A Pose-Based Pipeline for Annotation of Headshakes in Sign Language Corpora

Gustaf Gren, Nikolaus Riemer Kankkonen
Stockholm University
gustaf.gren@ling.su.se, nikolaus.kankkonen@ling.su.se

Abstract

This paper introduces a pose-based pipeline designed to support scalable annotation of headshakes in sign language corpora. Motivated by the scarcity of annotated datasets and the need for quantitative typological research, the study evaluates whether automated detection can reduce human annotation effort. The system operates on yaw trajectories extracted with MediaPipe Holistic and uses sliding-window segmentation with neural sequence models (LSTM/CNN) to surface candidate segments for review. Training and evaluation are conducted on a subset of the German Sign Language (DGS) Corpus annotated to target grammatical headshakes functioning as negation rather than for every instance of headshakes. On the DGS dataset the best performing LSTM model achieves an F_2-score of 0.45, recall of 0.63. Despite the narrow annotation scope, the pipeline reduces search space: annotators need review only 13% of frames to recover 87% of labeled instances. Error analysis indicates that many false positives correspond to plausible head movements excluded by the annotation criteria. A pilot transfer to Swedish Sign Language shows reduced effectiveness without adaptation, underscoring the need for alignment in cross-lingual transfer scenarios.

Keywords: sign language, non-manual signs, non-manual sign detection, computational sign language, pose estimation, sign language resources

1. Introduction

Elements that convey linguistic meaning in sign languages without involving the hands are collectively known as nonmanual markers (Pfau and Quer, 2010). These markers can function either affectively (e.g. to express surprise, backchanneling) or grammatically (e.g. as an obligatory negation marker).

With respect to headshakes, Pfau (2015) argues that their grammaticalization as a negation marker originates from their use in spoken language as co-speech gestures expressing negation. Building on cross-linguistic variation, sign languages have been typologically divided into those that are manually dominant and those that are non-manually dominant for headshakes as a negation marker (Zeshan, 2004). Pfau (2015) adopts this distinction in his discussion. However, Johnston (2018) cautions against adopting this manual versus non-manual negation typology, based on the limited quantitative coverage of sign languages to test such typological claims.

Recent work in spoken language typology proposes treating features as gradient rather than binary, modeling them as continuous properties (Kann, 2025; Levshina et al., 2023). In the case of headshakes, this would entail examining the *distribution* of their grammatical use relative to their overall frequency in interaction, yielding a continuous value rather than a binary typological classification. The latter includes a range of discourse and interactional functions (e.g. backchanneling, emphasis, and other pragmatic uses). Investigating this degree of variation quantitatively requires corpus-based studies. Similar gradient-like approaches have been explored in sign language research, for example in corpus-based work on Turkish Sign Language (Makaroğlu, 2021).

Annotated data is scarce, in general, for many sign languages (De Sisto et al., 2022). Public datasets annotated for all instances of headshakes do not, to our knowledge, yet exist for any sign language. In this study, we investigate whether pose-based models can detect headshakes with sufficient recall to assist human annotation for sign language, with the goal of aiding broader data-driven typological investigations.

We frame headshake detection as an annotation-support task rather than a pure classification problem. To evaluate this approach we use German Sign Language (DGS) data annotated for headshakes used as negation. We compare a simple zero-crossing baseline with two neural architectures: a bidirectional LSTM and a 1D CNN. Both models operate on yaw trajectories extracted using MediaPipe Holistic. Model performance is assessed using both aggregate metrics and a targeted error analysis to distinguish between failures arising from representational limitations and those attributable to annotation. Lastly, we perform a pilot study on cross-lingustic transfer. We annotate 50 minutes of Swedish Sign Language data and evaluate the models trained for DGS without additional training.

Our contributions are:

- **A practical evaluation framework for non-manual annotation support.** We introduce an annotation-centric evaluation based on re-

Proceedings of the 12th Workshop on the Representation and Processing of Sign Languages, pages 193–202
16 May 2026.

view rate (the fraction of frames requiring human inspection) and demonstrate how recall trades off against annotation workload.

- **A pose-based pipeline for headshake candidate surfacing.** Using yaw and its temporal derivatives extracted with MediaPipe Holistic, we compare a zero-crossing baseline to BiLSTM and 1D CNN models, establishing performance bounds on DGS headshake-as-negation labels.

- **Analysis of failure modes under label scope and domain shift.** Confidence-stratified manual inspection reveals systematic discrepancies between model predictions and annotation targets, and a zero-adaptation pilot transfer to Swedish Sign Language shows degradation without calibration.

