Evaluation of Co-Speech Gesture Tracking Techniques in Naturalistic Interactions

Victoria Ivanova, Naomi Harte
Sigmedia Group, School of Engineering, Trinity College Dublin
{vivanova, nharte}@tcd.ie

Abstract

Hand gestures convey a significant portion of communicative meaning, making multimodal datasets essential for interaction research. However, annotating gestures remains a time-consuming and challenging task. To speed up the process, semi-automatic methods have been developed that identify segments with hand movement for annotators to refine. These typically combine a pose estimation model with a rule-based or statistical movement detection algorithm. However, most are validated on idealised, non-naturalistic datasets with minimal hand occlusions. We benchmark combinations of four pose estimation methods (OpenPose, MediaPipe, DeepLabCut, and Kinect) and two rule-based movement detection algorithms on two naturalistic, conversational datasets. The best pipelines combine the SPUDNIG displacement algorithm with OpenPose on MULTISIMO and with DeepLabCut on ECOLANG. These pipelines achieved Tversky scores of 0.57 on MULTISIMO and 0.65 on ECOLANG, with recall scores of 0.73 and 0.78, respectively. While off-the-shelf gesture detection systems can support annotation, performance remains limited on naturalistic data, and careful camera setup minimizing occlusions is essential.

Keywords: gestures, annotation, evaluation

1. Introduction

Interaction is a multimodal phenomenon – we convey information not only through speech but also through body posture, eye gaze, facial expressions and hand gestures. An estimated 50–70% of the information is encoded solely or jointly through hand gestures (Holler et al., 2018). Gestures are studied not only to better understand communication, but also because they provide helpful cues for speech recognition models (Kim et al., 2025) and turn-taking models (Kendrick et al., 2023).

Advancing gesture research depends on large, reliably annotated datasets, but annotation is labour-intensive and time-consuming. The effort varies with the number of annotators, annotation detail, scheme, and factors such as video quality, hand occlusions, and object handling (Grondin-Verdon et al., 2024). To reduce the burden, various systems have been developed that leverage pose estimation models and either machine learning or rule-based hand movement detection methods. Currently, no existing methods can distinguish between sporadic movements and meaningful gestures with high accuracy (Pouw et al., 2025). However, as we review in Section 2, some have had success in identifying movement peaks, onsets and offsets, as well as gesture phases to an extent. Often, systems are evaluated on isolated gestures (Benitez-Garcia et al., 2021) or ideal-angle videos (Ripperda et al., 2020). In the absence of a dedicated benchmark dataset (Pouw et al., 2025), no prior work has evaluated multiple methods on the same data. This paper offers a systematic benchmarking of semi-automatic gesture-detection methods on naturalistic, multimodal interaction data rich in co-speech gestures.

By performing this evaluation, we aim to answer the question: **Can off-the-shelf gesture detection systems reliably support the annotation of large-scale, naturalistic, multimodal interaction data?**

To answer this question, we leverage four pose estimation techniques – OpenPose, MediaPipe, DeepLabCut and Microsoft Kinect – in combination with two different rule-based movement detection methods, allowing systematic comparison of their performance on two different datasets – MULTISIMO (Koutsombogera and Vogel, 2018) and ECOLANG (Gu et al., 2025).

We review the challenges of annotation and solutions in Section 2. In Section 3, we describe the evaluated pipelines by first summarizing the coordinate estimation process, followed by the hand movement extraction methods. We present the results in Section 4. In Section 5, we conclude that these methods struggle with naturalistic data, particularly in the case of hand occlusions. To obtain the most benefit, filming setups, even in naturalistic settings, should be strategically designed.

2. Gesture annotation challenges and approaches

2.1. Annotation challenges and multimodal corpora

Despite the increasing interest in co-speech gesture analysis (Lai et al., 2024; Trujillo, 2024; Bar-

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 6277–6288
11–16 May 2026.

ros et al., 2024), there are currently no publicly available datasets that provide gesture segment annotations that were generated fully or semi-automatically, without direct human intervention.

Previous studies have reported on the considerable burden associated with capturing co-speech gestures in multimodal data. For example, Ripperda et al. (2020) report 35.4 minutes per 2-minute video on average for manual annotation, while Benitez-Garcia et al. (2025) note 70 minutes to annotate a 2.3 minute long video of non-speech gestures. Beugher et al. (2018) estimate that annotation time ranges from 10 to 50 minutes per minute of video, depending on the level of detail required. The difficulty of annotating multimodal corpora often limits the dissemination of gesture datasets and contributes to limited availability of high-quality data for researchers (Grondin-Verdon et al., 2024). Most existing datasets are tailored to specific tasks and lack ecological validity. For instance, datasets such as IPN HandS (Benitez-Garcia et al., 2021) and SHREC (Caputo et al., 2021) contain isolated, non-co-speech gestures drawn from predefined vocabularies, while EgoGesture (Zhang et al., 2018) contains egocentric recordings designed for first-person gesture recognition.

Manual annotation typically involves labelling gesture types or identifying phases of movement. Unlike isolated gestures such as *grab*, *swipe*, and *pinch*, which are common in human-computer interaction contexts (De Smedt et al., 2016), co-speech gestures are defined by their tight relationship with speech prosody and semantic content. They are often categorised into four types: iconic (mimicking typing when speaking about a *computer*), deictic (pointing to something referred to in speech), metaphoric (mimicking holding a heavy object to signify *responsibility*), and beat gestures (simple strokes that follow the rhythm of speech) (McNeill, 1992). Hand movements such as adjusting eyeglasses or hair are known as self-adaptors and are not considered part of co-speech gestures (Sekine and Hotta, 2025). Another common level of detail involves segmenting gestures into phases such as preparation, stroke (the meaningful part of the gesture), hold, and retraction (Kendon, 1980). These annotations follow established protocols (Allwood et al., 2007; Carletta et al., 2003; Bressem et al., 2013), typically involving multiple annotators and requiring high inter-rater agreement (Holle and Rein, 2015).

However, it is widely acknowledged that such pragmatic and semantic information, i.e. gesture types or phases, cannot be reliably extracted from motion data alone (Trujillo, 2024). Therefore, the goal of contemporary gesture detection systems is to determine intervals of hand movement, to which a researcher can later manually add information such as gesture type or phase. These intervals may also include movements such as self-adaptors that cannot be distinguished from co-speech gestures by automatic systems and therefore must be manually labelled.

