INDIAN SIGN LANGUAGE RECOGNITION IN THE BROWSER

CASE STUDY REPORT

Review 2 submission · Report Completion (50%)

Submitted by

MITHUNISH PRABHAKARAN

[Register No: RA2512049015065]

Under the Guidance of

Dr. S SARAVANAN

21CSC601T-Case Studies

Department of Computational Intelligence



FACULTY OF ENGINEERING AND TECHNOLOGY
SCHOOL OF COMPUTING
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR

SEPTEMBER 2026

Own Work* Declaration Form

This sheet must be completed, each condition ticked to show that it has been met, and signed and dated with the student's registration number before the work is submitted; work is not assessed without it.

To be completed by the student for all assessments

Degree / Course : **M.Tech in Artificial Intelligence**
Student Name : Mithunish Prabhakaran
Registration Number : RA2512049015065
Title of Work : Sanket: On-Device, Open-Vocabulary Indian Sign Language Recognition in the Browser

I hereby certify that this assessment complies with the University's Rules and Regulations relating to academic misconduct and plagiarism**, as listed in the University website, Regulations, and the Education Committee guidelines.

I confirm that all the work contained in this assessment is my own except where indicated, and that I have met the following conditions:

- ☐ Clearly referenced / listed all sources as appropriate
- ☐ Referenced and put in inverted commas all quoted text (from books, web, etc.)
- ☐ Given the sources of all pictures, data etc. that are not my own
- ☐ Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- ☐ Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- ☐ Complied with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalised in accordance with the University policies and regulations.

DECLARATION:

I am aware of and understand the University's policy on academic misconduct and plagiarism, and I certify that this assessment is my own work, except where indicated by referencing, and that I have followed the good academic practices noted above.

Student signature · RA2512049015065

Date

* "Own work" includes work carried out with the help of tools, which are disclosed where used. ** Plagiarism as defined in the University's academic regulations.

Table of Contents

Abstract	vii
Review 2 Status	viii
1 Introduction	1
1.1 Objectives of the project	1
1.2 Problem statement	2
1.3 Sustainable Development Goal alignment	2
1.4 Scope of the project	3
1.5 Significance of the project	3
1.6 Organisation of the report	4
2 Literature Survey	5
2.1 Word-level sign language datasets and where ISL stands	5
2.2 Appearance-based and pose-based recognition	5
2.3 The one-shot, open-vocabulary regime	5
2.4 Template matching	6
2.5 Latent-predictive self-supervision (JEPA)	6
2.6 Self-supervision on skeletons and on sign language	6
2.7 Knowledge distillation, label smoothing and calibration	6
2.8 Signer-independent evaluation and on-device inference	7
2.9 Research gap	7
3 System Design and Architecture	8
3.1 Design principles	8
3.2 Run-time architecture	8
3.3 Training architecture	9
3.4 Evaluation architecture	10
3.5 Data pipeline and review loop	11
4 Implementation of Algorithms and Techniques	12
4.1 Technology stack	12
4.2 Landmark extraction and canonicalisation	13
4.3 ISL-JEPA self-supervised pretraining	13
4.4 Fine-tuning recipe: infusion, synthetic signers, recover-and-resample	14
4.5 Ensemble to single-model knowledge distillation	15
4.6 Export and quantisation	16
4.7 Temperature calibration with cross-fitting	16
4.8 Mirror test-time augmentation	17
4.9 Full-vocabulary retrieval	18
4.10 Shape match: model-free dynamic time warping	18
4.11 Dataset quality system	18
4.12 Compute	18

5 Intermediate Results	20
5.1 The accuracy test	20
5.2 Model lineage, Review 0 to Review 2	20
5.3 Seed-level evidence for the dataset contract	22
5.4 What the user sees in the application	22
5.5 The honest reading	22
5.6 Negative results	23
5.7 Shape match as a helper, measured	23
5.8 Data findings	24
5.9 To be completed for the final review	25
6 Expected Outcomes and Roadmap	26
6.1 Outcomes by the final review	26
6.2 Targets against status	26
6.3 Roadmap: the next levers	26
6.4 Risks and mitigations	27
6.5 Research outcomes: paper seeds	27
6.6 Societal outcome (SDG 10)	27
7 Conclusion	28
A Screenshots of the Working System	29
B Reproducibility	34
B.1 Where the work lives	34
B.2 Commands	34
B.3 Evaluation discipline	34
B.4 Compute spend	34
C Glossary	36
Bibliography	41

List of Tables

Table 1	Review 2 deliverables and their location in this report.	viii
Table 2	Report completion by chapter at Review 2.	viii
Table 1.1	How the project maps to the SDGs it aligns with.	3
Table 2.1	What the literature provides and what this project adds.	7
Table 3.1	Run-time artifacts and their measured size and speed.	9
Table 4.1	Technology stack.	12
Table 4.2	Datasets and the role each plays.	12
Table 5.1	Model lineage on the accuracy test	20
Table 5.2	Accuracy test per seed: control versus the v4 pack	22
Table 5.3	Four-seed ensembles: control versus v4 (Phase B).	22
Table 5.4	In-app accuracy (int8 + mirror pass) and Practice crediting at a calibrated floor of 0.5.	22
Table 5.5	Negative results, measured.	23
Table 5.6	Shape match (model-free DTW) on the 260-class bench.	24
Table 5.7	Data findings from the Dataset tab and the review loop.	24
Table 6.1	Targets set at Review 1 against the status at Review 2.	26
Table 6.2	Risks and mitigations.	27
Table 6.3	Semester-2 paper seeds.	27
Table B.1	Commands that reproduce the numbers in this report.	34

List of Algorithms

Algorithm 4.1	ISL-JEPA pretraining step	14
Algorithm 4.2	Fine-tune with infusion, synthetic signers and recover-and-resample	15
Algorithm 4.3	Ensemble distillation	16
Algorithm 4.4	Temperature calibration with cross-fit	17
Algorithm 4.5	Mirror test-time augmentation decision	17

List of Figures

Figure 3.1	The system at run time	9
Figure 3.2	How a model is made	10
Figure 3.3	The accuracy test	10
Figure 3.4	The dataset review loop	11
Figure 4.1	Synthetic signers from real clips	15
Figure 4.2	Twelve models into one	16
Figure 5.1	Model lineage, Review 0 to Review 2	21
Figure 5.2	Accuracy test, RKMVU and CISLR from jepa-rr4 to the ensembles	21
Figure 5.3	Shape match as a helper, measured	24
Figure 6.1	What comes next	27
Figure A.1	Studio with the demo player	29
Figure A.2	Learn	30
Figure A.3	Dictionary look-up in Learn	30
Figure A.4	Practice	31
Figure A.5	Model Lab	31
Figure A.6	Eval bench: the accuracy test in the browser	32
Figure A.7	Dataset tab	32
Figure A.8	Review tab	33
Figure A.9	Settings	33

Abstract

Indian Sign Language (ISL) is the first language of a large Deaf community, yet almost every service a Deaf person uses — a bank counter, a clinic, a classroom — is organised around speech and text, and qualified interpreters are scarce relative to that need. Automatic recognition of isolated ISL signs could narrow part of this gap, but the systems in the literature either recognise a small fixed vocabulary or send camera video to a server, and almost all of them are evaluated on held-out clips of the very signers and studios they were trained on.

This case study builds **Sanket**, a web application that recognises ISL signs from an ordinary webcam entirely on the user’s device. MediaPipe reduces each frame to 107 canonical body, hand and face landmarks; the frame itself is discarded, and every later stage sees only a 64-frame landmark window. Two recognisers run side by side: a closed-set classifier over a **260-sign set** that returns calibrated probabilities, and a retrieval path over a **full vocabulary** of 8,188 words (8,165 reachable through the stored examples). A third, model-free **shape match** aligns hand-crafted kinematics by dynamic time warping. No video leaves the device.

The closed-set model is the centre of this report. It is a 4.57 MB int8 ONNX encoder pretrained with ISL-JEPA — a latent-predictive self-supervised objective over a 123,340-clip landmark pool — and then fine-tuned with gloss-matched dictionary clips, geometric synthetic signers and a recover-and-resample temporal augmentation, ensembled, distilled into one student, and calibrated with a single temperature that must transfer across corpora before it ships. Accuracy is measured by one pre-stated **accuracy test**: clip-weighted top-1 over 599 clips from a signer the model has never seen (RKMVU, 220 clips) and a dictionary test role that was never trained on (CISLR, 379 clips). Between Review 1 and Review 2 the deployed model rose from .4591 to .5509 on this test, with the unseen-signer number moving from .4818 to .5000; the report attributes +7.0 pp of the gain to a reviewed dataset contract (4 of 4 seeds improved) and the remainder to ensembling and distillation, and it reports the negative results that shaped the plan as carefully as the positive ones.

The work is aligned with **SDG 10 — Reduced Inequalities**: a free, private ISL learning and recognition tool that runs in any modern browser.

Keywords: Indian Sign Language · on-device inference · self-supervised learning · JEPA · knowledge distillation · calibration · cross-signer evaluation · accessibility

Review 2 Status

Review 2 (Sunday, 27 September 2026, in person at SRM Kattankulathur) asks for four deliverables. This page maps each one to where it lives in this report, and states how complete the report is.

Table 1. Review 2 deliverables and their location in this report.

#	Deliverable (per the course guidelines)	Where it is answered
1	Implementation of algorithms / techniques	Chapter 4 — eleven techniques, five algorithm boxes, the loss and calibration formulas, the technology stack; Chapter 3 for the architecture they sit in
2	Intermediate results obtained	Chapter 5 — the accuracy test, model lineage, seed-level and in-app results, negative results, shape-match and data findings; Appendix A shows the working system
3	Report completion (50%)	This document — Chapters 1–4 complete, Chapters 5–6 substantially complete, Chapter 7 outlined (table below)
4	Expected outcomes	Chapter 6 — outcomes by the final review, targets against status, roadmap, risks, paper seeds, SDG impact

Table 2. Report completion by chapter at Review 2.

Part	Status	What remains
Front matter, Abstract	Done	Abstract numbers refreshed after the model freeze
1 Introduction	Done	—
2 Literature survey	Done	Minor additions if the final experiments need them
3 System design and architecture	Done	—
4 Implementation of algorithms	Done	—
5 Intermediate results	Done for Review 2	Frozen model recorded (jepa-x12); final multi-seed runs of the next levers
6 Expected outcomes and roadmap	In progress	Updated after the Review 2 panel’s feedback
7 Conclusion	Final review	Outline present; written for 11 Oct 2026 (tentative)
Appendix A Screenshots	Done	Refreshed with the frozen model
Appendix B Reproducibility	Done	Final command list at freeze
Appendix C Glossary	Done	—

By page count and by content the report is a little over half complete: the design, the implementation and the evidence behind every claim are written; what remains is the final measured row, the discussion that follows from it, and the conclusion.

CHAPTER 1

Introduction

A signed language is a complete natural language, with its own grammar and word order; Indian Sign Language is not a manual encoding of Hindi or English. For the Deaf community in India it is the language of home, school and friendship. The surrounding society, however, runs on speech and writing. A Deaf customer at a bank counter, a Deaf patient describing symptoms, or a Deaf student in a mainstream classroom depends on an interpreter being present, and interpreters are few relative to the number of people who need them. Technology that can recognise signing — even a single sign at a time — can help in two ways: it can let hearing people and learners practise and check their own signing, and it can give a Deaf signer a way to be understood by a device when no interpreter is present.

Most published recognisers are not built for either use. They classify a small, fixed set of signs; they are evaluated on held-out clips of the same few signers and studios they were trained on; and many of them need the camera stream to reach a server. This project takes the opposite position on all three: the vocabulary should be large and growable, the evaluation should be on a signer the model has never seen, and the camera stream should never leave the user's device.

1.1 Objectives of the project

1.1.1 Recognise ISL signs entirely on the user's device

Pose extraction, recognition and the user interface run inside a browser tab. Frames are reduced to landmarks in a worker and discarded; no image, video or landmark stream is uploaded. Privacy is a property of the construction, not a policy promise.

1.1.2 Cover the vocabulary a learner actually meets

Two complementary recognisers: a calibrated classifier for a 260-sign set that it names outright, and a retrieval path over the full 8,188-word vocabulary whose examples come from public ISL dictionaries. New words enter the full vocabulary by adding examples, not by retraining.

1.1.3 Measure accuracy the way a user would experience it

One pre-stated accuracy test built from a signer the model never saw and dictionary clips it never trained on, computed through the same numeric contract the browser executes, with every claim backed by at least four training seeds.

1.1.4 Close the cross-signer gap with methods that fit the data regime

ISL has thousands of words but only one or two reference clips for most of them. The objective is to make the encoder signer-invariant through self-supervision on a large unlabelled landmark pool, targeted data (gloss-matched dictionary clips, synthetic signers), and ensembling distilled back into one browser-sized model.

1.1.5 Ship a useful product around the recogniser

Learn (a figure signs each word from the model's own training clips), Practice (the model scores the learner), a recognition Studio, and developer surfaces that keep the dataset and the evaluation honest.

1.2 Problem statement

1.2.1 Closed vocabularies

The best-resourced ISL dataset for word-level recognition, INCLUDE, has 263 classes [1]. A recogniser limited to such a list forces every other sign onto its nearest listed word. The realistic ISL regime is thousands of words with one or two references each [2].

1.2.2 Server-side inference

A recogniser that needs the camera feed on a server imposes connectivity, cost and a privacy exposure precisely on the users who most need it to be free and private.

1.2.3 The cross-signer gap

At Review 1 this project measured the gap directly: a supervised model scoring 0.82 on held-out signers of its own corpus fell to 0.33 on a signer from a different corpus, and to 0.12 when that signer was left-hand dominant. Within-corpus evaluation, even signer-disjoint, did not predict this.

1.2.4 Research gap

No prior system combines open-vocabulary ISL recognition, on-device execution, and an evaluation on unseen signers computed through the deployed pipeline (Chapter 2, [Table 2.1](#)). This case study builds that combination and measures what limits it.

Problem statement in one sentence

Build an ISL recogniser that runs entirely in the browser, answers over a large vocabulary, and is measured — and improved — on signers it has never seen, with the measurement computed exactly as the deployed application computes it.

1.3 Sustainable Development Goal alignment

The course asks every case study to align with one of the United Nations Sustainable Development Goals (SDGs), the seventeen goals adopted by the UN member states in 2015

for the period to 2030. This project aligns with **SDG 10 — Reduced Inequalities** [3], and secondarily with **SDG 4 — Quality Education** [4].

SDG 10 asks countries to reduce inequality within and among themselves. One of its targets (10.2) is the social, economic and political inclusion of all people irrespective of attributes that include disability. Language access is a direct instance: a Deaf person’s participation in ordinary transactions depends on whether the hearing side can understand signing. A recogniser that is free, works offline in a browser, and never uploads the user’s video lowers the cost of that access to the cost of a webcam.

SDG 4 (secondary) concerns inclusive and equitable education. Learning ISL is how hearing family members, teachers and classmates meet a Deaf person halfway; the Learn and Practice surfaces turn the recogniser into a patient practice partner that scores each attempt on the learner’s own device.

Table 1.1. How the project maps to the SDGs it aligns with.

Goal	Concern relevant here	What Sanket contributes
SDG 10 (primary)	Inclusion irrespective of disability (target 10.2)	Free, private, on-device ISL recognition; no interpreter, server or data plan needed
SDG 4 (secondary)	Inclusive education and lifelong learning	Learn and Practice surfaces for ISL learners, scored by the same model

1.4 Scope of the project

In scope. Word-level (isolated) recognition of ISL signs from a webcam; the 260-sign closed set and the 8,188-word full vocabulary; on-device inference in a browser; the training pipeline that produces the models (self-supervised pretraining, fine-tuning, distillation, calibration, export); the evaluation harness; the dataset-quality system; and the learner-facing application.