2. Background

Human pose estimation models identify human body parts and build representations such as body or hand skeletons, or facial mesh, from images or videos (Zheng et al., 2023). Because sign languages rely on coordinated facial, manual, and bodily movements, these models offer a way around the dimensionality challenges of raw video. They reduce the input from a time series of pixel matrices to a time series of keypoints corresponding to specific body parts (e.g. left pinky knuckle, left elbow). Pose estimation models have been used extensively in sign language research. For example, they have been applied to automatically identify whether someone is signing (Moryossef et al., 2020) and to recognize fingerspelling in ASL (Shin et al., 2021).

The number of keypoints depends on the model. Some pose estimation models have less granularity, like BlazePose (Bazarevsky et al., 2020) which produces 33 body keypoints in total, in order to focus on speed. Others have many more. MediaPipe Holistic (Lugaresi et al., 2019), comparatively, can detect up to 468 keypoints just for the face.

Computational studies of headshakes within sign language have mostly focused on their phonetics and distribution. Chizhikova and Kimmelman (2022) studied the phonetics of headshakes in Russian Sign Language utilizing OpenFace (Baltrusaitis et al., 2018), showing that negative headshakes are optional and relatively rare in Russian Sign Language, appearing in under 30% of negative sentences, and typically consist of one or two small, fast head turns closely aligned with manual negation signs. Kimmelman et al. (2024) also used OpenFace for the Sign Language of the Netherlands, to analyze whether linguistic properties of headshake influence its phonetic form. These studies demonstrate the feasibility of pose-based phonetic analysis but still rely on manual annotation.

The closest related work is Rijbroek (2023), who evaluated automatic headshake detection for the Sign Language of the Netherlands utilizing OpenPose (Cao et al., 2017), achieving the best results using a Hidden Markov Model (F_1-score of 0.19, precision of 0.12 and recall of 0.54).

3. Data

For training, we base our work on a subset of the DGS Corpus (Konrad et al., 2020), a publicly available collection of German Sign Language recordings from 330 signers. The released videos are provided at 360p resolution and 50 FPS. Headshake annotations were provided by Kimmelman et al. (in preparation) and target grammatical headshake usage in negative clauses rather than all headshakes. Consequently, the labels reflect a narrower linguistic definition than the detection objective used in this study. To better understand the impact of this label mismatch, we conduct an error analysis on model predictions (Section 4.4) to identify systematic inference patterns and assess how annotation discrepancies influences performance.

Data were split at the file level prior to windowing into 72% training, 8% development, and 20% testing sets to ensure that windows from the same recording appeared in only one partition, thereby minimizing temporal leakage from overlapping windows. While a small number of signers contributed multiple recordings, these instances were rare, limiting potential overfitting to signer-specific motion patterns. Because the split was not strictly signer-independent, residual signer effects cannot be ruled out. File-level splitting resulted in a small shift in the global headshake-to-non-headshake ratio (±0.003). The median headshake duration is 0.8 seconds.

To assess performance in transfer scenarios, approximately 50 minutes of video from the Swedish Sign Language Corpus (SSLC) (Öqvist et al., 2020) were manually annotated for headshake events. The annotation was carried out by a single annotator with prior experience working with the corpus. Headshakes were identified based on visible lateral head movements in a frontal view, with a practical threshold of approximately ± 10 degrees from neutral position. Headshake duration was measured from one side of the movement to the other, excluding the lead-in and settling phases. The annotated headshakes have a median duration of 0.4 seconds.

See Table 1 for the full data rundown.

	Files	Total	HS	non-HS	HS/Total
all	86	2883202	37543	2845659	0.0130
train	61	2030449	26490	2003959	0.0130
dev	7	297304	4915	292389	0.0165
test	18	555449	6138	549311	0.0111
sslc	10	82211	2841	79370	0.0345

Table 1: File/frame statistics of each split and headshake frames (HS) and non-headshake frames (non-HS), including the domain separate annotated SSLC test-set.

4. Methodology

We base our methodology on the pipeline of Pouw (2025), who performed end-to-end multi-gesture detection using a sliding window approach. The approach involves using Mediapipe Holistic (Lugaresi et al., 2019) to extract face, body, and hand kinematic features from video, normalizing the keypoints, and trains a Convolutional Neural Network (CNN) on labeled clips for multigesture detection. Most relevant to headshake detection is the features related to deriving Euler angles using rotation matrix decomposition formulas from the facial keypoints. There are three Euler angles:

- **Pitch**: Head tilting up/down (nodding)
- **Yaw**: Head turning left/right (shaking)
- **Roll**: Head tilting ear-to-shoulder

In practice, deriving Euler angles in this way gives us an array with elements describing where the head is pointing for each given frame where keypoints are found. See Figure 1 for an example, where Euler angles are illustrated using a stick pointing in the direction of the derived angles.

