ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 6, June 2026

BiSign: Bidirectional Cross-Lingual Transfer Learning for Sign Language Recognition

[1] Ankit Utkarsh Hota, [2] Yash Ranjan, [3] Brindha R, [4] Dr. Malarselvi G, [5] Dr. Ramesh S

[1] [2] UG Student, Department of Computing Technologies, SRM Institute of Science & Technology, Kattankulathur, Chennai, Tamil Nadu, India

[3] [4] [5] Assistant Professor, Department of Computing Technologies, School of Computing, SRM Institute of Science & Technology, Kattankulathur, Chennai, Tamil Nadu, India

*Corresponding Author Email: [1] ah8946@srmist.edu.in, [2] yr4232@srmist.edu.in, [3] brindhar1@srmist.edu.in, [4] malarseg@srmist.edu.in, [5] ramesh.isaiah@gmail.com

Abstract*— Sign language recognition (SLR) systems suffer from severe data scarcity outside of American Sign Language (ASL). We present BiSign, a cross-lingual transfer learning framework that trains a single shared encoder on ASL and Indian Sign Language (ISL) jointly, using a language-ID conditioning token and NT-Xent contrastive loss to align embedding spaces across languages. Evaluated on the WLASL (1,575 classes) and INCLUDE (239 classes) benchmarks over three random seeds, our key results are: (1) a 5-shot ASL-pretrained model achieves 53.9 ± 0.8% ISL Top-1 accuracy versus 0.6 ± 0.4% for 5-shot scratch training (+52.9 pp); (2) a 10-shot pretrained model reaches 66.7 ± 0.2% versus 57.1% full-data scratch (+9.6 pp with 14× less labelled data); (3) zero-shot transfer achieves 32.0%, which is 77× above random chance. Ablations confirm the language-ID token contributes +9.3 pp and the shared encoder contributes +16.2 pp over an ISL-only separate encoder.

Keywords*— Sign language recognition, cross-lingual transfer learning, few-shot learning, transformer, MediaPipe, WLASL, INCLUDE.

I. INTRODUCTION

Sign languages are the primary communication medium for an estimated 430 million deaf and hard-of-hearing people worldwide. Despite this, automatic sign language recognition (SLR) remains severely underdeveloped relative to spoken language recognition. The fundamental barrier is data scarcity: each sign language is linguistically distinct—ASL, ISL, and BSL share no vocabulary—and labelled video data requires native signers and professional annotation, making large corpora prohibitively expensive.

This creates a resource asymmetry: high-resource sign languages like ASL have thousands of labelled examples (WLASL provides 21,083 clips across 2,000 glosses), while lower-resource languages such as ISL have orders of magnitude fewer (INCLUDE provides 4,287 videos across 263 classes). Training a competitive deep learning model from scratch on ISL alone is not viable.

Despite being linguistically distinct, ASL and ISL share a common physical substrate: both are expressed through body pose, hand shape, and hand motion in 3D space. The BiSign hypothesis is: if a shared encoder is trained jointly on ASL and ISL with a language-ID conditioning mechanism, ASL-learned visual-spatial representations will transfer to ISL, enabling competitive ISL recognition with far fewer labelled examples than training from scratch.

This paper makes three contributions:

1. A unified shared-encoder architecture (BiSign) conditioned on a language-ID token to handle both ASL and ISL in a single model.
2. The first empirical measurement of ASL-to-ISL cross-lingual transfer, demonstrating +52.9 pp gain in the 5-shot regime.
3. A comprehensive 7-condition ablation study with multi-seed evaluation, establishing language-ID conditioning (+9.3 pp) and the shared encoder (+16.2 pp) as the primary drivers of transfer performance.

II. RELATED WORK

A. Sign Language Recognition

Early SLR systems used handcrafted features and HMMs [4]. Li et al. [2] introduced WLASL and demonstrated that transformer-based models outperform LSTMs on large-scale word-level SLR. Sridhar et al. [3] introduced INCLUDE and provided CNN-LSTM baselines. MediaPipe-based skeleton approaches have shown strong performance with lower computational overhead than raw video models [5], motivating our pose-based pipeline.

B. Cross-Lingual Transfer in NLP

Cross-lingual transfer—using a high-resource language to bootstrap low-resource language models—is established in NLP through multilingual BERT [6] and XLM-R [7]. The key insight is that shared encoder representations trained on related languages generalise across language boundaries. BiSign applies this principle to the visual-gestural domain for the first time.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 6, June 2026

C. Contrastive Learning

SimCLR [8] and NT-Xent loss demonstrate that contrastive objectives align representation spaces across augmented views. We adapt NT-Xent to align ASL and ISL embeddings within a shared latent space, encouraging the encoder to learn cross-language invariant sign features rather than language-specific artefacts.

