Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Suvana: An Integrated Platform for Two-Way Deaf–Hearing Communication in Sri Lankan Sign Language

Lahiru Chamara

lahiruchamara1112@gmail.com

Sri Lanka Institute of Information Technology

Research Article

Keywords: Sri Lankan Sign Language, low-resource sign

Posted Date: October 29th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11538389/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations:
The authors declare no competing interests.

Figures 3,4,5,6 are available in the Supplementary Files section.

Suvana: An Integrated Platform for Two-Way Deaf–Hearing Communication in Sri Lankan Sign Language

***Abstract*—Sri Lankan Sign Language (SSL) is documented as under-served: interpreter provision is limited, no fully standardised national variety exists, and sign-language technology overwhelmingly targets American Sign Language. We present *Suvana*, a platform that brings four SSL components under one deployment: isolated sign recognition with Sinhala speech output, Sinhala speech to a signing avatar, on-device ambient-sound alerting, and an in-browser practice tutor for hearing learners. Our contribution is twofold. First, we describe an integration topology that lets four independently developed research prototypes — three web applications, one mobile application and two Python services — be served as a single product without merging their codebases, including a documented exception for the component whose transport cannot be proxied. Second, we report a calibrated Dynamic Time Warping (DTW) grader for the practice module and give measured evidence that open-set identification on the same distance metric would be unsafe: the closest pair of *distinct* SSL signs in our reference library sits 0.135 apart in normalised hand-size units, while two correct renditions of the *same* sign sit a median 0.458 apart. We therefore grade a specified sign rather than classify an unknown one. We report the practice module's discrimination as ROC AUC 0.744 against a 65.4% majority-class baseline, and state explicitly which of our targets remain unmeasured. We also report the recognition module's accuracy under the signer-dependent protocol it was trained with, and decline to interpret it as generalisation performance. All four authors are hearing; no Deaf signer has yet validated any component, and we set out what that means for the claims we make.**

***Index Terms*—Sri Lankan Sign Language, low-resource sign languages, dynamic time warping, sign language tutoring, accessibility, system integration**

I. Introduction

Udugama *et al.* [1], surveying deaf and hard-of-hearing provision across all nine provinces of Sri Lanka, report that the country "lacks a fully developed common sign language," that interpreter shortages are severe, and that deaf Sri Lankans remain "marginalized, under-supported." A recent review of a decade of SSL technology work [2] identifies persistent obstacles of the same shape: small datasets, regional sign variation, absent linguistic standards, and a lack of expressive synthesis. The two published SSL recognition systems we could locate [3], [4] both address recognition alone.

We do not claim to remove a communication barrier. Desai *et al.* [5], in a Deaf-led review of 101 sign language AI papers, name exactly that framing as the field's most common bias, and we take the point. What we claim is narrower and technical: that four separately built SSL prototypes can be operated as one product, and that one of them — a practice tutor — can be calibrated against measured data and characterised honestly.

Contributions.

1) An integration topology (Sec. III) under which four prototypes with incompatible stacks share one origin, one visual identity and one account, without codebase merging, and with the one transport-level exception documented rather than hidden.
2) A DTW-based grader for SSL practice whose distance-to-score map is *fitted* to measured data rather than assumed, and whose feature weighting was chosen against, not with, the metric that a classification objective would have maximised (Sec. IV-D).
3) Measured evidence that the same distance metric cannot safely support open-set identification, which motivates grading a specified sign instead of recognising an unknown one (Sec. V-B).
4) A staged evaluation statement that separates what is measured from what is instrumented but not yet measured, and a positionality statement (Sec. VI) covering the absence of Deaf involvement.

II. Related Work

A. Sign language recognition for low-resource languages

Landmark-based pipelines — pose estimation followed by a sequence model — now dominate isolated recognition for languages without large corpora. Alyami *et al.* [6] extract MediaPipe hand and face keypoints for Arabic SL and compare LSTM, temporal convolution and transformer heads; De Coster *et al.* [7] show that keypoint normalisation and missing-keypoint imputation materially improve transfer for Flemish SL, and that keypoint models generalise across signers better than image models. Shahgir *et al.* [8] report the same framing for Bangla SL, where a word-level dataset of 611 videos over 40 words was, at the time, the first of its kind. The SSL work we found [3], [4] follows the same landmark-plus-LSTM pattern.