Out of scope. Continuous signing and sentence translation (the application assembles recognised glosses with rules only); fingerspelling as a dedicated modality; non-manual grammar (facial expression is tracked but not modelled as grammar); dialect coverage beyond the source dictionaries.

1.5 Significance of the project

The project matters in three ways. **For users**, it is a working tool: a learner can open a browser, watch a sign, try it, and be scored, with nothing uploaded. **For the field**, it contributes an evaluation discipline — a small, fixed, unseen-signer test computed through the deployed pipeline and reported beside every aggregate — and a documented set of negative results, which are rarely published but decide where effort should go. **For the method**, it shows that a 4.57 MB model trained with latent-predictive self-supervision, targeted data and ensemble distillation can run at browser speed while improving on a signer it has never seen.

1.6 Organisation of the report

Chapter 2 surveys the literature the design rests on. Chapter 3 presents the architecture at run time, in training, in evaluation and in the data pipeline. Chapter 4 details the eleven algorithms and techniques implemented. Chapter 5 reports intermediate results, including negative ones. Chapter 6 states the expected outcomes and the roadmap to the final review. Chapter 7 will conclude. Appendix A shows the working system, Appendix B the reproducibility record, and Appendix C a glossary.

CHAPTER 2

Literature Survey

This chapter consolidates the surveys submitted at Review 0 and Review 1, condensed around the questions the implementation had to answer. Every factual statement about prior work carries a citation verified against its publishing venue; no claim is added that was not verified.

2.1 Word-level sign language datasets and where ISL stands

Word-level recognition matured on American Sign Language corpora: WLASL provides 2,000 glosses in more than 21,000 videos [5], and MS-ASL 1,000 classes from over 200 signers [6]. For ISL, INCLUDE offers 4,287 studio recordings of 263 words in 15 categories [1] — enough for supervised classification, but far below the size of the language. CISLR inverted the trade-off: 7,050 videos of roughly 4,700 words, most with a single reference, framing large-vocabulary ISL as one-shot retrieval [2]. iSign added more than 118,000 video-sentence pairs for sentence-level ISL [7]. The Indian Sign Language Research and Training Centre (ISLRTC) publishes per-word dictionary videos that this project treats as first-class supervision. The realistic regime for ISL is therefore thousands of words with one or two references each; methods that need dozens of examples per class do not fit it.

2.2 Appearance-based and pose-based recognition

Video networks such as I3D [8] set the closed-set baselines on WLASL and INCLUDE, but they consume raw RGB: heavy for in-browser inference, tied to studio appearance, and requiring the camera stream as model input. Pose-based work replaces pixels with joints. ST-GCN introduced graph convolutions over joint sequences [9]; SAM-SLR carried skeleton ensembles to the top of sign recognition challenges [10]; SPOTER showed that a small pose transformer is competitive [11]. OpenHands demonstrated pose-based pretraining across several sign languages including ISL [12]. MediaPipe supplies on-device body and hand landmarks at camera rate [13], [14], which is what makes the browser a viable target. RTMW is a stronger whole-body landmark model that this project evaluated as an alternative front end (§5.6) [15].

2.3 The one-shot, open-vocabulary regime

With one reference per word, classification becomes matching: learn an embedding in which performances of the same sign cluster, and recognise by distance to class prototypes [16]. Additive-margin losses such as ArcFace [17] and CurricularFace [18] make such embeddings discriminative. CISLR’s baseline follows this framing [2]. Reported open-vocabulary accuracies stay low in absolute terms, which the literature attributes to signer variation against single references — a data problem at least as much as a model problem.

2.4 Template matching

Before deep learning, gesture recognition was alignment: dynamic time warping (DTW) finds the least costly monotone alignment between two sequences [19]. Audio fingerprinting showed at industrial scale that retrieval against a reference library can beat open-ended learning when query and reference are performances of the same object [20]. This project keeps a DTW recogniser beside the neural ones, both as a fallback and as an instrument that separates what is hard because of the model from what is hard because of the data.

2.5 Latent-predictive self-supervision (JEPA)

Joint-Embedding Predictive Architectures learn by predicting the latent representation of hidden content from visible context, rather than reconstructing pixels or coordinates. I-JEPA established the approach for images [21]; V-JEPA extended it to video with temporal span masking [22]; V-JEPA 2 scaled it to large video pretraining with downstream planning [23]; V-JEPA 2.1 added dense predictive losses and released smaller distilled checkpoints [24]. The relevance to signing is structural: a predictor can only succeed on what is predictable across the corpus, so idiosyncratic signer style is treated as noise by the objective.

2.6 Self-supervision on skeletons and on sign language

S-JEPA predicts the latents of masked joints with an exponential-moving-average (EMA) target encoder and target centring to prevent collapse, with its largest gains in the low-label regime [25]. Skeleton2vec [26] and an IJCV benchmark of skeleton self-supervision [27] support the same finding: latent targets beat coordinate reconstruction for skeletal action features. For sign language, SignBERT+ introduced hand-model-aware masked pretraining [28]; SHuBERT predicts masked multi-stream clusters over about a thousand hours of ASL and transfers across several tasks [29]; SignMAE argues that masking should follow the articulators [30]; SSL-SLR cautions that contrastive negatives are ill-posed because different signs share sub-movements [31]; and Uni-Sign pretrains a unified sign model at scale [32]. YouTube-SL-25 released video identifiers for more than 3,000 hours of signing across 25+ languages, including ISL [33].

2.7 Knowledge distillation, label smoothing and calibration

Distillation compresses a cumbersome model or ensemble into a single deployable one by training the student on the teacher’s temperature-softened output distribution, with the soft-target term scaled by the square of the temperature [34]. Label smoothing replaces one-hot targets with a mixture of the target and a uniform distribution, which regularises over-confident classifiers [35]. Modern networks are often miscalibrated, and a single scalar temperature fitted on held-out data is a simple and effective remedy that cannot change the ranking of classes [36]. Integer quantisation makes such models small and fast enough for commodity CPUs [37].

2.8 Signer-independent evaluation and on-device inference

Evaluation methodology has converged on the concern at the centre of this project: the MSLR workshop runs an explicit signer-independent track [38], and continuous recognition work treats each signer as a domain whose identity should be removed from the features [39]. On the deployment side, ONNX Runtime Web executes quantised models in WebAssembly with SIMD and threads [40]. Sign language translation [41] is a separate field; this project cites it only to mark the boundary of its scope.

2.9 Research gap

Table 2.1. What the literature provides and what this project adds.

Prior work gives	Prior work does not give
Closed-set ISL classification at 263 classes [1]	Recognition over the dictionary-scale vocabulary, deployed on end-user devices
One-shot retrieval framing for ISL [2]	That framing running in the browser at camera rate
Latent-predictive SSL for images, video and skeletons [21], [25]	A JEPA on sign-language landmarks, or any ISL application of one
Multi-stream sign SSL at ASL scale [29]	The same for ISL, at a browser-deployable size
Signer-independent evaluation ethos [38]	An unseen-signer test computed through the exact deployed numeric contract
Distillation and temperature calibration [34], [36]	A calibration that must transfer across two corpora before it ships

CHAPTER 3

System Design and Architecture

The system is three architectures bound by one numeric contract. The **run-time** architecture is what a user's browser executes. The **training** architecture produces the models. The **evaluation** architecture decides whether a model may replace the one in production. A fourth loop — the **data pipeline** — keeps the training data honest. The same canonicalisation (mid-shoulder anchor, shoulder-width scale with an upper-median clamp, aspect correction, absent groups zeroed) is executed by the browser at inference, by the extraction jobs, and by the evaluation harness, so a schema change reaches all three or none.

3.1 Design principles

1. **Nothing leaves the device.** Frames become landmarks in a worker and are discarded. Models and banks are static files cached by the browser.
2. **The bundle is the contract.** Every threshold, label list and landmark index is read from the model bundle at run time, so a retrained model is a file drop, not an application release.
3. **Measure what the user sees.** Evaluation replays the same int8 graph, the same canonical tensors and the same mirror pass the application runs.
4. **Name things by their size.** The interface says *260-sign set*, *full vocabulary* and *shape match*, so that no label can be read backwards.

3.2 Run-time architecture

Figure 3.1 shows the application at run time. A webcam frame enters a pose worker, where MediaPipe extracts body, hand and face landmarks; these are canonicalised into 107 points and pushed into a 64-frame ring. Each window is scored by three recognisers:

- **260-sign set.** The ISL-JEPA classifier (int8 ONNX, 4.57 MB) runs on the window and on its mirror image; the more confident side answers, and a fitted temperature turns its output into a calibrated probability.
- **Full vocabulary.** The supervised v3.5.2 encoder (9.9 MB) maps the window to a 256-dimensional embedding, compared by cosine similarity against a prototype bank of 8,497 rows that reaches 8,165 of the 8,188 vocabulary words.
- **Shape match.** A model-free DTW over hand-crafted kinematics runs beside them, with no learned weights.

The surfaces — Learn, Practice, Dictionary look-up and the Studio — read from these recognisers. The application is served as static files from Cloudflare at `studio.sanket.kadal.cc`; the server never sees a frame.

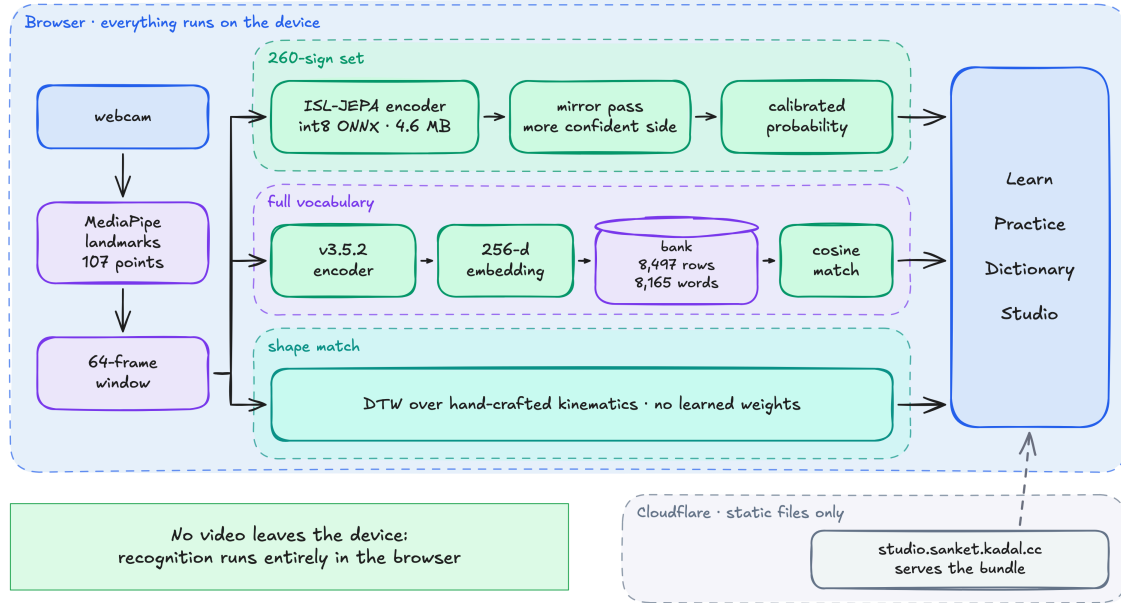


Figure 3.1. The system at run time. Everything inside the browser region runs on the user’s device; the only network traffic is the static download of the application and model files.

Table 3.1. Run-time artifacts and their measured size and speed.

Artifact	Role	Measured
ISL-JEPA classifier (int8 ONNX)	260-sign set: calibrated probabilities, mirror pass	4.57 MB · 12.5 ms/window (jepa-rr4)
Supervised encoder v3.5.2 (int8 ONNX)	Full vocabulary: 256-d embedding	9.90 MB · 34.6 ms/window
Prototype bank	Full vocabulary: 8,497 rows, cosine	8,165 of 8,188 words reachable
Shape-match references	DTW over kinematics	no learned weights

3.3 Training architecture

Figure 3.2 shows how a model is made. Corpora are extracted once through the browser-parity landmark contract into a canonical pack of 107 points $\times$ 64 frames. An articulator-token transformer is pretrained with the ISL-JEPA objective over the 123,340-clip pool. The encoder is cut at layer 4 (the layers above it stopped transferring in the Phase B run), fine-tuned on INCLUDE with gloss-matched dictionary clips, synthetic signers and recover-and-resample augmentation, repeated over several seeds and encoders, and the resulting ensemble is distilled into one student. The student is exported to int8 ONNX with the softmax inside the graph, calibrated, packaged as a Lab bundle, and promoted only through the accuracy test.

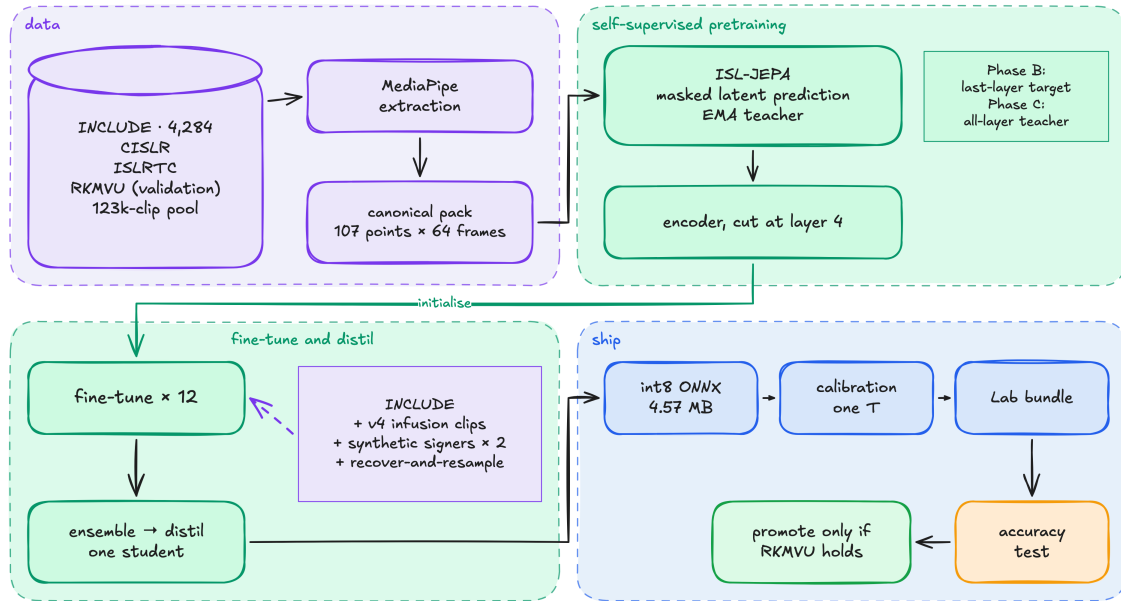


Figure 3.2. How a model is made: data, pretraining, fine-tuning and shipping stages. Each stage writes a versioned artifact that the next stage reads.

3.4 Evaluation architecture

Figure 3.3 shows the accuracy test and its side checks. Two sets feed one number. RKVMU-220 is 220 dictionary clips from one signer who appears nowhere in training — *the stranger number*. The CISLR test role is 379 clips that were never trained on, but whose presenters also appear in CISLR training clips. Their clip-weighted top-1 over 599 clips is the accuracy test (called *the gate* in the training ledgers); one clip is worth 0.17 percentage points. The side checks are fixed in advance: at least four seeds per claim, a leak check against training rows, calibration cross-fitted across the two corpora, and in-app parity through the int8 graph with the mirror pass. The promotion rule is that the unseen-signer number must not regress.