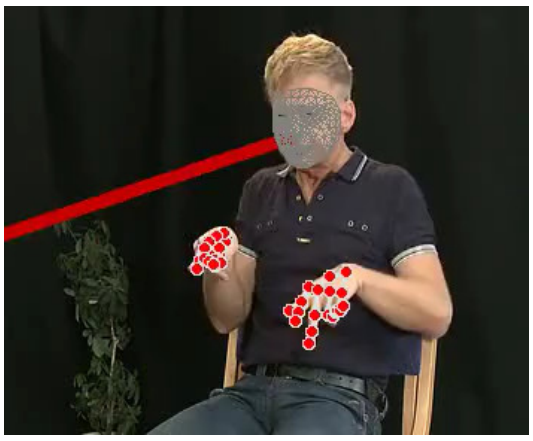


Figure 1: Example frame of video with face mask, hand keypoint estimations, and derived Euler angles from MediaPipe Holistic. Video still from the Swedish Sign Language Corpus.

Since we're doing headshake detection, we reduce dimensionality further, and only use the yaw values because headshakes is primarily horizontal rotation. Initially, we experimented using all three Euler angles, with poor results, with pitch and roll only adding noise. We also resample our video arrays to 25FPS. This is partly to reduce dimensionality further, but primarily to give an idea of future adaptability, given many sign language datasets are recorded at lower speeds, such as 30 or 25FPS. For example, the Swedish Sign Language Corpus (Öqvist et al., 2020) is recorded in 25FPS.

For each video clip, we derive keypoints for every frame. We use MediaPipe Holistic as the pose estimation model for our work because, while it has some limitations in estimating eyebrow movement (Kuznetsova and Kimmelman, 2024), Sargano et al. (2024) found it to be the most accurate among several pose estimation models for yaw estimation. If MediaPipe fails to find keypoints for a frame, then we set it to the same value as the previous frame. We do this instead of more advanced interpolation simply because there are few missing frames, and because it ensures a one-to-one mapping between the yaw array and the frame index. Continuous yaw sequences are segmented into sliding windows with stride. Stride is the number of frames you advance the start of each sliding window (e.g., with a window size of 75 and a stride of 5, windows start every 5 frames, so they overlap by 70 frames). Windows are normalized across clips. Each window $\tilde{y}_t$ is z-normalized using the clip's yaw mean μ_c and standard deviation σ_c:

$$\tilde{y}_t = \frac{y_t - \mu_c}{\sigma_c + \epsilon}$$

Where ϵ is a small constant to avoid division by zero.

4.1. Zero crossing baseline

As a simple baseline, we classify fixed-length yaw windows using their zero-crossing count. The hypothesis is that headshakes resemble a sinusoidal waveform, and should periodically cross 0. Assuming accurate pose estimation and a stable resting head position, 0 should represent the person looking straight ahead.

Given a windowed yaw signal $y_{1:T}$, we define the number of zero-crossings as

$$Z = \sum_{t=2}^{T} \mathbf{1}[\text{sign}(y_t) \neq \text{sign}(y_{t-1})]$$

A window is labeled positive during tuning if it contains at least one positive frame annotation. On the development set, we compute Z for all positive

windows and set a single decision threshold

$$k = \text{round}\left(\frac{1}{N_+}\sum_{i=1}^{N_+} Z_i\right)$$

where N_+ is the number of positive windows. At inference time, a window is classified as a headshake if $Z \geq k$, and negative otherwise.

4.2. Neural pipeline

Because this is a single-gesture detection task, we simplify the pipeline in Pouw (2025) and reduce it to use three features: **yaw angle**, **velocity** (derivative of the normalized yaw-array), **acceleration** (second derivative of the normalized yaw-array). Our training data then gets the shape $(B, W, 3)$, with B being our batch size, W our window size. In addition these input vectors are augmented during training, adding small random noise with a probability of 50% during training for each batch.

We test two types of neural models which we implement in `PyTorch`(Paszke et al., 2019)[1]: (i) a bidirectional LSTM with dropout of 0.2 or (ii) a 1D CNN with temporal convolutions. The LSTM was chosen for its ability to model temporal dependencies. It is also a suggestion for future research by Rijbroek (2023), but is to our knowledge still untested for headshake detection in sign languages. The CNN is kept as close to Pouw (2025) as possible. Both models produce per-frame logits for each window, and are trained with a weighted focal loss to address the heavy class imbalance.

A targeted hyperparameter search was performed. Using the LSTM as the target model, the search identified learning rate and window size as the dominant factors; other parameters had limited effect. Therefore we focused our CNN search on learning rate. This resulted in a learning rate of 1e-3 for the LSTM and 1e-4 for the CNN, and a window size of 75 and stride of 5 frames. Other parameters are kept identical between the CNN and LSTM: AdamW optimizer with weight decay set to 1e-5, FocalLoss with α=0.25;γ=2.0, and a ReduceLROnPlateau with a patience of 2 and reduction of 0.5.