B. Evaluation protocol

The benchmark literature is unambiguous that signers must be disjoint across splits. AUTSL [9] was built so that "the signers in the test set do not appear in the training set," and reports, on identical data and models, 95.95% under a random split against 62.02% under the user-independent benchmark. MS-ASL [10] is designed for the same reason. Alyami *et al.* [6] report 99.74% signer-dependent against 68.2% signer-independent on KArSL-100. FluentSigners-50 [11] publishes three splits and urges reporting on the signer-independent one. Akdag and Baykan [12] report three protocols side by side, and Podder *et al.* [13] demonstrate that leave-one-signer-out cross-validation is viable with as few as four signers. Pal *et al.* [14] argue the general principle that controlling for overlap is required for a realistic estimate. Separately, WLASL [15] shows top-1 accuracy falling from 65.89% at 100 glosses to 32.48% at 2000, which bounds what is plausible at any given vocabulary size. Section V-C states where our own recognition result sits against this literature.

C. Deaf participation

Bragg *et al.* [16] open their interdisciplinary workshop report with the call to "involve Deaf team members throughout," and state that "an all-hearing team lacks the lived experience of Deafness." Desai *et al.* [5] find the field "dominated by hearing non-signing researchers" and warn that claimed Deaf collaboration has itself become tokenising. De Meulder [17] argues the ideologies behind these technologies need interrogation; De Meulder *et al.* [18] distinguish co-creation from "participation as consultation," where deaf users are positioned as testers of pre-built systems. Bragg *et al.* [19] set out dataset ethics, including community claims to ownership of sign language data. The WFD and WASLI [20] caution against signing avatars as a replacement for human signers, permitting narrow use only for pre-recorded static content and only where deaf people have advised on the signed sentences. Yin *et al.* [21] argue for treating signed languages as full natural languages rather than gesture-to-word maps, and Angelini *et al.* [22] report what deaf participants themselves ask of technology. We return to all of this in Sec. VI.

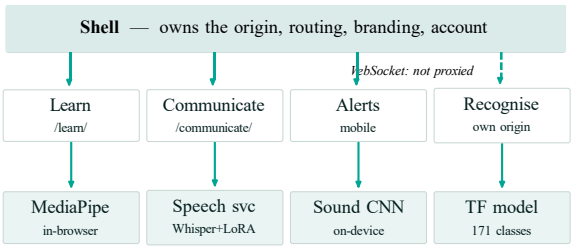


Fig. 1. Suvana's topology. The shell owns the domain root and proxies three modules by path prefix. Recognise (dashed) runs on its own origin because WebSocket upgrades cannot pass through the platform's path rewrites.

III. System Architecture

Suvana is a monorepo of four applications and three supporting services (Fig. 1). The four were built independently, in different stacks, before integration: a static shell, a React single-page application, a Next.js full-stack application, a React Native mobile application, and two Python services. The integration constraint was therefore to present one product without rewriting any of them.

One origin, four prefixes. The shell owns the domain root. Learn builds with a Vite `base` of `/learn/`; Communicate runs with a Next.js `basePath` of `/communicate`, which namespaces its pages, assets and API routes together. Platform-level rewrites forward each prefix to its own deployment. Unsetting the base path variable collapses Communicate to standalone behaviour, so integration is reversible.

The exception, stated. Recognise streams camera frames over a WebSocket. Path rewrites on our deployment platform do not proxy WebSocket upgrades, so Recognise runs on its own origin and serves its own frontend. We report this rather than describing a uniformity the system does not have; it is the kind of detail that determines whether an integration claim is reproducible.

Client-side asset paths. Under a base path, a bare absolute fetch bypasses the prefix. Learn resolves runtime asset paths through the build tool's injected base URL, and Communicate routes API calls through a helper rather than string literals. Both are integration couplings that a naive re-sync from an upstream standalone repository silently removes; we maintain them as an explicit list.

IV. Components

A. Recognise — signing to Sinhala speech

Camera frames are captured in the browser and streamed to a FastAPI service. Per frame, MediaPipe yields a 1629-dimensional landmark vector; sequences of 30 frames are classified over 171 SSL gloss classes. The deployed classifier is a stacked LSTM (256/128/64 units, followed by dense layers of 256 and 128 and a softmax), trained for 50 epochs with a batch size of 32 and a learning rate of 10^{-3}. A confidence threshold of 0.7 and a smoothing window of 5 frames gate the output, which is spoken in Sinhala.