In this paper, we distinguish between several related concepts. *Pose estimation* (or body tracking) refers to the extraction of body part coordinates from video frames using systems such as OpenPose, MediaPipe, DeepLabCut, or Kinect. Based on these pose trajectories, our pipelines perform *gesture detection*, that is, identifying time intervals in which the hands of the speaker are in motion. This differs from *gesture annotation* in the traditional multimodal corpus sense, where gesture segments are assigned semantic labels (iconic, deictic, metaphoric) or phases (preparation, stroke, hold). The present work therefore evaluates automatic gesture detection rather than semantic gesture annotation.

To evaluate the performance of our chosen gesture detection methods, we test their performance against human-labelled data, which we consider ground truth. To do so, we sought manually annotated, multimodal, conversational co-speech datasets. Some datasets in this category include MUNDEX (Schade et al., 2024) and SaGA (Lücking et al., 2013). Based on the availability of the data, its ecological validity and the presence of Kinect data, we selected MULTISIMO and ECOLANG (Koutsombogera and Vogel, 2018; Gu et al., 2025).

The MULTISIMO corpus consists of triadic conversations in a semi-naturalistic environment and focuses on collaborative multimodal behaviour in group settings. Crucially for our purposes, the corpus includes recordings from the Kinect v2 sensor. The ECOLANG dataset also consists of recordings of multimodal interaction, but in a dyadic setting and it includes annotations of object manipulation by the speakers.

2.2. Pose estimation methods

The process of automatic gesture detection begins with using a pose estimation technique to establish the position of the relevant body parts in each frame. These techniques are divided into markered and markerless (Trujillo, 2024).

Markered tracking methods involve attaching physical sensors or markers to the body, such as electromagnetic devices, accelerometers, or gloves worn on the hands and wrists (Jiang et al., 2017; Qian et al., 2025). For example, the BEAT dataset (Liu et al., 2022), developed for gesture synthesis, uses full-body motion capture suits to achieve precise tracking. However, such setups

require a controlled studio environment and restrict natural interaction, making them impractical for most interaction research scenarios that aim for ecological validity. Given these limitations, markered methods fall outside the scope of the present study.

In contrast, markerless pose estimation methods—such as OpenPose, MediaPipe, and DeepLabCut—apply computer vision models to standard RGB video to detect human body keypoints. These methods are widely adopted due to their flexibility and low setup cost (Trujillo, 2024). OpenPose is an open-source framework for multi-person 2D pose estimation, developed by Cao et al. (2019). It predicts 2D coordinates of keypoints for the full human body, face, hands, and feet from RGB video input. The method is based on Part Affinity Fields (PAFs), which encode both the location and orientation of body parts, allowing for robust multi-person detection even under some occlusion. Nonetheless, strong occlusions can still lead to mislabelled keypoints. Similar to OpenPose, DeepLabCut (Mathis et al., 2018) is another open-source 2D pose estimation system. Originally designed for animal pose estimation, it has since been used successfully for human hand and body movement studies (Nath et al., 2019; Panconi et al., 2025). It is based on deep convolution networks and the toolbox provides some pretrained models for animal pose tracking. Unlike OpenPose, the setup requires the user to annotate a small set of frames needed for the fine-tuning of a backbone network. The fine-tuning with the annotated frames leads to a potentially more robust performance in specific scenarios than OpenPose.

In addition to these, we also evaluate MediaPipe, a lightweight, real-time framework for pose, hand, and face tracking developed by Google (Kim et al., 2023). MediaPipe is designed for high performance on both desktop and mobile platforms, making it highly accessible. It has been widely used in gesture studies, often paired with machine learning movement detection methods (Biswas et al., 2024; PDF, 2024).

A second class of markerless methods utilise specialised infrared emitter devices and depth sensors to capture depth data before applying computer vision models. While more sensitive to occlusion and body orientation, they are less affected by lighting conditions. We evaluate Microsoft Kinect v2 – a widely used device, due to its low cost and portability (Cai et al., 2019). Originally developed for gesture-based gaming, it has since been adopted in research areas such as biomechanics (Kurillo et al., 2022) and gesture studies (Trujillo et al., 2021). It generates real-time depth maps using time-of-flight infrared sensing and predicts body part positions via a deep randomised decision forest classifier (Shotton et al., 2011).

2.3. Detection methods

After obtaining the coordinates of the hands, the next step is to determine where movement occurs. Broadly, movement detection methods fall into two categories: rule-based and machine learning-based approaches.

As an example of a machine learning method, Benitez-Garcia et al. (2025) compare a lightweight LSTM model against MediaPipe Hands, reporting 90.1% accuracy and 67.1% recall. However, this evaluation is based on the IPN HandS dataset (Benitez-Garcia et al., 2021), which contains only isolated, non-speech gestures such as ”zoom in” and ”click.” As such, it does not reflect the complexity of naturalistic, co-speech interactions.

A more domain-specific approach is presented by Ghaleb et al. (2024), who frame co-speech gesture detection as a multi-phase sequence labelling task, distinguishing between preparation, stroke, and retraction phases. Their model combines a spatio-temporal graph convolutional network, a Transformer encoder, and Conditional Random Fields, and is trained on manually annotated naturalistic data. While this architecture achieves an F1 score of 58.4 for stroke detection, performance remains moderate, particularly for the transitional phases.

Pouw et al. (2025) propose a method for distinguishing between sporadic hand movements and meaningful gestures. Using coordinates from MediaPipe Hands and a CNN trained on five multimodal datasets (among which MULTISIMO and ECOLANG), they report gesture detection accuracies ranging from 73% to 78%. However, like other machine learning-based systems, this model shows limited generalizability to new data sources and contexts.

As these examples show, current machine learning-based approaches have yet to achieve consistently strong results for gesture detection. More importantly, they often struggle to generalise across datasets or recording conditions, particularly when trained on narrowly defined or isolated gesture types (Pouw et al., 2025). In contrast, rule-based methods offer a simpler, more transparent alternative, and have shown promising results in supporting gesture annotation.

