# REAL-TIME SIGN LANGUAGE TRANSLATION BY PROTOTYPE-ASSISTED CAUSAL STREAMING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Sign language translation (SLT) for live captioning must produce text while the signer is still signing: each decision may use only frames already observed, and text shown to the user cannot be revised. Decisions are therefore made from short windows in which a sign may be only partly visible and pose detectors lose joints, so window-level recognition must be stabilised before it is committed. We propose a real-time, strictly causal, pose-based SLT pipeline with three components. A confidence-aware masked pose autoencoder uses detector confidence to gate unreliable joints and unreliable reconstruction targets. A part-specific vector-quantised tokenizer, with separate body, hand, and face codebooks trained by video-level contrast, block-to-video alignment, and angular-margin and centre objectives, maps short windows to discriminative sign units. A causal streaming decoder retrieves the nearest sign prototype for each window, aggregates retrievals by past-only voting, and passes the gloss stream through a causal boundary head to a Transformer translator; it reads no future frames and commits each output once. We further formalise streaming SLT through window end-times, zero lookahead, and commit-on-emit, and report time-to-first-token, token and end-to-end latency, and the quality and coverage of committed text. On Phoenix-2014, Phoenix-2014T, CSL-Daily, How2Sign, and OpenASL, the system improves BLEU-4 over the strongest online baseline by 5.55 on Phoenix-2014T and 6.29 on CSL-Daily, keeps a 2.59 BLEU-4 lead at matched time-to-first-token, and has committed 84% of its final output after 80% of the input. Controlled ablations isolate each component, the train-side word segmentation used to build the prototype bank, and the pose detector.

## 1 INTRODUCTION

Sign language translation (SLT) maps continuous sign-language video to spoken-language text (Camgöz et al., 2018; 2020) and supports communication between deaf and hearing people. Live captioning, video calls, and assistive devices place it in a streaming regime: text must appear while the signer is still signing, each decision may use only frames already observed, and text shown to the user cannot be taken back (Yin et al., 2021; Zuo et al., 2024). Every decision is then made from a short window of the past. Within such a window a sign may be only partly visible, a pose detector may lose a hand under occlusion or motion blur (Jiang et al., 2023), and adjacent windows often disagree about which sign is being produced. Streaming SLT is therefore a problem of reaching stable decisions from short, noisy context and committing them early enough to be useful.

This view exposes a limitation of the dominant formulations. Offline gloss-based pipelines (Camgöz et al., 2018; 2020; Chen et al., 2022b) and gloss-free systems (Zhou et al., 2023; Hwang et al., 2025; Wong et al., 2024; Uthus et al., 2023; Rust et al., 2024; Li et al., 2025; Gueuwou et al., 2025) encode the whole video and decode full sentences, so their recognisers are trained on long sequences with sentence-level supervision (Graves et al., 2006). Online systems (Zuo et al., 2024; Yin et al., 2021) restore causality with sliding windows or a wait-$k$ policy, yet their units are still learned from long sequences: a short window receives an unstable prediction, and the translator propagates that instability into the text. Latency is also reported in different terms, without a common definition of when a window ends, how much lookahead is allowed, and whether displayed text may change.

Our key observation is that a streaming decision needs a unit that is recognisable from a short window, together with a way to accumulate evidence for it using only the past. We therefore represent signing



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

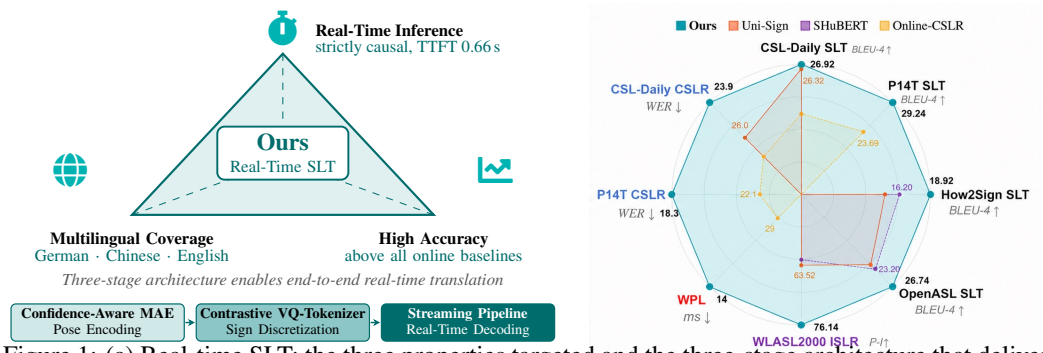


Figure 1: (a) Real-time SLT: the three properties targeted and the three-stage architecture that delivers them. (b) Comparison with online and offline systems on Phoenix-2014T, CSL-Daily, How2Sign, OpenASL, and WLASL2000 (BLEU-4, WER, WPL, P-I; outer ring is better).

as discrete sign units retrieved against a bank of sign prototypes, and stabilise the retrieved units with past evidence before they are committed (Fig. 1a, Fig. 2). A confidence-aware masked pose autoencoder uses detector confidence to gate unreliable joints and unreliable reconstruction targets, so short windows receive stable features. A part-specific vector-quantised tokenizer, with separate body, hand, and face codebooks and discriminative objectives on word- and sentence-level clips of each gloss, maps these features to compact units that stay close to the correct prototype even when the window covers only part of a sign. A causal streaming decoder retrieves the nearest prototype for each window, aggregates retrievals by past-only voting and deduplication, and passes the resulting gloss stream through a causal boundary head to a small Transformer translator. The same architecture and protocol serve German, Chinese, and English benchmarks; only the prototype bank is built per benchmark, from its training split.

We summarize our contributions as follows: (i) a confidence-aware pose representation for sign language, in which confidence masks unreliable joints and reconstruction targets and masked reconstruction recovers missing structure, giving stable short-window features across pose detectors; (ii) a part-specific VQ tokenizer trained with video-level contrast, block-to-video alignment, angular-margin classification, and centre regularisation, which turns short pose windows into discriminative sign units for retrieval; and (iii) a strictly causal streaming pipeline in which prototype retrieval, past-only voting, and a causal boundary head produce text from past frames only and never revise emitted output. We further formalise streaming SLT through window end-times, zero lookahead, and commit-on-emit, and report time-to-first-token, token and end-to-end latency, and the quality and coverage of committed text. On Phoenix-2014, Phoenix-2014T, CSL-Daily, How2Sign, and OpenASL, the system improves over the strongest online baseline (Fig. 1b), including at matched time-to-first-token, and controlled ablations isolate each component, the train-side word segmentation used to build the prototype bank, and the pose detector.

## 2 Related Work

### 2.1 Sign Language Recognition and Translation

Continuous sign language recognition (CSLR) is usually trained with sentence-level labels and the CTC loss (Graves et al., 2006; Zuo & Mak, 2022; Zheng et al., 2023; Min et al., 2021; Chen et al., 2022b; Hu et al., 2023b), while isolated sign recognition treats each sign as a classification problem (Hu et al., 2023a; Zuo et al., 2023; Lee et al., 2023); stage-optimisation methods transfer isolated-sign supervision back into continuous recognisers (Cui et al., 2019; Pu et al., 2019; Zhou et al., 2020). SLT builds on these recognisers with NMT-style encoder–decoder models (Camgöz et al., 2020; Chen et al., 2022a; Zhang et al., 2023; Gan et al., 2023; Chen et al., 2024), and gloss-free systems learn directly from video–text pairs, most recently with large pretrained representations (Zhou et al., 2023; Wong et al., 2024; Rust et al., 2024; Li et al., 2025; Gueuwou et al., 2025). VQ-VAEs (van den Oord et al., 2017) provide discrete motion vocabularies for motion generation (Zhang et al., 2024) and sign language production, where SOKE (Zuo et al., 2025) learns separate codebooks for body and hands. These systems model a sentence from its full video. Our tokenizer instead learns part-specific units for recognition from short causal windows, with discriminative objectives that tie each window to a sign prototype.

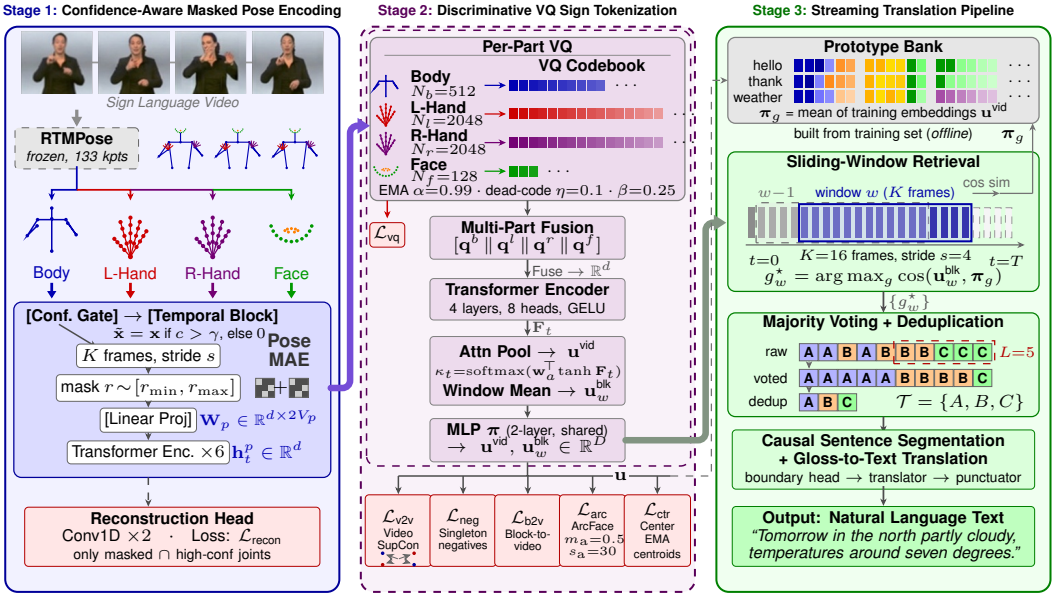


Figure 2: Pipeline overview. *Stage 1*: confidence-gated pose autoencoder; confidence is used only for input gating and reconstruction supervision, never as a feature embedding. *Stage 2*: part-specific VQ-VAE codebooks with EMA updates and dead-code reset, trained jointly with discriminative losses on word- and sentence-level clips that share one gloss label. *Stage 3*: past-only retrieval against a prototype bank built from training-split clips, majority voting, consecutive deduplication, a causal boundary head, and a small Transformer mapping gloss streams to fluent text. Inference parameters (Sec. 3.1): window length $K$=16, stride $s$=4, past-only vote horizon $L$=5. In Stage 3 the dashed window is the previous window $w-1$, and the vote rows show one worked example of the past-only majority followed by deduplication.

### 2.2 Streaming Recognition and Translation

Online CSLR and SLT run a recogniser over sliding windows. Online-CSLR (Zuo et al., 2024) shows that causal windowing can be accurate and efficient, and SimulSLT (Yin et al., 2021) brings wait-$k$ simultaneous translation to sign video with average-lagging evaluation. Sign spotting locates an isolated sign in continuous video, often with the help of dictionaries, mouthings, or attention cues (Momeni et al., 2020; Albanie et al., 2020; Varol et al., 2021; 2022). In speech, streaming recognition is mature, with CNN, Transformer, and hybrid architectures under CTC, sequence-to-sequence, and hybrid objectives (Pratap et al., 2020; He et al., 2019; Miao et al., 2020; An et al., 2022; Fan et al., 2019; Miao et al., 2019). Our pipeline recognises each window by prototype retrieval, stabilises retrievals with past-only voting before any text is emitted, and is evaluated under a protocol that fixes window end-times, zero lookahead, and commit-on-emit and scores the committed text.

## 3 Method

The system is a strictly causal three-stage pipeline summarised in Fig. 2. §3.1 fixes the streaming protocol and notation; §3.2–§3.3 describe offline tokenizer training; §3.4 the causal streaming decoder, with per-video control flow in Alg. 1. Derived equations and the Stage-2 discriminative objective are deferred to App. A.2.

### 3.1 Streaming Protocol and Notation

Frames $\{x_t\}_{t=1}^{T}$ arrive in real time; at decision time $t_0$ the system uses only $x_{1:t_0}$ and its own prior outputs. We use sliding window length $K$=16 and stride $s$=4 (window $w$ committed at the end-time $(w-1)s+K$, never centred). Allowed lookahead is 0 frames; allowed lookback is unbounded since memory is not the bottleneck. Reported latencies cover *time-to-first-token* (TTFT, $e_1-0$), *mean token latency* $\bar{\ell}_{\text{tok}}$ (mean offset between emission and forced-aligned mid-time), *end-to-end p50/p95/p99* latency including pose extraction, and *revision rate*, which is zero by construction under commit-on-emit; the correctness of the committed text is measured separately (Sec. 4.5). Per-frame

variables follow App. A: $\mathbf{x}^p_{t,v}\in\mathbb{R}^2$ denotes joint $v$ in part $p\in\{\text{body, lhand, rhand, face}\}$ at frame $t$, with confidence $c^p_{t,v}\in[0,1]$, encoder output $\mathbf{h}^p_t\in\mathbb{R}^d$, codebook entry $\mathbf{e}^p_k$, quantised vector $\mathbf{q}^p_t$, window embedding $\mathbf{u}^{\text{blk}}_w$, and prototype $\boldsymbol{\pi}_g$ for gloss $g$.