We note for accuracy that although the project's working title refers to transformer models, the deployed sign classifier

is recurrent; the only transformer in the platform is the Whisper encoder-decoder in Sec. IV-B. A transformer variant is scaffolded but untrained, and we make no claim for it.

B. Communicate — Sinhala speech to a signing avatar

Audio is transcribed by `whisper-medium` with a Sinhala-fine-tuned LoRA adapter, merged for inference and decoded greedily under a 96-token cap. Transcripts are tokenised with a Sinhala tokeniser and mapped to SSL glosses through a stored lexicon; a 3D avatar then renders the gloss sequence. A prosody classifier is intended to condition delivery on speaker emotion.

Two statements of scope follow the WFD/WASLI position [20]. First, the avatar's poses were authored by hand by a hearing student and have not been reviewed by a Deaf signer; we present the avatar as an illustration of the pipeline, not as validated SSL, and we make no intelligibility claim. Second, at the time of writing the trained prosody classifier artefact is not deployed, and the service falls back to an off-the-shelf English model; we therefore report no emotion result at all rather than one obtained from a language-mismatched model.

C. Alerts — ambient sound

A convolutional classifier runs entirely on the handset over seven ambient classes: car horn, crying baby, dog, wooden door knock, footsteps, glass breaking and siren. The model ships in two forms, a 9.1 MB ONNX graph and a sharded browser-runtime model. Detection raises a visual and haptic alert naming the sound; an escalation flow shares location. This component is a mobile application rather than a web application because it requires background notification, location and messaging capabilities.

D. Learn — practice with corrective feedback

The learner selects a gloss, views a reference recorded by a fluent signer, performs it to a webcam, and receives a score in $[0, 100]$ with written correction. All processing is in-browser; attempts are stored locally and no video leaves the device.

Features. MediaPipe returns 21 landmarks per hand per frame. Each frame is split into a *handshape* block — landmarks made wrist-relative and divided by hand size — and a *trajectory* block, the wrist path, centred and scaled, with aspect-ratio correction. Both are invariant to where in frame the learner stands and how far from the camera.

Alignment. DTW aligns the attempt against the reference and returns a mean per-frame distance d in hand-size units. Time warping absorbs speed differences, so a slow but correct rendition is not penalised for being slow; pace is judged separately. Critically, DTW also returns an alignment path, so a deviation is attributable to a specific joint at a specific moment — which is what makes "your ring finger is over-extended" expressible. A classifier returns a label and a confidence, from which no such correction follows without inventing it. Every attempt is additionally scored mirrored, and the better result taken, so that a left-dominant learner performing a right-handed reference is not penalised.

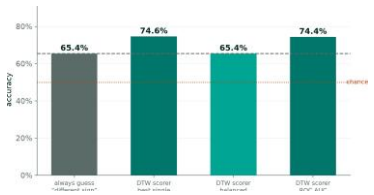


Fig. 2. The distance metric's discrimination stated against its majority-class baseline. Raw accuracy is reported only alongside the baseline, because the pair set is imbalanced 524:992.

Fitted score scale. d maps linearly to $[0, 100]$ between anchors $d_{100} = 0.22$ and $d_0 = 0.77$. These are not chosen: they are the 10th and 90th percentiles of the measured distance between two correct renditions, over 557 takes. The previous anchors were assumed values, and placed d_0 *below* the median correct rendition, so that a genuinely correct attempt scored zero. Calibrating against data is what exposed that.

Weighting, chosen against the metric. Handshape and trajectory are combined at $0.8/0.2$. A grid search over 11 weightings on 24 signs peaks at 78.6% separation with all weight on handshape — that is, with movement discarded entirely. We ship 0.8, at 74.2%. The search optimises "is this the same sign?", a classification objective, whereas the scorer grades a sign the learner was already asked for; movement is a phonological parameter of SSL, and a scorer blind to it would award full marks to a learner with the right handshape and the wrong movement. We took the measured gain to 0.8 and stopped.