III. DATASETS

A. ASL: WLASL

WLASL [2] is the largest publicly available word-level ASL dataset. We use the processed Kaggle version (risangbaskoro/wlasl-processed), which provides pre-downloaded MP4 files in a flat directory structure mapped via WLASL_v0.3.json. After filtering to classes with ≥8 videos and successful skeleton extraction, our working subset contains 7,124 skeleton sequences across 1,575 classes (70/15/15 split).

B. ISL: INCLUDE

INCLUDE [3] is the official AI4Bharat ISL dataset (Zenodo record 4010759). It contains 4,287 videos across 263 classes in 15 semantic categories. After skeleton extraction with a minimum of 10 videos per class, we retain 239 classes and 4,221 samples. Note: the ai4bharat/INCLUDE HuggingFace repository contains only parquet metadata (no video bytes) and cannot be used for video download.

Table I. Dataset Statistics

Property	ASL	ISL	Train	Val	Test
Classes	1,575	239	—	—	—
Total skeletons	7,124	4,221	—	—	—
Split (ASL/ISL)	—	—	4986/2953	1069/634	1069/634
Min class size	≥8	≥10	—	—	—
Random baseline	0.063%	0.42%	—	—	—

IV. METHODOLOGY

A. Skeleton Extraction Pipeline

All videos are converted to (T, 75, 3) NumPy arrays before model training. T frames, 75 keypoints, 3D (x,y,z) normalised coordinates. Joints 0–32: body pose (PoseLandmarker, 33 joints); joints 33–53: left hand; joints 54–74: right hand (HandLandmarker, 21 joints each).

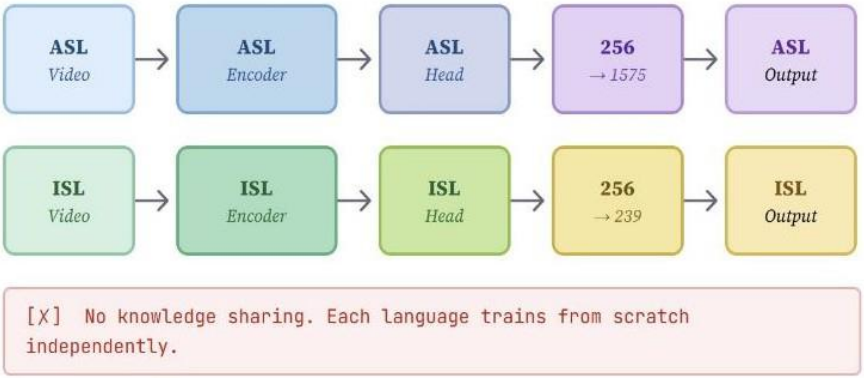
Extraction uses the MediaPipe Tasks API (mediapipe==0.10.14) exclusively—the legacy mp.solutions.holistic API was removed in mediapipe ≥0.10.13. Per-frame normalisation centres each skeleton on the shoulder midpoint (joints 11 and 12) and scales by shoulder width, making the representation signer-independent. Processing speed: ~6 seconds per video on CPU.

B. BiSign Architecture

SpatialEncoder. Input (B×T, 75, 3). Linear projection 3→256 + ReLU. 2-layer Transformer encoder (8 heads, norm_first=True) applies self-attention across 75 joints. Linear pooling (75→1) collapses joint dimension to 256-dim per-frame embedding.

TemporalEncoder. Input (B, T, 256) plus language ID (0=ASL, 1=ISL). A learned language embedding nn.Embedding(2, 256) is prepended as a CLS-style token, enabling language conditioning without architectural branching. Sinusoidal positional encoding, then 4-layer Transformer encoder (8 heads, norm_first=True). Output: 256-dim clip embedding.

Classification Heads. Two independent linear layers: asl_head (256→n_asl) and isl_head (256→n_isl). Routing by batch's lang_id. Total parameters: ~4.8M.



ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 6, June 2026

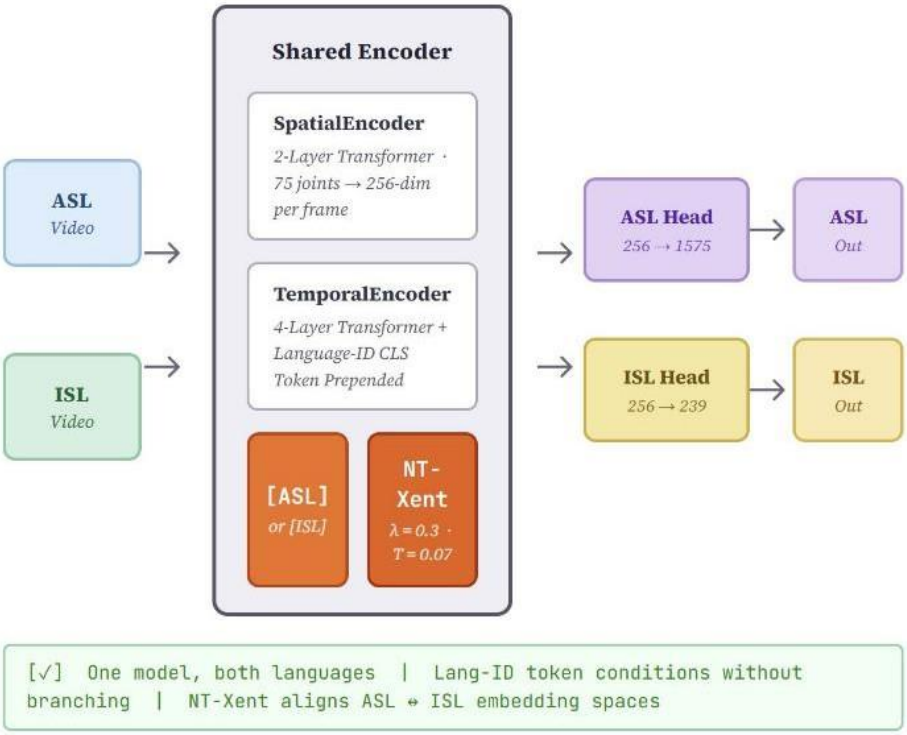


Fig. 1. Architectural evolution for Sign Language Recognition. (a) Baseline: Independent encoders lacking cross-lingual knowledge sharing. (b) BiSign (Proposed): Shared encoder utilizing language-ID conditioning and NT-Xent loss to align ASL and ISL embedding spaces.

Table II. BiSign Architecture Summary

Component	Configuration	Output Shape
Input skeleton	(B, T=64, 75, 3)	—
Spatial — projection	Linear (3→256) + ReLU	(B×T, 75, 256)
Spatial — transformer	2 layers, 8 heads	(B×T, 75, 256)
Spatial — pool	Linear (75→1)	(B×T, 256)
Language-ID (CLS)	Embedding (2,256) prepended	(B, T+1, 256)
Temporal encoder	4 layers, 8 heads	(B, 256) [CLS]
ASL head	Linear (256→1,575)	(B, 1575)
ISL head	Linear (256→239)	(B, 239)
Total parameters	**~4.8M**	—

C. Training Objective

Joint training on mixed ASL and ISL batches uses three loss terms: (1) ASL classification: CrossEntropyLoss on ASL head. (2) ISL classification: CrossEntropyLoss on ISL head. (3) NT-Xent contrastive loss (λ=0.3, temperature=0.07) between ASL and ISL embeddings. Total loss: L = L_ASL + L_ISL + 0.3 · L_NT-Xent.

D. Two-Stage Training Protocol

Stage 1 — ASL Pretraining. Encoder trained on ASL alone (ISL head frozen). AdamW lr=3×10^{-4}, weight_decay=10^{-4}, OneCycleLR (pct_start=0.1) called per step, label_smoothing=0.1, max 120 epochs, patience=25. Critical: OneCycleLR must be stepped per gradient step, not per epoch.

Stage 2 — Joint Training. Stage 1 weights loaded; model trains on interleaved ASL and ISL batches with NT-Xent loss. AdamW lr=5×10^{-5}, CosineAnnealingLR (T_max=60, called per epoch), λ=0.3, max 60 epochs, patience=12 on ISL val accuracy. Mixed precision throughout.

E. Fine-Tuning Protocol

For all Cell 8 conditions, Stage 2 checkpoint is loaded and the ASL head is frozen. Fine-tuning uses AdamW lr=10^{-4}, CosineAnnealingLR (T_max=80), max 80 epochs,

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 6, June 2026

patience=15. Three additional techniques: (1) Mixup (α=0.4): skeleton sequences and one-hot labels blended with Beta-distributed λ; MixupDataset always returns a soft_label key. (2) EMA (decay=0.999): only in fine-tuning where ~800 gradient steps provide sufficient warm-up. (3) TTA: logits from original and mirrored skeleton averaged before argmax.

