

Exploring Pose-based Sign Language Translation: Ablation Studies and Attention Insights

Tomáš Železný, Jakub Straka, Václav Javorek, Ondřej Valach, Marek Hruží, Ivan Gruber
Univerzity of West Bohemia,
Faculty of Applied Sciences, Department of Cybernetics
Univerzitní 2732/8, 301 00 Plzeň, Czech Republic

zeleznyt@kky.zcu.cz, strakajk@kky.zcu.cz, javorek@kky.zcu.cz
valacho@kky.zcu.cz, mhruz@ntis.zcu.cz, grubiv@ntis.zcu.cz

Abstract

Sign Language Translation (SLT) has evolved significantly, moving from isolated recognition approaches to complex, continuous gloss-free translation systems. This paper explores the impact of pose-based data preprocessing techniques — normalization, interpolation, and augmentation — on SLT performance. We employ a transformer-based architecture, adapting a modified T5 encoder-decoder model to process pose representations. Through extensive ablation studies on YouTubeASL and How2Sign datasets, we analyze how different preprocessing strategies affect translation accuracy. Our results demonstrate that appropriate normalization, interpolation, and augmentation techniques can significantly improve model robustness and generalization abilities. Additionally, we provide a deep analysis of the model’s attentions and reveal interesting behavior suggesting that adding a dedicated register token can improve overall model performance. We publish our code on our GitHub repository¹, including the preprocessed YouTubeASL data.

1. Introduction

Sign language translation has witnessed remarkable progress over the past few decades, transitioning from early isolated sign language recognition systems to more complex continuous recognition frameworks. Early methods largely depended on gloss-based approaches—relying on intermediary linguistic annotations to bridge the visual and textual modalities—while recent research has increasingly shifted toward gloss-free techniques. These gloss-free methods seek to directly map visual inputs into textual outputs, leveraging advances in multi-modal learning and large language models to enhance translation accuracy.

Despite these advancements, gloss-free systems still face notable challenges. Variations in signer position, scale, and background dynamics together with no direct alignment between the input and output languages contribute to the performance gaps when compared to their gloss-based counterparts. In response, our work systematically investigates a series of data preprocessing techniques including keypoint extraction, normalization, and augmentation aiming to mitigate the issues of spatial variation and improve the robustness of the translation pipeline.

We present a comprehensive evaluation of these techniques within a transformer-based framework, specifically adapting a modified T5 encoder-decoder architecture for the task of SLT. Extensive ablation studies are conducted on challenging datasets such as YouTubeASL [35] and How2Sign [10], revealing that a thoughtful combination of normalization and augmentation strategies can substantially enhance model performance. Our analysis not only demonstrates improvements in translation accuracy but also provides valuable insights into the interplay between visual preprocessing and model architecture.

Ultimately, this work contributes to the broader goal of developing more accurate and efficient SLT systems, paving the way for enhanced accessibility and communication between the Deaf and hearing communities.

2. Related Work

Sign Language Translation has progressed through dynamic evolution over the years, beginning with work on Isolated Sign Language Recognition (ISLR) [15, 23] and progressing more towards Continuous Sign Language Recognition (CSLR) [6, 14], with early efforts primarily focused on isolated sign language (SL) datasets [9, 22] and more recent studies advancing with continuous data that capture the dynamic and nature of sign language communication [4, 10, 32, 34, 35]. Building on this, SLT has devel-

¹https://github.com/zeleznyt/T5_for_SLT

opped in two main approaches: gloss-based and gloss-free methods. Gloss-based approaches utilize structured linguistic representations of signs to learn the alignment between sign language (glosses) and text [2, 6, 7, 41, 47], while gloss-free methods directly map visual features to text, aiming to bypass the need for intermediate linguistic annotations (glosses) [5, 13, 16, 27, 40, 45]. Gloss-free methods often introduce innovative approaches as for example utilization of self-supervised fine-tuning [17], sign pose quantization [18] or pseudo-translation tasks [44]. Although gloss-based techniques benefit from the transparent supervision, gloss-free approaches have become increasingly popular thanks to advancements in multi-modal learning, with the integration of Large Language Models (LLMs) enhancing the translation accuracy by utilizing better pretrained textual representations [24, 31, 36].

Thanks to this increasing popularity, we have seen innovations in gloss-free approaches such as Sign2GPT [36] using large-scale pretrained visual and language models, GFSLT-VLP [45] integrating contrastive language-image pretraining and masked self-supervised learning. Innovations went all the way into the topic of diffusion models with DiffSLT [28], a diffusion-based generative approach, transforming random noise into the target latent representation. Furthermore, we also saw SignLLM [11] applying vector-quantization to convert sign videos into discrete tokens, and SignCL [39] introducing a sign contrastive loss to reduce representation density in dense visual sequences. Moreover, there were innovations such as GASLT [40], which incorporates gloss-attention mechanisms, and CS-GCR [43], which utilizes custom word verification. Despite these developments and the overall potential, gloss-free SLT methods continue to face a performance gap when compared to their gloss-based alternatives.