We train both models for a maximum of ten epochs, while evaluating against the dev-set three times per epoch. The best checkpoint in regards to F_2 against the dev-set is used for the final evaluation against the test-set. Best checkpoint occurred at the third epoch for the LSTM, and fifth for the CNN.

Overlapping windows results in multiple predictions per frame (except a few in the beginning and at the end of a clip). Therefore, we need a temporal aggregation step for inference. We evaluate two strategies during training: mean pooling (mean probability across covering windows for a given frame), and max pooling (maximum probability across covering windows for a given frame).

4.3. Evaluation Metrics

We evaluate using precision, recall, F_1-scores, and F_2-scores. In annotation-support settings, false positives impose a small verification cost, whereas false negatives require full video inspection. Therefore, high recall with moderate precision can still produce efficiency gains. F_2-scores reflect this aim, which weighs the score towards recall. During training we set the classification threshold on the dev-set.

Let precision P and recall R be defined as

$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN}$$

where TP, FP, and FN denote true positives, false positives, and false negatives. The F$_\beta$ score is given by

$$F_\beta = (1 + \beta^2)\,\frac{PR}{\beta^2 P + R}$$

We use $\beta = 2$, yielding

$$F_2 = 5\,\frac{PR}{4P + R}$$

4.4. Error Analysis

We conduct error analysis to assess whether classification performance was driven by discrepancies between model goals and annotation goals, as well as uncover potential inference patterns. Predictions were categorized by model confidence **(high vs. low)** and correctness **(correct vs. incorrect)**.

Using the best-performing model, we extract 25 representative prediction windows from each category of prediction. For example, for the high confidence false positives we sort our false positives and take top k of the predictions by confidence level. We first turn our frame-level ground truth annotations into window-level and sort out windows where headshakes occur <50% of the frames in the window size, in order to avoid windows without much movement information. Otherwise we would have windows where, for example, the last two frames are headshakes. Such windows wouldn't be meaningful to humans. To derive the window-level predictions we use the same frame-level threshold set during training on the dev-set. These filtered window predictions are inspected manually for patterns.

[1]We tested utilizing a Hidden Markov Model similar to Rijbroek (2023), but it was rejected since it was difficult to make it work consistently with `PyTorch`.

4.5. Annotation Burden Analysis

To evaluate the feasibility of using the models for annotation work, we calculate the relationship between how many frames the annotator has to go through, compared to the headshakes correctly identified.

For each model we go through each window in the test-set and convert window predictions to per-frame probabilities by *mean* aggregation, taking the mean probability among all windows covering a frame. We then sweep classification thresholds from 0.0 to 1.0, converting probabilities into predicted frames. Finally, we add a dilation of 10 frames before and after predicted ranges to account for context frames that would be added during manual annotation.

Let F_{pred} be the number of predicted headshake frames after dilation, and F the total set of frames. We then calculate Review Rate (RR) and compare with recall:

$$RR = \frac{|F_{pred}|}{|F|}$$

For each model we calculate the best weigh-off point as

$$i^* = \arg\max_i (R_i - RR_i)$$

Where R_i is recall and RR_i is review rate at threshold i.

4.6. Domain Transfer

Finally, we investigate the effect of transferring the best-performing model to another sign language without additional training. We evaluate the annotated SSLC clips described in Section 3 in the same way as for the DGS data, using the models trained for the DGS data. The short length of headshakes in our small SSLC dataset makes a window analysis approach, as in Section 4.4, unfeasible. Instead, we present plots for all frames which aren't true negatives alongside their respective confidence scores in two randomly chosen clips.

5. Results

Neural approaches outperformed the baseline for headshake detection in DGS, which achieved very low precision despite moderate recall, resulting in poor overall F-scores. Among the tested architectures, LSTM-based models produced the strongest results, with the mean-pooled variant achieving the highest precision (0.21), recall (0.63), and F_2 score (0.45), while tying for the best F_1 score (0.31) with the max-pooled LSTM. CNN models showed competitive but lower performance, particularly in recall. See Table 2 for full metrics result table.

	P	**R**	F_1	F_2
Baseline	0.02	0.59	0.04	0.08
LSTM_{max}	0.20	0.61	**0.31**	0.43
CNN_{max}	0.16	0.53	0.24	0.36
$\text{LSTM}_{\text{mean}}$	**0.21**	**0.63**	**0.31**	**0.45**
CNN_{mean}	0.19	0.50	0.27	0.37

Table 2: Precision, recall, F_1, and F_2 for each model and pooling strategy on the DGS data. Threshold set at 0.9, calculated on dev-set.