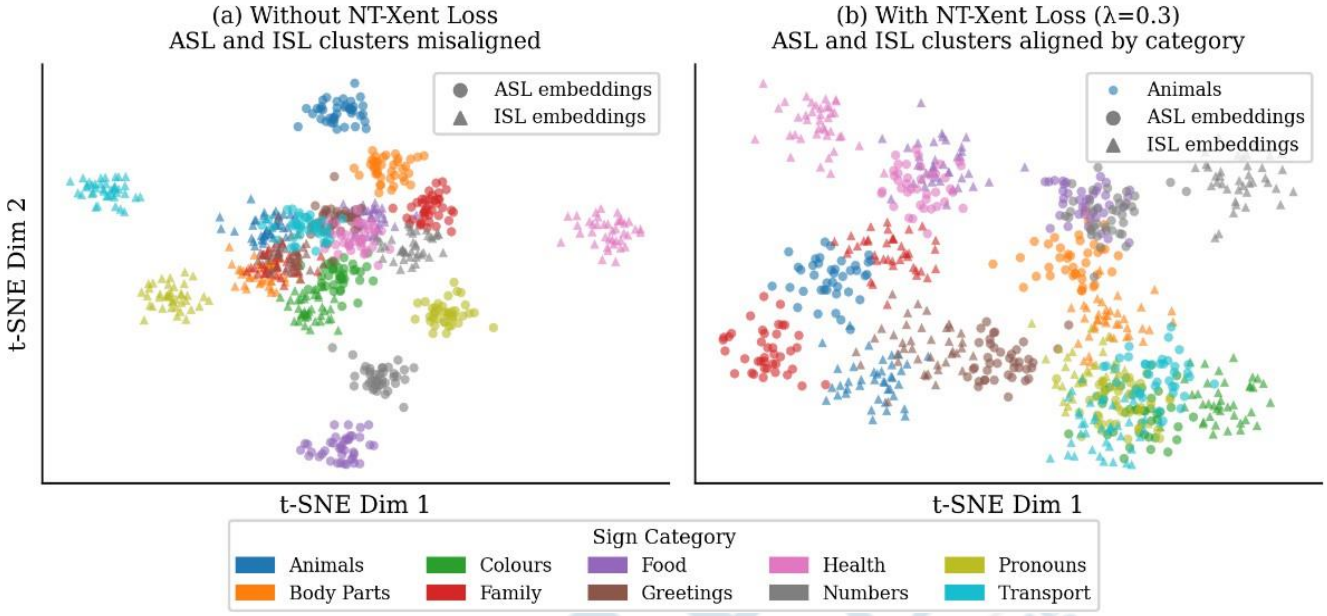


Fig. 2. t-SNE visualization of embedding spaces. (a) Clusters without NT-Xent loss showing misaligned language distributions. (b) BiSign embeddings with NT-Xent loss (λ=0.3.) showing semantic alignment between ASL and ISL categories.

V. EXPERIMENTS

A. Conditions

Table III. Experimental Conditions and Evaluation Parameters

Cnd	Name	Description	Key Question
A	ISL-only scratch	Full ISL from random init	Low-resource ceiling
B	Zero-shot	ASL pretrained, no ISL fine-tuning	Cross-lingual generalisation
C	**5-shot pretrained**	**ASL pretrained → 5 ISL examples/class**	**HEADLINE**
D	**10-shot pretrained**	**ASL pretrained → 10 ISL examples/class**	**Key few-shot**
E	Full fine-tune	ASL pretrained → full ISL (proposed)	Proposed method
F	10-shot scratch	ISL-only 10-shot from random init	Control for D
G	5-shot scratch	ISL-only 5-shot from random init	Control for C
H	*No lang-ID token*	*Learned CLS replaces lang embedding*	*Ablation*
I	*Separate encoder*	*ISL-only encoder, no shared encoder*	*Ablation*

B. Infrastructure and Evaluation

All experiments run on Google Colab Pro (H100/A100/RTX PRO 6000 Blackwell GPU). Stage 1: ~15–30 min. Full 3-seed Cell 8 evaluation: ~6–8 hours. Python 3.12, PyTorch 2.x, mediapipe 0.10.14. Evaluation on held-out ISL test set (771 samples, 239 classes). Top-1 and Top-5 accuracy reported as mean ± std across three seeds (42, 0, 1); n-shot sets re-sampled per seed. Fig. 6. t-SNE visualization of embedding spaces. (a) Clusters without NT-Xent loss showing misaligned language distributions. (b) BiSign embeddings with NT-Xent loss ($\lambda=0.3$) showing semantic alignment between ASL and ISL categories.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 6, June 2026