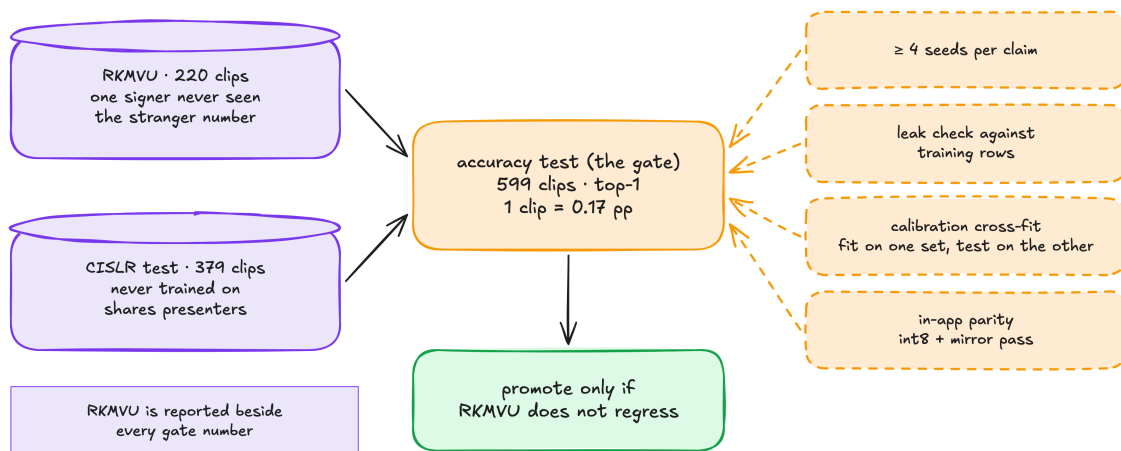


Figure 3.3. The accuracy test: two sets, one number, four side checks, and the promotion rule.

3.5 Data pipeline and review loop

Figure 3.4 shows the loop that turned dataset quality into accuracy. The Dataset tab shows every clip of four corpora with its video, extracted landmarks and a figure rendering. Marks — at clip, word or corpus-rule level, from an agent or a person, with a `needsReview` flag — are compiled by one command (`dataset-marks.ts apply`) into the contract the training packs are built from. Ambiguous cases go to a vision-model review of stick-figure packets; what it cannot settle is escalated to a blind RGB review with hidden controls; and what remains goes to a person in the Review tab, whose decisions flow back into the marks.

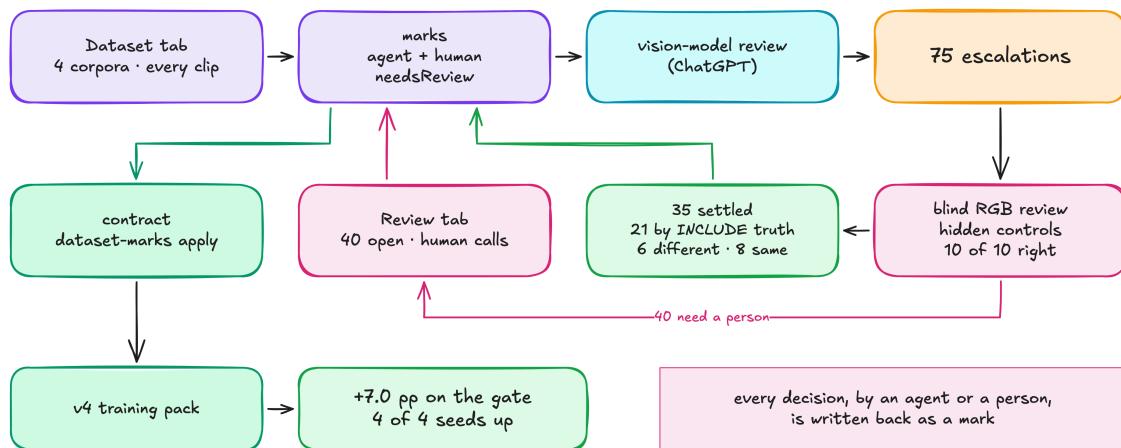


Figure 3.4. The dataset review loop, from marks to a rebuilt training pack. The arrow from the Review tab back to the marks closes the loop.

CHAPTER 4

Implementation of Algorithms and Techniques

This chapter is Review 2 deliverable 1. It describes the eleven techniques implemented, each with what it is, why it is needed, and how it is built. Five are given as algorithm boxes. Hyperparameters that are part of a technique’s definition are stated; the remaining ones live in the committed training scripts listed in Appendix B.

4.1 Technology stack

Table 4.1. Technology stack.

Layer	Technology	Used for
Application	TypeScript, React, Vite, TanStack Router	The studio single-page application and its surfaces
In-browser inference	onnxruntime-web (WebAssembly, SIMD, threads) [40]	int8 classifier and encoder; workers per recogniser
Landmarks	MediaPipe (pose, hands, face) [13], [14]	107 canonical points per frame, on-device
Training	PyTorch	ISL-JEPA pretraining, fine-tuning, distillation, export
Export	ONNX (opset 17), int8 dynamic quantisation [37]	Browser-shippable graphs with the softmax inside
Compute	Modal (A10G GPUs) until 27 Sep, then the local Mac GPU; earlier Colab Pro and Beam	Fine-tune and distillation gauntlet; corpus extraction
Hosting	Cloudflare Workers (static assets)	sanket.kadal.cc (docs) and studio.sanket.kadal.cc (app)
Evaluation	Playwright-driven Eval bench; Python harnesses	The accuracy test in-browser and offline

Table 4.2. Datasets and the role each plays.

Corpus	Size	Role in this project
INCLUDE [1]	4,287 videos, 263 classes, 15 categories; 4,284 extracted	Training corpus for the 260-sign set
CISLR [2]	7,050 videos, ~4,700 words	Gloss-matched infusion (training role) and the 379-clip test role (never trained on)
ISLRTC dictionary	4,804 of 11,993 extracted (download quota)	Infusion clips and full-vocabulary references
RKMVU (RKMVERI)	220 validation clips, one signer	The unseen-signer test; never trained on
Self-supervised pool	123,340 clips (grown from the Review 1 pool of 118,168 units, 6 sources)	ISL-JEPA pretraining only (no labels)

4.2 Landmark extraction and canonicalisation

What. MediaPipe extracts 33 pose points, 21 points for each hand, and a face subset; these are mapped to 107 canonical 2-D points per frame. **Why.** Landmarks remove appearance — skin, clothing, background, lighting — which a small training set would otherwise memorise, and they are the only representation the privacy constraint allows to persist. **How.** Each frame is translated to the mid-shoulder origin and scaled by shoulder width (with an upper-median clamp against tracker jitter), corrected for the camera’s aspect ratio, and frames where a hand is absent have that hand’s points zeroed. A clip is resampled to 64 frames. The same code path runs in the browser worker, the extraction jobs and the evaluation harness.

4.3 ISL-JEPA self-supervised pretraining

What. A joint-embedding predictive objective on landmark tokens. Each frame contributes three articulator tokens (body, left hand, right hand), giving 192 tokens for a 64-frame window; an 8-layer transformer encodes them. **Why.** Labelled ISL is scarce and signer-specific. A pool of 123,340 unlabelled clips covers far more signers, and predicting latents (not coordinates) rewards what is predictable across signers and discards individual style [21], [25]. **How.** A context encoder sees a masked or corrupted view and a predictor must produce the latents that an EMA target encoder assigns to the clean clip. Three masks are used: M1 hides an articulator stream, M2 hides temporal spans, and M3 feeds a deliberately corrupted view (mirrored, tempo-warped, rescaled, tracker-noised) whose latents must match the clean clip’s. Targets are layer-normalised and EMA-centred; a variance hinge on the latent standard deviation guards against collapse. Phase B predicts the teacher’s last layer; Phase C uses an all-layer teacher. In Phase B the last-layer target collapsed while layers up to 4 still transferred, so every shipped encoder is cut at layer 4.

The prediction losses are smooth- L_1 between predicted and target latents on the masked set $\mathcal{M}$:

$$\begin{aligned} \mathcal{L}_{\text{JEPA}} = & \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \text{SmoothL1}(\hat{z}_i, \text{sg}(\bar{z}_i - c)) \\ & + \lambda_v \mathcal{L}_{\text{view}} + \lambda_\sigma \sum_d \max(0, \gamma - \sigma_d(z)) \end{aligned} \quad (4.1)$$

where $\hat{z}_i$ is the predictor output, $\bar{z}_i$ the layer-normalised target latent, c the running centre, sg stop-gradient, $\mathcal{L}_{\text{view}}$ the M3 nuisance-view term, and σ_d the per-dimension standard deviation of the context latents.

Algorithm 4.1: ISL-JEPA pretraining step

Input: batch of canonical windows X (64 frames $\times$ 107 points); context encoder f_θ ; predictor g_φ ; EMA target encoder f_ξ ; centre c ; momentum τ

Output: updated θ, φ, ξ, c ; after training, the encoder is cut at layer 4

- 1: Tokenise each frame into three articulator tokens (body, left hand, right hand): 192 tokens per window.
 - 2: Sample a mask m : M1 hides one articulator stream, or M2 hides temporal spans.
 - 3: Build a nuisance view $\tilde{X}$ of the same windows (mirror, tempo warp, morphology rescale, tracker noise) for M3.
 - 4: Targets without gradient: $\bar{Z} \leftarrow \text{LN}(f_\xi(X)) - c$ (Phase B: last layer; Phase C: all layers).
 - 5: Context: $H \leftarrow f_\theta(X \odot \neg m)$; prediction $\hat{Z}_m \leftarrow g_\varphi(H, m)$.
 - 6: $\mathcal{L}_{\text{mask}} \leftarrow \text{SmoothL1}(\hat{Z}_m, \bar{Z}_m)$; $\mathcal{L}_{\text{view}} \leftarrow \text{SmoothL1}(q(f_\theta(\tilde{X})), \bar{Z})$.
 - 7: $\mathcal{L}_{\text{var}} \leftarrow \sum_d \max(0, \gamma - \sigma_d)$ over the context latents.
 - 8: Update θ, φ by one optimiser step on $\mathcal{L}_{\text{mask}} + \lambda_v \mathcal{L}_{\text{view}} + \lambda_\sigma \mathcal{L}_{\text{var}}$.
 - 9: EMA: $\xi \leftarrow \tau \xi + (1 - \tau) \theta$ with τ ramped towards 1; centre $c \leftarrow 0.99c + 0.01 \cdot \text{mean}(\bar{Z})$.
-

4.4 Fine-tuning recipe: infusion, synthetic signers, recover-and-resample

What. The recipe named `all+synth:full@2+rr` fine-tunes the cut encoder with an attentive head on the 260 classes. **Why.** INCLUDE’s signers are few and its studio is uniform; the unseen-signer test differs in presenter, framing, speed and where the sign starts and stops. **How.** Three additions, each measured before it was kept:

- **Gloss infusion.** Clips from CISLR and ISLRTC whose gloss matches one of the 260 classes (all 260 covered) are added to the training set.
- **Geometric synthetic signers.** Each real clip’s MediaPipe landmarks, including their depth coordinate, are lifted to 3-D, rotated in yaw and pitch, retargeted to different limb proportions, and projected back to 2-D — two variants per clip (Figure 4.1). On their own the synthetic signers add about nothing; stacked on the real infusion clips they lifted the 5-seed mean cross-validation from .3018 to .3927 (+9.1 pp).
- **Recover-and-resample (rr).** With probability 0.5 per batch, up to 25% of the frames are dropped independently from each end of the clip and the remainder is stretched back to 64 frames by linear interpolation, so the model sees signs that start late or end early — the way dictionary clips differ from INCLUDE takes.

one real clip in, two new signers out

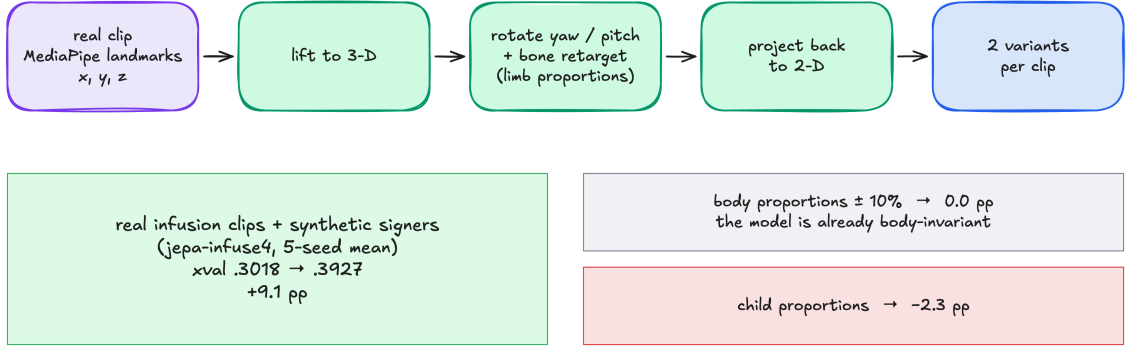


Figure 4.1. Synthetic signers from real clips: lift to 3-D, rotate and retarget, project back to 2-D. Alone they add about nothing; stacked on real infusion clips they add +9.1 pp; changing body proportions by ±10% changes nothing (0.0 pp).

Algorithm 4.2: Fine-tune with infusion, synthetic signers and recover-and-resample

Input: encoder f cut at layer 4; INCLUDE training clips D_I ; gloss-matched infusion clips D_G (CISLR + ISLRTC); synthesiser S ; epochs E

Output: one fine-tuned classifier per seed

- 1: $D \leftarrow D_I \cup D_G$.
 - 2: For each real clip $x \in D$: append two synthetic variants $S(x)$ (random yaw/pitch, bone retarget, re-projection to 2-D).
 - 3: For epoch $e = 1..E$, for each mini-batch $B \subset D$:
 - 4: apply the standard spatial augmentations and the mirror augmentation;
 - 5: with probability 0.5 (rr): draw $a, b \sim \mathcal{U}[0, 0.25T]$, keep frames $[a, T - b]$ if at least 8 remain, and resample them linearly to $T = 64$;
 - 6: logits $\leftarrow \text{head}(\text{attentive-pool}(f(B)))$; loss $\leftarrow \text{CE with label smoothing } 0.1$ [35];
 - 7: optimiser step on encoder and head.
 - 8: Repeat for each seed (≥ 4 per claim).
-

4.5 Ensemble to single-model knowledge distillation

What. Several fine-tuned models are averaged into a teacher and distilled into one browser-sized student [34]. **Why.** Averaging fine-tunes scores higher than any single one — each four-seed ensemble in Table 5.3 is above every seed of Table 5.2 — but twelve models do not fit a browser budget. **How.** The online teacher is the mean softmax of the N fine-tunes, $\bar{p}$, sharpened or softened at temperature T by renormalising its powers; the student is trained on

$$\mathcal{L}_{\text{KD}} = (1 - \alpha) \cdot \text{CE}_\varepsilon(y, \text{softmax}(z_s)) + \alpha T^2 \text{KL}(\tilde{p}_T \parallel \text{softmax}(z_s/T)) \quad (4.2)$$

$$\tilde{p}_T(k) = \frac{\bar{p}(k)^{1/T}}{\sum_j \bar{p}(j)^{1/T}}, \quad \bar{p} = \frac{1}{N} \sum_{n=1}^N \text{softmax}(z_n) \quad (4.3)$$

with label smoothing $\varepsilon = 0.1$, $\alpha = 0.7$, $T = 2$ and 40 epochs. The T^2 factor keeps the soft-target gradients on the same scale as the hard-label term [34]. The current student, jepa-x12,

is distilled from twelve fine-tunes (four on Phase B encoders and eight on Phase C encoders) into a Phase B student cut at layer 4 (Figure 4.2).