For example, SPUDNIG (Ripperda et al., 2020) is a software tool designed to assist annotators by automatically detecting stretches of hand movement based on OpenPose-derived keypoints and a set of rule-based heuristics. It is a user-friendly, off-the-shelf application with a graphical user interface (GUI), allowing users to select a video for analysis and specify a confidence threshold for Open-

Pose's keypoint predictions. The movement detection algorithm assigns a gesture segment whenever selected hand keypoints (wrists, elbows, index finger, and thumb) move away from an adaptive rest position for a sustained period and then return to rest. Keypoints are taken into account if the model has predicted their position with a certain confidence. SPUDNIG merges detected movement segments across all tracked keypoints to produce a continuous gesture timeline. Another method that employs a similar pixel displacement algorithm is described by Beugher et al. (2018). Notably, it does not employ any of the previously described pose estimation techniques; instead, it relies on subject-specific skin-colour modelling to localise hands, which makes it particularly susceptible to illumination changes, occlusions, and skin-like backgrounds.

Another method to detect movements from pose estimation coordinates is through peaks in velocity. It provides a straightforward way to detect changes in position and has been previously proposed by Trujillo et al. (2019), where it was validated on a limited set of isolated, non-naturalistic gestures. Peak velocity, in particular, has been associated with the moment of maximum effort in a gesture and correlated with prosodic features such as F0 peaks (Pouw and Dixon, 2019). It has been reported as a useful feature in gesture detection, specifically in the context of Kinect (Elgendi et al., 2012).

2.4. Task and contribution

The task considered in this study is automatic gesture detection, defined here as the identification of time intervals in which hand movement occurs in conversational video data. These automatically detected intervals are intended to support human annotation by narrowing down the parts of the recording that are likely to contain gestures. To assess how well existing systems support this task, we evaluate combinations of pose estimation and movement detection methods on two naturalistic multimodal corpora, MULTISIMO and ECOLANG, using human-labelled gesture segments as ground truth. By comparing multiple pipelines on the same data, this work provides a systematic benchmark of off-the-shelf gesture detection approaches on conversational interaction data, highlighting their strengths and limitations in realistic research settings.

In total, we evaluate four techniques for pose estimation (OpenPose, DeepLabCut, MediaPipe, and Kinect) in conjunction with two movement detection algorithms (SPUDNIG and velocity peaks). The details of the evaluations are outlined in the next section.

3. Methodology

3.1. Data processing & pose estimation

We focus on the setup for each dataset in turn in the next two sections.

3.1.1. MULTISIMO

MULTISIMO is an English audiovisual corpus. Each of the 23 recorded sessions of the MULTISIMO corpus (average duration: 10 minutes) features two participants working together to solve a quiz, assisted by a facilitator. Each session yields three videos, with each video focusing on one of the participants. The sessions are fully transcribed and annotated with information about speech, speaking turns, manual gestures, and gaze direction by two expert annotators. The gaze and gesture labelling is done following the MUMIN annotation scheme (Allwood et al., 2007). Additional metadata includes emotional state annotations and self-assessed personality profiles of the participants.

The first AI pose estimation model we used to obtain 2D body and hand keypoint predictions was OpenPose (BODY25 model). OpenPose outputs 25 body keypoints per person, including the elbows, wrists, index fingers, and thumbs. We extracted one keypoint per joint: the wrist, elbow, tip of the thumb, and tip of the index finger, which were used as inputs to our movement detection algorithms. OpenPose was run frame-by-frame using the official Python API with default parameters.

Next, we used MediaPipe Hands, which was applied using the default configuration in the MediaPipe Python API. For each detected hand, the model outputs 21 2D landmarks. We used the wrist, tip of the thumb, and tip of the index finger.

Additionally, we used DeepLabCut (DLC) for pose estimation, which requires manual annotation of a small subset of frames for training. To ensure frame diversity, we employed a k-means-based method for frame selection and labelled 0.06% of the total frames (520 in total). Of these, 95% were used for training and 5% for testing.

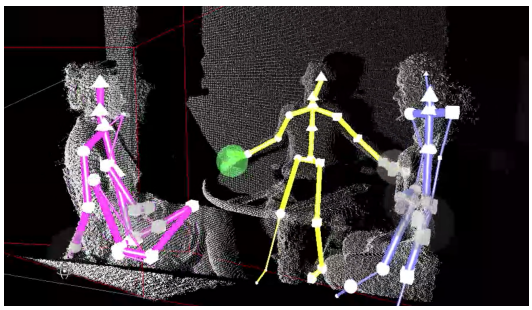

Figure 1: A shot from the Kinect data.

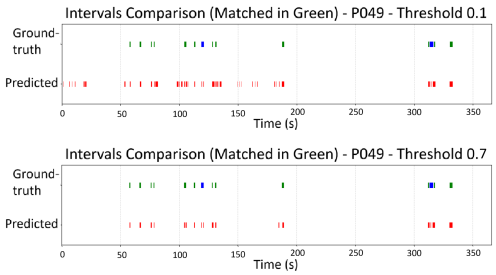

Figure 2: Ground truth (top) vs. OpenPose–SPUDNIG predictions (bottom) for MULTISIMO part. P049, confidence thresholds 0.1 and 0.7.

Given that participants often rest their hands on a table—leading to frequent occlusions—we placed estimated keypoints in manually annotated frames where necessary. This approach is supported by prior work showing DLC's ability to track occluded hands in sleight-of-hand contexts (Zaghi-Lara et al., 2019).

For training, we used a ResNet-101-based model (Insafutdinov et al., 2016) with default parameters and trained for 200 iterations. The best model (at 110 iterations) was selected based on validation performance. We applied the default p-cutoff of 0.6 to filter out low-confidence predictions during inference.

Preprocessing the Microsoft Kinect data required parsing a specialised file format, trimming and synchronizing recordings with the corresponding video and audio tracks, and identifying participants within each session. We also ensured consistent tracking coverage for each participant throughout the duration of the Kinect recordings (see Figure 1).

Due to these requirements, only a subset of the recordings could be retained. The final dataset used for analysis consists of 38 videos, with a total duration of 5 hours and 34 minutes (mean duration: 8 minutes 48 seconds; SD = 2 minutes 33 seconds). These videos include data from 29 unique participants, as facilitators often appear in multiple sessions.

A key limitation of the Kinect v2 data is that each session was recorded from a single device capturing all three participants simultaneously. In contrast, the standard video recordings used with the other pose estimation methods (OpenPose, MediaPipe, DeepLabCut) are camera views focused on individual participants. While Kinect's infrared depth camera provides 3D positional data, its performance is significantly affected by occlusions—particularly from tables and other participants—which frequently leads to noisy and unstable tracking. To mitigate this, we applied smoothing using a rolling mean filter with a window size of 30 frames.

After keypoint extraction, we apply the movement-detection methods from Section 3.2.