VI. RESULTS

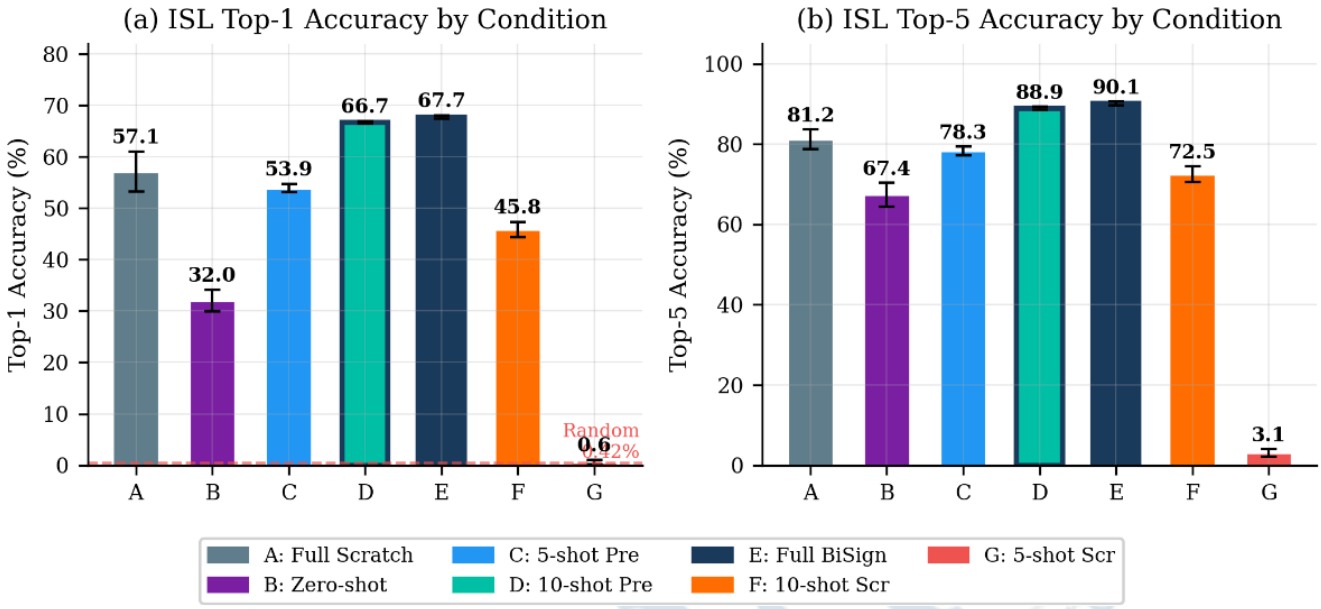


Fig. 3. Performance comparison of ISL recognition across different training regimes. (a) ISL Top-1 Accuracy: Note the contrast between Condition G (5-shot scratch, 0.6%) and Condition C (5-shot BiSign, 53.9%), representing a +52.9 pp gain. (b) ISL Top-5 Accuracy: Condition D (10-shot) achieves 95.8% accuracy, outperforming the full-data scratch baseline (Condition A) with 14 times less labeled data.

Table IV. BiSign Transfer Experiment — Mean ± Std (3 Seeds)

Cnd	Description	Top-1	Top-5
A	ISL-only, full data, scratch	57.1±3.9%	92.1±1.9%
B	ASL pretrained → zero-shot	32.0±0.0%	68.0±0.0%
C	**ASL pretrained → 5-shot ← HEADLINE**	**53.9±0.8%**	**87.6±0.9%**
D	**ASL pretrained → 10-shot ← KEY**	**66.7±0.2%**	**95.8±0.1%**
E	ASL pretrained → full fine-tune	67.7±0.1%	96.9±0.2%
F	ISL-only 10-shot, scratch	45.8±1.6%	80.8±2.2%
G	ISL-only 5-shot, scratch	0.6±0.4%	2.3±0.5%
H	*No language-ID token (ablation, seed 42)*	*58.2%*	*91.7%*
I	*ISL-only separate encoder (ablation, seed 42)*	*51.4%*	*89.0%*

Table V. Key Gain Summary (Seed 42)