5.1. Error Analysis

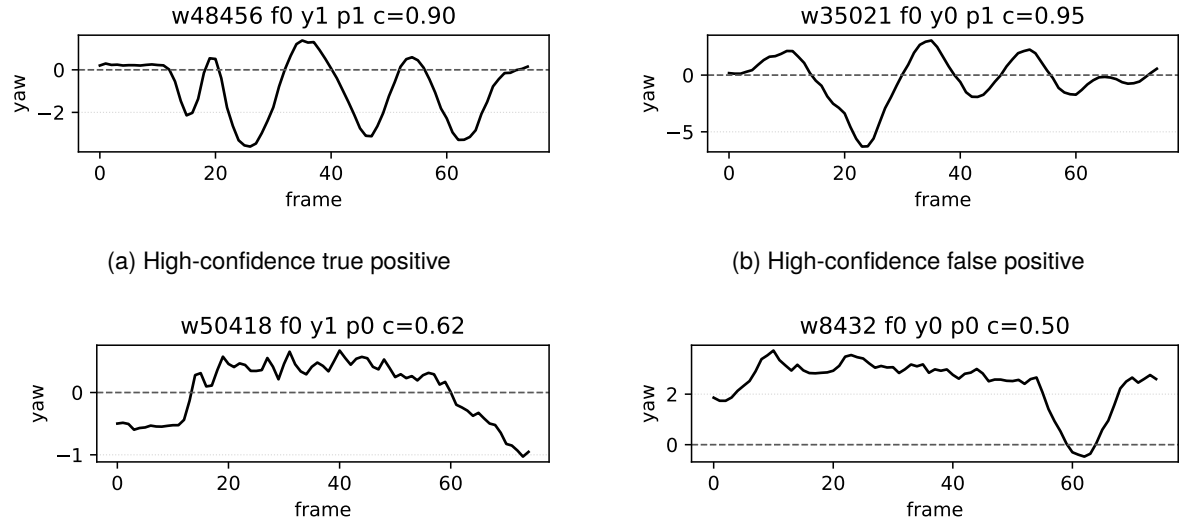
Figure 2 shows example yaw-angle trajectories for the four confidence/outcome categories. For the 25 most representative windows of each category, we generally observe:

- **High-confidence true positives:** Clear periodic waveforms with consistent oscillation between negative and positive yaw angles, matching expected headshake motion across all inspected windows.
- **High-confidence false positives:** Visually plausible headshakes, though the motion is not always centered around a neutral (zero) yaw position.
- **Low-confidence false negatives:** Two distinct patterns emerge in the 25 least confident windows for false negatives:
  1. *Subtle, high-frequency low-amplitude motion*. If you inspect the yaw-array (see Figure 2c) it appears this subtle nature is reflected in the array as well.
  2. *High deviation from resting position*. For example, one instance was observed where the subject had turned just before initiating the headshake. This made the window start highly off-center, even if the pattern was consistent with a headshake.
- **Low-confidence true negatives:** No consistent motion pattern, though they all contain motion. For example, a person with head movement in virtue of swaying left and right with the body.

5.2. Annotation Burden

The LSTM and CNN perform similarly when it comes to annotation burden metrics, requiring 13% review rate to achieve 87% recall, and 17% review rate to achieve 89% recall respectively. They both outperform the baseline, which has the best trade-off at a review rate of 36% to achieve 53% coverage.

The full plot is available in Figure 3.



(a) High-confidence true positive

(b) High-confidence false positive

(c) Low-confidence false negative

(d) Low-confidence true negative

Figure 2: Yaw angle trajectories for representative windows across prediction outcomes. .

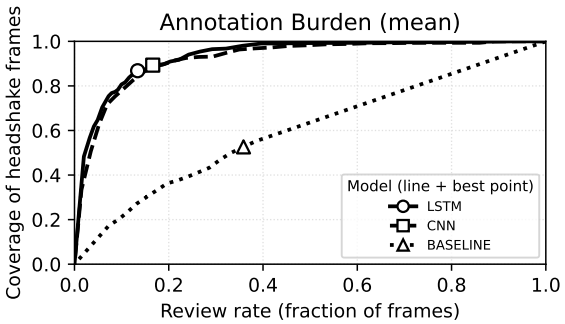


Figure 3: Proportion of required frame review rate of the test-set compared to headshake frame recall, utilizing mean inference pooling across frames.

	P	**R**	F_1	F_2
Baseline	0.05	**0.77**	0.09	0.18
LSTM_{max}	0.18	0.22	**0.20**	**0.21**
CNN_{max}	0.11	0.26	0.16	**0.21**
$\text{LSTM}_{\text{mean}}$	**0.36**	0.05	0.09	0.06
CNN_{mean}	0.14	0.11	0.12	0.11

Table 3: Precision, recall, F_1, and F_2 for each model and pooling strategy on the SSLC data.

5.3. Domain Transfer

The results get worse across all models except the baseline when testing on SSLC data. The baseline gets a recall of 0.77 and precision of 0.05, higher on both than for DGS. While the $\text{LSTM}_{\text{mean}}$ performed best for DGS, it here achieves very poor recall (0.05), though with high precision (0.36). Both neural models with max pooling perform best for F_2-scores.