3.1.2. ECOLANG

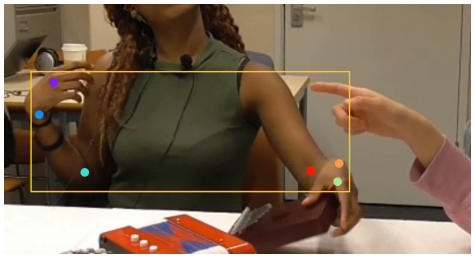

Figure 3: The top-down approach defining an area of interest, avoiding other hands

ECOLANG is also an English multimodal corpus. Unlike MULTISIMO, the ECOLANG task involves frequent interaction with physical objects, often leading to hand occlusion and more complex gesture contexts. The participants are British and American English-speaking adults interacting in semi-naturalistic conversations, either with their child (38 videos) or with a familiar adult (31 videos). Each session follows a structured information-transfer task: the speaker is required to convey previously learned information about four different sets of objects, across two distinct phases—one with the objects physically present and one without. Although each video was initially annotated by at least seven crowd-sourced annotators, low inter-rater agreement led to the appointment of an expert annotator to produce the final annotations.

To maintain consistency and reduce interference from other agents in the scene, we focused exclusively on the adult-oriented recordings and further narrowed our selection to sessions with minimal listener hand movement. This filtering reduced the dataset to 88 videos (eight per speaker), totaling 6 hours and 35 minutes of footage (mean duration: 4 minutes 32 seconds; SD = 58.15 seconds). From this pool, 13 randomly selected videos were reserved for the validation set.

Each session was divided into eight segments corresponding to the four object sets and the two presence conditions (objects present vs. absent). We cropped the video frames to primarily include the speaker, minimizing visual interference from the listener.

The application of OpenPose and MediaPipe was carried out following the same procedure as for the MULTISIMO dataset. For DeepLabCut (DLC), we followed the same training protocol as with MULTISIMO, manually annotating approximately 0.06% of frames. Of these, 95% were used for training and 5% for testing. We again used a ResNet-101-based network with default parameters. To handle frequent occlusions

of hands, wrists, or elbows—caused by object manipulation—we applied the same strategy as in MULTISIMO, placing estimated markers at expected keypoint locations when necessary.

However, to address high error from hand intrusions, we adopted a top-down approach that automatically defines a speaker region of interest (ROI) (Figure 3), retraining for 400 iterations, with the model from iteration 370 selected for evaluation.

3.2. Movement detection

After obtaining the coordinates of the hands, the next step is to determine where movement occurs. We use two rule-based approaches: a displacement-based method following SPUDNIG and a velocity peak detection method.

3.2.1. SPUDNIG algorithm

We adapt the displacement algorithm of SPUDNIG in order to apply it to the outputs of all pose estimation methods we use. We retained SPUDNIG's original settings for frame-to-frame displacement thresholds, which determine whether movement is occurring. These defaults are not configurable in the off-the-shelf tool, so we treat them as representative of a typical user setup. For pixel-based coordinate systems (OpenPose, DLC, and MediaPipe), the thresholds were set to 10 pixels for rest, 8 pixels for stillness, and 5 pixels for movement onset. When applying the logic to Kinect's 3D coordinates (expressed in meters), equivalent thresholds of 0.015 m, 0.02 m, and 0.035 m were used, respectively.

We tuned the confidence threshold on a validation set (N = 6 videos for MULTISIMO, N = 8 for ECOLANG). This threshold defines which keypoint coordinates are considered by the algorithm, based on the confidence scores assigned by the pose estimation models. Example outputs at low vs. high thresholds for MULTISIMO are shown in Figure 2.

3.2.2. Velocity peaks algorithm

Our second movement detection algorithm is based on determining peaks in velocity in the output coordinates. Our implementation calculates the velocity of each keypoint and identifies local maxima using peak detection based on minimum distance and prominence thresholds. To reduce noise, we retain only peaks whose amplitudes exceed a percentile threshold, determined empirically using a validation set. Gesture segments are then defined by expanding symmetrically from each peak until the signal drops below 60% of the peak amplitude (a drop ratio of 0.6), and by enforcing a minimum segment duration of 7.5 frames to eliminate spurious detections.

We used the validation set to determine the percentile threshold above the noise floor at which to detect peaks for each dataset, defining how prominent a movement must be to be considered as a possible gesture. Velocities were min–max normalised to $[0, 1]$ at the dataset level, using global minima and maxima computed across all files; the percentile threshold was then applied per file to the normalised signal.

For Kinect data, we performed an additional smoothing step after velocity calculation, as the data remained particularly noisy due to depth distortion and occlusions. A Savitzky–Golay filter (*window length:* 31, *polyorder:* 2) was applied to reduce jitter and preserve peak structure.

Once both algorithms were applied, we obtained movement onsets and offsets that could be compared to the ground truth gesture intervals. Since both MULTISIMO and ECOLANG label all hand movements as gestures, our predicted segments could be evaluated directly, even though our methods do not distinguish between sporadic movement and intentional gesture.

For an overview of the combinations of the data, pose estimation methods and hand movement detection algorithms, see Table 1.

	SPUDNIG	**Peak detection**
OpenPose	M, E	M, E
DLC	M, E	M, E
MediaPipe	M, E	M, E
Kinect	M	M

Table 1: Combinations tested (M = MULTISIMO, E = ECOLANG). Rows list pose estimators; columns list detection algorithms.

3.3. Metrics

The goal of these methods is to support and accelerate manual annotation rather than to replace it, and therefore their output is intended as a first-pass segmentation of hand movement for further human refinement (Trujillo, 2024). Consequently, recall is the most critical metric, as missing segments are more problematic for an annotator. In contrast, the precise placement of gesture boundaries is less crucial, as these are likely to be adjusted during manual review.

To evaluate system performance, we employ both segment-level and frame-level metrics. Segment-level metrics include counts of hits, deletions, and insertions. A predicted segment counts as a hit if it overlaps a ground truth segment by ≥60% of its duration (following Ripperda

et al. (2020)). This approach allows us to capture intuitive, interpretable errors at the segment level, though it may penalise pipelines that produce multiple short, overlapping segments (oversegmentation), which might still be acceptable in practical settings.