### 3.2 Stage 1: Confidence-Aware Masked Pose Autoencoder

A 2D pose estimator (RTMPose (Jiang et al., 2023), WholeBody-133, 34 FPS; other detectors in Sec. 4.6) returns keypoints with confidence. Joints are split into the four parts $\mathcal{P}$ and per-part normalised. Low-confidence joints are zeroed before encoding (*never* embedded as a feature, which would leak detector calibration into the representation):

$$\tilde{\mathbf{x}}^p_{t,v} = \mathbf{x}^p_{t,v}\,\mathbf{1}[c^p_{t,v}>\gamma], \qquad \gamma{=}0.3. \tag{1}$$

As in masked autoencoders (He et al., 2022), we Bernoulli-sample a mask $m^p_{t,v}\in\{0,1\}$ at rate $r$ ($r\in[0.15,0.20]$ continuous, $[0.30,0.35]$ isolated), zero masked joints, project visible joints to tokens with temporal+part embeddings, and run a 6-layer Transformer encoder shared across parts, followed by a 2-layer 1-D-conv decoder ($d\to d/2\to 2V_p$). The reconstruction loss is computed only on *masked* joints whose original confidence is reliable, so the target itself is not noisy:

$$\mathcal{L}_{\text{recon}} = \tfrac{1}{|\mathcal{P}|}\sum_p \frac{\sum_{t,v} m^p_{t,v}\mathbf{1}[c^p_{t,v}>\gamma]\,\|\hat{\mathbf{y}}^p_{t,v}-\mathbf{x}^p_{t,v}\|_2^2}{\sum_{t,v} m^p_{t,v}\mathbf{1}[c^p_{t,v}>\gamma]+\epsilon}. \tag{2}$$

The decoder is discarded after pretraining; only the encoder is retained.

### 3.3 Stage 2: Part-Specific VQ-VAE with Multi-Level Discriminative Training

Each part has its own codebook $\{\mathbf{e}^p_k\}_{k=1}^{N_p}$ with sizes 512/2048/2048/128 for body/lhand/rhand/face (mirroring hand articulation). Encoded features are quantised by nearest-neighbour assignment, with gradients flowing through the discrete bottleneck via the straight-through estimator (App. A.2):

$$k^*_t = \arg\min_k \|\mathbf{h}^p_t-\mathbf{e}^p_k\|_2^2, \qquad \mathbf{q}^p_t = \mathbf{e}^p_{k^*_t}. \tag{3}$$

Codebooks are updated by EMA at decay $\alpha{=}0.99$ with cluster-size $\hat{n}^p_k$ and feature-sum $\hat{\mathbf{s}}^p_k$ statistics, $\mathbf{e}^p_k\leftarrow\hat{\mathbf{s}}^p_k/(\hat{n}^p_k+\epsilon)$, and refreshed by dead-code reset (random feature replacement when EMA cluster size drops below $\eta{=}0.1$). A commitment term $\mathcal{L}_{\text{vq}}=\beta\sum_{p,t}\|\mathbf{h}^p_t-\text{sg}(\mathbf{q}^p_t)\|_2^2$ with $\beta{=}0.25$ keeps the encoder close to its assigned code. Quantised part tokens are concatenated and fused, $\mathbf{F}_t=\text{Fuse}([\mathbf{q}^{\text{body}}_t\|\mathbf{q}^{\text{lhand}}_t\|\mathbf{q}^{\text{rhand}}_t\|\mathbf{q}^{\text{face}}_t])$, by a 4-layer 8-head Transformer; a video-level embedding uses learned attention pooling and a block-level embedding uses a uniform mean over each window, both projected by a 2-layer MLP $\pi$ to produce $\mathbf{u}^{\text{vid}},\mathbf{u}^{\text{blk}}_w\in\mathbb{R}^D$.

**Multi-granularity supervision.** The training set mixes word-level clips, cut by the train-side segmenter of Sec. 3.5, and sentence-level clips. Both carry the gloss label $g\in\{1,\dots,G\}$: a word-level and a sentence-level instance of the same gloss are positives for each other and share one class, one ArcFace centre, and one prototype. The level indicator $\ell\in\{0,1\}$ records whether a clip is word- or sentence-level. With positive set $\mathcal{N}^+(i)=\{j\neq i\mid g_j=g_i\}$ and temperature $\tau$, the video-level supervised contrastive term over the anchors $\mathcal{A}$ that have at least one positive is

$$\mathcal{L}_{\text{v2v}} = -\frac{1}{|\mathcal{A}|}\sum_{i\in\mathcal{A}}\frac{1}{|\mathcal{N}^+(i)|}\sum_{j\in\mathcal{N}^+(i)}\log\frac{\exp(\mathbf{u}_i^\top\mathbf{u}_j/\tau)}{\sum_{k\neq i}\exp(\mathbf{u}_i^\top\mathbf{u}_k/\tau)}, \tag{4}$$

and, for anchors without a positive ($i\in\mathcal{S}$, the common case in word vocabularies where most glosses appear in a single video), a singleton fallback with negative set $\mathcal{N}^-(i)=\{j\neq i\mid g_j\neq g_i\}$ pushes them away from every other class,

$$\mathcal{L}_{\text{neg}} = \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\log\Big(\tfrac{1}{|\mathcal{N}^-(i)|}\sum_{j\in\mathcal{N}^-(i)}\exp(\mathbf{u}_i^\top\mathbf{u}_j/\tau)\Big). \tag{5}$$

The block-to-video term ties each block $i\in\mathcal{B}$ (batch of $N$ videos) to its own video's embeddings $\mathcal{V}(i)$,

$$\mathcal{L}_{\text{b2v}} = -\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\log\frac{\sum_{j\in\mathcal{V}(i)}\exp(\mathbf{u}_i^{\text{blk}\top}\mathbf{u}_j^{\text{vid}}/\tau)}{\sum_{j=1}^{N}\exp(\mathbf{u}_i^{\text{blk}\top}\mathbf{u}_j^{\text{vid}}/\tau)}, \tag{6}$$

and an angular-margin classifier (ArcFace, margin $m_{\mathrm{a}}$=0.5, scale $s_{\mathrm{a}}$=30) with an EMA centre regulariser supplies $\mathcal{L}_{\text{arc}}$ and $\mathcal{L}_{\text{ctr}}$. The Stage-2 objective is

$$\mathcal{L}_2 = \mathcal{L}_{\text{recon}} + \lambda_{\text{vq}}\mathcal{L}_{\text{vq}} + \lambda_{\text{cl}}(\mathcal{L}_{\text{v2v}} + \mathcal{L}_{\text{neg}} + \lambda_{\text{b2v}}\mathcal{L}_{\text{b2v}}) + \lambda_{\text{arc}}\mathcal{L}_{\text{arc}} + \lambda_{\text{ctr}}\mathcal{L}_{\text{ctr}}, \tag{7}$$

with default weights $(\lambda_{\text{vq}}, \lambda_{\text{cl}}, \lambda_{\text{b2v}}, \lambda_{\text{arc}}, \lambda_{\text{ctr}})=(1.0, 2.0, 1.0, 1.0, 0.01)$. ArcFace and centre-loss formulations are given in App. A.2. We initialise from the Stage-1 encoder and train for 150 epochs with Adam at $10^{-3}$.

### 3.4 Stage 3: Causal Streaming Decoder

For each gloss $g$, the prototype is the mean projection embedding over its training clips, $\boldsymbol{\pi}_g = |\mathcal{D}_g|^{-1}\sum_{i\in\mathcal{D}_g}\mathbf{u}_i^{\text{vid}}$, where $\mathcal{D}_g$ holds the word- and sentence-level clips labelled $g$. The bank is built once from the official training split and held fixed during evaluation. Window $w$ becomes available at $(w-1)s+K$, and the per-window retrieval is

$$g_w^\star = \arg\max_g \frac{\mathbf{u}_w^{\text{blk}} \cdot \boldsymbol{\pi}_g}{\|\mathbf{u}_w^{\text{blk}}\|\,\|\boldsymbol{\pi}_g\|}. \tag{8}$$

Past-only majority voting aggregates over a horizon $L$ ending at $w$ (*not* centred at $w$, in contrast with prior bidirectional designs) so the system is strictly causal:

$$\hat{g}_w = \arg\max_g \sum_{w'=\max(1,w-L+1)}^{w} \mathbf{1}[g_{w'}^\star{=}g]. \tag{9}$$

We use $L$=5 (the red dashed span in Fig. 2), with consecutive deduplication ($\hat{g}_w$ enters $\mathcal{T}$ iff $\hat{g}_w{\neq}\hat{g}_{w-1}$). A causal boundary head, a 2-layer causal Transformer ($d$=192), reads the running gloss stream and predicts a boundary probability $\sigma_n = \text{sigmoid}(\mathbf{w}_b^\top \mathbf{a}_n + b_b)$ (emit when $\sigma_n{>}\theta_b$), trained on transcript boundary labels $y_n^b$ taken from official train-split transcripts only; a 4+4-layer Transformer $p_\theta$ ($d$=384, 8 heads, FFN 1536) maps the bounded gloss segment $\mathcal{T}_{n_1:n_2}$ to target text $z$, causal in $z_{<k}$:

$$\mathcal{L}_{\text{seg}} = -\sum_n \big(\omega_+ y_n^b \log\sigma_n + (1{-}y_n^b)\log(1{-}\sigma_n)\big), \qquad \omega_+{=}4, \tag{10}$$

$$\mathcal{L}_{\text{trans}} = -\sum_k \log p_\theta(z_k \mid z_{<k}, \mathcal{T}_{n_1:n_2}). \tag{11}$$

Training pairs are constructed by running the same retrieve–vote–dedup pipeline with frozen Stage-2 weights on every training video and pairing $\mathcal{T}$ with the official transcript. For How2Sign and OpenASL the decoded text is additionally passed through the off-the-shelf ONNX model 'pcs_en' from the open-source `punctuators` package (Carroll, 2023), which restores casing and inserts terminal punctuation only and never inserts content tokens. The per-video streaming control flow is summarised in Alg. 1; full equations for the discriminative Stage-2 objective and the gloss-to-text decoder are in App. A.2.

### 3.5 Train-Side Word-Level Segmentation

The prototype bank requires word-level training clips, and we use a different segmenter per regime. On Phoenix-2014/2014T and CSL-Daily, where gloss annotations are available, we run TwoStream-SLT (Chen et al., 2022b) CSLR on the training split and read off (start, end, gloss) triples from its predicted gloss timestamps. On How2Sign and OpenASL, where no gloss exists, we follow the dictionary-retrieval procedure of SOKE (Zuo et al., 2025) (Sec. 3.1): each training-transcript content word is looked up in the public SOKE word-level sign dictionary (Sec. 3.4 of SOKE) and DTW-aligned against the training video. Neither segmenter is applied to the dev or test split. Train-side word segmentation is distinct from the causal boundary head of Sec. 3.4: the former cuts training clips offline to build the bank, the latter is a learned component that groups the retrieved gloss stream at inference using only past outputs. App. B gives implementation details, a 500-clip boundary-quality study, and the full training/inference information flow (Tab. 12); Sec. 4.4 measures how results depend on segmentation quality.