Algorithm 4.3: Ensemble distillation

Input: fine-tuned teachers $M_1 \dots M_N$; student S (Phase B encoder, cut at 4); $\alpha = 0.7$, $T = 2$, 40 epochs

Output: one student model, exported to int8 ONNX

- 1: For each epoch and mini-batch B (with the same augmentations as fine-tuning):
 - 2: teacher: $\bar{p} \leftarrow \frac{1}{N} \sum_n \text{softmax}(M_n(B))$ (no gradient);
 - 3: tempered target $\tilde{p}_T \leftarrow \bar{p}^{1/T} / \sum \bar{p}^{1/T}$;
 - 4: student logits $z_s \leftarrow S(B)$;
 - 5: $\mathcal{L} \leftarrow (1 - \alpha) \text{CE}_{0.1}(y, z_s) + \alpha T^2 \text{KL}(\tilde{p}_T \parallel \text{softmax}(z_s/T))$;
 - 6: optimiser step on S .
-

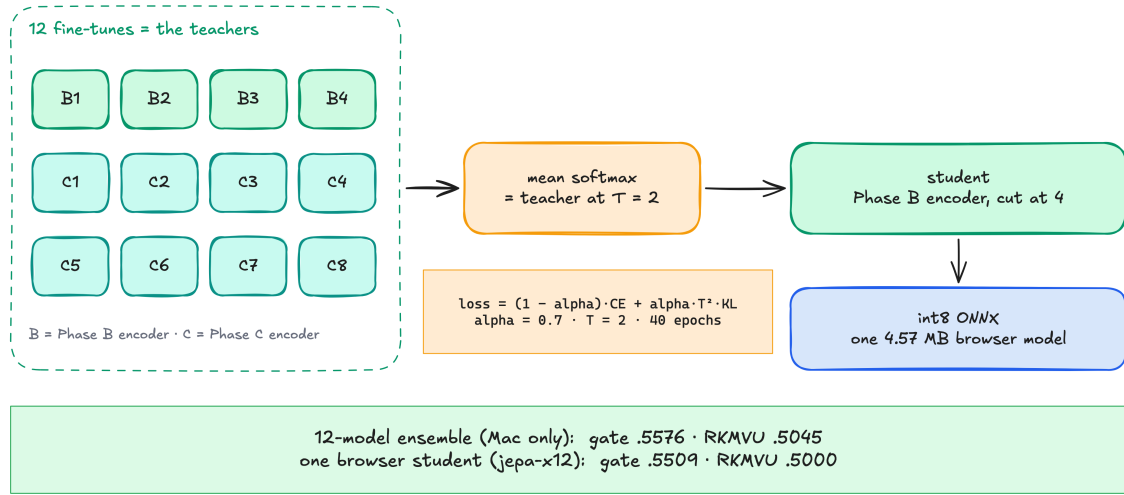


Figure 4.2. Twelve models into one: the mean softmax of twelve fine-tunes teaches a single Phase B student. The ensemble scores .5576 on the accuracy test (RKMVU .5045); the student keeps .5509 (RKMVU .5000) in one 4.57 MB browser model.

4.6 Export and quantisation

What. The student is exported to ONNX (opset 17) with the softmax baked into the graph, then quantised to int8 with dynamic quantisation [37]. **Why.** The browser needs one self-contained graph of a few megabytes that produces probabilities directly. **How.** After export, a parity check compares the PyTorch and ONNX outputs on real windows; the maximum difference is at most 1.3×10^{-6} . The int8 graph is then scored on the accuracy test like any other candidate; int8 and fp32 numbers are reported separately (Chapter 5).

4.7 Temperature calibration with cross-fitting

What. One scalar temperature per bundle turns the classifier’s output into a probability that means what it says [36]. **Why.** The JEPa classifiers are *under*-confident: Practice credits a sign only above a confidence floor, so an uncalibrated model refused learners who had signed

correctly. **How.** The temperature acts on the log-probabilities of exactly the posterior the user sees — the mirror-pass posterior of §4.8:

$$q_i = \frac{p_i^{1/T}}{\sum_j p_j^{1/T}}, \quad T^* = \arg \min_{T \in \mathcal{G}} -\frac{1}{n} \sum_{k=1}^n \log q_{y_k}^{(k)}(T) \quad (4.4)$$

where $\mathcal{G}$ is a grid of 400 log-spaced values in $[0.2, 3.0]$. Because the only labelled cross-signer data are the accuracy-test clips, the fit is cross-checked: T fitted on RKMVU is scored on CISLR and vice versa, using the expected calibration error (ECE, 10 bins)

$$\text{ECE} = \sum_{b=1}^{10} \frac{|B_b|}{n} |\text{acc}(B_b) - \text{conf}(B_b)|. \quad (4.5)$$

The pooled temperature ships only if each one-corpus fit lowers the ECE on the other corpus. A temperature cannot change which class ranks first, so it cannot move top-1 on any benchmark. For jepa-x12 the shipped temperature is $T = 0.788$ ($T < 1$ sharpens the distribution).

Algorithm 4.4: Temperature calibration with cross-fit

Input: classifier M ; accuracy-test clips split by corpus R (RKMVU) and C (CISLR); grid $\mathcal{G}$

Output: a temperature T^* or a documented refusal

- 1: For every clip x : $p \leftarrow M(x)$, $p' \leftarrow M(\text{mirror}(x))$; keep the side with the larger maximum probability (the posterior the app shows).
 - 2: $T_R \leftarrow \arg \min_{T \in \mathcal{G}} \text{NLL}_R(T)$; $T_C \leftarrow \arg \min_{T \in \mathcal{G}} \text{NLL}_C(T)$.
 - 3: transfers $\leftarrow \text{ECE}_C(T_R) < \text{ECE}_C(1)$ and $\text{ECE}_R(T_C) < \text{ECE}_R(1)$.
 - 4: If transfers: $T^* \leftarrow \arg \min_{T \in \mathcal{G}} \text{NLL}_{R \cup C}(T)$; write T^* into the bundle.
 - 5: Otherwise: report the disagreement and ship the bundle uncalibrated.
-

4.8 Mirror test-time augmentation

What. Every window is scored twice — as captured and mirrored — and the more confident side answers. **Why.** A left-hand-dominant signer produces the mirror image of a right-dominant performance; at Review 1 such a signer scored 0.12 against 0.33, and the mirror pass restored the full 0.33. **How.** The canonical normalisation is symmetric about the body midline, so mirroring is exact: negate the horizontal coordinate and swap every left–right pair (shoulders, elbows, wrists, hips, and the two hands).

Algorithm 4.5: Mirror test-time augmentation decision

Input: canonical window x ; classifier M with temperature T^*

Output: a word and a calibrated confidence

- 1: $x' \leftarrow x$ with the horizontal coordinate negated and left/right landmark pairs swapped.
 - 2: $p \leftarrow M(x)$; $p' \leftarrow M(x')$.
 - 3: If $\max_k p'_k > \max_k p_k$ then $\hat{p} \leftarrow p'$ else $\hat{p} \leftarrow p$.
 - 4: $q \leftarrow \text{temper}(\hat{p}, T^*)$ (Eq. (4.4)); answer $\arg \max_k q_k$ with confidence $\max_k q_k$.
 - 5: Practice credits the target word when $q_{\text{target}} \geq 0.5$ (calibrated floor).
-

4.9 Full-vocabulary retrieval

What. The full vocabulary is reached by similarity, not by classification. **Why.** Most of the 8,188 words have one or two reference clips; a classifier cannot be trained for them, but an embedding can compare against them [16]. **How.** A supervised encoder trained with an ArcFace margin loss [17] maps a window to a 256-dimensional unit vector. The prototype bank holds 8,497 rows; a window’s answer is the word of its most similar row, emitted only above a confidence bar on the cosine. Because 833 words are signed in more than one way, the bank keeps one variant per clip rather than one averaged prototype per word; rebuilding it that way gained +10.7 pp.

4.10 Shape match: model-free dynamic time warping

What. A recogniser with no neural network: hand-crafted kinematic features (hand shape, orientation, position relative to the body, motion) aligned by DTW [19] against reference performances (`fingerprint.ts`). **Why.** It is a fallback that needs no training, and it is an instrument: when the model and the matcher fail on the same clip, the data are suspect; when only the model fails, the learning is. **How.** A dwell-weighted, orientation-aware alignment scores every reference; the best word answers. On the 260-sign version of the accuracy test it improved from 7.0% to 16.7% and, on the current all-260 bench, to 25.0%. Its use as a helper to the model is measured in §5.7.

4.11 Dataset quality system

What. A system that records problems with clips and words and compiles them into the training data. **Why.** Several of the largest measured gains came from data, not from models. **How.** Marks are attached to a clip, a word, or a corpus-wide rule, drawn from a catalogue of issue types, created by an agent or a person, and flagged `needsReview` when uncertain. One command (`dataset-marks.ts apply`) compiles the marks into the contract that the training packs are built from. Ambiguous sense conflation — a clip labelled with a specific sense such as “rough (surface)” standing for a general word — are reviewed first by a vision model on stick-figure packets, then escalated to a blind RGB review with hidden controls (all 10 committed controls answered correctly), and finally, for what remains, to a person in the Review tab (Appendix A).

4.12 Compute

Fine-tunes and distillations run on Modal A10G GPUs through a gauntlet harness (`gauntlet.py`) that executes the unmodified training scripts, one GPU per arm: about 5 minutes per fine-tune seed and about 17 minutes per distillation. Earlier stages used Colab Pro and Beam; corpus extraction ran on cloud CPU task queues. The Review 1 record measured the pilot spend at about \$2 of GPU and the corpus build at about \$12 of CPU. The Modal workspace stopped being available at about 01:15 on 27 September (workspace disabled); the one remaining

distillation (a 16-teacher student) was moved to the local Mac's GPU, where it could not finish before the freeze; the model presented at Review 2 is therefore jepa-x12.

CHAPTER 5

Intermediate Results

This chapter is Review 2 deliverable 2. Every number is measured; targets are labelled as targets. The unseen-signer number (RKMVU) is reported beside every accuracy-test number.

5.1 The accuracy test

The accuracy test (the *gate* in the training ledgers) is clip-weighted top-1 over 599 clips: RKMVU-220, one signer the model never saw, and the CISLR test role of 379 clips, never trained on. One clip is 0.17 pp, which is why claims rest on at least four seeds. RKMVU is reported separately because it is the honest stranger number: the CISLR test clips share presenters with CISLR training clips, so CISLR gains partly reflect presenter familiarity.

5.2 Model lineage, Review 0 to Review 2

Table 5.1 lists every model that reached the accuracy test, with fp32 numbers unless stated, and Figure 5.1 places them on the review timeline. Review 0 reported in-corpus accuracy only (INCLUDE-263 top-1 0.958; CISLR 8,165-word open-vocabulary top-1 0.2429 and top-20 0.4105). Review 1 measured the cross-corpus gap (0.82 in-corpus versus 0.33 cross-corpus) and the mirror fix for a left-dominant signer (0.12 to 0.33).

Table 5.1. Model lineage on the accuracy test (fp32 unless stated). Bold rows are the model deployed since 4 September and the best single browser model at Review 2.

Model	Date	Gate	RKMVU	CISLR	Notes
Supervised v3.5.2 (retrieval encoder; old 260 lane)	Aug	.1469 ^b	.3318	.0396	9.90 MB, 34.6 ms/window; ^b browser
jepa-ship4 (Phase B JEPA, cut@4)	1 Sep	—	.3273 (int8 .3318)	—	4.57 MB
jepa-infuse4 (+ gloss infusion + synthetic signers)	2 Sep	.4124	.4318	—	5-seed mean xval .3018 → .3927
jepa-distil4 (4-seed ensemble → student)	3 Sep	.4424	.4545	.4354	
jepa-rr4 (+ recover-and-resample)	3 Sep	.4591	.4818 (int8 .4864)	.4459	deployed since 4 Sep; 12.5 ms/window, 4.57 MB
jepa_c4 (Phase C all-layer teacher)	4 Sep	.4524	—	—	a wash against rr4
jepa-v4 (reviewed v4 pack, 4 teachers → student)	26 Sep	.5359	.4818 (int8 .5000)	.5673	in-app .5412
jepa-x12 (12 teachers → Phase B student)	26 Sep	.5509	.5000 (int8 .4909)	.5805	in-app .5546; T = 0.788; RKMVU top-5 .6909

12-model ensemble (not browser-shippable)	26 Sep	.5576 (+mirror .5609)	.5045	.5884	Mac only
16-model ensemble (8 Phase B + 8 Phase C fine-tunes; not browser-shippable)	27 Sep	.5693 (+mirror .5626)	.5136 (top-5 .6955)	.6016	strongest measured configuration; Mac only; 16× the compute of one model
Frozen for Review 2: jepa-x12	27 Sep	.5509	.5000 (int8 .4909)	.5805	the pre-stated rule kept it: the 16-teacher student could not finish once Modal ended

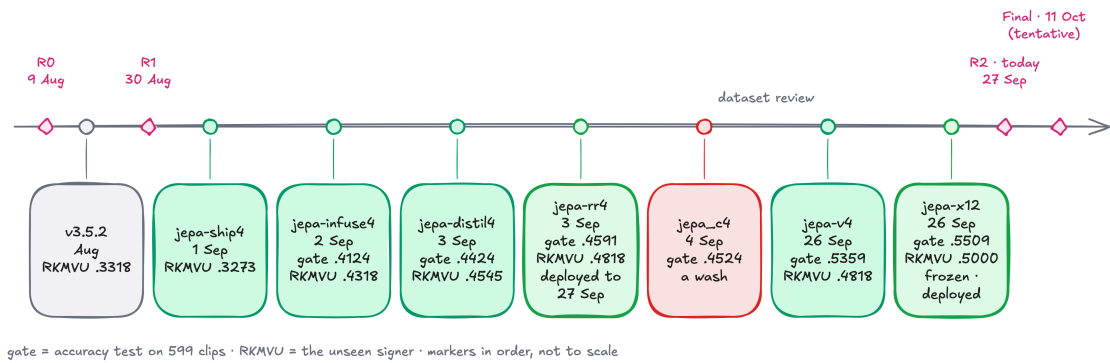


Figure 5.1. Model lineage from Review 0 to Review 2 on the review timeline (Review 0, 9 Aug; Review 1, 30 Aug; Review 2, 27 Sep; final review, 11 Oct, tentative).