V. Evaluation

A. Data

The practice module's reference library holds 501 recordings over 490 distinct glosses, drawn from two public corpora: 141 under CC0 and 360 under CC BY-NC-SA 4.0. Landmark sequences are shipped; no corpus video is redistributed, and a test fails if a reference lacks its licence record. All calibration uses the CC BY-NC-SA corpus: 557 takes over 33 signs, of which 32 are usable, since one sign has a single take and cannot form a positive pair. There is a single signer. The corpus labels itself — two takes of one sign are a correct rendition performed twice, and takes of different signs are a wrong-sign attempt — so no human annotation is involved and none is claimed. This yields 524 positive and 992 negative pairs.

B. Practice module: discrimination

Table I reports discrimination between a correct rendition and a different sign. We lead with ROC AUC because it is threshold-free and insensitive to the class imbalance; accuracy appears only beside its baseline (Fig. 2). At the best single threshold the operating point is TPR = 0.357, FPR = 0.048 — the optimum buys accuracy by rejecting, which is what an imbalanced set rewards, and one more reason the product applies no threshold.

Grading, not identifying. Over 1766 distinct-sign pairs, the two closest sit 0.135 and 0.140 apart, while the median

TABLE I
PRACTICE MODULE DISCRIMINATION, 524 POSITIVE / 992 NEGATIVE PAIRS

Measure	Value
ROC AUC (threshold-free)	0.744
Majority-class baseline	65.4%
Best single-threshold accuracy	74.6% at $d = 0.370$
Gain over baseline	+9.2 points
Balanced accuracy at that threshold	65.4%
Operational score gap (correct − different)	28.7 points
Correct renditions scoring ≥ 50	60.5%
Different signs scoring ≥ 50	20.7%

distance between two correct takes of the same sign is 0.458. Some genuinely different signs are therefore closer together than two correct renditions of one sign typically are. A high score is not evidence that the right sign was made. This is a property of the task, not a deficiency of the implementation, and it is why the shipped system grades a sign the learner was asked for and never identifies an unknown one. No threshold is applied anywhere in the deployed system; the threshold in Table I is an evaluation device for characterising the distance metric.

What the learner sees. On the shipped [0, 100] scale, correct renditions average 56.7 and different signs 28.0, a gap of 28.7 points. This, not the AUC, is the quantity a user experiences.

Cost. The scoring stage — DTW between a completed attempt and the reference — has a median cost of 1.8 ms and a 95th percentile of 12.6 ms over $n = 40$ references, against a 300 ms design budget. This excludes landmark inference, framework commit and browser paint, and was measured in Node rather than a participant's browser; it is not end-to-end feedback latency, which is instrumented in the application but not yet measured over enough real attempts to report.

C. Recognition module: a protocol statement

The recognition classifier records a test accuracy of 0.9878 (loss 0.0532) over 171 classes. We report this as a *signer-dependent* result obtained under a random 0.2 validation split, and we decline to present it as generalisation performance.

Our reason is the literature summarised in Sec. II-B. On identical data and models, AUTSL falls from 95.95% to 62.02% [9] and KArSL-100 from 99.74% to 68.2% [6] when signers are made disjoint. A figure in the high nineties under a random split is what these systems report immediately before that correction. WLASL [15] independently bounds what is plausible as vocabulary grows. We therefore treat a signer-independent re-evaluation — a fixed held-out signer set, or leave-one-signer-out cross-validation following Podder *et al.* [13] — as required work before any generalisation claim, and we report both figures side by side when it is done, as in [12].

D. What is not measured

Of four targets set for the practice module, one is measured. Feedback latency is met at the scoring stage. Feedback agreement with expert judgment is *not* measured: the discrimination result above answers a different question, and the target requires learner attempts graded by an SSL teacher, which do not exist for SSL. Learning gain is instrumented — every attempt carries a session identifier — but has zero participants. System usability has neither participants nor a prepared instrument. The recognition, communication and alerting components have no user-facing evaluation at all.

VI. POSITIONALITY AND LIMITATIONS

All four student authors are hearing, and none is a member of the Sri Lankan Deaf community. We do not have the lived experience or the SSL fluency to judge whether what this system produces or accepts is intelligible or acceptable to Deaf signers. Bragg *et al.* [16] state the consequence directly: an all-hearing team "lacks the lived experience of Deafness, and is removed from the use cases and contexts within which sign language software must function," and even hearing people with strong community ties "are not in a position to speak for Deaf needs." Mack *et al.* [23] find that 90.1% of accessibility papers with user studies included participants with disabilities; this work is in the remaining minority.