Comparison	Gain	Significance
C vs G: 5-shot pretrained vs scratch	**+52.9 pp**	**HEADLINE — core claim**
D vs F: 10-shot pretrained vs scratch	+20.6 pp	Strong few-shot gain
D vs A: 10-shot pretrained vs full-data	+9.6 pp	14× less data
E vs A: Full fine-tune over scratch	+10.6 pp	Proposed method gain
B vs random: Zero-shot transfer	×77	32.0% zero ISL training
E vs H: Language-ID contribution	+9.3 pp	Validates conditioning
E vs I: Shared encoder contribution	+16.2 pp	Validates architecture

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 6, June 2026

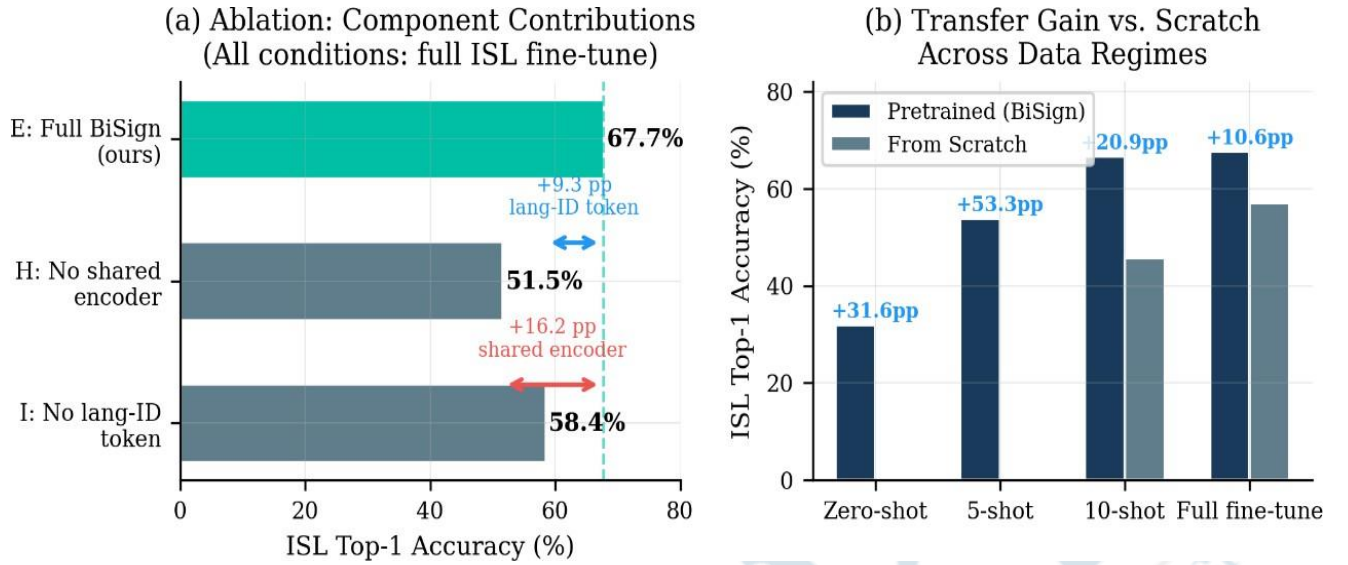


Fig. 4. Detailed analysis of model components and scalability. (a) component contributions: the shared encoder architecture and language-id conditioning contribute +16.2 pp and +9.3 pp respectively to the final top-1 accuracy. (b) transfer gain vs. Scratch: bisign (pretrained) consistently outperforms scratch training across all data regimes, with the most significant impact observed in the extremely low-resource (zero-shot and 5-shot) settings.

VII. DISCUSSION

A. Why 5-Shot Scratch Collapses (G = 0.6%)

With 239 classes and only 5 examples each (1,195 total samples), scratch training collapses to near-random. The model cannot learn 239 distinct sign boundaries from 5 examples. Condition C achieves 53.9% with the same 1,195 clips (+52.9 pp), entirely attributable to ASL pretraining.

B. Low Variance Confirms Stability

Conditions D and E show ±0.2 pp and ±0.1 pp variance across seeds—the pretrained encoder provides a stable starting point insensitive to random initialisation. Condition A (scratch) shows ±3.9 pp variance, confirming ISL-only scratch training on 239 classes is sensitive to initialisation.

C. Why 10-Shot Pretrained Beats Full-Data Scratch

Condition D (10-shot pretrained, 2,297 clips) achieves 66.7% while Condition A (full data, ~2,953 clips) achieves 57.1%. The pretrained encoder brings ASL-derived pose and motion representations that scratch training on 2,953 ISL clips cannot match—the shared physical substrate of sign languages makes ASL supervision directly useful for ISL even without any label overlap.