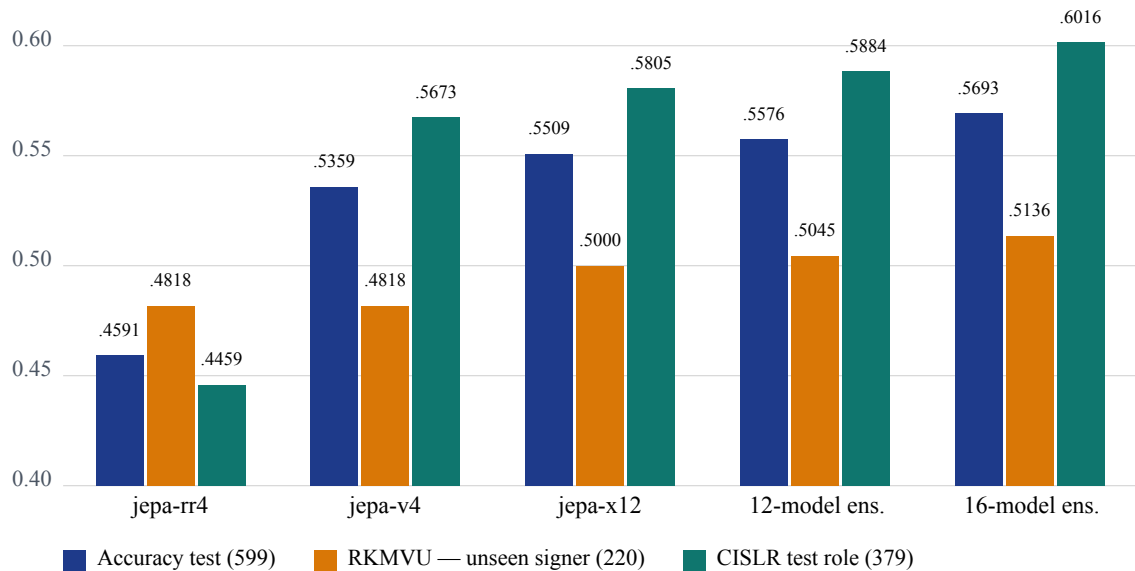


Figure 5.2. Accuracy test, unseen-signer (RKMVU) and CISLR test-role top-1 (fp32) from the deployed jepa-rr4 to the Review 2 models. The y-axis starts at 0.40. The gate gain is carried mostly by CISLR (+13.5 pp from rr4 to x12); the unseen-signer gain is +1.8 pp. The two ensembles (right) run on a workstation and are not browser-shippable as they stand.

5.3 Seed-level evidence for the dataset contract

The v4 pack is the training pack rebuilt from the reviewed dataset contract (§4.11). To separate its effect from seed noise, the champion recipe was trained four times on the previous pack (control) and four times on the v4 pack, in the same run.

Table 5.2. Accuracy test per seed: control recipe versus the reviewed v4 pack (4 seeds each, same recipe, same run).

Pack	Seed 1	Seed 2	Seed 3	Seed 4	Mean
Control (champion recipe)	.4057	.4174	.4124	.4107	.4116
v4 pack	.4825	.4942	.4691	.4808	.4817

All four seeds improved; the mean rose by +7.0 pp. The same comparison at the ensemble level:

Table 5.3. Four-seed ensembles: control versus v4 (Phase B).

4-seed ensemble	Gate	RKMVU	CISLR
Control	.4574	.4727	.4485
v4 (Phase B)	.5376	.4909	.5646

5.4 What the user sees in the application

The application runs the int8 graph with the mirror pass. Measured through that path on the same 599 clips, the in-app accuracy rose from .4538 (jepa-rr4) to .5412 (jepa-v4) and .5546 (jepa-x12). Calibration changes what Practice credits: with a floor of 0.5 on the calibrated confidence, jepa-x12 credits 69% of correct signs at 83% precision, against 61% at 82% for jepa-rr4.

Table 5.4. In-app accuracy (int8 + mirror pass) and Practice crediting at a calibrated floor of 0.5.

Model	In-app accuracy	Correct signs credited	Precision of credit
jepa-rr4	.4538	61%	82%
jepa-v4	.5412	—	—
jepa-x12	.5546	69%	83%

5.5 The honest reading

- **The gate rose +9.2 pp** (.4591 → .5509) with one browser model of the same size and speed.
- **The stranger number rose +1.8 pp** (.4818 → .5000). Most of the gate gain is CISLR (+13.5 pp), and part of that reflects presenter familiarity, since the CISLR test clips share presenters with CISLR training clips.
- **Where the gain came from.** The dataset contract (sense-conflation review) accounts for +7.0 pp of gate over four seeds; ensembling and distillation account for the rest.
- **The ceiling measured so far.** The 16-model ensemble (8 Phase B and 8 Phase C fine-tunes) is the strongest configuration measured: gate .5693, RKMVU .5136 (top-5 .6955), CISLR .6016. It needs sixteen times the compute of one model and is not browser-shippable

as it stands; jepa-x12 remains the single-model headline until the model freeze, and the ensemble is the target a distilled student aims at.

- **Against the Review 1 target.** The Phase B target of 0.50 is met on the gate (.5509) and is borderline on RKMVU alone (.5000 fp32, .4909 int8).

5.6 Negative results

These results did not improve the model. They are reported because each one closed a direction or changed the plan.

Table 5.5. Negative results, measured.

Direction	Measurement	Consequence
RTMW whole-body landmarks instead of MediaPipe [15]	Same trainer, 5 seeds, all 599 gate clips: MediaPipe .257 vs RTMW .238. RTMW's +13 pp on INCLUDE's own signer split (.497 → .625) does not transfer (CISLR −3 to −8 pp); RTMW invents legs on waist-framed video.	Stay on MediaPipe, which the browser must run anyway
Photometric pre-processing for missing hands (gamma, CLAHE, sharpen, denoise, upscale)	Null or negative. Hands go missing because they <i>move</i> (wrist speed 5.3 → 12.5 px/frame in 97% of clips) or leave the frame (68% of misses).	No image pre-pass; the problem is motion and framing
Shape match as an RL reward or pseudo-label verifier	Worse than the model's own confidence: 55.3% vs 68.0% precision at equal coverage.	Not used as a reward
Body proportions ±10%	0.0 pp (child proportions −2.3 pp).	The model is already body-invariant
Phase C objective (all-layer teacher)	A wash at full budget; Phase C students lose RKMVU (.4591, .4636).	Phase C used only as ensemble teachers
Boundary head (sign start/end)	Labels 94.9% of frames “inside a sign”, so it cannot trim idle lead-in and tail; the v3.5 export dropped the boundary output.	Idle trimming needs a new approach

5.7 Shape match as a helper, measured

The model-free shape match was measured both as a recogniser and as a helper to the model (Figure 5.3). Across all 260 classes it names 25.0% of clips; restricting it to the model's top-3 shortlist raises that to 34.7%, but the model alone reaches 45.7% on the same bench, so the shortlist never beats the model. As a label checker for pseudo-labelling or reinforcement-learning reward, its agreement is less precise than the model's own confidence (55.3% versus 68.0% at equal coverage). The verdict is that shape match stays a transparent fallback and diagnostic instrument, not a training signal.

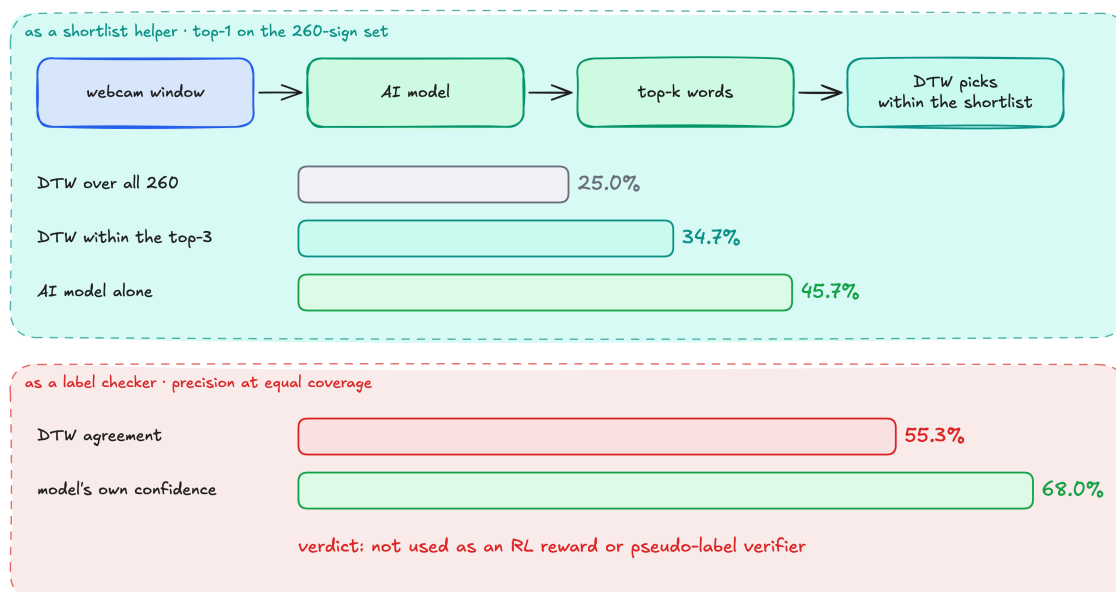


Figure 5.3. Shape match as a helper, measured: as a re-ranker inside the model’s shortlist (25.0% → 34.7%, against 45.7% for the model alone), and as a label checker (55.3% versus 68.0% precision at equal coverage).

Table 5.6. Shape match (model-free DTW) on the 260-class bench.

Configuration	Top-1
Shape match, first version	7.0%
Shape match, improved	16.7%
Shape match, all 260 classes (current bench)	25.0%
Shape match within the model’s top-3 shortlist	34.7%
AI model alone (same bench)	45.7%

5.8 Data findings

The dataset tools measured problems that no model change could fix.

Table 5.7. Data findings from the Dataset tab and the review loop.

Corpus / process	Finding
ISLRTC	64% of files exceed 8 s — lessons rather than single words; median clip 14.7 s; 17% silently truncated at 36 s; 767 phantom words created from file names
INCLUDE	76% of frames are idle (hands at rest)
Vocabulary	833 words are signed more than one way; keeping a variant per clip rebuilt the bank (+10.7 pp)
Vision review	75 escalations; 35 settled by blind RGB review (21 by INCLUDE ground truth, 6 confirmed different, 8 rejected as the same sign); 40 open in the Review tab

5.9 To be completed for the final review

To be completed for the final review, 11 October 2026 (tentative). The model frozen for Review 2 is jepa-x12 (last row of [Table 5.1](#); in the app .5546). Still to come: the 16-teacher student and a second student seed per recipe; multi-seed results for the next levers of §6.3; per-class error analysis on RKMVU; measured browser latency of the frozen bundle on a mid-range device.

CHAPTER 6

Expected Outcomes and Roadmap

This chapter is Review 2 deliverable 4.

6.1 Outcomes by the final review

By the final review (11 October 2026, tentative) the project is expected to deliver:

1. **A frozen model in production** with its accuracy-test and unseen-signer numbers measured and stated in the application’s own model card.
2. **The complete report** — this document with Chapters 5–7 finished.
3. **A live demonstration** on a webcam, in the browser, with nothing uploaded.

6.2 Targets against status

Table 6.1. Targets set at Review 1 against the status at Review 2.

Metric	Target (R1)	Status (R2)	Verdict
Accuracy test (599 clips)	≥ 0.50	.5509 (jepa-x12, fp32)	Met
Unseen signer (RKMVU-220)	≥ 0.50	.5000 fp32 · .4909 int8	Borderline
Mirrored signer	parity	mirror pass in every path	Held
Model size (260-sign set)	$\leq$ deployed	4.57 MB int8	Held

6.3 Roadmap: the next levers

The levers are ranked by expected value per unit of work, and every one is gated on the unseen-signer number ([Figure 6.1](#)).

1. **Settle the 40 open escalations and rebuild the pack.** The v4 gain came from exactly this kind of cleanup (+7.0 pp over four seeds).
2. **A canonical 3-D body frame** from MediaPipe world landmarks, so that camera angle stops being a nuisance the model has to learn around.
3. **A motion-style prior from the 123k pool**, used to re-perform labelled clips in the styles of the pool’s signers — synthetic signers for motion rather than body shape.
4. **Per-user adaptation from Practice attempts**, where the target word is a free, correct label for every attempt the learner chooses to contribute.

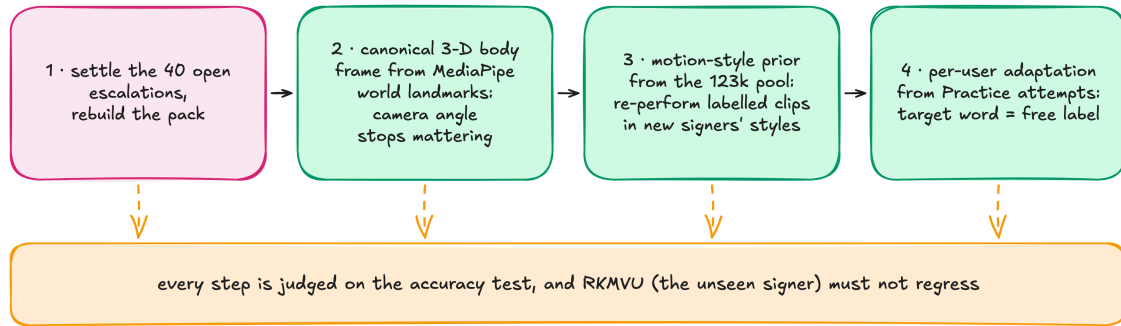


Figure 6.1. What comes next: four steps, each gated on the unseen-signer number.

6.4 Risks and mitigations

Table 6.2. Risks and mitigations.

Risk	Mitigation
Gate gains that do not reach unseen signers	RKMVU is reported beside every gate number and the promotion rule forbids an RKMVU regression
Seed noise on a 599-clip test (1 clip = 0.17 pp)	At least four seeds per claim; seed-level tables published
Test-set reuse through calibration	One scalar only, cross-fitted across corpora, shipped only if both directions transfer
Label noise in dictionary corpora	Dataset marks, vision review with blind controls, and human review of what remains
Tracker failure bounds any landmark model	Measured (motion and framing, not lighting); the plan targets framing and 3-D normalisation
Compute availability	The Modal workspace was disabled on 27 September; remaining distillations run on the local Mac GPU, and the training scripts run unmodified on either

6.5 Research outcomes: paper seeds

Table 6.3. Semester-2 paper seeds.

Seed	Working claim	Candidate venues
Systems and accessibility	On-device ISL recognition in the browser, with a measured cost model and an unseen-signer evaluation	ASSETS, W4A, CHI late-breaking work
Method and analysis	A cross-corpus audit of ISL recognition; ISL-JEPA; the reviewed data contract as a measured lever	LREC-COLING, ACL/EMNLP Findings, ICVGIP

6.6 Societal outcome (SDG 10)

The outcome this project is accountable to is a free, private, on-device ISL learning and recognition tool for India’s Deaf community and for the people learning to sign with them. No video leaves the device, the application runs in any modern browser, and its accuracy is reported on a signer it has never seen, so a user can know what to expect.

CHAPTER 7

Conclusion

To be completed for the final review, 11 October 2026 (tentative). This section is present as a headed outline; the bullets below fix its scope so the final text only fills it in.

- **Summary of contributions** — the on-device system; the accuracy test computed through the deployed contract; ISL-JEPA with the cut-at-4 finding; the fine-tuning recipe and ensemble distillation; the cross-fitted calibration; the dataset quality system.
- **Answer to the problem statement** — the frozen model’s accuracy-test and unseen-signer numbers against the Review 1 targets.
- **Limitations** — the unseen-signer test is one signer; CISLR shares presenters; the full-vocabulary path is not yet retrained on the invariant encoder; continuous signing is out of scope.
- **Future work** — the roadmap levers not completed by the final review; the signer study; the two paper seeds.

APPENDIX A

Screenshots of the Working System

All screenshots were taken on 27 September 2026 from the running local studio (development build, which also shows the developer-only tabs) at 1488×834 pixels, light theme. Where a camera is shown, the input is a public RKMVERI dictionary clip played into the pipeline in place of a webcam.