To complement this, we also compute *frame-level metrics*: the Jaccard index and the Tversky index, both of which measure the overlap between predicted and annotated gesture frames over time. The Jaccard index (Eq. (1)) is defined as the size of the intersection between predicted and annotated positive frames ($|A \cap B|$) divided by the size of their union ($|A \cup B|$). The Tversky index (Eq. (2)) is a weighted generalization of the Jaccard index. It is calculated as the size of the intersection ($|A \cap B|$) divided by the sum of the intersection and a weighted penalty for false positives ($|A \setminus B|$) and false negatives ($|B \setminus A|$), controlled by parameters α and β (Tversky and Kahneman, 1992). In our evaluation, we set $\alpha = 0.7$ and $\beta = 0.3$ to prioritise recall over precision, reflecting our annotation-support use case.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

$$T_{\alpha,\beta}(A, B) = \frac{|A \cap B|}{|A \cap B| + \alpha\,|A \setminus B| + \beta\,|B \setminus A|} \tag{2}$$

These frame-level metrics offer a fine-grained view of how closely the predicted gesture timeline aligns with the human-annotated one, accounting for partial overlaps and temporal misalignment. However, they are less sensitive to structural errors such as fragmented or overextended segments, and may not fully reflect qualitative differences in gesture segmentation.

4. Results

4.1. Markerless pose estimation

4.1.1. MULTISIMO

Method	Cond.	Recall	Precision	F1	Tversky
OpenPose	Peak	0.279	0.396	0.288	0.142
	SP	**0.732**	0.438	**0.497**	**0.570**
DLC	Peak	0.684	0.432	0.475	0.127
	SPUDNIG	0.547	0.374	0.396	0.527
MediaPipe	Peak	0.356	**0.484**	0.368	0.410
	SPUDNIG	0.170	0.305	0.183	0.329
Kinect	Peak	0.507	0.231	0.278	0.342
	SPUDNIG	0.386	0.299	0.298	0.344

Table 2: Performance of all methods on the MULTISIMO dataset, averaged over all files. Best results are in **bold**.

In this subsection, we describe the performance of the three AI-based pose estimation methods and their combinations with the two movement detection algorithms on the MULTISIMO dataset. Figure 4 presents both frame-wise and segment-wise evaluation metrics for each pipeline variant. These scores are averaged across all analysed videos.

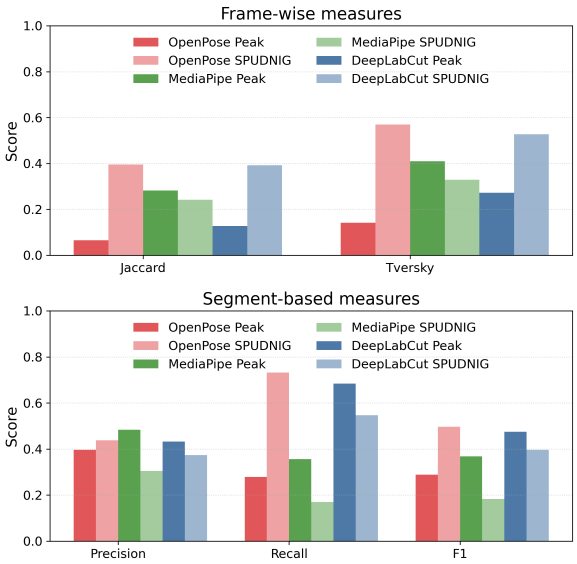

Figure 4: Frame-wise and segment-wise measures of the performance of the pipeline on the MULTISIMO task

As illustrated in Figure 4, the best performing combination on the MULTISIMO task is the OpenPose and SPUDNIG pipeline with a Tversky index of 0.570 and recall of 0.732. Even though we prioritise recall, it is worth noting that F1 and precision scores are lower at 0.497 and 0.438 respectively. These results are based on a confidence threshold of 0.3, selected on the validation set (N=6). This threshold determines the minimum confidence score that OpenPose must assign to a keypoint for it to be considered in the movement detection stage. The performance indicates that the algorithm is likely too sensitive to non-significant movements at a relatively low confidence threshold (0.3). Possibly, the small size of the dataset and, consequently, of the validation set, makes parameter tuning more susceptible to overfitting or instability.

This performance is lower than the 87% agreement between SPUDNIG and a human annotator previously reported by Ripperda et al. (2020), whose metric is most comparable to our F1 score. Notably, the tool was validated on more straightforward data for OpenPose to analyse, with no obstructions of the hands. The worst performing combination is MediaPipe and the SPUDNIG algorithm, achieving a Tversky index of only 0.329 and recall of 0.170. See a summary of the results in Table 2.

The performance differences visible in Figure 4 and Table 2 were formally assessed using non-parametric statistical tests, treating each video as a repeated observation across the different pipeline configurations.

Specifically, Friedman tests on the 31 videos with complete results for all eight pipelines revealed significant differences across all evaluation metrics (Tversky: $\chi^2(7)=128.15$, $p<.001$; Recall: $\chi^2(7)=106.84$, $p<.001$; Precision: $\chi^2(7)=25.40$, $p=.00065$; F1: $\chi^2(7)=56.24$, $p<.001$). Effect sizes were large for Tversky (Kendall's W=0.59) and Recall (W=0.49), small for Precision (W=0.12), and small-to-moderate for F1 (W=0.26). Post-hoc Nemenyi tests indicated widespread pairwise differences for Tversky and Recall, particularly involving the OpenPose and SPUDNIG and OpenPose and peak detection pipelines, while far fewer differences were observed for Precision. For F1, significant differences were primarily driven by the MediaPipe and SPUDNIG pipeline, which differed from several alternatives (DLC–Peak, DLC–SPUDNIG, MediaPipe–Peak, and OpenPose–SPUDNIG; all $p\leq.001$).