**Algorithm 1** Causal streaming inference (one video).

**Require:** Frame stream $\{x_t\}$, prototype bank $\{\boldsymbol{\pi}_g\}$, frozen Stage 1+2 encoder, boundary head $\sigma$, decoder $p_\theta$
**Ensure:** Emitted text segments $\mathbf{z}$

$w \leftarrow 0$; $\mathcal{T} \leftarrow [\,]$; pending $\leftarrow [\,]$
**for** $t = 1, 2, \ldots$ **do**
  **if** $t \bmod s = 0$ **and** $t \geq K$ **then**
    $w \leftarrow w + 1$
    Encode window $\mathbf{u}_w^{\text{blk}}$ from $x_{(w-1)s+1\,:\,(w-1)s+K}$ ▷ Stage 1+2
    $g_w^\star \leftarrow \arg\max_g \cos(\mathbf{u}_w^{\text{blk}}, \boldsymbol{\pi}_g)$ ▷ Eq. (8)
    $\hat{g}_w \leftarrow$ majority over past $L$ retrievals ▷ Eq. (9)
    **if** $\hat{g}_w \neq \hat{g}_{w-1}$ **then** ▷ dedup
      append $\hat{g}_w$ to $\mathcal{T}$; pending $\leftarrow$ pending $\cup \{\hat{g}_w\}$
      **if** $\sigma(\mathcal{T}_{1:|\mathcal{T}|}) > \theta_b$ **then** ▷ causal boundary head
        emit $\mathbf{z} \leftarrow \arg\max p_\theta(\cdot \mid \text{pending})$; pending $\leftarrow [\,]$
      **end if**
    **end if**
  **end if**
**end for**

# 4 Experiments

## 4.1 Setup, Datasets, and Reporting Protocol

**Datasets.** We evaluate on Phoenix-2014 (Koller et al., 2015), Phoenix-2014T (Camgöz et al., 2018), CSL-Daily (Zhou et al., 2021), and How2Sign (Duarte et al., 2021). OpenASL (Shi et al., 2022) follows the How2Sign setup (Tab. 3); an isolated-sign-recognition (ISLR) variant on WLASL2000 (Li et al., 2020) is reported in App. H.

**Train-side word segmentation.** Each benchmark's prototype bank is built from its own training split, with word clips from TwoStream-SLT (Chen et al., 2022b) on Phoenix-2014/2014T and CSL-Daily and from SOKE retrieval (Zuo et al., 2025) on How2Sign and OpenASL (Sec. 3.5, App. B).

**Metrics.** CSLR: WER. SLT: BLEU-1/2/3/4 and ROUGE-L; for English additionally BLEURT (Sellam et al., 2020). Streaming: TTFT, $\bar{\ell}_{\text{tok}}$, end-to-end p50/p95/p99, prefix-BLEU, revision rate, window-processing latency (WPL, pose extraction latency excluded). $\bar{\ell}_{\text{tok}}$ averages $e_i - \mu_i$, where $e_i$ is the commit time of token $i$ and $\mu_i$ the midpoint of its span from a forced aligner constrained to the official gloss sequence; jittering $\mu_i$ by up to $\sigma_{\text{j}}$=12 frames changes $\bar{\ell}_{\text{tok}}$ by at most 0.08 s (App. C.2).

**Implementation.** RTMPose (Jiang et al., 2023) WholeBody-133 (34 FPS); Pose-MAE $d$=256 with two stride-2 1D-conv layers; $\gamma$=0.3. VQ codebooks 512/2048/2048/128 (body/lhand/rhand/face), EMA decay 0.99, commitment $\beta$=0.25, dead-code reset $\eta$=0.1. Inference parameters match Fig. 2: window $K$=16, stride $s$=4, past-only vote $L$=5. Translator: 4+4 layers, $d$=384, 8 heads, FFN 1536, Adam $5 \times 10^{-4}$, 80 epochs. All reported numbers are means over 5 seeds ({10, 20, 30, 40, 50}).

## 4.2 CSLR Results

Tab. 1 compares CSLR WER on Phoenix-2014, Phoenix-2014T, and CSL-Daily under matched window size 16. Online-CSLR (Zuo et al., 2024) is the strongest online baseline under this protocol.

Table 1: CSLR results. WER (lower is better) on Phoenix-2014, Phoenix-2014T, and CSL-Daily under matched window size 16. "–" = not reported in the original paper.

| Method | Mode | Window | Phoenix-2014 | | Phoenix-2014T | | CSL-Daily |
|---|---|---|---|---|---|---|---|
| | | | Dev | Test | Dev | Test | Dev/Test |
| TwoStream-SLR (Chen et al., 2022b) | Online | 16 | 30.3 | 31.6 | 38.4 | 39.3 | 101.4/103.3 |
| Online-CSLR (Zuo et al., 2024) | Online | 16 | 22.6 | 22.1 | 22.2 | 22.1 | 30.2/29.3 |
| TwoStream-SLR (Chen et al., 2022b) | Offline | full | — | — | 17.7 | 19.3 | 25.4/25.3 |
| Uni-Sign (Li et al., 2025) | Offline | full | — | — | — | — | 26.7/26.0 |
| **Ours** | Online | 16 | **17.2** | **17.4** | **18.1** | **18.3** | **24.7/23.9** |

Under the same 16-frame window and a strict-causal vote, WER improves by **3.8** points on Phoenix-2014T test over Online-CSLR; offline rows use full-video context as reference.

### 4.3 SLT RESULTS

We report SLT on Phoenix-2014T (German), CSL-Daily (Chinese), and How2Sign (English). OpenASL follows the same setup; both English benchmarks are in Tab. 3.

Table 2: SLT results on Phoenix-2014T and CSL-Daily. "Vis.Ft." = visual fine-tuning on sign data. Best online results in bold.

| Setting | Method | Mode | Vis.Ft. | Phoenix-2014T | | | | | CSL-Daily | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | B1 | B2 | B3 | B4 | RG | B1 | B2 | B3 | B4 | RG |
| Gloss-based offline | MMTLB (Chen et al., 2022a) | Off | ✗ | 53.97 | 41.75 | 33.84 | 28.39 | 53.48 | 53.31 | 40.41 | 30.87 | 23.92 | 53.25 |
| | TS-SLT (Chen et al., 2022b) | Off | ✓ | 54.90 | 42.43 | 34.46 | 28.95 | 52.11 | 55.42 | 42.59 | 32.87 | 25.79 | 55.72 |
| Gloss-free offline | Uni-Sign (Li et al., 2025) | Off | ✓ | — | — | — | — | — | 55.08 | 42.14 | 32.98 | 26.36 | 56.51 |
| Online prior | SimulSLT (Yin et al., 2021) | On | ✓ | — | — | — | 23.14 | — | — | — | — | 13.88 | — |
| | Online-CSLR (Zuo et al., 2024) | On | ✗ | 52.49 | 38.43 | 30.26 | 23.69 | 50.63 | 51.87 | 36.22 | 27.04 | 20.63 | 49.76 |
| **Ours** | — | On | ✗ | **63.15** | **51.46** | **37.78** | **29.24** | **62.43** | **57.64** | **45.86** | **34.32** | **26.92** | **60.29** |

Under the same 16-frame window, BLEU-4 improves over Online-CSLR by **5.55** on Phoenix-2014T test and **6.29** on CSL-Daily test.

Table 3: English SLT on How2Sign and OpenASL. Offline rows are reference. Train-side word segmentation by SOKE retrieval (Zuo et al., 2025); its effect is analysed in Sec. 4.4.

| Method | Mode | B1 | B4 | RG | BLRT |
|---|---|---|---|---|---|
| *How2Sign* | | | | | |
| GloFE-VN (Lin et al., 2023) | Off | 14.9 | 2.2 | 12.6 | 31.7 |
| YT-ASL (Uthus et al., 2023) | Off | 37.8 | 12.4 | — | 46.6 |
| SSVP-SLT (Rust et al., 2024) | Off | 43.20 | 15.50 | 38.40 | 49.60 |
| SHuBERT (Gueuwou et al., 2025) | Off | — | 16.2 | — | 49.9 |
| Uni-Sign (Li et al., 2025) | Off | 40.2 | 14.9 | 36.0 | 49.4 |
| **Ours** | On | **48.54** | **18.92** | **43.87** | **51.85** |
| *OpenASL* | | | | | |
| GloFE-VN (Lin et al., 2023) | Off | 21.56 | 7.06 | 21.75 | 36.35 |
| SHuBERT (Gueuwou et al., 2025) | Off | — | 23.20 | — | 60.60 |
| Uni-Sign (Li et al., 2025) | Off | 49.35 | 23.14 | 43.22 | 60.40 |
| **Ours** | On | **56.86** | **26.74** | **47.34** | **61.34** |

Table 4: Attribution on Phoenix-2014T (B-4 / WER) and How2Sign (B-4), one component toggled per row.

| Configuration | P14T B-4 ↑ | P14T WER ↓ | H2S B-4 ↑ |
|---|---|---|---|
| Full pipeline | **29.24** | **18.3** | **18.92** |
| linear cls. instead of bank | 25.30 | 22.7 | 15.40 |
| random-label bank | 8.20 | 47.6 | 6.10 |
| oracle gloss in (translator only) | 36.85 | 8.2 | 24.62 |
| $L=1$ (no vote) | 26.15 | 21.4 | 16.20 |

Tab. 4 changes one component at a time. Replacing the prototype bank with a linear classifier lowers BLEU-4 from 29.24 to 25.30 on Phoenix-2014T and from 18.92 to 15.40 on How2Sign, and a bank with shuffled labels falls to 8.20, so the gain comes from the labelled prototypes. Feeding the translator the official gloss sequence raises Phoenix-2014T BLEU-4 to 36.85; the remaining 7.6 points are therefore recognition error (App. D, Tab. 20). The prototypes are averaged from word clips cut by a train-side segmenter; Sec. 4.4 measures how prototype quality depends on that segmentation. App. E discusses why a pose-only system is competitive with offline RGB systems on Phoenix-2014T; qualitative outputs against SHuBERT on How2Sign are in App. G.

### 4.4 SENSITIVITY TO TRAIN-SIDE WORD SEGMENTATION

The prototype bank and the Stage-2 word clips come from an external segmenter applied to the training split only (Sec. 3.5). A segmenter can err in two ways. Within a video, a word's boundaries shift or the word is missed. Across videos, the same word is cut with different extents, and a smaller lookup dictionary leaves fewer clips to average into each prototype. Tab. 5 perturbs each error type, rebuilds the bank, and finally gives our segmenter to the strongest online baseline.

Table 5: Train-side word segmentation. *Left*: Phoenix-2014T test, boundary jitter (zero-mean Gaussian, $\sigma_{\mathrm{j}}$ in frames) and clip removal applied before the bank is rebuilt, and Online-CSLR (Zuo et al., 2024) trained with our segmenter. *Right*: How2Sign test, SOKE dictionary coverage, whole-clip prototypes, and oracle word boundaries (App. B).

| Phoenix-2014T condition | B-4 ↑ | WER ↓ |
|---|---|---|
| Ours, TwoStream segments (default) | **29.24** | **18.3** |
| jitter $\sigma_{\mathrm{j}}=2$ | 29.08 | 18.5 |
| jitter $\sigma_{\mathrm{j}}=4$ | 28.54 | 19.8 |
| jitter $\sigma_{\mathrm{j}}=8$ | 27.23 | 21.6 |
| drop 10% of clips | 29.12 | 18.5 |
| drop 20% of clips | 28.37 | 20.1 |
| Online-CSLR, original segmenter | 23.69 | 22.1 |
| Online-CSLR, our segmenter | 22.16 | 23.8 |

| How2Sign condition | B-4 ↑ |
|---|---|
| No word segmentation (whole clips) | 14.80 |
| 25% dictionary coverage | 16.12 |
| 50% dictionary coverage | 17.34 |
| SOKE retrieval, full (default) | 18.92 |
| Oracle word boundaries | 19.62 |

On Phoenix-2014T, boundary jitter of $\sigma_j$=4 frames lowers BLEU-4 by 0.70; for reference, the mean boundary error of the SOKE segmenter on How2Sign/OpenASL is 3.4 frames (App. B). Jitter of $\sigma_j$=8 frames and removal of 20% of the clips give 27.23 and 28.37, above every online baseline in Tab. 2. On How2Sign, BLEU-4 grows steadily with segmentation coverage and precision: 14.80 with whole-clip prototypes, 16.12 and 17.34 with 25% and 50% of the dictionary, 18.92 with the full dictionary, and 19.62 with oracle word boundaries. Approximate word regions suffice, and prototype quality degrades gradually as segmentation becomes noisier or sparser. Trained with our segmenter, Online-CSLR reaches 22.16 BLEU-4 and 23.8 WER, close to its original 23.69 and 22.1 and 7.08 BLEU-4 below our system with the same segmenter. With segmentation held fixed, the remaining gap lies in the other components, which Tab. 4 and App. D isolate one at a time.

### 4.5 Streaming Evaluation

All numbers report the strictly causal pipeline (past-only voting, commit-on-emit). Tab. 6 compares us against online baselines at the same 16-frame window and against the offline TwoStream-SLR upper bound on Phoenix-2014T.