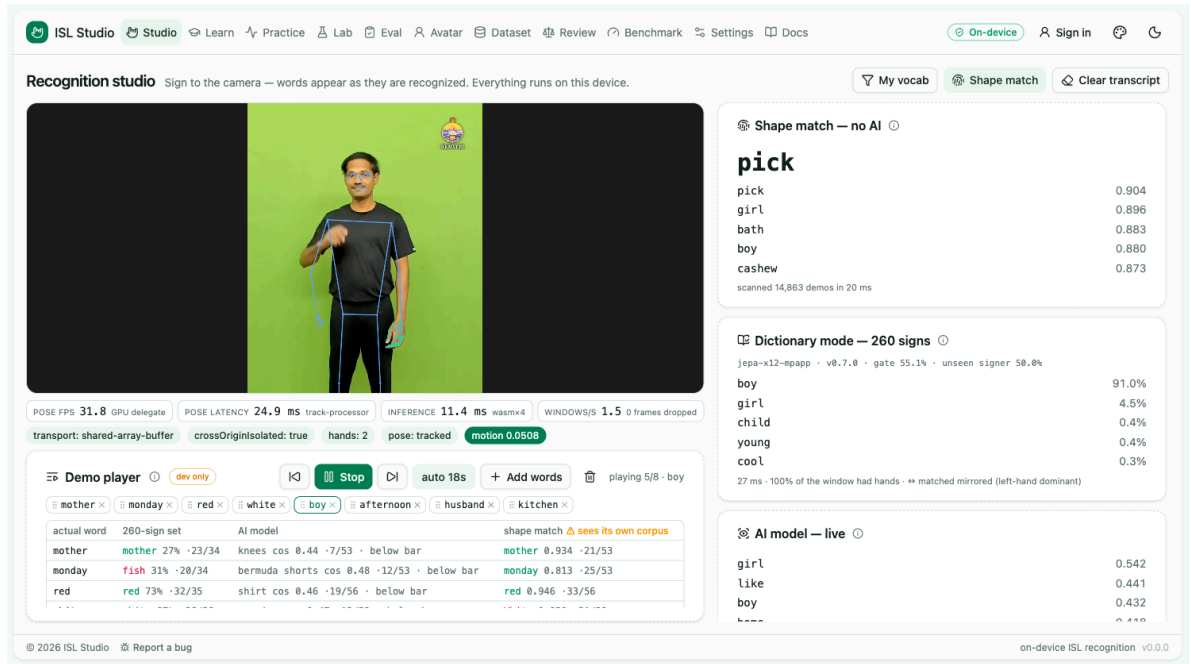


Figure A.1. Studio with the demo player: dictionary clips play through the real pipeline. The *Dictionary mode — 260 signs* panel shows jepa-x12 (the frozen model) naming *boy* at 91%, and the table below the video records, per clip, the actual word against each recogniser: the 260-sign set named mother, red, white and boy and missed monday. The full-vocabulary column (“AI model”) had repeated one word for every clip until a normalisation bug was fixed on 27 September (hand depth was anchored to the shoulders, which training never does); it now varies with the sign but is weak on live video and mostly stays below its confidence bar. That lane is not part of the Review 2 claims.

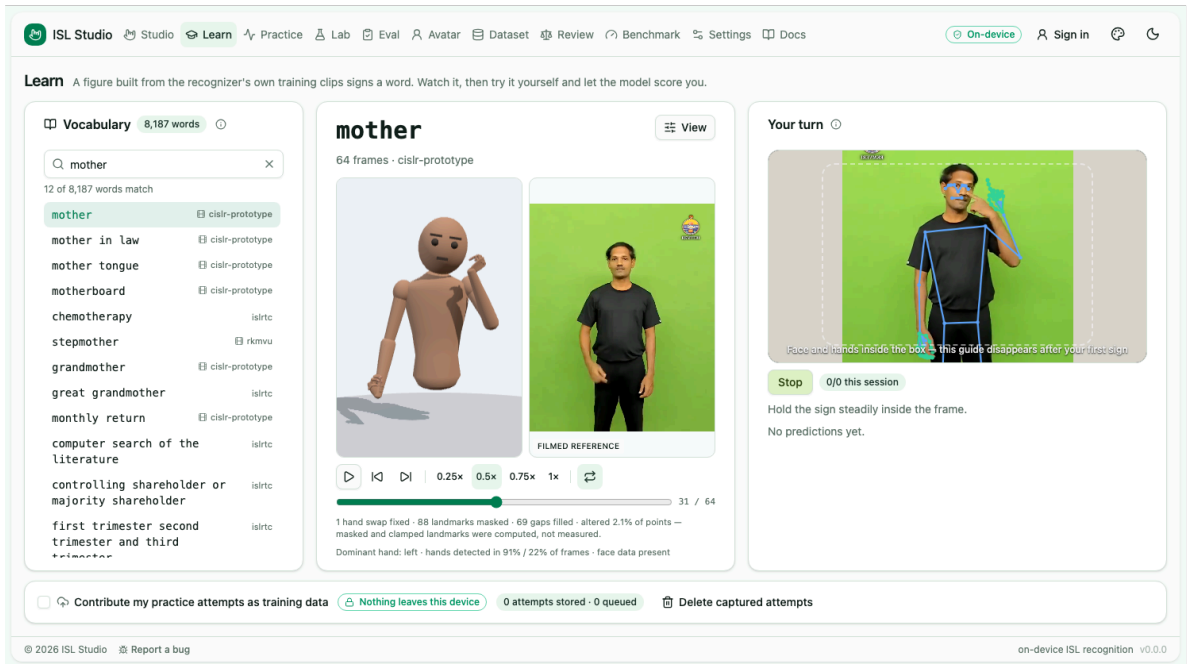


Figure A.2. Learn: the vocabulary list (8,187 words), a figure signing *mother* from the recogniser’s own training clip beside the filmed reference, and *Your turn* with the camera running.

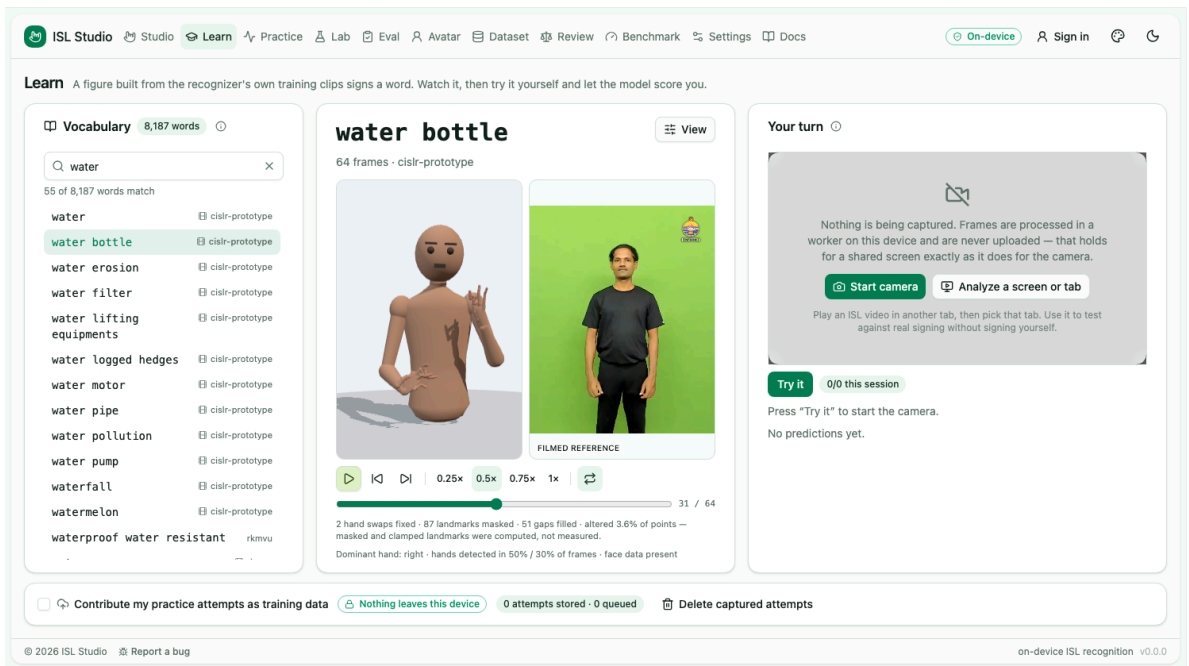


Figure A.3. Dictionary look-up in Learn: searching “water” returns 55 of the 8,187 words with their source corpus; the selected entry is demonstrated by the figure and the filmed reference.

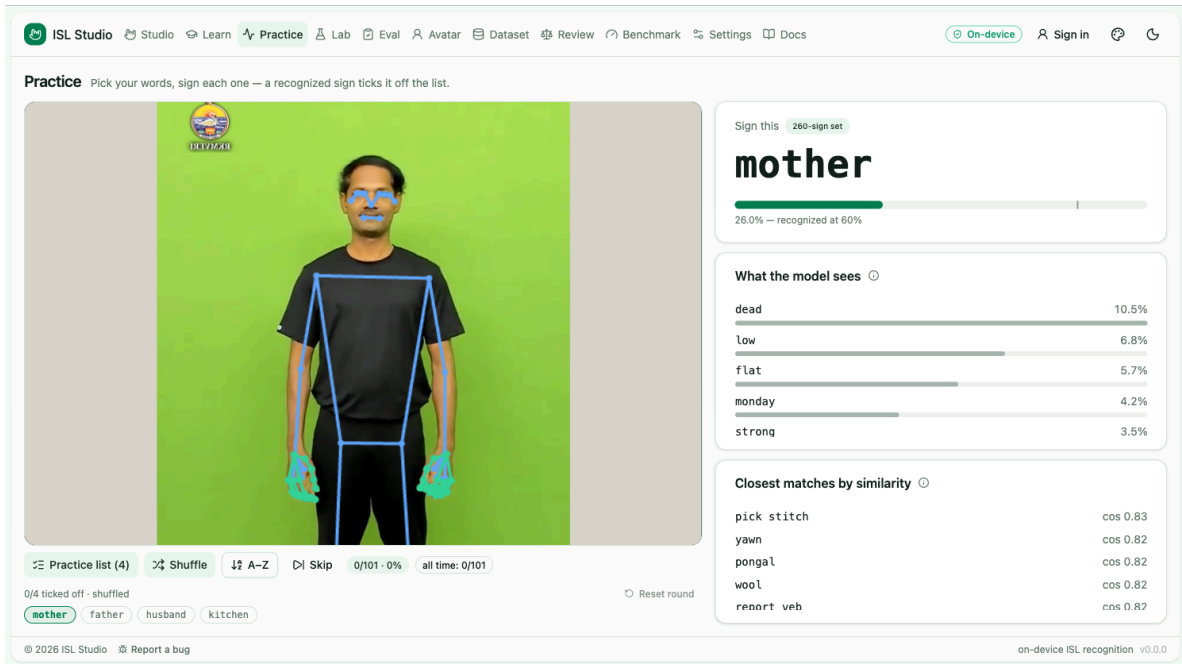


Figure A.4. Practice: a four-word list, the target word with live progress towards the credit bar, the 260-sign set’s top five, and the closest full-vocabulary matches.

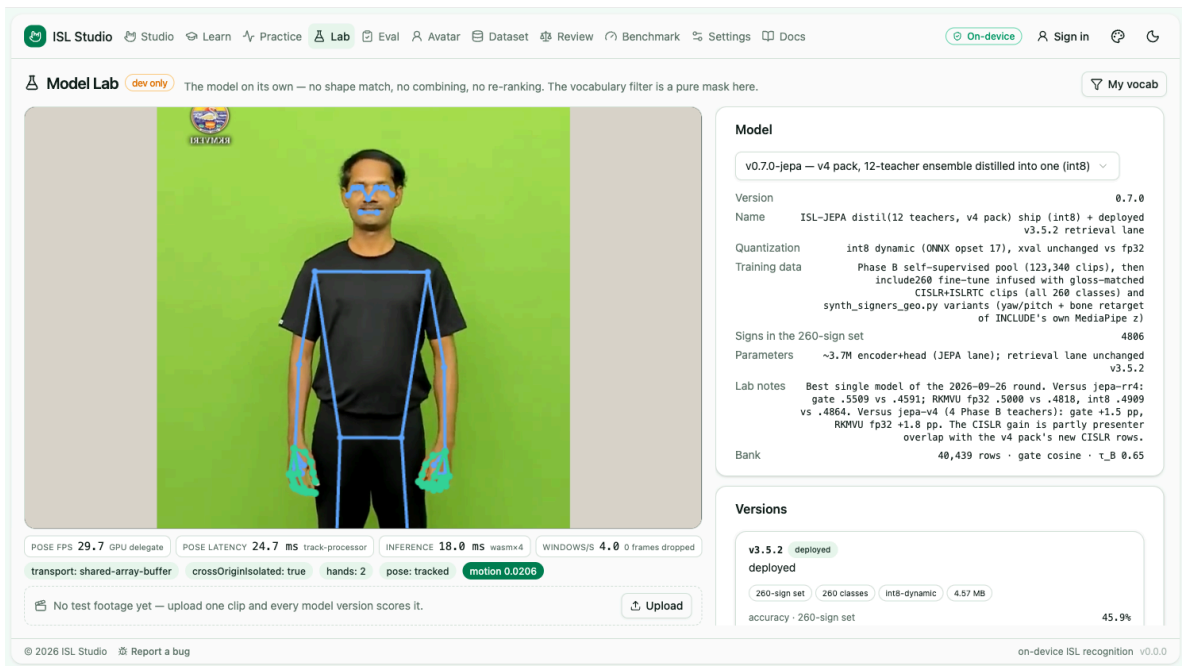


Figure A.5. Model Lab (developer only): the jepa-x12 bundle’s model card read from the bundle itself, running on its own with no shape match or re-ranking.

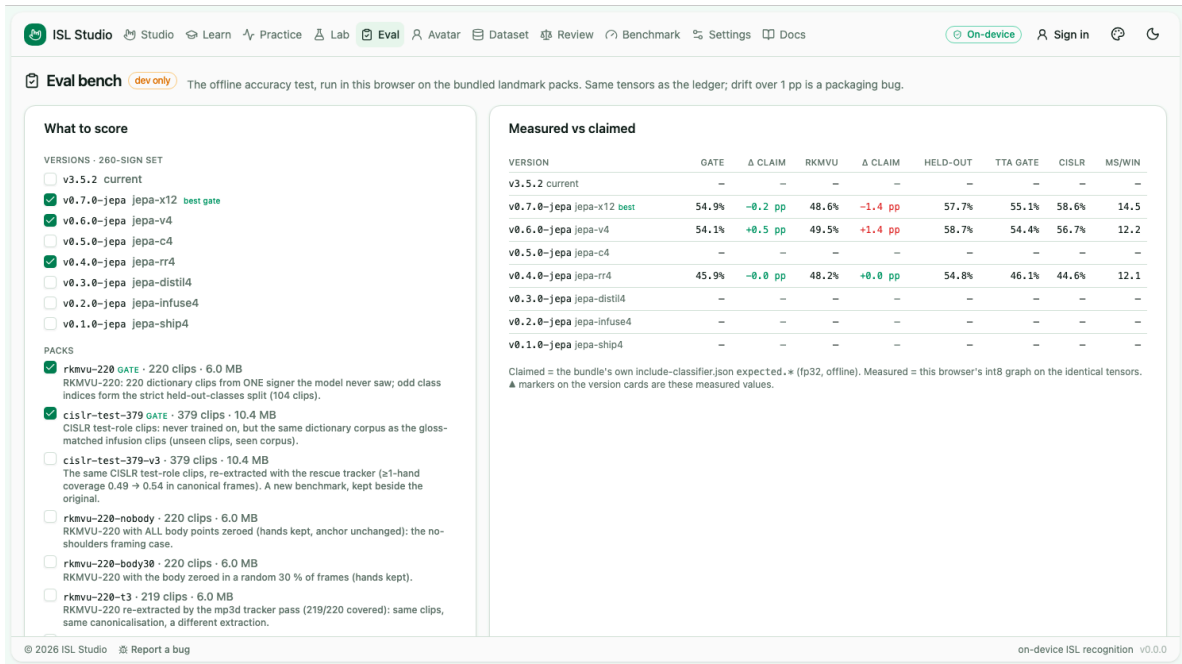


Figure A.6. Eval bench (developer only): a fresh in-browser run of the accuracy test on the int8 graphs of jepa-x12, jepa-v4 and jepa-rr4. “Δ claim” is the browser measurement minus the bundle’s own fp32 claim; drift beyond 1 pp would indicate a packaging bug. The tables of Chapter 5 quote the training ledger.