4.1.2. ECOLANG

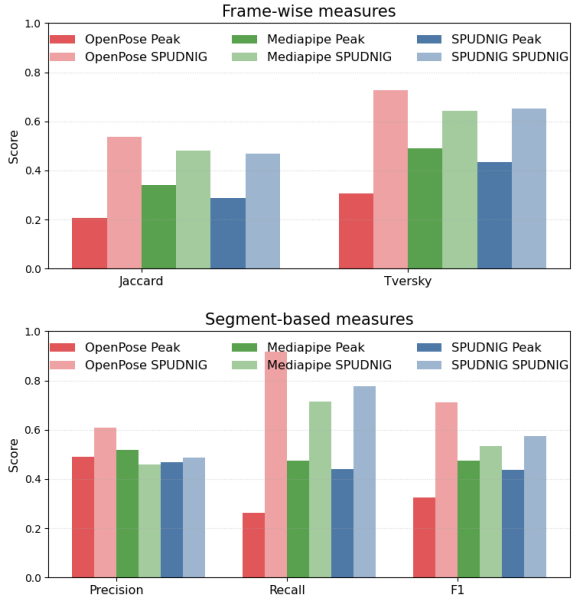

Figure 5: Frame-wise and segment-wise measures of the performance of the pipeline on the ECOLANG task

Following the MULTISIMO analysis, we now analyse the performance of the AI-based tracking methods and movement detection algorithm combinations on the ECOLANG dataset. This dataset is more challenging, and we expected lower overall performance. Figure 5 presents both frame-wise and segment-wise evaluation metrics for each pipeline variant.

Method	Cond.	Recall	Precision	F1	Tversky
OpenPose	Peak	0.263	0.491	0.324	0.307
	SPUDNIG	**0.918**	**0.608**	**0.710**	**0.726**
DLC	Peak	0.441	0.469	0.437	0.433
	SPUDNIG	0.778	0.486	0.574	0.653
MediaPipe	Peak	0.475	0.519	0.476	0.490
	SPUDNIG	0.713	0.459	0.533	0.642

Table 3: Performance of all methods on the ECOLANG dataset, averaged over all files. Best results are in **bold**.

The best-performing pipeline on the ECOLANG task is OpenPose combined with the SPUDNIG movement detection algorithm, achieving a Tversky index of 0.726, recall of 0.918, precision of 0.608, and F1 score of 0.710. Notably, this performance exceeds that of the best-performing pipeline on MULTISIMO, indicating that ECOLANG may be more amenable to automatic segmentation despite its complex interaction setting. In contrast, OpenPose combined with the velocity peak algorithm yields the lowest performance, with a Tversky index of 0.307 and recall of 0.263. A summary of all results is provided in Table 3.

Friedman tests revealed significant differences between pipelines across all evaluation metrics (Tversky: $\chi^2(5) = 219.08$, $p < .001$; Recall: $\chi^2(5) = 231.57$, $p < .001$; Precision: $\chi^2(5) = 47.23$, $p < .001$; F1: $\chi^2(5) = 126.33$, $p < .001$). Effect sizes were large for Tversky (Kendall's W = 0.65) and Recall (W = 0.69), moderate for F1 (W = 0.38), and small for Precision (W = 0.14). Post-hoc Nemenyi tests indicated several significant pairwise differences between pipelines. In particular, SPUDNIG-based pipelines consistently outperformed their peak-detection counterparts, while OpenPose combined with peak detection performed significantly worse than several alternative pipelines. These statistical differences align with the performance pattern shown in Table 3, where OpenPose combined with SPUDNIG achieves the highest overall scores across evaluation metrics.

Surprisingly, the models handled this task relatively well. This may be due to the annotation characteristics – there are often very long gesture segments when the participants are holding a certain object, making the data less sparse than MULTISIMO. Therefore, it is easier for a relatively permissive model to score higher on the recall and precision scales. As a result, permissive detection models can achieve higher recall and overlap-based metrics.

4.2. Device-based pose estimation

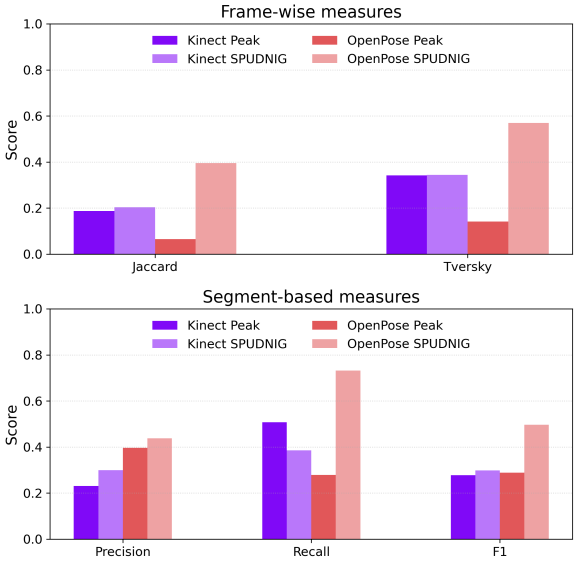

Figure 6: Kinect v2 pipelines vs. the best AI baseline on MULTISIMO: frame-wise and segment-wise metrics.

This subsection presents the results for our only device-based method, Kinect v2. Since the Kinect data was only available for the MULTISIMO dataset, we compare to the best performing AI-based method on the MULTISIMO data, OpenPose with the SPUDNIG algorithm (Figure 6).

The overall performance of the Kinect methods is poorer, even though we leverage three-dimensional data. Despite the smoothing, the noisy data resulted in low precision (0.278) and higher recall (0.507) for the better performing peak algorithm, suggesting a lot of jitter causing false positives, even when only considering peaks in the 98th percentile above the noise floor.

This pattern is consistent with the statistical analysis reported in Section 4.1.1. Post-hoc Nemenyi tests following the Friedman analysis revealed that the Kinect pipelines differed significantly from several AI-based pipelines across multiple evaluation metrics, particularly for Tversky and recall. These results support the visual comparison in Figure 6, indicating that the device-based approach performs less reliably than the best AI-based pipeline on the MULTISIMO task.

5. Discussion and conclusion

The goal of this paper was to assess whether off-the-shelf gesture-tracking pipelines can realistically support human annotation. Overall, naturalistic recordings—with frequent occlusions, hand instrusions and object manipulation—remain challenging for all systems.

Across datasets, the strongest pipelines combined OpenPose or DeepLabCut with the SPUDNIG detection algorithm. The Kinect data proved to be challenging to process and was affected by significant jitter and noise.

Is this technology usable? For our best-performing combination on ECOLANG, an average 92% of ground truth gestures were matched by a predicted gesture segment, overlapping by at least 60%. Given ≈29 gestures per video on average in the ECOLANG data, this corresponds to about 27 correctly matched segments and 19 insertions per video. With $\alpha = 0.7$ and $\beta = 0.3$, a Tversky index of 0.726 indicates that—after penalizing misses more than false alarms—roughly three-quarters of the weighted positive frames are correctly predicted. These results suggest that, under suitable recording conditions, current systems can provide strong practical support for annotation workflows.