Table 6: Streaming evaluation on Phoenix-2014T test. WPL, FPS: window-processing latency and throughput without pose extraction, as in Online-CSLR (Zuo et al., 2024); E2E p95 includes pose. "Reprod. pose front-end": baseline on our RTMPose features. Bold: best online value.

| Method | Win. | WER ↓ | B-4 ↑ | TTFT ↓ | $\bar{\ell}_{\text{tok}}$ ↓ | WPL ↓ | E2E p95 ↓ | FPS ↑ | Mem. ↓ |
|---|---|---|---|---|---|---|---|---|---|
| TwoStream-SLR (entire video) | full | 19.3 | 28.95 | 4.30 s | 4.30 s | 261 ms | 5.20 s | 3.8 | 15.0 GB |
| TwoStream-SLR (sliding 16) | 16 | 39.3 | — | **0.32** s | 0.65 s | 31 ms | 0.51 s | 32 | 5.1 GB |
| Online-CSLR (Zuo et al., 2024) | 16 | 22.1 | 23.69 | **0.32** s | **0.61** s | 29 ms | 0.49 s | 34 | 5.1 GB |
| Online-CSLR (reprod. pose front-end) | 16 | 22.5 | 22.85 | 0.45 s | 0.72 s | 24 ms | **0.40** s | 42 | 4.2 GB |
| SimulSLT (Yin et al., 2021) (reprod. pose front-end) | 16 | — | 22.42 | 0.55 s | 0.85 s | 28 ms | 0.46 s | 36 | 4.6 GB |
| **Ours**, strict causal ($K$=16, $L$=5) | 16 | **18.3** | **29.24** | 0.66 s | 0.78 s | **14** ms | 0.41 s | **83** | **1.4** GB |

At the default operating point our TTFT is 0.66 s and $\bar{\ell}_{\text{tok}}$ is 0.78 s, against 0.32 s and 0.61 s for Online-CSLR, while our window-processing latency is half as large (14 against 29 ms). Each prediction uses a 16-frame window, and the first token is committed once the five-window past-only vote has formed; the TTFT therefore reflects a deliberate latency–accuracy choice rather than execution cost, and Tab. 7 varies it through $K$ and $L$ on the test split. At $K$=8, $L$=3 the TTFT is 0.37 s, within 0.05 s of Online-CSLR, with 2.59 higher BLEU-4 and 1.5 lower WER; $K$=16, $L$=5 is the accuracy-oriented default used in all other tables. The past-only vote at $L$=5 matches a non-causal bidirectional vote on Phoenix-2014T (18.3 WER, 29.24 BLEU-4); the vote-horizon sweep is in App. C. Per window, pose extraction takes 38.2 ms (median) and each learned stage stays below 10 ms (App. C, Tab. 14).

Table 7: Operating points on Phoenix-2014T test ($K$: window length, $L$: past-only vote horizon).

| System | TTFT ↓ | WER ↓ | B-4 ↑ |
|---|---|---|---|
| Online-CSLR | **0.32** s | 22.1 | 23.69 |
| Ours, $K$=8, $L$=3 | 0.37 s | 20.6 | 26.28 |
| Ours, $K$=8, $L$=5 | 0.56 s | 19.4 | 28.50 |
| Ours, $K$=16, $L$=5 | 0.66 s | **18.3** | **29.24** |

Table 8: Committed output on Phoenix-2014T test. At each input prefix, only text already emitted is scored, against the reference aligned to the same prefix. Coverage: share of the final output tokens already committed.

| Input prefix | 20% | 40% | 60% | 80% | 100% |
|---|---|---|---|---|---|
| Prefix-BLEU-1 | 38.5 | 48.2 | 56.7 | 60.8 | 63.15 |
| Prefix-BLEU-4 | 12.4 | 19.7 | 24.8 | 27.6 | 29.24 |
| Prefix-ROUGE-L | 38.7 | 49.3 | 57.1 | 60.5 | 62.43 |
| Token coverage | 22% | 44% | 65% | 84% | 100% |

**Reliability of committed output.** Under commit-on-emit the revision rate is zero, so displayed text is stable by construction; its correctness and timeliness are measured on the committed text itself (Tab. 8). After 80% of the video the system has committed 84% of its final tokens, and this text reaches 27.6 prefix-BLEU-4 against 29.24 on the full video. Most of the translation is on screen before the video ends, at close to its final quality. Before commitment, the past-only vote corrects unstable window retrievals: over groups of five consecutive windows inside one forced-aligned gloss span, 76.8% of the raw top-1 retrievals match the gloss, and the voted decision is correct for 88.2% of the groups (App. C.4).

### 4.6 Robustness via Pose-Quality Reframing

Because an apparently broken video can still yield reasonable poses (and vice versa), we frame robustness as a question of pose quality and vary pose quality in two ways: by augmenting How2Sign video before pose extraction (Fig. 3, Tab. 9) and by replacing the pose detector (Tab. 10). Both ask whether BLEU tracks pose quality or drifts independently of it.

Table 9: Per-augmentation pose quality and BLEU on How2Sign. MPJPE in pixels; Missing = fraction of frames with $> 50\%$ joints below $\gamma$; Finger conf. over the 21 hand keypoints; Bone-viol. rate ($> 15\%$ deviation from per-signer reference); Hand-jit. = median normalised inter-frame hand-keypoint jitter. Bottom row: Pearson with $\Delta$B-4.

| Aug. | MPJPE↓ | Miss.↓ | Finger↑ | B-viol.↓ | Jit.↓ | B-4↑ |
|---|---|---|---|---|---|---|
| Orig. | 0.0 | 0.04 | 0.78 | 0.05 | 0.012 | 18.92 |
| Bg. repl. | 1.8 | 0.06 | 0.74 | 0.07 | 0.014 | 18.75 |
| Body repl. | 3.5 | 0.09 | 0.69 | 0.11 | 0.018 | 18.32 |
| View $\pm 15°$ | 4.2 | 0.07 | 0.71 | 0.10 | 0.016 | 19.04 |
| View $\pm 30°$ | 8.7 | 0.13 | 0.62 | 0.16 | 0.024 | 18.67 |
| $\rho(\Delta$B-4) | $-0.78$ | $-0.71$ | $+0.74$ | $-0.69$ | $-0.66$ | — |

Table 10: Pose detector on How2Sign test. FPS: detector throughput; B-viol.: bone-violation rate as in Tab. 9.

| Detector | FPS ↑ | B-viol. ↓ | B-4 ↑ |
|---|---|---|---|
| MediaPipe Holistic (Lugaresi et al., 2019) | **65** | 0.108 | 17.39 |
| RTMPose (default) (Jiang et al., 2023) | 34 | 0.050 | 18.92 |
| ViTPose++ (Xu et al., 2023) | 27 | 0.041 | 19.07 |
| DWPose (Yang et al., 2023) | 18 | **0.036** | **19.28** |

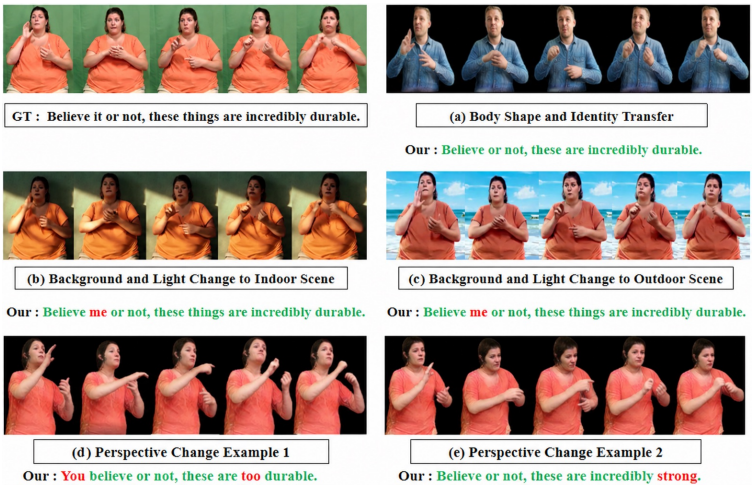


Figure 3: Sensitivity analysis on How2Sign. Representative panels for the four augmentation categories of Tab. 9: (a) body-shape and identity transfer, (b) indoor and (c) outdoor background+illumination change, (d) and (e) perspective change at two view angles. Each clip runs through the same strictly causal pipeline; "Ours" against ground truth "GT".

Across the How2Sign augmentations, all five pose-quality signals in Tab. 9 correlate with $\Delta$BLEU-4 at $|\rho| \in [0.66, 0.78]$, with MPJPE the strongest single predictor. Replacing the detector gives the same ordering with the pipeline unchanged: BLEU-4 rises from 17.39 to 19.28 as the bone-violation rate falls from 0.108 (MediaPipe) to 0.036 (DWPose). All four detectors work with the same pipeline, and quality follows detector accuracy. RTMPose is our default because it has the lowest bone-violation rate among the detectors that run above 30 FPS; ViTPose++ and DWPose add 0.15 and 0.36 BLEU-4 at 27 and 18 FPS. Detector confidence acts as a reliability mask: with coordinates fixed, the full confidence design reaches 76.14 WLASL2000 accuracy, against 67.07 without the gate and 65.52 when confidence is embedded as an input feature (App. D, Tab. 18). App. F adds RGB perturbations before pose extraction on Phoenix-2014T and OpenASL (at most 0.71 BLEU-4 lower), a failure-case taxonomy, and an OOV stress test.

## 5 Conclusion

We presented a strictly causal, pose-based SLT pipeline in which a confidence-aware pose representation, a part-specific discriminative VQ tokenizer, and a prototype-retrieval decoder with past-only voting and a causal boundary head produce text from past frames only and commit each output once. Evaluated under a streaming protocol that scores latency and the committed text, the system improves over the strongest online baseline on German, Chinese, and English benchmarks, both at its default operating point and at matched time-to-first-token; its accuracy degrades gradually under noisy train-side segmentation, carries over across pose detectors, and follows pose quality. The pipeline runs in real time on a single A100: a window costs 51.9 ms in total, 38.2 ms of it in pose extraction and under 10 ms in each learned stage (Tab. 14). Limitations, including prototype-bank coverage of unseen and fingerspelled signs, are in App. I.

### AI USE STATEMENT

We used large language model assistants for editing (grammar, spelling, and word choice). All AI-assisted text were reviewed by the authors, and we take responsibility for the final content of this work.

### ETHICS STATEMENT

This work uses only public sign-language benchmarks (Phoenix-2014, Phoenix-2014T, CSL-Daily, How2Sign, OpenASL, WLASL2000) under their original licences; the SOKE retrieval segmenter is run against the public word-level sign dictionary released by SOKE (Zuo et al., 2025), built from MS-ASL (Joze & Koller, 2019) and WLASL (Li et al., 2020). No new human-subjects data were collected; the 500-clip boundary subset (App. B) was annotated internally by the authors on training-split clips. Real-time SLT is intended to support live captioning and accessibility for deaf and hard-of-hearing users. Because the system operates on 2D pose only, it is signer-anonymous by construction and cannot be repurposed for face- or identity-level recognition. However, the system is bounded by its prototype vocabulary and inherits pose-detector failure modes (App. I), so deploying it without these caveats could give incorrect translations to users who depend on them; it should complement, not replace, human interpreters.

### REPRODUCIBILITY STATEMENT

Sec. 3 specifies the full architecture and the streaming inference procedure (Alg. 1); Sec. 4.1 lists all hyperparameters, optimisers, and inference parameters. App. A gives the complete variable table and every loss term, and App. B documents both train-side word segmenters end-to-end. All reported numbers are means over five seeds, and all latency measurements are on a single NVIDIA A100 (Tab. 6, Tab. 14). Code will be released.

## REFERENCES

Samuel Albanie, Gül Varol, Liliane Momeni, Triantafyllos Afouras, Joon Son Chung, Neil Fox, and Andrew Zisserman. BSL-1K: Scaling up co-articulated sign language recognition using mouthing cues. In *ECCV*, pp. 35–53, 2020.

Keyu An, Huahuan Zheng, Zhijian Ou, Hongyu Xiang, Ke Ding, and Guanglu Wan. CUSIDE: Chunking, simulating future context and decoding for streaming ASR. In *Interspeech*, pp. 2103–2107, 2022.

Necati Cihan Camgöz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. Neural sign language translation. In *CVPR*, pp. 7784–7793, 2018.

Necati Cihan Camgöz, Oscar Koller, Simon Hadfield, and Richard Bowden. Sign language transformers: Joint end-to-end sign language recognition and translation. In *CVPR*, pp. 10023–10033, 2020.

Shane Carroll. punctuators: Onnx punctuation/capitalization/segmentation models. `https://github.com/1-800-BAD-CODE/punctuators`, 2023.

Yutong Chen, Fangyun Wei, Xiao Sun, Zhirong Wu, and Stephen Lin. A simple multi-modality transfer learning baseline for sign language translation. In *CVPR*, pp. 5120–5130, 2022a.

Yutong Chen, Ronglai Zuo, Fangyun Wei, Yu Wu, Shujie Liu, and Brian Mak. Two-stream network for sign language recognition and translation. In *NeurIPS*, volume 35, pp. 17043–17056, 2022b.

Zhigang Chen, Benjia Zhou, Jun Li, Jun Wan, Zhen Lei, Ning Jiang, Quan Lu, and Guoqing Zhao. Factorized learning assisted with large language model for gloss-free sign language translation. In *LREC-COLING*, pp. 7071–7081, 2024.