We see this performance difference reflected in Figure 4, where many frames annotated as headshakes exhibit low confidence. This trend is further illustrated in the annotation burden plot for SSLC (Figure 5), which shows a less steep improvement as thresholds become more generous compared to DGS.

6. Discussion

Across models for the DGS data, results indicate that headshake detection is achievable with moderate recall but limited precision, suggesting that current pose-based approaches are better suited for assisting human annotation than for fully automated labeling. The most confident false positives during our error analysis correspond to genuine headshakes, in this case pointing to a mismatch between annotation targets (headshakes used as grammatical negation) and what our aim is (general headshake detection). This suggests the metrics above could potentially *under-represent* the models' true performance for general headshake detection, at least for DGS.

Even so, annotation-support systems do not require near-perfect classification performance to be valuable; their utility derives from reliably narrowing the search space for human annotators. The models appear capable of accelerating annotation for headshakes used as negation: the LSTM model requires annotators to review only 13% of the total frames to achieve 87% headshake coverage in DGS. However, this is not consistent with the small

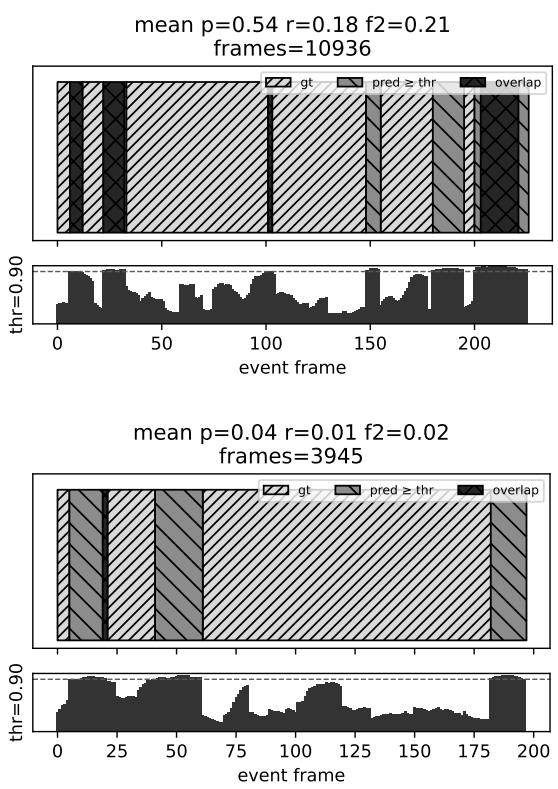


Figure 4: Event-only visualization of frame-level predictions for two SSLC clips using mean probabilities. For each clip, the upper panel shows contiguous segments of frames where the ground truth is positive but the prediction is negative (gt), the prediction exceeds the threshold but the ground truth is negative (pred $\geq$ thr), and frames where prediction and ground truth overlap (overlap). The lower panel shows the corresponding frame-level confidence scores for the same displayed frames, with the decision threshold indicated by the dashed horizontal line. True negatives are omitted; therefore, the x-axis is labeled "event frame" and represents only frames for which either the prediction or the ground-truth label is positive. The title above each plot reports clip-level precision, recall, F_2 score, and the total number of frames in the full clip.

transfer scenario we tested, where performance dropped, and therefore also the proposed annotation benefit. While both neural models still were more annotation efficient than the baseline, the gap between them narrowed. This might be because of several factors. Firstly, the length of the median headshake is vastly different from our two datasets. In DGS it is 0.8s, 0.4s for SSLC, which could be cross-lingual difference, or just as likely being due to the difference in annotation goals (i.e. having a different *definition* of a headshake). In a similar line of thought, it could be that the neural models overfit to other aspects of headshakes used as negation

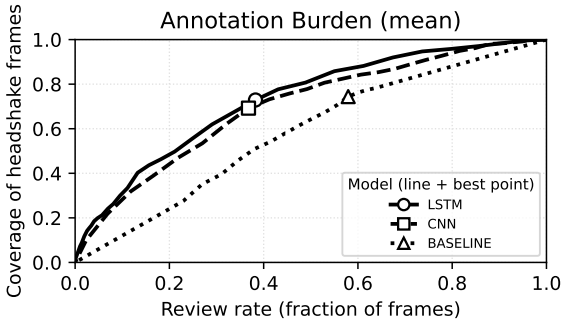


Figure 5: Proportion of required frame review rate of the SSLC data compared to headshake frame recall, utilizing mean inference pooling across frames.

in DGS; such as the amplitude and frequency, for example. One possibility is that headshakes associated with negation in DGS exhibit distinct phonological patterns that are distinct from headshakes more broadly. If so, the models may be capturing these phonological features rather than learning more generalizable representations. Should this interpretation hold, the most direct remedy would be the collection of additional, more diverse training data.