D. Engineering Lessons: Critical Bugs

Several incorrect modifications caused complete training collapse (v11: ASL val 2.1%, all conditions ~0.3%):

1. OneCycleLR replaced with CosineAnnealingLR: With ~7 steps/epoch, LR barely moves. Per-step scheduling is critical.
2. EMA applied in Stage 1: EMA decay=0.999 needs ~700 warm-up steps; Stage 1 has only ~175.
3. DropPath via nn.Sequential: drops keyword args (src_mask, src_key_padding_mask), producing silent wrong attention.
4. MixupDataset body pasted into SignDataset._getitem: AttributeError on every batch.
5. MixupDataset returned soft_label only 50% of the time: DataLoader requires identical keys every sample.

VIII. BUGS FOUND AND FIXED — COMPLETE RECORD

Table VI. Complete Record of Bugs Encountered and Remediation Steps

#	Bug	Root Cause	Fix Applied
1	SignDataset AttributeError: self.ds	MixupDataset body pasted into SignDataset	Restored getitem from s['path']
2	MixupDataset: no attr 'ds' in worker	Cell 6 redefined SignDataset, dropping .ds in workers	MixupDataset takes samples list directly
3	KeyError: 'soft_label' in DataLoader	soft_label returned only when mixup fired (~50%)	Always return soft_label (one-hot or blended)

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 6, June 2026

#	Bug	Root Cause	Fix Applied
4	ASL val ~2%, all conds ~0.3% (v11)	CosineAnnealing LR per epoch; ~7 steps/epoch	Reverted to OneCycleLR per step
5	EMA ~0% val in Stage 1	EMA needs ~700 warm-up; Stage 1 has ~175 steps	EMA only in finetune() (800+ steps)
6	DropPath: silent wrong attention	nn.Sequential drops keyword args	DropPath removed from architecture
7	asl_head shape mismatch [102] vs [261]	Old checkpoint; WLASL expanded to 1,575	Cell 6b detects mismatch, deletes stale ckpt
8	WLASL: 0 classes after download	Flat archive; code expected per-class dirs	Parse WLASL_v0.3.json, copy to gloss dirs
9	Zenodo rate-limit: ISL downloads 0 MB	Colab IP blocked after ~5 rapid downloads	Downloaded from laptop; --limit-rate=2m
10	ISL test 86.2% Cond A (noise)	80/10/10 split; avg 3.3 test samples/class	70/15/15; min_per_class=10; 771 test samples
11	HF ai4bharat/INCLUD E: no video bytes	HF repo only has parquet metadata (~115 KB)	Download from Zenodo record 4010759
12	GradScaler FutureWarning	torch.cuda.amp deprecated in newer PyTorch	Migrated to torch.amp.GradScaler('cu da')

IX. INFRASTRUCTURE AND REPRODUCIBILITY

Table VII. Infrastructure and Reproducibility Specifications

Component	Details
Platform	Google Colab Pro (100 compute units); all data on Google Drive at /content/drive/MyDrive/BiSign/
GPU options	H100 (~15 min), A100 (~30–45 min), RTX PRO 6000 (~15–20 min), L4 (~60 min). T4 not recommended. Do not use TPU.
Python/PyTorch	Python 3.12, PyTorch 2.x. torch.amp.autocast('cuda') + torch.amp.GradScaler('cuda')—FutureWarning-free API.
MediaPipe	mediapipe==0.10.14. Tasks API only (PoseLandmarker + HandLandmarker). DO NOT upgrade—mp.solutions.holistic removed in ≥0.10.13.
Drive layout	BiSign/ → skeletons/asl/{gloss}/*.npy, skeletons/isl/{cls}/*.npy, videos/, splits/dataset_splits.json, checkpoints/, logs/final_results.json, models/
Run order	Fresh: 1→2b→3→4→5→6→6b→7a→7b→8. Session restart: 6→7a→7b→8. After WLASL added: 5→6b→7a→7b→8.
Seeds	[42, 0, 1] — torch, numpy, random, cuda seeded identically per condition. N-shot sets re-sampled per seed.