Runpeng Cui, Hu Liu, and Changshui Zhang. A deep neural framework for continuous sign language recognition by iterative training. *IEEE TMM*, 2019.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *CVPR*, 2019.

Amanda Duarte, Shruti Palaskar, Lucas Ventura, Deepti Ghadiyaram, Kenneth DeHaan, Florian Metze, Jordi Torres, and Xavier Giro-i Nieto. How2Sign: A large-scale multimodal dataset for continuous American Sign Language. In *CVPR*, 2021.

Ruchao Fan, Pan Zhou, Wei Chen, Jia Jia, and Gang Liu. An online attention-based model for speech recognition. In *Interspeech*, pp. 4390–4394, 2019.

Shiwei Gan, Yafeng Yin, Zhiwei Jiang, Kang Xia, Lei Xie, and Sanglu Lu. Contrastive learning for sign language recognition and translation. In *IJCAI*, pp. 763–772, 2023.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *ICML*, 2006.

Shester Gueuwou, Xiaodan Du, Greg Shakhnarovich, Karen Livescu, and Alexander H. Liu. SHuBERT: Self-supervised sign language representation learning via multi-stream cluster prediction. In *ACL*, pp. 28792–28810, 2025.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

Yanzhang He, Tara N Sainath, Rohit Prabhavalkar, Ian McGraw, Raziel Alvarez, Ding Zhao, David Rybach, Anjuli Kannan, Yonghui Wu, Ruoming Pang, et al. Streaming end-to-end speech recognition for mobile devices. In *ICASSP*, pp. 6381–6385, 2019.

Hezhen Hu, Weichao Zhao, Wengang Zhou, and Houqiang Li. SignBERT+: Hand-model-aware self-supervised pre-training for sign language understanding. *IEEE TPAMI*, 45(9):11221–11239, 2023a.

Lianyu Hu, Liqing Gao, Zekang Liu, and Wei Feng. Continuous sign language recognition with correlation network. In *CVPR*, pp. 2529–2539, 2023b.

Eui Jun Hwang, Sukmin Cho, Junmyeong Lee, and Jong C. Park. An efficient gloss-free sign language translation using spatial configurations and motion dynamics with LLMs. In *NAACL*, 2025.

Tao Jiang, Peng Lu, Li Zhang, Ningsheng Ma, Rui Han, Chengqi Lyu, Yining Li, and Kai Chen. RTMPose: Real-time multi-person pose estimation based on MMPose. *arXiv preprint arXiv:2303.07399*, 2023.

Hamid Reza Vaezi Joze and Oscar Koller. MS-ASL: A large-scale data set and benchmark for understanding American Sign Language. In *BMVC*, 2019.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, 2020.

Oscar Koller, Jens Forster, and Hermann Ney. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding*, 141:108–125, 2015.

Taeryung Lee, Yeonguk Oh, and Kyoung Mu Lee. Human part-wise 3D motion context learning for sign language recognition. In *ICCV*, pp. 20740–20750, 2023.

Dongxu Li, Cristian Rodriguez, Xin Yu, and Hongdong Li. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *WACV*, 2020.

Zecheng Li, Wengang Zhou, Weichao Zhao, Kepeng Wu, Hezhen Hu, and Houqiang Li. Uni-Sign: Toward unified sign language understanding at scale. In *ICLR*, 2025.

Kezhou Lin, Xiaohan Wang, Linchao Zhu, Ke Sun, Bang Zhang, and Yi Yang. Gloss-free end-to-end sign language translation. In *ACL*, pp. 12904–12916, 2023.

Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. MediaPipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*, 2019.

Haoran Miao, Gaofeng Cheng, Pengyuan Zhang, Ta Li, and Yonghong Yan. Online hybrid CTC/attention architecture for end-to-end speech recognition. In *Interspeech*, pp. 2623–2627, 2019.

Haoran Miao, Gaofeng Cheng, Changfeng Gao, Pengyuan Zhang, and Yonghong Yan. Transformer-based online CTC/attention end-to-end speech recognition architecture. In *ICASSP*, pp. 6084–6088, 2020.

Yuecong Min, Aiming Hao, Xiujuan Chai, and Xilin Chen. Visual alignment constraint for continuous sign language recognition. In *ICCV*, pp. 11542–11551, 2021.

Liliane Momeni, Gül Varol, Samuel Albanie, Triantafyllos Afouras, and Andrew Zisserman. Watch, read and lookup: learning to spot signs from multiple supervisors. In *ACCV*, 2020.

Vineel Pratap, Qiantong Xu, Jacob Kahn, Gilad Avidov, Tatiana Likhomanenko, Awni Hannun, Vitaliy Liptchinsky, Gabriel Synnaeve, and Ronan Collobert. Scaling up online speech recognition using ConvNets. In *Interspeech*, pp. 3376–3380, 2020.

Junfu Pu, Wengang Zhou, and Houqiang Li. Iterative alignment network for continuous sign language recognition. In *CVPR*, pp. 4165–4174, 2019.

Phillip Rust, Bowen Shi, Skyler Wang, Necati Cihan Camgöz, and Jean Maillard. Towards privacy-aware sign language translation at scale. In *ACL*, 2024.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. BLEURT: Learning robust metrics for text generation. In *ACL*, 2020.

Bowen Shi, Diane Brentari, Gregory Shakhnarovich, and Karen Livescu. Open-domain sign language translation learned from online video. In *EMNLP*, 2022.

Dave Uthus, Garrett Tanzer, and Manfred Georg. YouTube-ASL: A large-scale, open-domain American Sign Language-English parallel corpus. In *NeurIPS*, volume 36, 2023.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NeurIPS*, 2017.

Gül Varol, Liliane Momeni, Samuel Albanie, Triantafyllos Afouras, and Andrew Zisserman. Read and attend: Temporal localisation in sign language videos. In *CVPR*, pp. 16857–16866, 2021.

Gül Varol, Liliane Momeni, Samuel Albanie, Triantafyllos Afouras, and Andrew Zisserman. Scaling up sign spotting through sign language dictionaries. *IJCV*, 130(6):1416–1439, 2022.

Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, pp. 499–515, 2016.

Ryan Wong, Necati Cihan Camgöz, and Richard Bowden. Sign2GPT: Leveraging large language models for gloss-free sign language translation. In *ICLR*, 2024.

Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. ViTPose++: Vision transformer for generic body pose estimation. *IEEE TPAMI*, 2023.

Zhendong Yang, Ailing Zeng, Chun Yuan, and Yu Li. Effective whole-body pose estimation with two-stages distillation. In *ICCV Workshops*, 2023.

Aoxiong Yin, Zhou Zhao, Jinglin Liu, Weike Jin, Meng Zhang, Xingshan Zeng, and Xiaofei He. SimulSLT: End-to-end simultaneous sign language translation. In *ACM MM*, 2021.

Biao Zhang, Mathias Müller, and Rico Sennrich. SLTUNET: A simple unified model for sign language translation. In *ICLR*, 2023.

Yaqi Zhang, Di Huang, Bin Liu, Shixiang Tang, Yan Lu, Lu Chen, Lei Bai, Qi Chu, Nenghai Yu, and Wanli Ouyang. Motiongpt: Finetuned llms are general-purpose motion generators. In *AAAI*, volume 38, pp. 7368–7376, 2024.

Jiangbin Zheng, Yile Wang, Cheng Tan, Siyuan Li, Ge Wang, Jun Xia, Yidong Chen, and Stan Z Li. CVT-SLR: Contrastive visual-textual transformation for sign language recognition with variational alignment. In *CVPR*, 2023.

Benjia Zhou, Zhigang Chen, Albert Clapés, Jun Wan, Yanyan Liang, Sergio Escalera, Zhen Lei, and Du Zhang. Gloss-free sign language translation: Improving from visual-language pretraining. In *ICCV*, pp. 20871–20881, 2023.

Hao Zhou, Wengang Zhou, Yun Zhou, and Houqiang Li. Spatial-temporal multi-cue network for continuous sign language recognition. In *AAAI*, pp. 13009–13016, 2020.

Hao Zhou, Wengang Zhou, Weizhen Qi, Junfu Pu, and Houqiang Li. Improving sign language translation with monolingual data by sign back-translation. In *CVPR*, pp. 1316–1325, 2021.

Ronglai Zuo and Brian Mak. C2SLR: Consistency-enhanced continuous sign language recognition. In *CVPR*, 2022.

Ronglai Zuo, Fangyun Wei, and Brian Mak. Natural language-assisted sign language recognition. In *CVPR*, 2023.

Ronglai Zuo, Fangyun Wei, and Brian Mak. Towards online continuous sign language recognition and translation. In *EMNLP*, pp. 11050–11067, 2024.

Ronglai Zuo, Rolandos Alexandros Potamias, Evangelos Ververas, Jiankang Deng, and Stefanos Zafeiriou. Signs as tokens: A retrieval-enhanced multilingual sign language generator. In *ICCV*, 2025.

# APPENDIX

This appendix collects materials that complement the main paper: variable-and-symbol consistency (App. A); the train-side word-segmenters used to build the prototype bank, including the boundary-quality study for the SOKE-style segmenter on How2Sign / OpenASL (App. B); the full streaming protocol (App. C); full attribution and ablation tables (App. D); an account of why a pose-only pipeline is competitive on RGB-rich benchmarks (App. E); a pose-quality framing of the robustness study plus failure cases (App. F); and qualitative translation examples on How2Sign (App. G). App. H reports the WLASL2000 isolated-sign-recognition variant of the tokenizer, and App. I states the limitations. The lookup dictionary used by the SOKE retrieval-based segmenter on the English benchmarks is the public word-level sign dictionary released by SOKE (Zuo et al., 2025) (Sec. 3.4 of SOKE), built from MS-ASL (Joze & Koller, 2019) and WLASL (Li et al., 2020); we do not introduce a new dictionary.

## A VARIABLE TABLE, ALGORITHMS, AND PSEUDO-GLOSS PIPELINE

This appendix is the single source of truth for every symbol used in the main paper. App. A (this section) gives the variable table; App. A.3 gives the pseudo-gloss construction algorithm (Alg. 2) used by the SOKE retrieval segmenter of Sec. 3.5 and App. B. Wherever the main paper uses a symbol without local re-definition, it refers back to Tab. 11.

### A.1 COMPLETE VARIABLE TABLE

Tab. 11 lists every symbol in the order it first appears in the main paper, together with its mathematical domain and a one-line meaning. Symbols are grouped implicitly by stage: pose input, MAE, VQ tokenizer, prototype bank, streaming decoder, and evaluation metrics.

Table 11: Single-source variable table. Every symbol used in the main paper is defined here once.