Transformer-based models, such as the T5 [30], have shown great multilingual capability. Recent literature have explored T5’s flexibility in handling multimodal inputs [12, 38, 42], which showed its potential to address the translation of embedded visual sign language input into text. Additionally, studies using encoder-decoder models that integrate pretrained visual encoders with advanced text decoders — like GFSLT-VLP [45] based on mBART [25] — indicate that utilizing strong language priors without relying on gloss annotations is an interesting approach to further investigate. Our work uses T5 as model for SL translation and conducts extensive ablation studies which cover areas such as pose augmentation and sign space pose normalization. Recent research shows that while increasing model scale tends to boost performance, using well-curated data and a thoughtfully designed approach is equally important [20].

There are few recent papers related to these topics that explore the utilization of unique pose normalization aiming for encoder-only transformer in SL modeling [37], face

swapping, and other image (mostly affine) augmentations of SL data which report positive effects during training [29]. Two studies dive into an attention analysis and attention-based sign language recognition built upon decoupled graph and temporal self-attention [1, 33]. These studies showcase some interesting observations, for example, that transformer models for SLT learn to attend to sequential clusters rather than individual frames [1], which will be referred to more in Section 5.3.

3. Methods

In this section, we describe different parts of our processing parts with emphasis on the parts relevant to the following ablation studies.

3.1. Data preprocessing

Data preprocessing is important, especially when working with uncurated datasets. In our experiments, we use YouTubeASL [35] and How2Sign [10]. YouTubeASL consists of videos captured in the wild and is uncurated, meaning signers appear in various positions, sizes, and resolutions, sometimes alongside other people. In contrast, How2Sign is recorded in a controlled setting with a single signer positioned in front of the camera. However, signers can still shift across videos or appear at different distances.

To address these variations, we first extract keypoints and then evaluate multiple normalization strategies. In both cases, we first split videos into clips based on the captions and work only with the clips.

3.1.1. Keypoint Extraction

We use a two-stage approach for keypoint detection: first, we detect a person in the frame, and then we predict keypoints within the detected area. Detecting the person first is crucial, as the signer may occupy only a small portion of the screen (e.g., a news interpreter).

Instead of using a standard object detection model for person detection, we employ a lightweight keypoint detection model. We then define a bounding box around the signer based on the signing space. Signing space is a concept from linguistics, which we define as a rectangle centered between the shoulders, with a width and height four times the shoulder distance. All signing should happen in this area, we make the box slightly bigger than is necessary to ensure that all keypoints are in the box. This guarantees that the signer remains centered, occupies the majority of the frame, and maintains a consistent size across the clip.

We exclude clips containing multiple people, as tracking all individuals across frames and identifying the signer introduces potential errors. To simplify processing, we omit such clips.

Our keypoint extraction pipeline consists of the following steps: 1. We start by detecting pose using

YOLOv8-nano [19], if the clip contains multiple people we discard it. 2. Based on the detected poses we create the signing space. 3. Next, we spatially crop frames based on the sign space, this ensures that all excessive background is removed and frames are roughly centered on the signer. 4. Lastly, we use MediaPipe [26] to predict body pose, hand pose, and face mesh in the spatially cropped clip.

We do not use all keypoints from MediaPipe. For the body pose, we omit leg keypoints, and for the face, we select only a small subset representing prominent facial features. In total, we extract 104 keypoints, this includes 21 keypoints for each hand, 25 for the body pose, and 37 for the face². We use the x and y coordinates generated by MediaPipe, resulting in a final 208-dimensional vector per frame.

3.1.2. Pose Normalization

The main step in preprocessing is keypoint normalization, which aims to make keypoints invariant to translation and scale. Although we centered frames on the signer during keypoint extraction, some shifts or size differences may still occur. We evaluate three normalization strategies: two based on the YouTubeASL paper and one based on our signing space approach based on the work [2].

In the YouTubeASL paper, normalization is applied by scaling keypoints to fit within a unit bounding box across the entire duration of the clip. We refer to this method as *yasl_c*. This approach ensures that the signer remains of consistent size across all frames but does not account for the changing position within the frame.