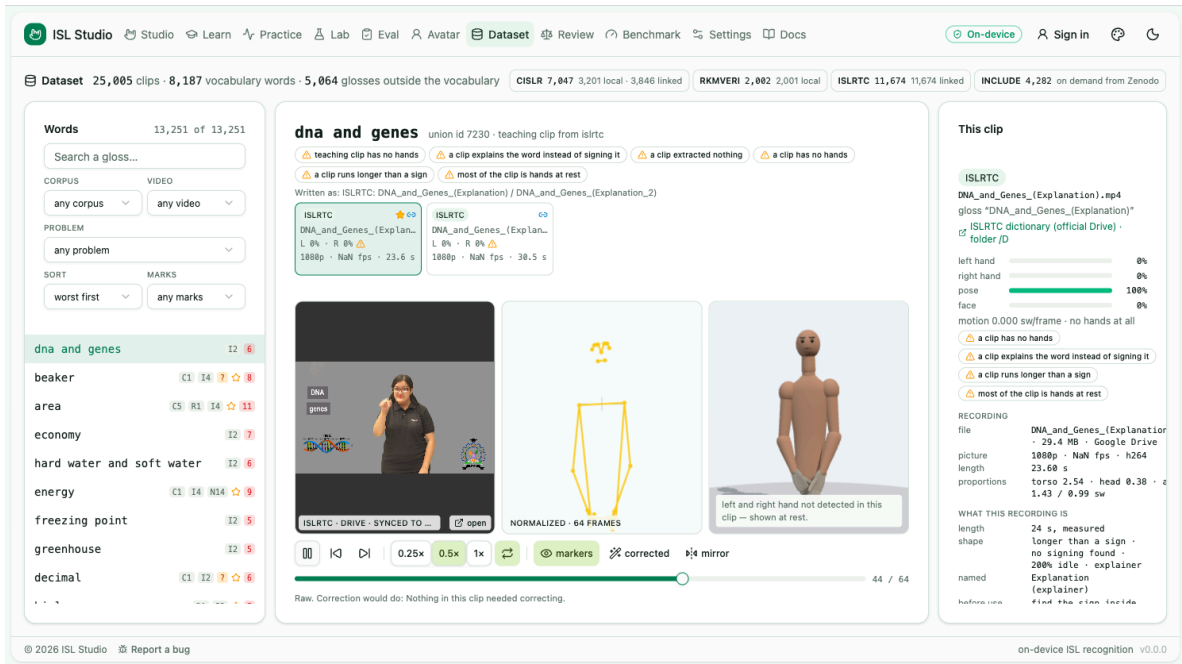


Figure A.7. Dataset tab (developer only): 25,005 clips across four corpora, sorted worst-first, with the clip’s video, its normalised 64-frame landmarks and a figure rendering side by side, and the marks attached to it.

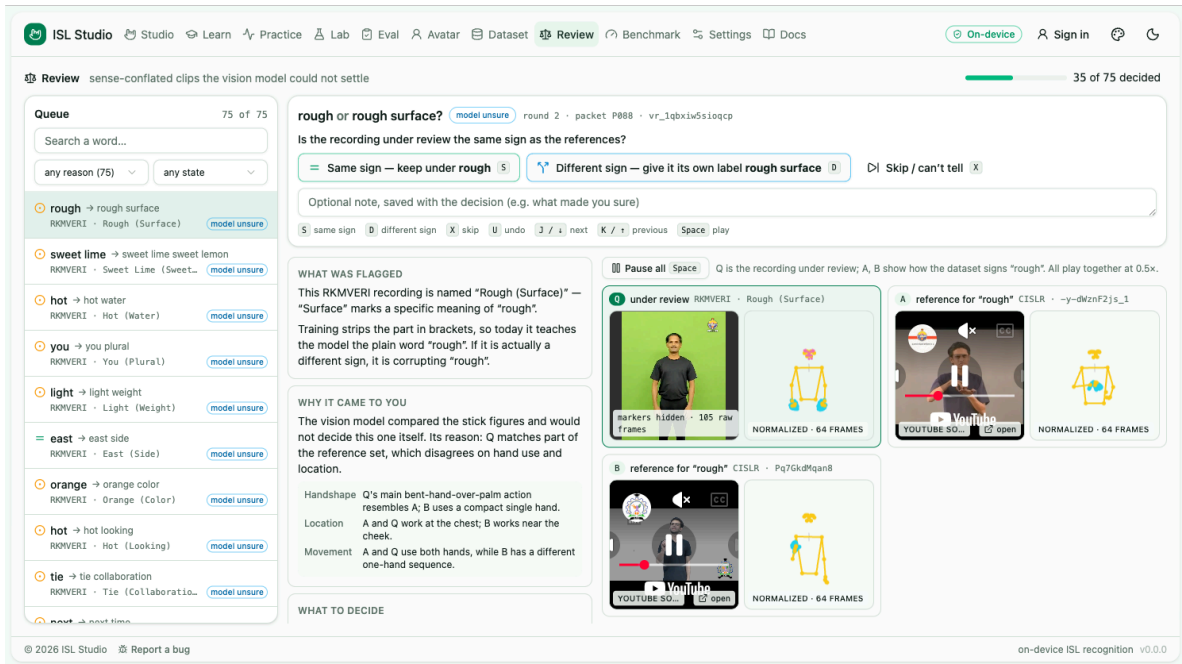


Figure A.8. Review tab (developer only): a sense-conflated clip the vision model could not settle, with the reason it was escalated and the reference clips it is compared against; 35 of 75 decided.

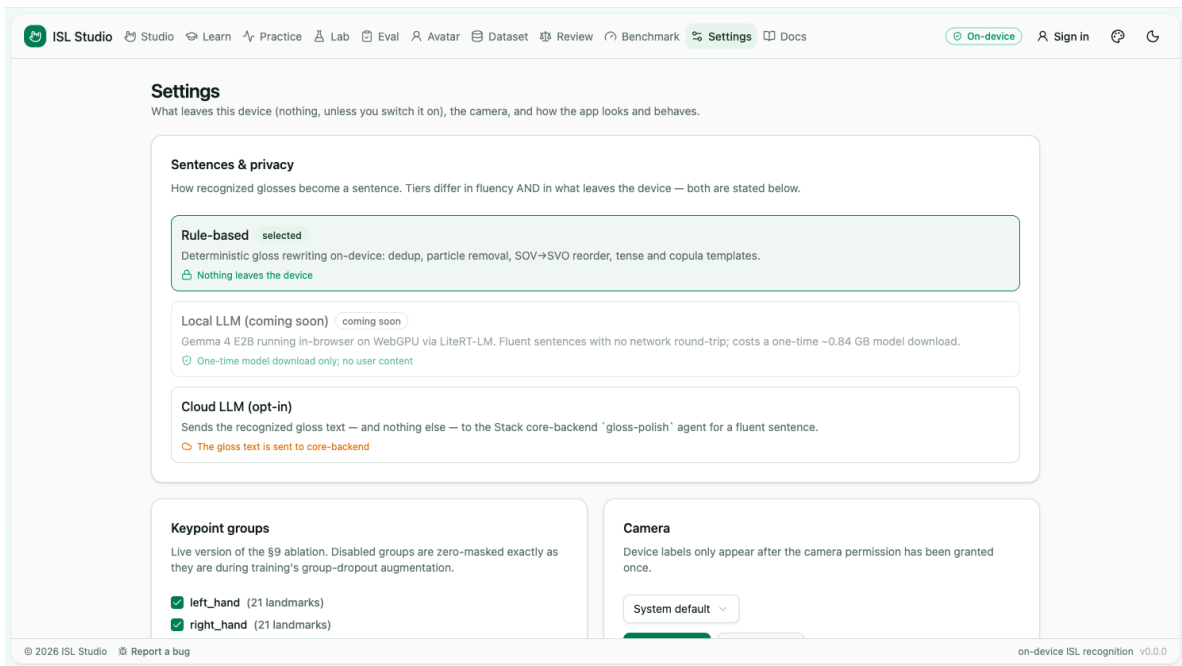


Figure A.9. Settings: what leaves the device (nothing, unless the user opts in), the sentence-assembly tiers, and the keypoint groups.

APPENDIX B

Reproducibility

B.1 Where the work lives

- **Application and documentation.** The studio is `apps/isl-spa` and the landing and documentation site is `apps/isl-landing` in the candidate’s monorepo; this report is `apps/isl-landing/docs-src/review-2-report.typ`. Live: `sanket.kadal.cc` (documentation, with the Review 0 and Review 1 records under `/docs`) and `studio.sanket.kadal.cc` (application).
- **Training and extraction.** The case-study tree `project/: local/` holds the ISL-JEPA trainer (`train_isl_jepa.py`), the fine-tune and distillation scripts (`phaseb_infuse_ft.py`, `phaseb_distil.py`) and export (`phaseb_ship_export.py`); `modal_jobs/` holds the extraction jobs and the gauntlet harness (`gauntlet.py`).

B.2 Commands

To be completed for the final review. The exact command list, commit identifiers and bundle hashes for the frozen model; the total GPU spend for the Review 2 round.

Table B.1. Commands that reproduce the numbers in this report.

Step	Command (abridged)
Fine-tune one seed on Modal	<code>modal run project/modal_jobs/gauntlet.py --script phaseb_infuse_ft.py --env 'INFUSE_ARMS=all+synth:full@2+rr;INFUSE_SEEDS=1'</code>
Calibrate a bundle	<code>uv run --with onnxruntime python tools/fit-dictionary-calibration.py [--write]</code>
Accuracy test in the browser	<code>bun tools/eval-drive.mjs jepa-x12</code> against the running studio (Eval bench)
Compile dataset marks	<code>bun tools/dataset-marks.ts apply</code>
Build this report	<code>typst compile docs-src/review-2-report.typ public/docs/sanket-review-2-report.pdf</code>

B.3 Evaluation discipline

Every accuracy-test number in Chapter 5 is computed on the same 599 tensors; the browser Eval bench replays them through the shipped int8 graph and flags any drift over 1 pp from the training ledger. Calibration uses the cross-fit rule of Algorithm 4.4. Claims rest on at least four seeds. The CISLR test role is never trained on, and a leak check against training rows runs before any number is recorded.

B.4 Compute spend

Review 1 recorded the measured spend for the pilots (about \$2 of GPU) and the corpus build (about \$12 of CPU). The Review 2 gauntlet ran at about 5 minutes per fine-tune seed and 17

minutes per distillation on an A10G, until the Modal workspace was disabled at about 01:15 on 27 September; the remaining distillations run on the local Mac GPU.

APPENDIX C

Glossary

The glossary below is generated from the same source as the documentation site’s glossary, so the two cannot drift apart.

Glossary

Every technical, sign-language, dataset, and institutional term the documents use, each explained in a sentence or two. The same entries power the docs site’s glossary and its search.

Sign language & ISL

Indian Sign Language (ISL) The sign language used by the Deaf community across India — a complete natural language with its own grammar and word order, not a signed encoding of Hindi or English.

Gloss The written label for a single sign, conventionally in capitals (YESTERDAY, SCHOOL). A gloss names the sign; it is not a translation of the sentence the sign appears in.

Word-level (isolated) sign language recognition Recognising one sign at a time from a video clip or a live window — the task this case study addresses. Contrast with continuous recognition.

Continuous sign language recognition Recognising an unsegmented stream of signing, where sign boundaries must be discovered. Harder than word-level recognition and out of this case study’s scope.

Sign language translation (SLT) Producing grammatical spoken-language sentences from signing — a sequence-to-sequence task beyond recognition. This project stops at gloss streams plus rule-based assembly.

Fingerspelling Spelling out words letter by letter using a manual alphabet, typically for names and loanwords. Scoped out of this case study as a dedicated modality.

Signing space The region in front of the signer’s torso and face where signs are articulated. This project uses it operationally: hands below signing space are treated as idle.

Dominant hand The hand a signer leads with; left- and right-dominant signers produce mirrored performances of the same sign, which recognition must treat as identical.

Chirality Handedness of a performance — whether it is produced (or captured) left- or right-handed, including mirror flips introduced by selfie cameras. The matcher is chirality-invariant because handedness is not part of a sign’s identity.

Static hold (held sign) A sign whose content is a sustained handshape and orientation rather than movement — most number signs, for example. Holds are first-class citizens in this project’s matcher.

Non-manual markers Facial expression, mouth shape, head and torso movement that carry grammatical meaning alongside the hands. Tracked as face reference points here, but not yet modelled as grammar.

Institutions & programmes

ISLRTC The Indian Sign Language Research and Training Centre, a Government of India institute under the Ministry of Social Justice and Empowerment. Its official ISL dictionary videos are one of this project’s reference sources.

SRMIST SRM Institute of Science and Technology — the university this M.Tech (Artificial Intelligence) case study is conducted at.

AI4Bharat An IIT Madras research initiative building open AI resources for Indian languages; the group behind the INCLUDE dataset and the OpenHands sign-language toolkit.

SDG 10 — Reduced Inequalities The UN Sustainable Development Goal this case study aligns with: technology access should not depend on hearing, connectivity, or the availability of an interpreter.

Review 0 / 1 / 2 / final review The SRM case-study review cadence: proposal (Review 0), mid-term progress (Reviews 1–2) with the support faculty, and the final review conducted by the main faculty.

Datasets & benchmarks

INCLUDE The standard ISL word-level benchmark (AI4Bharat, ACM Multimedia 2020): 4,287 studio recordings over 263 word classes in 15 categories. This project’s closed-set head is trained and evaluated on it.

CISLR Corpus for Indian Sign Language Recognition (EMNLP 2022): 7,050 videos across roughly 4,700 words, mostly one performance per word — the dataset that frames large-vocabulary ISL as one-shot retrieval, and the source of this project’s open-vocabulary protocol.

WLASL Word-Level American Sign Language dataset (WACV 2020): 2,000 glosses across 21,000+ videos — the scale reference for word-level recognition in a resource-rich sign language.

MS-ASL A Microsoft ASL benchmark (BMVC 2019) of 1,000 classes from over 200 signers in unconstrained recording conditions.

iSign An ISL processing benchmark (ACL Findings 2024) of 118K+ video–sentence pairs with a suite of sentence-level tasks. Cited here to mark the boundary between recognition and translation.

ISLRTC dictionary The official public ISL dictionary programme: thousands of per-word demonstration videos by certified signers. This project processes them into landmark reference clips — the video itself is never redistributed.

RKMVU validation set (the unseen signer) 220 clips of one signer who appears nowhere in any training data — the project’s stranger test. Its top-1 is reported beside every accuracy-test number because it is the only part of the test no training clip resembles.