We state the consequences without resolving them.

No Deaf validation. No Deaf signer has reviewed any reference in the library, any avatar pose, or any item of feedback text. Our reported numbers measure agreement with corpus labels, not communicative adequacy.

Consultation is not co-design. Our plan is to record references with a teacher at a school for the Deaf. De Meulder *et al.* [18] name this tier accurately as participation as consultation, with users positioned as testers of a pre-built system. We describe it as that, and note that Desai *et al.* [5] warn that claimed Deaf collaboration has itself become tokenising.

Data provenance. Both corpora are public and performed by real signers, but neither was collected for this project and neither carries community consent for this use. Bragg *et al.* [19] record that Deaf communities may claim ownership of sign language datasets. One corpus is non-commercial by licence, which constrains what this project can become.

Which variety? Sri Lanka lacks a fully standardised national SSL [1]. A hearing team encoding one variety as canonical is making a standardisation decision it has no standing to make. We invented no vocabulary, gloss, handshape or sentence order at any point; every sign comes from a published corpus.

Single signer. All fitted constants derive from one fluent signer's correct renditions. The measured spread is narrower than a learner's, so every constant is optimistic. This locates the score scale; it does not substantiate a grading-accuracy claim.

Non-manual markers. The build tracks hands only, so facial expression, head tilt and body movement are not captured. Their weight was reallocated rather than scored as zero, and the deviation is disclosed in the application itself. Treating signed languages as full natural languages [21] requires these; we have not.

VII. Conclusion and Future Work

We have described a platform that operates four SSL components as one product, and a practice tutor whose scoring is calibrated against measured data rather than assumed. The tutor's central design decision — grading a specified sign rather than identifying an unknown one — is not a convenience but follows from a measurement: distinct signs in our library can lie closer together than two correct renditions of one sign. The ordering of future work follows from what is unmeasured. First, signer-independent re-evaluation of the recognition module. Second, a learner study with expert grading, which is the only route to the agreement target and would produce a corpus of SSL learner attempts, which does not currently exist. Third, references recorded with, and validated by, Deaf signers — which bounds everything the scorer can do and is the precondition for treating any output as authoritative. Fourth, restoring non-manual markers, which requires face and pose tracking and a second calibration. Work with the Deaf community should precede, not follow, further capability.

Acknowledgment

The authors thank the School for the Deaf, Ratmalana. The reference corpora are used under their respective licences; no corpus video is redistributed.

References

[1] L. S. K. Udugama, R. Nethsinghe, J. Southcott, S. Kularathna, T. D. T. L. Dhanapala, and K. A. C. Alwis, "Sign language usage of deaf or hard of hearing Sri Lankans," *The Journal of Deaf Studies and Deaf Education*, vol. 29, no. 2, pp. 187–198, 2024.

[2] P. Ahinsa, S. Thrimahavithana, and K. Karunanayaka, "A comprehensive review of Sri Lankan sign language recognition and Sinhala text/speech to sign language translation technologies," in *2025 7th International Conference on Advancements in Computing (ICAC)*. IEEE, 2025, pp. 1–6.

[3] L. Thennakon, D. C. Kahawearachchi, and R. Panchendrarajan, "Real-time recognition and translation of Sinhala sign language," in *2025 5th International Conference on Advanced Research in Computing (ICARC)*. IEEE, 2025, pp. 1–6.

[4] R. J. Herath and P. Ishanka, "An approach to Sri Lankan sign language recognition using deep learning with MediaPipe," in *Digital Technologies and Applications (ICDTA 2022)*, ser. Lecture Notes in Networks and Systems. Springer, 2022, pp. 449–459.

[5] A. Desai, M. De Meulder, J. A. Hochgesang, A. Kocab, and A. X. Lu, "Systemic biases in sign language AI research: A deaf-led call to reevaluate research agendas," in *Proceedings of the LREC-COLING 2024 11th Workshop on the Representation and Processing of Sign Languages*. Torino, Italy: ELRA and ICCL, 2024, pp. 54–65.

[6] S. Alyami, H. Luqman, and M. Hammoudeh, "Isolated Arabic sign language recognition using a transformer-based model and landmark keypoints," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 23, no. 1, pp. 1–19, 2024.