We also evaluate a frame-wise normalization strategy, where keypoints are normalized independently in each frame to fit within a unit bounding box. While this method eliminates shifts in the frame and distributes keypoints more evenly within the bounding box, it can cause the signer’s size to fluctuate across frames. We refer to this normalization as *yasl_f*. Examples of *yasl_c* and *yasl_f* normalized keypoints are shown in Figure 1a and Figure 1b, respectively.

The third normalization method (denoted as *SignSpace*) we evaluate is based on the signing space we defined in Subsubsection 3.1.1. We normalize body pose keypoints by creating a bounding box centered between the shoulders, with its width and height set to three times the distance between the shoulders. Keypoints within the signing space are then scaled to be in the range $\langle -1, 1 \rangle$. After scaling, keypoints are shifted so that the center of the signing space is at position $[0, 0]$. This normalization is applied frame by frame and we consider it as global, as it preserves the relation between the individual body parts.

Global normalization is applied only to body pose keypoints. For hands and face, we use local normalization, meaning we normalize each hand and face separately by

scaling them to range $\langle -1, 1 \rangle$ while maintaining their aspect ratio. Additionally, we add a 10% border from each side around them to suppress the effect of inaccuracies in the pose estimation model. Local normalization ensures a focused view of individual parts, independent of their absolute position. The absolute position and relationship between different body parts are instead captured through global body pose normalization. Example of keypoints normalized by this method is depicted in Figure 1c.

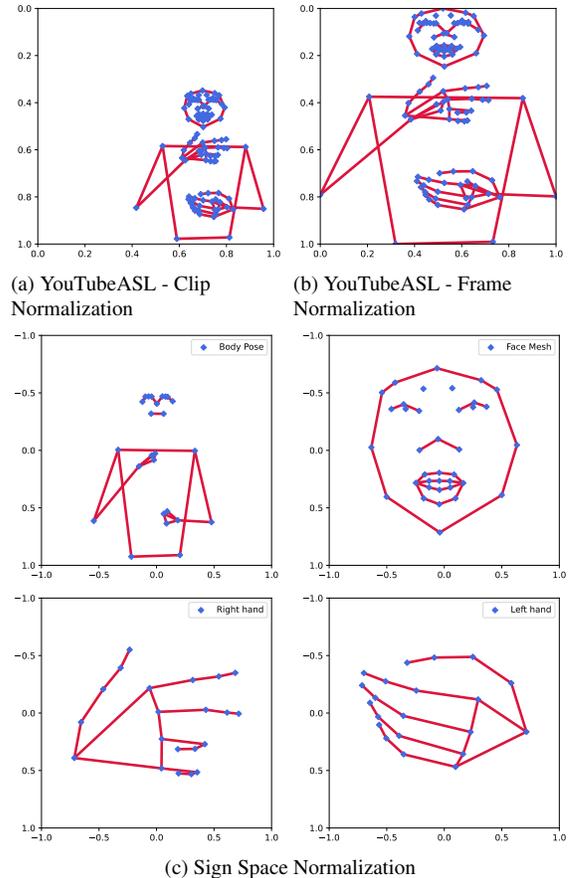


Figure 1. Examples of evaluated normalization methods. We compare multiple approaches: (a) shows the normalization proposed in the YouTubeASL paper, where poses are scaled to fit within a unit box across the entire clip. (b) shows an alternative method where normalization is applied separately to each frame. Finally, (c) illustrates our approach, which normalizes the body pose globally using signing space while applying local normalization separately to the hands and face.

3.1.3. Missing Values

One important issue that is necessary to handle during the normalization are miss-detections that result in missing values. Some of the keypoints may not be detected, or the signer may use only one hand, with the other hand out of the frame. In the YouTubeASL paper, missing values are

²Same as in YouTubeASL paper [35].

handled by replacing them with a large negative value. We adopt this approach, but we also propose to linearly interpolate keypoints if the frame gap between the detected keypoints is short.

In our case, the gap is two frames or shorter in almost 60% of cases, and three frames or less in almost 75% of cases. We assume that the change between close frames is small, which means that interpolated keypoints should be close enough to retain their semantic meaning. We compare this to the approach where all the missing values are replaced with constant values.

3.1.4. Augmentations

Augmentations are commonly used to enrich datasets. However, in the SLT task, it is essential to ensure that the augmentation process does not change the semantic meaning of the pose. We evaluated three augmentation strategies, each varying the probability and intensity of augmentations by scaling the default strategy values. The default strategy is *heavy*, the second is *medium* (scaled by 0.75), and the last is *light* (scaled by 0.5).

We use mainly geometric augmentations, which include: rotation, shear, perspective, arm rotation, and additive Gaussian noise. The same augmentations are applied to all frames in the clip.