Insufficient tuning to the new domain could also be an explanation for the poor transfer performance. We tested against the best classification threshold from training (0.9). But we see in Figure 4 that many of the ground truth headshake frames would have been captured with a lower threshold. In fact, if we set the threshold to 0.5 we effectively "flip" the precision and recall, and get a precision of 0.11 and recall of 0.36 for the mean-pooled LSTM in the transfer scenario. While we can't tune the threshold like this automatically in a transfer scenario (since we would have no ground truth to tune it to), it does show the effect a little bit of human intervention can have on this task.

Also, a more detailed hyperparameter search could improve performance further; for example utilizing a more complex model design, attention, more layers. We used a simple feature set. Including more hand-crafted features, e.g. from the frequency domain (like Fast Fourier Transform), could be a better fit for modeling headshakes.

The results hint at several future research paths. Firstly, given that the high-confidence false positives predictions contained headshakes, introducing a human in the loop to manually re-label or enhance the DGS dataset for general headshakes seems feasible. A human in the loop could also alleviate the major performance drop seen during our testing of cross-lingual transfer. We could consider an active learning approach to maximize annotation efficiency and increase dataset size: querying the model for least confident negative/positive predic-

tions and manually labeling them. In other annotation workflows, this can drastically lower annotation time (Tharwat and Schenck, 2023).

7. Conclusion

This study demonstrates that pose-based models can support the automatic detection of grammatical headshakes in German Sign Language, with neural architectures outperforming a simple baseline and reducing annotation effort. While precision remains limited, error analysis shows that many predicted instances were consistent with headshake movements. The models narrow the search space for human annotators, showing their practical application for corpus construction and typological research. However, the sharp decline in cross-linguistic transfer underscores the sensitivity of such systems, suggesting that adaptation or human-in-the-loop strategies will be necessary for broader applicability in cross-lingual studies. Overall, the findings point to the promise of pose-driven pipelines as scalable tools for investigating headshakes in sign languages.

Limitations

Several limitations constrain the generalizability of these findings. First, the models rely solely on yaw-based features, potentially overlooking multimodal cues such as facial expressions or body movement that contribute to headshake interpretation. Second, annotations in the DGS corpus target grammatical negation rather than general headshakes, creating a label mismatch that complicates evaluation and may underestimate true performance. The dataset split is not fully signer-independent, leaving residual risk of overfitting to individual motion patterns. Cross-linguistic transfer results are based on a small Swedish dataset with differing headshake durations and definitions, limiting strong conclusions about generalization. Additionally, the models require threshold tuning and hyperparameter exploration that were only partially optimized.

Acknowledgments

This work has been funded by the Swedish national research infrastructure Språkbanken, in turn jointly funded by its 10 partner institutions and the Swedish Research Council (2025–28, grant 2023-00161). We thank the anonymous reviewers for their insightful comments, which helped improve the paper. Additionally, Marc Schulder for providing helpful comments about formatting for the final paper. We also thank Robert Östling for his input, as well as endless patience during discussions.

Code Availability

Code is publicly available at:
https://github.com/skogsgren/shakepose

8. Bibliographical References

Tadas Baltrusaitis, Amir Zadeh, Yao Chong Lim, and Louis-Philippe Morency. 2018. OpenFace 2.0: Facial behavior analysis toolkit. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 59–66.

Valentin Bazarevsky, Ivan Grishchenko, Karthik Raveendran, Tyler Zhu, Fan Zhang, and Matthias Grundmann. 2020. BlazePose: On-device real-time body pose tracking.

Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Anastasia Chizhikova and Vadim Kimmelman. 2022. Phonetics of negative headshake in Russian Sign Language: A small-scale corpus study. In *Proceedings of the LREC2022 10th Workshop on the Representation and Processing of Sign Languages: Multilingual Sign Language Resources*, pages 29–36, Marseille, France. European Language Resources Association.

Mirella De Sisto, Vincent Vandeghinste, Santiago Egea Gómez, Mathieu De Coster, Dimitar Shterionov, and Horacio Saggion. 2022. Challenges with Sign Language datasets for Sign Language recognition and translation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2478–2487, Marseille, France. European Language Resources Association.

Trevor Johnston. 2018. A corpus-based study of the role of headshaking in negation in Auslan (Australian Sign Language): Implications for signed language typology. *Linguistic Typology*, 22(2):185–231.

Amanda Kann. 2025. Are translated texts useful for gradient word order extraction? In *Proceedings of the 7th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 177–182, Vienna, Austria. Association for Computational Linguistics.