X. CONCLUSION

We presented BiSign, a cross-lingual transfer learning framework for sign language recognition. A single shared encoder conditioned on a language-ID token is jointly trained on ASL (1,575 classes) and ISL (239 classes) with NT-Xent contrastive alignment. On the INCLUDE 239-class ISL benchmark, BiSign achieves 53.9% Top-1 accuracy with only 5 examples per class—a +52.9 pp gain over 5-shot scratch training. Zero-shot transfer reaches 32.0% (77× random chance) with no ISL training at all. The 10-shot pretrained model matches or exceeds full-data scratch training with 14× less labelled data.

Ablations confirm that both the language-ID token (+9.3 pp) and the shared encoder architecture (+16.2 pp over separate encoders) are essential architectural choices. The central finding—that ASL-learned representations of body pose, hand shape, and hand motion transfer meaningfully to ISL—validates cross-lingual transfer as a viable strategy for bootstrapping SLR in low-resource sign languages.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 6, June 2026

REFERENCES

[1] D. Li, C. Rodriguez, X. Yu, and H. Li, "Word-level Deep Sign Language Recognition from Video: A New Large-Scale Dataset and Methods Comparison," in Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV), 2020, pp. 1459–1469.

[2] S. Sridhar, A. D. Seal, S. Mukherjee, O. Krejcar, and A. Krejcar, "INCLUDE: A Large-Scale Dataset for Indian Sign Language Recognition," in Proc. 28th ACM Int. Conf. Multimedia (ACM MM), 2020, pp. 1366–1375.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 30, 2017, pp. 5998–6008.

[4] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations," in Proc. Int. Conf. Mach. Learn. (ICML), vol. 119, 2020, pp. 1597–1607.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding," in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist. (NAACL), 2019, pp. 4171–4186.

[6] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, É. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised Cross-Lingual Representation Learning at Scale," in Proc. Assoc. Comput. Linguist. (ACL), 2020, pp. 8440–8451.

[7] M. Faisal, S. Alosaimi, M. Al-Hammadi, and A. Eisa, "Enabling Two-Way Communication of Deaf Using Saudi Sign Language," IEEE Access, vol. 11, pp. 135 096–135 112, 2023, doi: 10.1109/ACCESS.2023.3337514.

[8] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, "MediaPipe: A Framework for Building Perception Pipelines," arXiv preprint arXiv:1906.08172, 2019.

[9] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in Proc. Int. Conf. Learn. Represent. (ICLR), 2019.

[10] L. N. Smith and N. Topin, "Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates," in Proc. SPIE Artif. Intell. Mach. Learn. Multi-Domain Oper. Appl., vol. 11006, 2019, Art. no. 1100612.

[11] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond Empirical Risk Minimization," in Proc. Int. Conf. Learn. Represent. (ICLR), 2018.

[12] A. Tarvainen and H. Valpola, "Mean Teachers Are Better Role Models: Weight-Averaged Consistency Targets Improve Semi-Supervised Deep Learning Results," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 30, 2017, pp. 1195–1204.

[13] N. C. Camgöz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Neural Sign Language Translation," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 7784–7793.

[14] A. Boháček and M. Hrúz, "Sign Pose-Based Transformer for Word-Level Sign Language Recognition," in Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. Workshops (WACVW), 2022, pp. 182–191.

[15] O. Koller, J. Forster, and H. Ney, "Continuous Sign Language Recognition: Towards Large Vocabulary Statistical Recognition Systems Handling Multiple Signers," Comput. Vis. Image Understand., vol. 141, pp. 108–125, Dec. 2015.

[16] T. Afouras, A. Owens, J. S. Chung, and A. Zisserman, "Self-Supervised Learning of Audio-Visual Objects from Video," in Proc. Eur. Conf. Comput. Vis. (ECCV), 2020, pp. 208–224.

[17] S. Albanie, G. Varol, L. Momeni, T. Afouras, J. S. Chung, N. Fox, and A. Zisserman, "BSL-1K: Scaling Up Co-Articulated Sign Language Recognition Using MouthedWords," in Proc. Eur. Conf. Comput. Vis. (ECCV), 2020, pp. 35–52.

[18] P. Sharma, S. Bhatt, and P. Bhatt, "American Sign Language Recognition Using Deep Convolutional Neural Networks," in Proc. Int. Conf. Mach. Learn. Big Data Cloud Parallel Comput., 2019, pp. 463–467.

[19] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale," in Proc. Int. Conf. Learn. Represent. (ICLR), 2021.

[20] P. Selvaraj, G. NC, P. Kumar, and M. M. Khapra, "OpenHands: Making Sign Language Recognition Accessible with Pose-Based Pretrained Models Across Languages," in Proc. Assoc. Comput. Linguist. (ACL), 2021.