A key practical implication is that recording setups should be optimised for these tools. This need not compromise naturalism, as we make a distinction between task naturalism and capturing data strategically. Careful choices such as camera placement and framing preserve ecological validity while enabling reliable automatic annotation.

A limitation in our analysis was that we focused on rule-based movement detection rather than machine learning approaches. In our view, the marginal performance gains from ML are not yet large enough to offset their reduced transferability and fine-tuning requirements. Nonetheless, machine learning is likely to lead to better results with careful tuning to specific data, as shown in Ghaleb et al. (2024). Additionally, in the longer term, it has the potential to predict gesture phases and distinguish between sporadic movement and meaningful gestures (Pouw and Dixon, 2019).

In short, current off-the-shelf systems can support, but not replace, human annotation. SPUDNIG has been shown to speed up annotation on straightforward data (Ripperda et al., 2020) when it reaches 87% agreement with a human annotator. As a useful comparator in semi-automatic speech annotation, ASR becomes useful only when word error rates fall below 30% and annotators correct rather than erase predictions (Papadopoulou et al., 2021). Our results show these methods can be valuable as a first pass, especially with strategic data capture.

6. Acknowledgments

This publication has emanated from research conducted with the financial support of Taighde Éireann – Research Ireland under Grant number 22/FFP-A/11059.

7. Bibliographical References

Camila Antônio Barros, Jorge Francisco Ciprián-Sánchez, and Saulo Mendes Santos. 2024. A Tool for Determining Distances and Overlaps between Multimodal Annotations. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pages 1705–1714.

Gibran Benitez-Garcia, Jesus Olivares-Mercado, Gabriel Sanchez-Perez, and Hiroki Takahashi. 2025. IPN HandS: Efficient Annotation Tool and Dataset for Skeleton-Based Hand Gesture Recognition. Applied Sciences, 15(11):6321.

Stijn De Beugher, Geert Brône, and Toon Goedemé. 2018. A semi-automatic annotation tool for unobtrusive gesture analysis. Language Resources and Evaluation, 52(2):433–460.

Sougatamoy Biswas, Anup Nandy, Asim Kumar Naskar, and Rahul Saw. 2024. MediaPipe with LSTM Architecture for Real-Time Hand Gesture Recognization. In Harkeerat Kaur, Vinit Jakhetiya, Puneet Goyal, Pritee Khanna, Balasubramanian Raman, and Sanjeev Kumar, editors, Computer Vision and Image Processing, volume 2010, pages 422–431. Springer Nature Switzerland, Cham. Series Title: Communications in Computer and Information Science.

Laisi Cai, Ye Ma, Shuping Xiong, and Yanxin Zhang. 2019. Validity and Reliability of Upper Limb Functional Assessment Using the Microsoft Kinect V2 Sensor. Applied Bionics and Biomechanics, 2019(1):7175240.

Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2019. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. IEEE transactions on pattern analysis and machine intelligence, 43(1):172–186.

Quentin De Smedt, Hazem Wannous, and Jean-Philippe Vandeborre. 2016. Skeleton-Based Dynamic Hand Gesture Recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 1206–1214, Las Vegas, NV, USA. IEEE.

Mohamed Elgendi, Flavien Picon, and N. Magenant-Thalmann. 2012. Real-time speed detection of hand gesture using, Kinect. In Proc. Workshop on Autonomous Social Robots and Virtual Humans, The 25th Annual Conference on Computer Animation and Social Agents (CASA 2012), pages 1–15.

Esam Ghaleb, Ilya Burenko, Marlou Rasenberg, Wim Pouw, Peter Uhrig, Judith Holler, Ivan Toni, Aslı Özyürek, and Raquel Fernández. 2024. Co-Speech Gesture Detection through Multi-Phase Sequence Labeling. In 2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pages 3995–4003, Waikoloa, HI, USA. IEEE.

Mickaëlla Grondin-Verdon, Domitille Caillat, and Slim Ouni. 2024. Qualitative study of gesture annotation corpus : Challenges and perspectives. In Companion Proceedings of the 26th International Conference on Multimodal Interaction, pages 147–155, San Jose Costa Rica. ACM.

Judith Holler, Kobin H. Kendrick, and Stephen C. Levinson. 2018. Processing language in face-to-face conversation: Questions with gestures get faster responses. Psychonomic Bulletin & Review, 25(5):1900–1908.

Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. 2016. DeeperCut: A Deeper, Stronger, and Faster Multi-person Pose Estimation Model. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, Computer Vision – ECCV 2016, volume 9910, pages 34–50. Springer International Publishing, Cham. Series Title: Lecture Notes in Computer Science.

Shuo Jiang, Bo Lv, Weichao Guo, Chao Zhang, Haitao Wang, Xinjun Sheng, and Peter B. Shull. 2017. Feasibility of wrist-worn, real-time hand, and surface gesture recognition via sEMG and IMU sensing. IEEE Transactions on Industrial Informatics, 14(8):3376–3385.

Adam Kendon. 1980. Gesticulation and speech: Two aspects of the relationship of verbal and nonverbal communication. In The Relationship of Verbal and Nonverbal Communication, pages 207–227. Walter de Gruyter. Mouton The Hague.

Kobin H. Kendrick, Judith Holler, and Stephen C. Levinson. 2023. Turn-taking in human face-to-face interaction is multimodal: gaze direction and manual gestures aid the coordination of turn transitions. Philosophical Transactions of the Royal Society B: Biological Sciences, 378(1875):20210473.

Jong-Wook Kim, Jin-Young Choi, Eun-Ju Ha, and Jae-Ho Choi. 2023. Human pose estimation using mediapipe pose and optimization method based on a humanoid model. Applied sciences, 13(4):2700.

Seungbae Kim, Daeun Lee, Brielle Stark, and Jinyoung Han. 2025. Gesture-Aware Zero-Shot Speech Recognition for Patients with Language Disorders. ArXiv:2502.13983 [eess].

Gregorij Kurillo, Evan Hemingway, Mu-Lin Cheng, and Louis Cheng. 2022. Evaluating the Accuracy of the Azure Kinect and Kinect v2. Sensors (Basel, Switzerland), 22(7):2469.

Kenneth Lai, Richard Brutti, Lucia Donatelli, and James Pustejovsky. 2024. Encoding Gesture in Multimodal Dialogue: Creating a Corpus of Multimodal AMR. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pages 5806–5818.