Vadim Kimmelman, Anastasia Bauer, Carl Börstell, Jan Bulla, Allah Bux, Laurence Crettenand, Lorena Figueiredo, Anna Kuder, Hannah Lutzenberger, Marloes Oomen, and Roland Pfau. in preparation. Kinematics of negative headshake in seven Sign Languages. Unpublished manuscript, available upon request.

Vadim Kimmelman, Marloes Oomen, and Roland Pfau. 2024. Headshakes in NGT: Relation between phonetic properties & linguistic functions. In *Proceedings of the LREC-COLING 2024 11th Workshop on the Representation and Processing of Sign Languages: Evaluation of Sign Language Resources*, pages 159–167, Torino, Italia. ELRA and ICCL.

Anna Kuznetsova and Vadim Kimmelman. 2024. Testing MediaPipe holistic for linguistic analysis of nonmanual markers in Sign Languages. Preprint. arxiv:2403.10367.

Natalia Levshina, Savithry Namboodiripad, Marc Allassonnière-Tang, Mathew Kramer, Luigi Talamo, Annemarie Verkerk, Sasha Wilmoth, Gabriela Garrido Rodriguez, Timothy Michael Gupton, Evan Kidd, Zoey Liu, Chiara Naccarato, Rachel Nordlinger, Anastasia Panova, and Natalia Stoynova. 2023. Why we need a gradient approach to word order. *Linguistics*, 61(4):825–883.

Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. 2019. MediaPipe: A framework for perceiving and processing reality. In *Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR) 2019*.

Bahtiyar Makaroğlu. 2021. A corpus-based typology of negation strategies in Turkish Sign Language. *Dilbilim Araştırmaları Dergisi*, 32(2):111–147.

Amit Moryossef, Ioannis Tsochantaridis, Roee Aharoni, Sarah Ebling, and Srini Narayanan. 2020. Real-time Sign Language detection using human pose estimation. In *European Conference on Computer Vision*, pages 237–248. Springer.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Roland Pfau. 2015. The grammaticalization of headshakes: From head movement to negative head. In Andrew D.M. Smith, Graeme Trousdale, and Richard Waltereit, editors, *New Directions in Grammaticalization Research*, pages 9–50. John Benjamins Publishing Company.

Roland Pfau and Josep Quer. 2010. Nonmanuals: their grammatical and prosodic roles. In Diane Brentari, editor, *Sign Languages*, Cambridge Language Surveys, pages 381–402. Cambridge University Press.

Wim Pouw. 2025. EnvisionBox: End-to-end human behavioral classification using Convolutional Neural Networks. Software. GitHub repository. Commit 4f01dd6.

Cas van Rijbroek. 2023. Automatically detecting head-shakes in NGT conversations. Master's thesis, Radboud University Nijmegen.

Allah Bux Sargano, Sébastien Vandenitte, Tommi Jantunen, and Vadim Kimmelman. 2024. Evaluation of head pose estimation algorithms for Sign Language analysis. In *2024 International Conference on IT and Industrial Technologies (ICIT)*, pages 1–6. IEEE.

Jungpil Shin, Akitaka Matsuoka, Md. Al Mehedi Hasan, and Azmain Yakin Srizon. 2021. American Sign Language alphabet recognition by extracting feature from hand pose estimation. *Sensors*, 21(17).

Alaa Tharwat and Wolfram Schenck. 2023. A survey on active learning: State-of-the-art, practical challenges and research directions. *Mathematics*, 11(4):820.

Ulrike Zeshan. 2004. Hand, head, and face: Negative constructions in Sign Languages. *Linguistic Typology*, 8(1):1–58.

Ce Zheng, Wenhan Wu, Chen Chen, Taojiannan Yang, Sijie Zhu, Ju Shen, Nasser Kehtarnavaz, and Mubarak Shah. 2023. Deep learning-based human pose estimation: A survey. *ACM Comput. Surv.*, 56(1).

9. Language Resource References

Reiner Konrad, Thomas Hanke, Gabriele Langer, Dolly Blanck, Julian Bleicken, Ilona Hofmann, Olga Jeziorski, Lutz König, Susanne König, Rie Nishio, Anja Regen, Uta Salden, Sven Wagner,

Satu Worseck, Oliver Böse, Elena Jahn, and Marc Schulder. 2020. MEINE DGS – annotiert. Öffentliches Korpus der Deutschen Gebä rdensprache, 3. Release / MY DGS – annotated. Public Corpus of German Sign Language, 3rd release. Dataset. Universität Hamburg.

Zrajm Öqvist, Nikolaus Riemer Kankkonen, and Johanna Mesch. 2020. STS-korpus: A Sign Language web corpus tool for teaching and public use. In *Proceedings of the LREC2020 9th Workshop on the Representation and Processing of Sign Languages: Sign Language Resources in the Service of the Language Community, Technological Challenges and Application Perspectives*, pages 177–180, Marseille, France. European Language Resources Association (ELRA).