| Symbol | Range / Domain | Meaning |
|---|---|---|
| $T$ | $\mathbb{N}$ | Number of frames in the input video |
| $K, s$ | $\mathbb{N}$ | Sliding-window length and stride (default $K=16$, $s=4$) |
| $L$ | $\mathbb{N}$ | Past-only majority-vote horizon (default $L=5$) |
| $\mathcal{P}$ | {body, lhand, rhand, face} | Body-part partition |
| $V_p$ | $\mathbb{N}$ | Number of joints in part $p$ |
| $\mathbf{x}_{t,v}^p$ | $\mathbb{R}^2$ | 2D coord. of joint $v$ in part $p$ at frame $t$ |
| $c_{t,v}^p$ | $[0,1]$ | Detector confidence |
| $\gamma$ | $[0,1]$ | Confidence threshold (default 0.3) |
| $\tilde{\mathbf{x}}_{t,v}^p$ | $\mathbb{R}^2 \cup \{\mathbf{0}\}$ | Confidence-gated coordinate (Eq. (1)) |
| $m_{t,v}^p$ | $\{0,1\}$ | MAE mask indicator (1 = masked) |
| $r$ | $[0,1]$ | MAE mask rate |
| $\hat{\mathbf{x}}_{t,v}^p$ | $\mathbb{R}^2$ | Encoder input after gating + masking (App. A.2) |
| $\mathbf{h}_t^p$ | $\mathbb{R}^d$ | Per-frame part-encoded feature |
| $\hat{\mathbf{y}}_{t,v}^p$ | $\mathbb{R}^2$ | Reconstruction-decoder output |
| $N_p$ | $\mathbb{N}$ | Codebook size for part $p$ |
| $\mathbf{e}_k^p$ | $\mathbb{R}^d$ | $k$-th codebook entry of part $p$ |
| $k_t^*, \mathbf{q}_t^p$ | $\mathbb{N}, \mathbb{R}^d$ | Assigned code index and quantized vector |
| $\hat{n}_k^p, \hat{\mathbf{s}}_k^p$ | $\mathbb{R}, \mathbb{R}^d$ | EMA-tracked cluster size and feature sum |
| $\alpha, \beta, \eta$ | $[0,1]$ | EMA decay, commitment weight, dead-code threshold |
| $\mathbf{F}_t$ | $\mathbb{R}^d$ | Fused part token at frame $t$ |
| $\kappa_t, \mathbf{w}_a$ | $[0,1], \mathbb{R}^d$ | Attention-pooling weight and its projection vector |
| $\mathbf{u}^{\text{vid}}, \mathbf{u}_w^{\text{blk}}$ | $\mathbb{R}^D$ | Projection-head outputs |
| $g \in \{1,\dots,G\}$ | $\mathbb{N}$ | Gloss / class index |
| $g_i \in \{1,\dots,G\}$ | $\mathbb{N}$ | Gloss label of training clip $i$ (shared by word- and sentence-level clips) |
| $\ell \in \{0,1\}$ | — | Clip level (0 = word, 1 = sentence); bookkeeping only, not a class |
| $\mathcal{N}^+(i), \mathcal{N}^-(i)$ | sets | Positive and negative sets of anchor $i$ |
| $\mathcal{A}, \mathcal{S}, \mathcal{B}, \mathcal{V}(i)$ | sets | Anchors with a positive, singleton anchors, blocks in a batch, video embeddings of block $i$'s video |
| $\mathcal{D}_g$ | set | Train-split clips labelled $g$ |
| $\boldsymbol{\pi}_g$ | $\mathbb{R}^D$ | Prototype embedding for gloss $g$ (Sec. 3.4) |
| $g_w^*, \hat{g}_w$ | $\mathbb{N}$ | Per-window retrieval and post-vote prediction |
| $\mathcal{T}$ | sequence | Deduplicated emitted gloss stream |
| $\sigma_n, y_n^b, \omega_+, \theta_b$ | $[0,1], \{0,1\}, \mathbb{R}_+, [0,1]$ | Boundary prob., label, positive weight, emission threshold |
| $\mathbf{z}$ | sequence | Output text sentence |
| $\tau$ | $\mathbb{R}_+$ | Contrastive temperature |
| $e_i, \mu_i$ | $\mathbb{R}_+$ | Token emission time and reference mid-time |
| TTFT, $\bar{\ell}_{\text{tok}}$ | $\mathbb{R}_+$ | Time-to-first-token, mean token latency |

### A.2 DETAILED EQUATIONS FOR STAGES 1–3

The main paper compresses Stages 1–3 around the design-load-bearing equations only. We collect the deferred equations here in the order they appear in Sec. 3.

**Stage 1 (encoder/decoder).** After confidence gating (Eq. (1)), masked joints are zeroed, $\hat{\mathbf{x}}^p_{t,v} = (1-m^p_{t,v})\,\tilde{\mathbf{x}}^p_{t,v}$, projected to tokens, augmented with temporal+part embeddings, and processed by the shared 6-layer Transformer encoder $\mathbf{h}^p_t = \mathrm{Enc}(\hat{\mathbf{x}}^p_{1:T})_t \in \mathbb{R}^d$. The 2-layer 1-D-conv decoder produces $\hat{\mathbf{y}}^p_t = \mathrm{Dec}(\mathbf{h}^p_t) \in \mathbb{R}^{V_p \times 2}$, supervised by Eq. (2) of the main paper.

**Stage 2 (codebook updates and fusion).** Gradients flow through the discrete bottleneck via the straight-through estimator $\tilde{\mathbf{q}}^p_t = \mathbf{h}^p_t + \mathrm{sg}(\mathbf{q}^p_t - \mathbf{h}^p_t)$. With cluster-size and feature-sum EMAs $\hat{n}^p_k$, $\hat{\mathbf{s}}^p_k$ at decay $\alpha$=0.99:

$$\hat{n}^p_k \leftarrow \alpha\hat{n}^p_k + (1-\alpha)n^p_k, \quad \hat{\mathbf{s}}^p_k \leftarrow \alpha\hat{\mathbf{s}}^p_k + (1-\alpha)\mathbf{s}^p_k, \quad \mathbf{e}^p_k \leftarrow \hat{\mathbf{s}}^p_k/(\hat{n}^p_k + \epsilon). \tag{12}$$

The commitment loss is $\mathcal{L}_{\text{vq}} = \beta \sum_{p,t} \|\mathbf{h}^p_t - \mathrm{sg}(\mathbf{q}^p_t)\|_2^2$ with $\beta$=0.25. Quantised part tokens are concatenated and fused, $\mathbf{F}_t = \mathrm{Fuse}([\mathbf{q}^{\text{body}}_t \| \mathbf{q}^{\text{lhand}}_t \| \mathbf{q}^{\text{rhand}}_t \| \mathbf{q}^{\text{face}}_t])$. Video- and block-level embeddings use attention pooling and a uniform mean; both pass through the 2-layer MLP $\pi$:

$$\kappa_t = \frac{\exp(\mathbf{w}_a^\top \tanh(\mathbf{F}_t))}{\sum_{t'} \exp(\mathbf{w}_a^\top \tanh(\mathbf{F}_{t'}))}, \quad \mathbf{u}^{\text{vid}} = \pi\left(\sum_t \kappa_t \mathbf{F}_t\right), \quad \mathbf{u}^{\text{blk}}_w = \pi\left(\tfrac{1}{K}\sum_{t\in w} \mathbf{F}_t\right). \tag{13}$$

**Stage 2 (discriminative losses).** The video-level term $\mathcal{L}_{\text{v2v}}$ (SupCon (Khosla et al., 2020)), the singleton fallback $\mathcal{L}_{\text{neg}}$, and the block-to-video term $\mathcal{L}_{\text{b2v}}$ are Eqs. (4)–(6) of the main paper.

The angular-margin term (ArcFace (Deng et al., 2019), margin $m_{\text{a}}$=0.50, scale $s_{\text{a}}$=30) and the centre regulariser (Wen et al., 2016) (centres updated by EMA, $\eta_c$) over the $G$ gloss classes are

$$o_j = \begin{cases} s_{\text{a}} \cos(\theta_j + m_{\text{a}}) & j{=}g_i \\ s_{\text{a}} \cos\theta_j & j{\neq}g_i \end{cases}, \quad \mathcal{L}_{\text{arc}} = -\log \frac{\exp(o_{g_i})}{\sum_j \exp(o_j)}, \quad \mathcal{L}_{\text{ctr}} = \tfrac{1}{N}\sum_i \|\mathbf{u}_i - \boldsymbol{\mu}_{g_i}\|_2^2. \tag{14}$$

**Stage 3 (causal boundary head and translation).** The causal boundary head, a 2-layer causal Transformer ($d$=192), reads the running gloss stream and emits boundary probabilities $\sigma_n = \mathrm{sigmoid}(\mathbf{w}_b^\top \mathbf{a}_n + b_b)$ trained with the weighted cross-entropy of Eq. (10) of the main paper. The 4+4-layer Transformer translator $p_\theta$ is causal in $z_{<k}$ but conditions on the entire emitted segment $\mathcal{T}_{n_1:n_2}$, Eq. (11).

### A.3 Pseudo-gloss pipeline (algorithmic spec)

Alg. 2 specifies how an English training transcript is turned into the content-word sequence that the SOKE retrieval segmenter (App. B) looks up in the sign dictionary.

**Algorithm 2** Pseudo-gloss construction from an English transcript.

**Require:** Transcript sentence $S$ (string), ASL multi-word-sign vocabulary $V_{\text{asl}}$, ASL-keep set $K_{\text{asl}}$
**Ensure:** Pseudo-gloss sequence $G$ (list of underscore-tokens, ASL-grammar order)

$S \leftarrow$ contraction-expand($S$) ▷ didn't → did not, etc.
$S' \leftarrow$ tokenize-spaCy($S$)
$G \leftarrow [\,]$
$i \leftarrow 0$
**while** $i < |S'|$ **do** ▷ phrase-aware tokenisation
  $j^\star \leftarrow$ longest $j > i$ s.t. $S'[i{:}j] \in V_{\text{asl}}$ (else $j^\star = i+1$)
  $tok \leftarrow$ underscore-join($S'[i{:}j^\star]$)
  **if** $tok \in V_{\text{asl}}$ **or** POS($tok$) $\in$ {NOUN, VERB, PROPN, NUM, ADJ, ADV, PRON} **then**
    **if** $tok \notin$ STOP$\backslash K_{\text{asl}}$ **and** not pure-digits **then**
      $G \leftarrow G \,\|\, \mathrm{lemma}(tok)$.upper()
    **end if**
  **end if**
  $i \leftarrow j^\star$
**end while**
$G \leftarrow$ ASL-reorder($G$) ▷ TIME+TOPIC+SUBJECT+VERB+OTHER+NEG+WH
**return** $G$

**ASL multi-word-sign vocabulary.** 27,080 entries from the SignASL.org public dictionary; entries have at most 4 tokens. Examples: `HOW_ARE_YOU`, `AT_LEAST`, `NEW_YORK`, `ICE_CREAM`.

**ASL-keep set.** {what, who, where, when, why, how, which, not, no}—function words that are real ASL signs and survive STOP filtering.

**ASL-reorder rule.** Implemented over a spaCy dependency parse: identify TIME (date/temporal NPs), TOPIC (object that begins the ASL phrase), SUBJECT (subject NP), VERB (head verb), OTHER (residual content), NEG (*not*, *no* when present), WH (terminal interrogative). Concatenate in the listed order; ties broken by source token index.

**No test-side leakage.** The pipeline reads the training-split transcripts only; vocabulary $V_{\text{asl}}$ is constructed from external ASL dictionary resources, not from any evaluation transcripts.

# B TRAIN-SIDE WORD-LEVEL SEGMENTERS

This appendix gives the implementation details of the two segmenters in Sec. 3.5 and the boundary-quality study used to support the How2Sign / OpenASL prototype bank.

## B.1 TWOSTREAM-SLT SEGMENTER (PHOENIX-2014/2014T, CSL-DAILY)

We use the public TwoStream-SLT (Chen et al., 2022b) CSLR system without architectural changes, trained on the official training split of each benchmark. The trained recogniser is run on each training video; the predicted gloss-aligned timestamps yield (start, end, gloss) triples that we take as ground-truth segmentation for the prototype bank. We do not run the recogniser on dev/test; word-level prototypes are therefore derived purely from training-split recognitions of training-split video.

## B.2 SOKE RETRIEVAL SEGMENTER (HOW2SIGN, OPENASL)

We follow Sec. 3.1 of SOKE (Zuo et al., 2025): for each training transcript $S$ we enumerate its content words after the pseudo-gloss pipeline of Alg. 2 (App. A.3), and for each word $w_i$ we retrieve the nearest sign clip from a precomputed isolated-sign reference set. The retrieved clip is aligned to the training-split video by dynamic time warping over our 2D-pose features, producing a (start, end) span on the H2S/OpenASL training split. The reference set we use is the public word-level sign dictionary released by SOKE (Sec. 3.4 of SOKE), constructed from MS-ASL (Joze & Koller, 2019) and WLASL (Li et al., 2020); we introduce no new word-level data. For the German and Chinese benchmarks no SOKE-style segmentation is used, since CSLR boundaries are available. The dictionary is consulted only as a training-side segmentation lookup; each benchmark's prototype bank is built from its own training-split clips.

**Boundary-quality study.** On a 500-clip subset annotated internally by the authors (drawn evenly across How2Sign and OpenASL test-disjoint training-split clips; no crowd workers and no new human subjects involved) we report six quantities: (i) boundary-$F_1$ at $\pm1$ and $\pm3$ frame tolerance; (ii) per-clip word-level precision/recall; (iii) mean boundary error in frames; (iv) *invalid-rate*, the fraction of predicted segments whose duration is outside the plausible word-duration range $[3, 40]$ frames at 30 fps; (v) *coverage*, the fraction of gold-annotated word boundaries with at least one predicted boundary within $\pm3$ frames; (vi) *spurious-boundary rate*, predicted boundaries with no gold within $\pm5$ frames per video minute. Numbers: boundary-$F_1$ at $\pm1$ frame = **0.62**; at $\pm3$ frames = **0.83**; per-clip word-level precision/recall = **0.74/0.69**; mean boundary error = **3.4** frames; invalid-rate = **0.087**; coverage = **0.91**; spurious-boundary rate = **4.2 per video minute**. The oracle-boundary condition of Tab. 5 uses these annotated boundaries and extends them to the rest of the training split with audio-aligned word timestamps.