Alexander Mathis, Pranav Mamidanna, Kevin M. Cury, Taiga Abe, Venkatesh N. Murthy, Mackenzie Weygandt Mathis, and Matthias Bethge. 2018. DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. Nature neuroscience, 21(9):1281–1289.

David McNeill. 1992. Hand and mind: What gestures reveal about thought. The University of Chicago Press, Chicago.

Tanmay Nath, Alexander Mathis, An Chi Chen, Amir Patel, Matthias Bethge, and Mackenzie Weygandt Mathis. 2019. Using DeepLabCut for 3D markerless pose estimation across species and behaviors. Nature protocols, 14(7):2152–2176.

Giulia Panconi, Stefano Grasso, Sara Guarducci, Lorenzo Mucchi, Diego Minciacchi, and Riccardo Bravi. 2025. DeepLabCut custom-trained model and the refinement function for gait analysis. Scientific Reports, 15(1):2364.

Martha Maria Papadopoulou, Anna Zaretskaya, and Ruslan Mitkov. 2021. Benchmarking ASR systems based on post-editing effort and error analysis. In Proceedings of the Translation and Interpreting Technology Online Conference, pages 199–207.

PDF. 2024. Dynamic Gesture Recognition using a Transformer and Mediapipe. International Journal of Advanced Computer Science and Applications, 15(6). Place: West Yorkshire, United Kingdom.

Wim Pouw and James A. Dixon. 2019. Quantifying gesture-speech synchrony. Proceedings of the 6th Gesture and Speech in Interaction Conference. Version Number: 1.

Muyun Qian, Haitang Yan, Wanying Wang, Zelin Sun, Yaohui Dong, Xinyuan Wei, and Hanbin Wang. 2025. Dynamic gesture tracking using wearable data gloves with flexible FBGs. Sensors and Actuators A: Physical, 390:116622.

Kazuki Sekine and Hiroshi Hotta. 2025. The Role of Self-Adaptors in Lexical Retrieval. Languages, 10(9):209.

Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. 2011. Real-time human pose recognition in parts from single depth images. In CVPR 2011, pages 1297–1304. Ieee.

James Trujillo, Asli Özyürek, Judith Holler, and Linda Drijvers. 2021. Speakers exhibit a multimodal Lombard effect in noise. Scientific Reports, 11(1):16721.

James P. Trujillo. 2024. Motion-tracking technology for the study of gesture. In The Cambridge Handbook of Gesture Studies. Cambridge University Press.

James P. Trujillo, Julija Vaitonyte, Irina Simanova, and Asli Özyürek. 2019. Toward the markerless and automatic analysis of kinematic features: A toolkit for gesture and movement research. Behavior Research Methods, 51(2):769–777.

Amos Tversky and Daniel Kahneman. 1992. Advances in prospect theory: Cumulative representation of uncertainty. Journal of Risk and Uncertainty, 5(4):297–323.

Regina Zaghi-Lara, Miguel Ángel Gea, Jordi Camí, Luis M. Martínez, and Alex Gomez-Marin. 2019. Playing magic tricks to deep neural networks untangles human deception. ArXiv:1908.07446 [q-bio].

8. Language Resource References

Jens Allwood and Loredana Cerrato and Kristiina Jokinen and Costanza Navarretta and Patrizia Paggio. 2007. MUMIN Coding Scheme. Language Resources and Evaluation. PID https://doi.org/10.1007/s10579-007-9061-5.

Gibran Benitez-Garcia and Jesus Olivares-Mercado and Gabriel Sanchez-Perez and Keiji Yanai. 2021. IPN HandS Dataset & Tool. IEEE. PID https://gibranbenitez.github.io/IPN_Hand/.

Jana Bressem and Silva H. Ladewig and Cornelia Müller. 2013. LASG Annotation System. Body—Language—Communication: An international handbook on multimodality in human interaction. PID https://doi.org/10.1515/9783110261318.1098.

Ariel Caputo and others. 2021. SHREC 2021 Dataset. Computers & Graphics. PID https://doi.org/10.1016/j.cag.2021.07.007.

Jean Carletta and Stefan Evert and Ulrich Heid and Jonathan Kilgour and Judy Robertson and Holger Voormann. 2003. NITE XML Toolkit. Behavior Research Methods, Instruments, & Computers. PID https://groups.inf.ed.ac.uk/nxt/.

Yan Gu and Ed Donnellan and Beata Grzyb and Gwen Brekelmans and Margherita Murgiano and Ricarda Brieke and Pamela Perniss and Gabriella Vigliocco. 2025. ECOLANG Corpus. Nature Publishing Group UK London. PID https://www.nature.com/articles/s41597-025-04405-1.

Henning Holle and Robert Rein. 2015. EasyDIAg Tool. Behavior research methods. PID https://sourceforge.net/projects/easydiag/.

Maria Koutsombogera and Carl Vogel. 2018. MULTISIMO Corpus. European Language Resources Association (ELRA). PID https://multisimo.eu/datasets.html.

Haiyang Liu and Zihao Zhu and Naoya Iwamoto and Yichen Peng and Zhengqing Li and You Zhou and Elif Bozkurt and Bo Zheng. 2022. BEAT Dataset. Springer Nature Switzerland. PID https://pantomatrix.github.io/BEAT/.

Andy Lücking and Kirsten Bergman and Florian Hahn and Stefan Kopp and Hannes Rieser. 2013. SaGA Corpus. Journal on Multimodal User Interfaces. PID https://clarin.is/en/resources/sagacorpus/.

Wim Pouw and Bosco Yung and Sharjeel Ahmed Shaikh and James Trujillo and Antonio Rueda-Toicen and Gerard De Melo and Babajide Owoyele. 2025. EnvisionHGdetector Toolkit. PsyArXiv. PID https://doi.org/10.31234/osf.io/psg5f_v1.

Jordy Ripperda and Linda Drijvers and Judith Holler. 2020. SPUDNIG Toolkit. Behavior Research Methods. PID https://doi.org/10.3758/s13428-020-01350-2.

Leonie Schade and Nico Dallmann and Olcay Türk and Petra Wagner. 2024. MUNDEX Corpus. Proceedings of INTERSPEECH 2024. PID https://www.isca-archive.org/interspeech_2024/schade24_interspeech.html.

Yifan Zhang and Congqi Cao and Jian Cheng and Hanqing Lu. 2018. EgoGesture Dataset. IEEE Transactions on Multimedia. PID https://nlpr.ia.ac.cn/iva/yfzhang/datasets/egogesture.html.