[7] M. De Coster, E. Rushe, R. Holmes, A. Ventresque, and J. Dambre, "Towards the extraction of robust sign embeddings for low resource sign language recognition," 2023.

[8] H. S. Shahgir, K. S. Sayeed, M. T. Tahmid, T. A. Zaman, and M. Z. U. Alam, "Connecting the dots: Leveraging spatio-temporal graph neural networks for accurate Bangla sign language recognition," 2024.

[9] O. M. Sincan and H. Y. Keles, "AUTSL: A large scale multi-modal Turkish sign language dataset and baseline methods," *IEEE Access*, vol. 8, pp. 181 340–181 355, 2020.

[10] H. R. Vaezi Joze and O. Koller, "MS-ASL: A large-scale data set and benchmark for understanding American sign language," in *British Machine Vision Conference (BMVC)*, Cardiff, UK, 2019.

[11] M. Mukushev, A. Ubingazhibov, A. Kydyrbekova, A. Imashev, V. Kimmelman, and A. Sandygulova, "FluentSigners-50: A signer independent benchmark dataset for sign language processing," *PLOS ONE*, vol. 17, p. e0273649, 2022.

[12] A. Akdag and O. K. Baykan, "Enhancing signer-independent recognition of isolated sign language through advanced deep learning techniques and feature fusion," *Electronics*, vol. 13, no. 7, p. 1188, 2024.

[13] K. K. Podder, M. Ezeddin, M. E. H. Chowdhury, M. S. I. Sumon, A. M. Tahir, M. A. Ayari, P. Dutta, A. Khandakar, Z. B. Mahbub, and M. A. Kadir, "Signer-independent Arabic sign language recognition system using deep learning model," *Sensors*, vol. 23, no. 16, p. 7156, 2023.

[14] A. Pal, S. Huber, C. Chaabani, A. Manzotti, and O. Koller, "On the importance of signer overlap for sign language detection," 2023.

[15] 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 *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020, pp. 1448–1458.

[16] D. Bragg, O. Koller, M. Bellard, L. Berke, P. Boudreault, A. Braffort, N. Caselli, M. Huenerfauth, H. Kacorri, T. Verhoef, C. Vogler, and M. Ringel Morris, "Sign language recognition, generation, and translation: An interdisciplinary perspective," in *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, 2019, pp. 16–31.

[17] M. De Meulder, "Is "good enough" good enough? Ethical and responsible development of sign language technologies," in *Proceedings of the 1st International Workshop on Automatic Translation for Signed and Spoken Languages (AT4SSL)*. Association for Machine Translation in the Americas, 2021, pp. 12–22.

[18] M. De Meulder, D. Van Landuyt, and R. Omardeen, "Lessons in co-creation: the inconvenient truths of inclusive sign language technology development," 2024.

[19] D. Bragg, N. Caselli, J. A. Hochgesang, M. Huenerfauth, L. Katz-Hernandez, O. Koller, R. Kushalnagar, C. Vogler, and R. E. Ladner, "The FATE landscape of sign language AI datasets: An interdisciplinary perspective," *ACM Transactions on Accessible Computing*, vol. 14, no. 2, pp. 1–45, 2021.

[20] World Federation of the Deaf and World Association of Sign Language Interpreters, "WFD and WASLI statement on use of signing avatars," https://wasli.org/wp-content/uploads/2023/07/WFD-and-WASLI-Statement-on-Avatar-FINAL-14032018-Updated-14042018-1.pdf, 2018, updated 14 April 2018.

[21] K. Yin, A. Moryossef, J. Hochgesang, Y. Goldberg, and M. Alikhani, "Including signed languages in natural language processing," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, 2021, pp. 7347–7360.

[22] R. Angelini, K. Spiel, and M. De Meulder, "Speculating deaf tech: Reimagining technologies centering deaf people," in *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. ACM, 2025, pp. 1–18.

[23] K. Mack, E. McDonnell, D. Jain, L. L. Wang, J. E. Froehlich, and L. Findlater, "What do we mean by "accessibility research"? A literature survey of accessibility papers in CHI and ASSETS from 1994 to 2019," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, 2021, pp. 1–18.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Figure3.png
- Figure4.jpg
- Figure5.jpg
- Figure6.jpg