**Information flow.** Train-side word segmentation and the causal boundary head of Sec. 3.4 both delimit units but act at different times. Train-side word segmentation cuts approximate word clips from the official training split, offline and with an external model, to build the Stage-2 word clips and the fixed prototype bank. The causal boundary head is a learned component that, at inference, groups the already retrieved gloss stream into segments for the translator and reads only past outputs. Tab. 12 lists every resource and signal and where it is used. At inference, Alg. 1 reads only the current causal window, the frozen encoder and prototype bank, previous retrievals, the pending gloss buffer, the boundary head, and the translator; TwoStream-SLT, the SOKE dictionary, DTW, training transcripts, and word boundaries are never called or read.

Table 12: Resources and signals used for training (representation, prototype bank, boundary head, translator) and for dev/test inference.

| Resource or signal | Training | Dev/test inference |
|---|---|---|
| TwoStream-SLT predicted boundaries (Phoenix-2014/2014T, CSL-Daily) | training split only | not used |
| SOKE lookup + DTW (How2Sign, OpenASL) | training split only | not used |
| SOKE isolated-sign videos | lookup/alignment reference; not stored as prototypes | not used |
| Training glosses, transcripts, pseudo-glosses | clip labels; boundary-head and translator targets | not read |
| Benchmark training clips | encoder training, prototype-bank construction | not accessed |
| Prototype bank | built once from the training split | read-only lookup |
| Incoming dev/test pose | — | current and past frames only |
| Past retrievals | — | past-only vote |
| Causal boundary head, translator | trained on the training split | frozen, causal |
| Future frames, dev/test annotations | not used | not used |

# C Streaming Protocol Details

This appendix supports Sec. 4.5: the vote-horizon sweep and the per-stage runtime (App. C.1), the sensitivity of $\bar{\ell}_{\text{tok}}$ to its alignment reference (App. C.2), the protocol behind the committed-output evaluation of Tab. 8 (App. C.3), and the window-level stability of the past-only vote (App. C.4).

## C.1 Vote horizon and per-stage runtime

Tab. 13 sweeps the past-only vote horizon at $K$=16, $s$=4. Accuracy saturates at $L$=5, which reproduces the non-causal bidirectional vote. Tab. 14 decomposes wall-clock runtime into the five pipeline stages; pose extraction dominates, and every learned stage costs less than 10 ms per window.

Table 13: Past-only vote sweep on Phoenix-2014T test ($K$=16, $s$=4). The bidirectional-vote row is a non-causal reference, included for comparison only.

| Vote scheme | WER ↓ | B-4 ↑ |
|---|---|---|
| Bidirectional vote (original) | 18.3 | 29.24 |
| Past-only $L = 7$ | 18.6 | 29.10 |
| Past-only $L = 5$ (**default**) | **18.3** | **29.24** |
| Past-only $L = 3$ | 19.5 | 27.85 |
| $L = 1$ (no vote) | 21.4 | 26.15 |

Table 14: Per-stage runtime, single A100. Per-window cost amortised across stride-$s$=4 windows on a 25-word sentence; per-sentence cost at length 25 / 100 words. All values in milliseconds.

| Stage | Per-sentence (ms) | | Per-window (ms) | |
|---|---|---|---|---|
| | 25 w | 100 w | median | p95 |
| Pose extraction (RTMPose) | 1085 | 4256 | 38.2 | 52.4 |
| Pose-MAE + VQ + fusion | 106 | 514 | 4.5 | 7.2 |
| Prototype retrieval | 60 | 293 | 2.6 | 4.8 |
| Past-only voting + dedup | 238 | 1032 | 1.2 | 2.0 |
| Gloss-to-text (greedy) | 132 | 587 | 5.4 | 8.6 |
| Total | 1621 | 6682 | 51.9 | 75.0 |

## C.2 Sensitivity of token latency to the alignment reference

Phoenix-2014T provides gloss identities and order but no token timestamps, so $\bar{\ell}_{\text{tok}}$ needs an evaluation-only time reference. We constrain a forced aligner to the official gloss sequence, obtain a span $[a_i, b_i]$ for each token $i$, and set $\mu_i = (a_i + b_i)/2$; a token committed at video time $e_i$ has latency $e_i - \mu_i$. Tab. 15 perturbs only $\mu_i$ with zero-mean Gaussian jitter and recomputes $\bar{\ell}_{\text{tok}}$ on the test split; the tested range covers and exceeds the 3.4-frame mean boundary error of App. B. Because $e_i - \mu_i$ is signed, jitter can move the mean in either direction; the largest change is 0.08 s. We therefore read 0.78 s as an alignment-dependent estimate with this tolerance. This test changes only the latency reference; the boundary jitter of Sec. 4.4 instead rebuilds the prototype bank.

## C.3 Committed-output evaluation protocol

Tab. 8 evaluates each Phoenix-2014T test video at input prefixes $p \in \{20, 40, 60, 80, 100\}\%$. For a prefix, the model receives only the frames up to $p \cdot T$, and we collect exactly the text it has emitted by then; output produced later is excluded. That text is scored against the reference content up to the

Table 15: Mean token latency under jitter of the forced-aligned reference mid-times, Phoenix-2014T test, default operating point.

| Reference timing | $\bar{\ell}_{\text{tok}}$ | Change |
|---|---|---|
| Forced alignment (default) | 0.78 s | — |
| Jitter $\sigma_{\text{j}}$=4 frames | 0.81 s | +0.03 s |
| Jitter $\sigma_{\text{j}}$=8 frames | 0.86 s | +0.08 s |
| Jitter $\sigma_{\text{j}}$=12 frames | 0.83 s | +0.05 s |

same time, obtained by the forced alignment of App. C.2. Prefix-BLEU-1/4 and Prefix-ROUGE-L measure translation overlap of the committed text, and token coverage is the fraction of the final output tokens already committed. Under commit-on-emit this committed text is what the user sees, so the table measures the reliability of the irreversible output rather than of a later corrected sequence. At $p$=100% the numbers equal the sentence-level results of Tab. 2.

### C.4 Window-level stability of the past-only vote

This diagnostic measures how voting treats unstable retrievals before anything is committed. On Phoenix-2014T test we form groups of five consecutive windows whose centres all fall inside the same forced-aligned gloss span, and discard every group that crosses a gloss boundary, so a genuine sign transition is not counted as a retrieval error. For each retained group we compute the fraction of its five raw top-1 retrievals that match the reference gloss, and we score the single group-level decision of the past-only $L$=5 vote. On average 76.8% of the raw window retrievals match the reference gloss, and the voted decision is correct for 88.2% of the groups. The two numbers are measured at different levels: the first is per-window agreement, the second the accuracy of the five-window decision.

## D Full Ablation Tables

The ablations in this appendix expand on the attribution table in the main paper (Tab. 4) by isolating the design choices inside each stage individually. App. D.1 (Tab. 17) decomposes the Stage-2 discriminative loss; App. D.2 (Tab. 18) ablates the MAE input pathway including the deliberately-bad "confidence as feature" control; App. D.3 (Tab. 19) toggles the streaming-decoder components introduced in Sec. 3.4; App. D.4 (Tab. 20) splits the residual error into recognition vs. translation. Each table changes *exactly one* component relative to the corresponding full pipeline. Tab. 16 maps every module of the headline comparison to the experiment that isolates it.

Table 16: Where each component of the pipeline is isolated.

| Component | Origin | Isolating experiment |
|---|---|---|
| Train-side word segmentation | external (TwoStream-SLT / SOKE) | jitter, clip removal, coverage, segmenter transfer (Tab. 5) |
| Pose detector | external (RTMPose) | four detectors (Tab. 10) |
| Confidence gate, Pose-MAE | Stage 1 | Tab. 18 |
| Discriminative Stage-2 objective | Stage 2 | Tab. 17 |
| Prototype bank | Stage 3 | linear classifier, shuffled labels (Tab. 4) |
| Past-only vote | Stage 3 | $L$ sweep (Tab. 13) |
| Causal boundary head | Stage 3 | Tab. 19 |
| Gloss-to-text translator | Stage 3 | oracle gloss, identity decoder (Tab. 20) |

### D.1 Loss-component ablation

Tab. 17 ablates each term in the Stage-2 loss $\mathcal{L}_2$ defined in Eq. (7) on the WLASL2000 ISLR variant of the tokenizer (App. H). Removing the discriminative losses entirely ($\mathcal{L}_{\text{v2v}}$, $\mathcal{L}_{\text{b2v}}$, $\mathcal{L}_{\text{arc}}$, $\mathcal{L}_{\text{ctr}}$, $\mathcal{L}_{\text{neg}}$) leaves only reconstruction + VQ; the resulting drop quantifies how much of the representation is discriminative-training-induced versus reconstruction-induced. The singleton fallback ($\mathcal{L}_{\text{neg}}$) is included separately because it is the term that lets us train on vocabularies where most glosses appear in a single video.

### D.2 MAE design ablation

Tab. 18 ablates the four pieces of the Stage-1 MAE: confidence gating at the input (Eq. (1)), random masking, the alternative "confidence as feature" design (a deliberately bad control), and the reliability-weighted reconstruction target. The "confidence as feature" row is the failure mode we explicitly

Table 17: Loss-component ablation on WLASL2000 (ISLR variant). Each row removes a single loss component from the full Stage-2 objective in Eq. (7).

| Configuration | P-I | P-C |
|---|---|---|
| Full ($\mathcal{L}_2$ as in Eq. (7)) | **76.14** | **74.82** |
| w/o $\mathcal{L}_{\text{v2v}}$ (video contrastive) | 52.21 | 50.94 |
| w/o $\mathcal{L}_{\text{neg}}$ (singleton fallback) | 63.84 | 62.51 |
| w/o $\mathcal{L}_{\text{b2v}}$ (block-to-video) | 57.54 | 54.32 |
| w/o $\mathcal{L}_{\text{arc}}$ (ArcFace) | 69.78 | 67.03 |
| w/o $\mathcal{L}_{\text{ctr}}$ (centre) | 68.46 | 65.92 |
| w/o all discriminative (recon + VQ only) | 43.21 | 40.35 |

avoid because it causes the encoder to memorise the detector's confidence signature and degrade under distribution shift.

Table 18: MAE design ablation on WLASL2000 (ISLR variant).

| Configuration | P-I | P-C |
|---|---|---|
| Full | **76.14** | **74.82** |
| w/o confidence gating (Eq. (1)) | 67.07 | 65.30 |
| w/o random masking (no MAE pretraining) | 64.23 | 62.31 |
| confidence as feature (*ablation: bad design*) | 65.52 | 63.94 |
| w/o reliability-weighted reconstruction target | 69.17 | 66.45 |

### D.3 Streaming pipeline component ablation

Tab. 19 toggles each component of the Stage-3 streaming decoder: the past-only majority vote, the consecutive deduplication, the causal boundary head, the gloss-to-text Transformer, and the punctuator post-processor. Removing the boundary head passes the retrieved gloss stream to the translator without learned boundaries. The dependence on train-side word segmentation is analysed in Sec. 4.4.

Table 19: Streaming pipeline component ablation on Phoenix-2014T test / How2Sign test. Default vote horizon $L = 5$, past-only.

| Configuration | P14T WER ↓ | P14T B-4 ↑ | H2S B-4 ↑ |
|---|---|---|---|
| Full streaming pipeline | **18.3** | **29.24** | **18.92** |
| w/o majority voting ($L = 1$) | 21.4 | 26.15 | 16.20 |
| w/o consecutive dedup | 21.4 | 28.42 | 16.43 |
| w/o causal boundary head | — | 26.82 | 16.71 |
| w/o gloss-to-text Transformer (raw glosses + casing) | — | 11.50 | 7.84 |
| w/o punctuator post-processor | — | — | 17.61 |
| w/o pseudo-gloss reorder (En) | — | — | 16.74 |

### D.4 Recognition vs. translation error split

Tab. 20 attributes the residual error to recognition (gloss prediction) versus translation (gloss-to-text). The "oracle gloss in" row replaces the prototype-bank output with the official gloss sequence and measures only the gloss-to-text Transformer; the "identity decoder" row replaces the Transformer with a casing-only pass and measures only the recognition pipeline. The gaps from the full system to each oracle are BLEU-4 deltas that decompose the residual error into recognition and translation.

## E Why Pose-Only Is Competitive on RGB-Rich Benchmarks

A natural question raised by the SLT results in Tab. 2 is why a pose-only pipeline matches or exceeds RGB-trained *offline* systems on Phoenix-2014T, a benchmark for which appearance information is available. Three coupled mechanisms together account for the gap.