Machine learning

Pose estimation / landmarks Detecting body and hand keypoints (landmarks) in each video frame, reducing appearance to a skeleton. This project’s entire pipeline operates on an 85-point landmark skeleton, never on pixels.

Embedding A learned fixed-size vector representation (here, 256 dimensions per landmark window) in which distance means dissimilarity — the substrate that makes recognition-by-retrieval possible.

Metric learning Training an embedding so that performances of the same sign land close together and different signs land far apart — recognition then reduces to nearest-neighbour lookup instead of a fixed classifier.

ArcFace / CurricularFace (margin losses) Training losses from face recognition that enforce an angular margin between classes, producing highly discriminative embeddings; CurricularFace adds a curriculum that emphasises hard examples over time.

Prototype A single representative embedding standing for a class — here, the embedding of a word’s dictionary reference. Recognising by distance to prototypes is what lets one example per word suffice.

One-shot learning The regime where a class must be recognised from a single labelled example. Large-vocabulary ISL is effectively one-shot: most words have exactly one dictionary performance.

The 260-sign set The fixed list of signs a model is trained to name outright. Answers are probabilities and they are the most reliable the system produces — but the list cannot grow without retraining. Called closed-set recognition in the literature.

The full vocabulary All 8,188 words the system can reach, by comparing a sign against stored examples rather than naming it from a fixed list. Adding a word needs an example, not a retraining run. Scores here are similarity, not probability. Called open-vocabulary recognition in the literature.

Retrieval Answering “which word was signed?” by ranking stored references by similarity to the query, instead of running a fixed classifier — the formulation that scales to dictionary-sized vocabularies.

Calibration Making a model’s confidence scores mean what they say — a calibrated 0.9 is right about nine times in ten — so that emission thresholds behave predictably.

Quantisation (int8) Storing weights and reference vectors as 8-bit integers instead of 32-bit floats, shrinking artifacts roughly fourfold so an 8,165-word bank ships to a browser.

Fine-tuning Continuing training of an existing model on new data or a new objective. Three fine-tuning tracks in this project produced ≤ 0.3 pp — the measurement behind the data-not-model diagnosis.

Ablation Removing or swapping one component at a time to measure what each part actually contributes — the experimental discipline the semester’s retrain phase commits to.

Top-k accuracy / enrolled top-1 A prediction counts if the right word appears in the model’s k best guesses. “Enrolled” means every one of the 8,165 vocabulary words is a candidate — nothing is excluded to flatter the number.

Frozen protocol / held-out test An evaluation set and procedure fixed in advance and never trained or tuned on. This project treats its 2,285-query CISLR test as sacred for exactly this reason.

Domain shift The gap between the data a system was built on (studio dictionary recordings) and the data it meets in use (webcams, home lighting, untrained signers) — a named risk in the proposal.

JEPA / ISL-JEPA (self-supervised pretraining) Joint-embedding predictive architecture: hide parts of a clip and train the encoder to predict the hidden parts’ latent representation rather than their raw coordinates. ISL-JEPA applies this to landmark tokens over a large unlabelled pool before any labels are used.

EMA teacher (target encoder) A copy of the encoder whose weights trail the trained ones as an exponential moving average; it supplies the prediction targets in JEPA pretraining, which keeps them stable while the student learns.

Knowledge distillation (ensemble → student) Training one small model to match the averaged, temperature-softened answers of several larger or more numerous ones, alongside the true labels — how twelve fine-tunes become one 4.57 MB browser model.

Synthetic signers New training clips made from real ones by lifting the landmarks to 3-D, turning the body and changing limb proportions, then projecting back — two variants per clip. They help only when stacked on real extra clips.

Recover-and-resample (rr) A training augmentation: with probability 0.5, trim up to a quarter of the clip from either end and stretch the rest back to 64 frames, so the model stops depending on where in the window a sign starts.

System & project terms

Accuracy test (“the gate”) The one number that decides whether a model ships: top-1 over 599 held-out clips — 220 from an unseen signer (RKMVU) and 379 CISLR clips never trained on. One clip is 0.17 pp. Written “the gate” in earlier reports.

Mirror pass (mirror test-time augmentation) Scoring every window twice — as captured and flipped left to right — and keeping the more confident answer, so a left-hand-dominant signer is read as well as a right-handed one.

Excalidraw (hand-drawn diagrams) An open-source whiteboard whose drawings look hand-sketched. The Review 2 diagrams are Excalidraw scenes written once as code and shown in the deck, on these pages and in the report.

On-device inference Running the entire recognition pipeline on the user’s own hardware, inside the browser tab — no server round-trip, no video upload, and it works offline. In this project a construction property, not a policy.

WebAssembly (WASM) A portable binary format browsers execute at near-native speed; with SIMD and threads it is what makes running a neural network inside a tab practical.

ONNX Runtime Web Microsoft’s inference engine for the ONNX model format compiled to WebAssembly — the runtime this project’s quantised encoder executes on, at ~14 ms per window.

MediaPipe Google’s on-device perception framework; its holistic pose and hand models supply the per-frame landmarks this project consumes at camera rate.

Canonical space The normalised coordinate frame all landmarks are mapped into — origin at the mid-shoulder point, scale set by shoulder width — so camera distance, framing, and signer height stop mattering.

Recognition window A 64-frame slice of the landmark stream — the unit the encoder scores. Overlapping windows are voted over so a sign is recognised once, not once per window.

Prototype bank The matrix of 256-dimensional embeddings that full-vocabulary retrieval scans on every window: 8,497 rows covering 8,165 words, with one row per variant when a word is signed more than one way.

The two lists The AI model answers from two places: the 260-sign set, which it names outright with a probability, and the full vocabulary, which it matches by similarity. A word is only said when one of them clears its confidence bar. Written tier A / tier B in the code and the earlier reports.

Shape match (no AI) Recognition with no neural network at all: the movement of your hands and arms is compared frame by frame against stored example clips, the way Shazam matches a song. It runs beside the AI model as a second opinion, and as the instrument that separates what the model learned from what the data already gave away.

Fingerprint (trajectory) A compact orientation-aware feature track distilled from a reference clip or live window — wrist paths, forearm and finger directions, handshape ratios — the unit the direct-match aligner compares.

Dynamic time warping (DTW) A classic alignment algorithm that matches two sequences that unfold at different speeds. This project’s aligner descends from it, with sign-specific changes: condensed poses, dwell weighting, coverage penalties.

Dwell weighting Weighting each pose by how long it is held, so a sustained handshape carries its full duration of evidence while one-frame transitions barely count — how holds and movements coexist in one matcher.

Resting detection A cheap check run before anything else: are the hands raised into signing position, and is anything actually moving? If not the window is skipped, which is what stops idle video from producing words. Shown in the app as “resting”.

Audio fingerprinting (Shazam) Identifying a recording by matching a compact signature against a reference library rather than by open-ended learning — the industrial precedent the direct-match pipeline is modelled on.

Gloss stream / sentence assembly The sequence of recognised glosses, and the rule layer that reorders them into a readable sentence. An optional LLM polish exists strictly as user opt-in.

Temporal voting Aggregating predictions across overlapping windows before emitting a word, trading a few hundred milliseconds of latency for large stability gains.

Teach pack / .isl format The project’s 8,187 per-word landmark reference clips in a compact binary format (.isl). It drives the 3D-mannequin dictionary, the practice mode, and the direct-match fingerprints alike.

3D mannequin (Learn / Practice) A rigged 3D avatar that performs any dictionary word by replaying its landmark clip — the dictionary you can watch from any angle, and the reference the practice mode scores you against.

Natural-signing harness A committed regression suite that matches synthetic independently-constructed signers — not replayed references — against the pack, and fails the build if naturally-performed signs stop ranking first.

Research & publishing

Non-archival A document or venue whose publications do not count as prior publication — these review documents are non-archival, so future papers can build on their text without dual-publication conflict.

Self-overlap / dual publication Reusing one’s own published text or results across submissions. Venues restrict it; this project’s rule is to check the target venue’s policy at every submission.

Reproducibility The property that every reported number can be regenerated by running the committed evaluation harnesses — the standard this document set holds itself to, and papers inherit.

Informed consent Written agreement from study participants who understand what is collected and why, with the right to withdraw. The signer study proceeds only under it — and collects no video at all.

Dataset licence audit Re-reading every source dataset’s release terms and documenting what use they permit — scheduled in this project before any paper submission or public artifact.

Venues referenced CVPR, WACV, BMVC (computer vision); ACL, EMNLP, LREC-COLING (language); NeurIPS, AAAI (machine learning); ACM Multimedia, ISMIR (media); ASSETS, W4A, CHI (accessibility and HCI); ICVGIP (Indian vision & graphics). The publication path targets the accessibility and language groups.

Bibliography

- [1] A. Sridhar, R. G. Ganesan, P. Kumar, and M. Khapra, “INCLUDE: A Large Scale Dataset for Indian Sign Language Recognition,” in *ACM Multimedia*, 2020.
- [2] A. Joshi, A. Bhat, and P. S. Basu, “CISLR: Corpus for Indian Sign Language Recognition in a Low-Resource Setting,” in *EMNLP*, 2022.
- [3] United Nations, “Goal 10: Reduce inequality within and among countries.” [Online]. Available: <https://sdgs.un.org/goals/goal10>
- [4] United Nations, “Goal 4: Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all.” [Online]. Available: <https://sdgs.un.org/goals/goal4>
- [5] D. Li, C. Rodriguez-Opazo, X. Yu, and H. Li, “Word-level Deep Sign Language Recognition from Video: A New Large-scale Dataset and Methods Comparison,” in *WACV*, 2020.
- [6] H. R. Vaezi Joze and O. Koller, “MS-ASL: A Large-Scale Data Set and Benchmark for Understanding American Sign Language,” in *BMVC*, 2019.
- [7] A. Joshi, “iSign: A Benchmark for Indian Sign Language Processing,” in *Findings of ACL*, 2024.
- [8] J. Carreira and A. Zisserman, “Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset,” in *CVPR*, 2017.
- [9] S. Yan, Y. Xiong, and D. Lin, “Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition,” in *AAAI*, 2018.
- [10] S. Jiang, “Skeleton Aware Multi-modal Sign Language Recognition,” in *CVPR Workshops*, 2021.
- [11] M. Boháček and M. Hruš, “Sign Pose-based Transformer for Word-level Sign Language Recognition,” in *WACV Workshops*, 2022.
- [12] P. Selvaraj, G. NC, P. Kumar, and M. Khapra, “OpenHands: Making Sign Language Recognition Accessible with Pose-based Pretrained Models across Languages,” in *ACL*, 2022.
- [13] C. Lugaresi, “MediaPipe: A Framework for Building Perception Pipelines,” 2019. [Online]. Available: <https://arxiv.org/abs/1906.08172>
- [14] F. Zhang, “MediaPipe Hands: On-device Real-time Hand Tracking,” in *CVPR Workshop on Computer Vision for AR/VR*, 2020.
- [15] T. Jiang, X. Xie, and Y. Li, “RTMW: Real-Time Multi-Person 2D and 3D Whole-body Pose Estimation,” 2024. [Online]. Available: <https://arxiv.org/abs/2407.08634>

- [16] J. Snell, K. Swersky, and R. Zemel, “Prototypical Networks for Few-shot Learning,” in *NeurIPS*, 2017.
- [17] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “ArcFace: Additive Angular Margin Loss for Deep Face Recognition,” in *CVPR*, 2019.
- [18] Y. Huang, “CurricularFace: Adaptive Curriculum Learning Loss for Deep Face Recognition,” in *CVPR*, 2020.
- [19] H. Sakoe and S. Chiba, “Dynamic Programming Algorithm Optimization for Spoken Word Recognition,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1978.
- [20] A. Wang, “An Industrial-Strength Audio Search Algorithm,” in *ISMIR*, 2003.
- [21] M. Assran *et al.*, “Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture,” in *CVPR*, 2023.
- [22] A. Bardes *et al.*, “Revisiting Feature Prediction for Learning Visual Representations from Video (V-JEPA),” *TMLR*, 2024.
- [23] M. Assran, A. Bardes, D. Fan, Q. Garrido, and N. Ballas, “V-JEPA 2: Self-Supervised Video Models Enable Understanding, Prediction and Planning,” 2025. [Online]. Available: <https://arxiv.org/abs/2506.09985>
- [24] L. Mur-Labadia, M. Muckley, A. Bar, M. Assran, N. Ballas, and A. Bardes, “V-JEPA 2.1: Unlocking Dense Features in Video Self-Supervised Learning,” 2026. [Online]. Available: <https://arxiv.org/abs/2603.14482>
- [25] M. Abdelfattah and A. Alahi, “S-JEPA: A Joint Embedding Predictive Architecture for Skeletal Action Recognition,” in *ECCV*, 2024.
- [26] R. Xu, L. Huang, M. Wang, J. Hu, and W. Deng, “Skeleton2vec: A Self-supervised Learning Framework with Contextualized Target Representations for Skeleton Sequences,” 2024. [Online]. Available: <https://arxiv.org/abs/2401.00921>
- [27] J. Zhang, L. Lin, S. Wang, and J. Liu, “Self-Supervised Skeleton-Based Action Representation Learning: A Benchmark and Beyond,” *IJCV*, 2026.
- [28] H. Hu, W. Zhao, W. Zhou, and H. Li, “SignBERT+: Hand-Model-Aware Self-Supervised Pre-Training for Sign Language Understanding,” *IEEE TPAMI*, 2023.
- [29] S. Gueuwou, X. Du, G. Shakhnarovich, K. Livescu, and A. H. Liu, “SHuBERT: Self-Supervised Sign Language Representation Learning via Multi-Stream Cluster Prediction,” in *ACL*, 2025.
- [30] K. Xie, Z. Cai, and K. Stefanov, “SignMAE: Segmentation-Driven Self-Supervised Learning for Sign Language Recognition,” in *ICPR*, 2026.
- [31] A. Basso Madjoukeng, J. Fink, P. Poitier, E. B. Kenmogne, and B. Frenay, “SSL-SLR: Self-Supervised Representation Learning for Sign Language Recognition,” 2025. [Online]. Available: <https://arxiv.org/abs/2509.05188>

- [32] Z. Li, W. Zhou, W. Zhao, K. Wu, H. Hu, and H. Li, “Uni-Sign: Toward Unified Sign Language Understanding at Scale,” in *ICLR*, 2025.
- [33] G. Tanzer and B. Zhang, “YouTube-SL-25: A Large-Scale, Open-Domain Multilingual Sign Language Parallel Corpus,” in *ICLR*, 2025.
- [34] G. Hinton, O. Vinyals, and J. Dean, “Distilling the Knowledge in a Neural Network,” 2015. [Online]. Available: <https://arxiv.org/abs/1503.02531>
- [35] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” in *CVPR*, 2016.
- [36] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On Calibration of Modern Neural Networks,” in *ICML*, 2017.
- [37] B. Jacob *et al.*, “Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference,” in *CVPR*, 2018.
- [38] MSLR Workshop Organisers, “Multimodal Sign Language Recognition Workshop — signer-independent track.” [Online]. Available: <https://m-slrt.github.io/MSLR2026/>
- [39] R. Zuo and B. Mak, “Improving Continuous Sign Language Recognition with Consistency Constraints and Signer Removal,” *ACM TOMM*, 2024.
- [40] Microsoft, “ONNX Runtime Web — WebAssembly execution provider.” [Online]. Available: <https://onnxruntime.ai/>
- [41] N. C. Camgöz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, “Neural Sign Language Translation,” in *CVPR*, 2018.