**(i) Visual homogeneity of the corpus.** Phoenix-2014T is recorded entirely in a single newsroom studio with consistent backgrounds, lighting, and signer appearance. The variation an RGB encoder

Table 20: Recognition vs. translation error split (Phoenix-2014T and How2Sign). The gap from oracle gloss to system measures recognition error; the gap from official transcript to oracle gloss measures translation error.

| Configuration | B-4 (P14T) | B-4 (H2S) | Identifies |
|---|---|---|---|
| Full system | 29.24 | 18.92 | joint |
| replace retrieval by oracle gloss | 36.85 | 24.62 | translation only |
| replace decoder by identity (raw gloss) | 11.50 | 7.84 | recognition only |

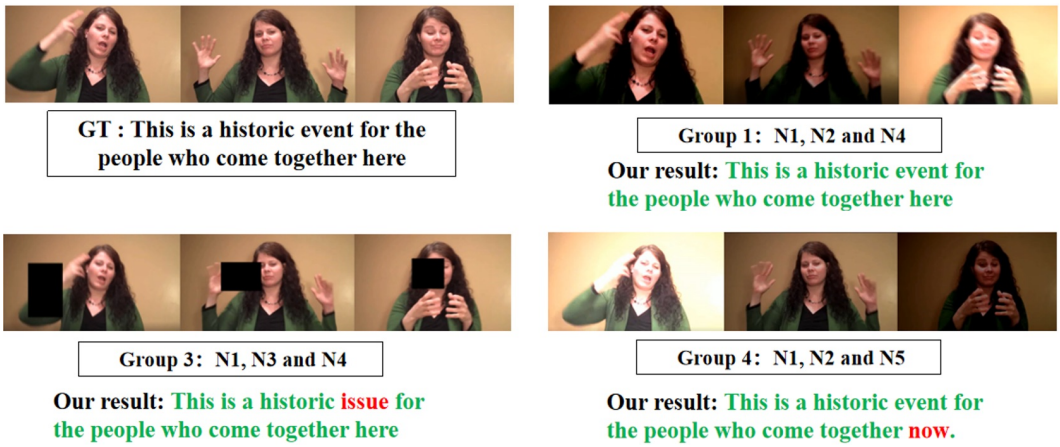


Figure 4: OpenASL examples of camera- and input-level perturbations applied *before* pose extraction. Top row: original; bottom rows: representative perturbation groups from Tab. 22 (definitions in Tab. 21). Group 1 (N1+N2+N4) and Group 4 (N1+N2+N5) maintain correct translations; Group 3 (N1+N3+N4) introduces partial occlusion and produces a minor semantic deviation ("event"→"issue"). Group 2 is omitted (qualitatively similar to Group 1); full numbers in Tab. 22.

can plausibly exploit beyond what 2D pose already encodes is small: the dataset's *visual* task is close to identity-invariant articulation recognition, which is precisely what RTMPose (Jiang et al., 2023) produces in normalised form.

**(ii) Underfitting under limited supervision.** Phoenix-2014T provides 7,096 training videos. RGB encoders that operate on ∼200–400 ms windows must allocate capacity to discount nuisance factors (background, lighting, signer appearance) that carry no information for the SLT task. With limited training data they typically underfit this denoising step before they can learn task-relevant motion patterns. Pose extraction performs the denoising up front: RTMPose is trained on large-scale general human-motion data and supplies a low-dimensional representation in which the nuisance factors are already factored out. The remaining capacity in our tokenizer can therefore be spent on *shape*-invariant articulation differences, where the supervision signal is concentrated.

**(iii) Retrieval prior as a substitute for supervision.** The attribution rows of Tab. 4 quantify what the prototype-retrieval prior contributes: replacing the bank with a linear classifier costs $-3.94$ B-4 on Phoenix-2014T, and a random-label bank costs $-21.04$ B-4. The retrieval prior acts as a strong nearest-neighbour structure over the training distribution; it lets a discriminative module trained on 7,096 labelled videos behave as if it had access to a continuous denoised manifold of training examples. RGB-based offline systems do not have an analogous structure unless they explicitly add retrieval, and most do not.

**Summary.** The combination of (i)–(iii) is not a claim that pose is universally superior to RGB—it is a claim that on a small, visually homogeneous corpus, a pose-only system with a retrieval prior operates closer to the data-distribution upper bound than an RGB-only offline system can without that prior.

## F Robustness and Failure Cases

This appendix supports the main-paper reframing of robustness as a pose-quality study (Sec. 4.6). App. F.1 defines the noise items and groups (Tab. 21) and reports translation quality under them

(Tab. 22): across the four groups, BLEU-4 falls by at most 0.71 on Phoenix-2014T and 0.69 on OpenASL. App. F.2 (Tab. 23) catalogues the five categorical failure modes we observe in the wild. App. F.3 (Tab. 24) quantifies the OOV/open-vocabulary stress test, calibrating the qualitative F2 (OOV) failure mode of Tab. 23.

### F.1 Noise group definitions

Tab. 21 defines the five noise items N1–N5 and the four groups G1–G4, all applied to the video before pose extraction; Tab. 22 reports BLEU-4 under each group on Phoenix-2014T and OpenASL, and Fig. 4 shows OpenASL examples.

Table 21: Noise definitions and evaluation groups for the pre-extraction RGB perturbation study (Tab. 22). All perturbations are applied before pose extraction.

| Item | Description |
|---|---|
| N1 | Scale variation by random rescaling of the signer region (far / close camera distance). |
| N2 | Illumination variation by random brightness and contrast jitter. |
| N3 | Partial occlusion by a random black rectangle covering 20–30% of the frame. |
| N4 | Blur by Gaussian blur together with random motion blur. |
| N5 | Temporal dropout by removing 15–25% of frames. |
| G1 | N1, N2, and N4 |
| G2 | N2, N3, and N5 |
| G3 | N1, N3, and N4 |
| G4 | N1, N2, and N5 |

Table 22: Translation quality under pre-extraction RGB noise on Phoenix-2014T and OpenASL (group definitions in Tab. 21). $\Delta$ = change vs. noise-free input.

| Setting | P14T | | OpenASL | |
|---|---|---|---|---|
| | B4 | $\Delta$B4 | B4 | $\Delta$B4 |
| Original | 29.24 | 0.00 | 26.74 | 0.00 |
| G1 | 28.85 | $-0.39$ | 26.12 | $-0.62$ |
| G2 | 28.68 | $-0.56$ | 26.44 | $-0.30$ |
| G3 | 28.71 | $-0.53$ | 26.05 | $-0.69$ |
| G4 | 28.53 | $-0.71$ | 26.27 | $-0.47$ |

### F.2 Failure case taxonomy

Tab. 23 catalogues the five categorical failure modes observed in qualitative analysis: boundary slip (F1), out-of-vocabulary signs (F2), fingerspelling (F3), profile-view signing (F4), and hand–hand self-occlusion (F5).

Table 23: Five failure-case categories observed in qualitative analysis of the test splits.

| Failure mode | Description |
|---|---|
| F1 – Boundary slip | Sign duration $> K$ frames; voting dilates the predicted gloss across two adjacent classes. |
| F2 – OOV | Sign not in prototype vocabulary; retrieval picks a phonetically similar sign. |
| F3 – Fingerspelling | Multi-letter spelled word; window misses or merges letters. |
| F4 – Profile view | Signer turned $> 60°$; RTMPose drops to $\sim 50\%$ confidence on hand joints. |
| F5 – Hand–hand self-occlusion | Two-hand sign with one hand occluding the other; right-hand keypoints flip. |

### F.3 OOV AND OPEN-VOCABULARY STRESS TEST

The qualitative F2 (OOV) entry of Tab. 23 is calibrated quantitatively in Tab. 24. We report three signals that bound how the prototype bank scales beyond the test-distribution sign vocabulary: *out-of-bank rate* (fraction of test windows whose top-1 retrieval cosine score is below $\theta_{\text{oob}} = 0.35$), *fingerspelled-letter accuracy* (per-letter and per-word, evaluated only on test sentences with $\geq 1$ fingerspelled name), and *rare-sign accuracy* (binned by training-set frequency). High out-of-bank rates indicate the bank cannot cover the test vocabulary; high rare-sign accuracy indicates the discriminative training generalises beyond per-class data scale.

Table 24: OOV / open-vocabulary stress test on Phoenix-2014T and How2Sign. "OOB" = out-of-bank rate at $\theta_{\text{oob}} = 0.35$. "FS letter / word" = fingerspelled-letter / fingerspelled-word accuracy on test sentences with at least one fingerspelled name. "Rare-sign acc." rows bin test signs by training-set frequency.

| Metric | Phoenix-2014T | How2Sign |
|---|---|---|
| Out-of-bank rate (OOB) | 8.4% | 23.6% |
| FS letter accuracy | 64.3% | 51.2% |
| FS word accuracy | 38.7% | 24.5% |
| Rare-sign acc. (1 occurrence in train) | 41.3% | 18.7% |
| Rare-sign acc. (2–5 occurrences) | 58.6% | 32.4% |
| Rare-sign acc. (6–20 occurrences) | 72.8% | 48.6% |
| Rare-sign acc. (21–100 occurrences) | 84.3% | 65.7% |
| Rare-sign acc. ($>$ 100 occurrences) | 91.7% | 79.4% |

## G QUALITATIVE TRANSLATION COMPARISON

To complement the quantitative results, we provide qualitative side-by-side translation comparisons on How2Sign. Tab. 25 lists four representative test sentences, comparing the outputs of SHuBERT (Gueuwou et al., 2025), the state-of-the-art offline baseline on How2Sign, with ours and the ground-truth (GT). The qualitative pattern is consistent with the BLEU-4 and BLEURT improvements in Tab. 3: SHuBERT occasionally misses local temporal details or produces less faithful sentence-level reconstructions even though it has full-video access, while our strictly causal pipeline produces more complete and temporally stable translations with better preservation of sentence-level semantics. Two design choices drive the gap. First, the prototype-retrieval recogniser is trained to recognise complete sign units rather than relying on short-window continuous CTC decoding, which reduces deletion errors caused by boundary ambiguity. Second, past-only majority voting and consecutive deduplication suppress unstable per-window predictions before they reach the gloss-to-text Transformer.

## H ISOLATED SIGN RECOGNITION (WLASL2000)

We additionally train an isolated-sign-recognition (ISLR) variant of the tokenizer on WLASL (Li et al., 2020) and evaluate it on the WLASL2000 test split (Tab. 26). This is reported here for completeness; ISLR is not the deployment target of this paper.

## I LIMITATIONS

The system has four limitations that we make explicit. (1) The fixed sliding window of length $K = 16$ frames trades off against the duration variability of natural sign language; long ASL classifier predicates and slow CSL signs occasionally cross window boundaries, producing boundary-slip errors that voting only partially corrects. (2) Our retrieval-based decoder is bounded by the prototype vocabulary; out-of-vocabulary signs (regional dialects, fingerspelled rare words) yield low-confidence retrievals that the gloss-to-text decoder cannot fully recover. (3) Pose-only operation makes the system signer-anonymous and bandwidth-light, but it inherits the failure modes of the underlying pose estimator (profile views, hand–hand self-occlusion); we do not claim semantic-level robustness to RGB-domain perturbations beyond what RTMPose itself can absorb. (4) The pseudo-gloss pipeline used for English data is rule-based and language-specific; extending the system to a new sign language

Table 25: Qualitative comparison of translation outputs on four representative How2Sign test sentences. "GT" = ground-truth translation; "SHuBERT" = the state-of-the-art offline baseline (Gueuwou et al., 2025); "Ours" = our strict-causal streaming pipeline. Each block corresponds to one test sentence.

| Source | Translation |
|---|---|
| GT | And that's a great vital point technique for women's self defense. |
| SHuBERT | If you're a bigger person we're talking about really self defense here. |
| Ours | That a good vital point skill for women defense. |
| GT | In this segment we're going to talk about how to load your still for distillation of lavender essential oil. |
| SHuBERT | In this clip we're going to talk about how to take our stick for disinfectant oil. |
| Ours | In this segment we discuss how to use still for distillation important oil. |
| GT | In this clip I'm going to show you how to tape your cables down. |
| SHuBERT | In this clip I'm going to show you how to brand out the cable strings. |
| Ours | In this case I show you how to tape the cables. |
| GT | You are dancing, and now you are going to need the veil and you are going to just grab the veil as far as possible. |
| SHuBERT | Her dancing and now she needs her feather to grab it with her foot as far as possible. |
| Ours | You dance, now you need the veil and grab the veil immediately. |

Table 26: ISLR on WLASL2000. P-I/P-C: per-instance / per-class Top-1 accuracy. External pretraining columns: Uni-Sign uses 1985 h CSL-News pretraining; SHuBERT 984 h YouTube-ASL.

| Method | P-I | P-C |
|---|---|---|
| SignBERT+ (Hu et al., 2023a) | 48.85 | 46.37 |
| NLA-SLR (Zuo et al., 2023) | 61.05 | 58.05 |
| SHuBERT (Gueuwou et al., 2025) | 60.90 | 58.01 |
| Uni-Sign (Li et al., 2025) | 63.52 | 61.32 |
| **Ours** | **76.14** | **74.82** |

without gloss annotations requires constructing a new ASL-style multi-word-sign vocabulary and re-tuning the reorder rules.