Arm rotation augmentation rotates all arm and hand keypoints around a shoulder, elbow, or wrist keypoint. This augmentation can be chained, which means that the entire arm can first rotate around the shoulder, then again around the elbow or wrist in successive transformations. In Figure 2, there are examples of some of the augmentations. Here, each augmentation is applied individually, but during the training, multiple augmentations can be applied to one frame.

3.2. Model

Our model setup follows the original baseline method of the YouTubeASL paper. We use a modified version of T5 [30] encoder-decoder-based transformer. In order to process the input of the 208-dimensional keypoint features, we employ a custom linear layer at the transformer’s encoder input instead of traditional tokenized text. Following standard embedding layer practices, our custom layer does not include an additive bias. Besides this change, our model follows a standard T5v1.1³ architecture. The T5 weights are initialized from T5X, while the custom layer uses the Xavier initialization.

³https://github.com/google-research/text-to-text-transfer-transformer/blob/main/released_checkpoints.md



Figure 2. Examples of individual augmentations. We show only body pose keypoints, during training all keypoints are augmented. To better illustrate their effects, we applied the same geometric augmentation (except for arm rotation) to the frame.

4. Experiments and Quantitative Results

In this section, we first describe our experimental setup. Then, we report and analyze the results of three different ablation studies.

4.1. Experimental Setup

In our experiments, we finetune the T5-based model on the YouTubeASL dataset using various data preprocessing techniques to evaluate their impact on overall model performance. For YouTubeASL we use a custom 90:10 train-val split, while for the How2Sign dataset we use the default split provided by the dataset. All our experiments are assessed based on the BLEU scores computed using sacrebleu v2.4.3 on the How2Sign dataset, a standard benchmark for gloss-free sign language translation systems. The performance on the How2Sign dataset is measured without any additional finetuning on this dataset. If not stated otherwise, the model is trained for a total of 200,000 iterations using an effective batch size of 256 and a constant learning rate of 0.0004. In the initial experiments, we observe high training volatility. To reduce this variability between training runs, we employ a warm-up phase for the first 5,000 training steps. Additionally, to ensure a fair comparison between different training setups, we run each experiment with three different seeds and report the best run. The YouTubeASL paper doesn’t provide the exact value to use in case of missing keypoint values. Inspired by their mention of a “large

Normalization	B-1	B-2	B-3	B-4
none	13.62	3.67	1.54	0.73
<i>yaslc</i>	13.00	3.90	1.59	0.66
<i>yaslf</i>	14.67	4.78	2.19	1.13
<i>SignSpace</i>	17.47	7.19	3.79	2.17

Table 1. Comparison of four different types of normalization techniques. Performance is measured by BLEU scores on the How2Sign dataset.

Interpolation	B-1	B-2	B-3	B-4
none	17.47	7.19	3.79	2.17
≤ 2 frames	16.91	7.35	4.06	2.43
≤ 3 frames	17.16	7.40	4.01	2.33

Table 2. Comparison of three different interpolation settings. Performance is measured by BLEU scores on the How2Sign dataset.

negative number,” we use a value -10 in our experiments. The training was conducted using 4 AMD MI250x GPU modules, split into 8 GCD for each experiment.

It should be noted that the trained models after 200,000 iterations are not ”fully trained”, and their performance would benefit from additional training; there are two reasons for this shorter training protocol. Firstly and more importantly, we believe the comparative performance after this shorter training protocol reflects the performance comparison of fully-trained models. The second reason is based on the restriction of computational resources available.

4.2. Normalization

In the first set of experiments, we analyze four different types of data normalization. Results can be seen in Table 1.

All the proposed normalization results in better performance when compared to the training without any normalization. Interestingly, the original *yaslc* performs worse than our modification *yaslf*. We argue that the speaker size change in the *yaslf* normalization is less *distracting* for the model than the shift in the speaker position in *yaslc*. The *SignSpace* normalization outperforms all other normalization approaches by a large margin. Based on this result, all the following experiments use the *SignSpace* normalization.

4.3. Interpolation

In the next series of experiments, we analyze the effect of using linear interpolation of the missing keypoints. We experiment with a total of 3 different settings: interpolate all gaps with size 2 or smaller, with gaps 3 or smaller, or don’t use interpolation at all, in which case all missing values are replaced with the default value equal to -10 . The results are in Table 2.

Both interpolation approaches result in slightly better re-

Augmentation	B-1	B-2	B-3	B-4
none	17.47	7.19	3.79	2.17
rotate	15.30	5.73	2.88	1.61
shear	17.19	7.25	3.86	2.2
perspective	16.07	6.83	3.70	2.17
rotate shoulder	16.39	6.97	3.75	2.17
rotate elbow	17.48	7.38	3.89	2.28
rotate wrist	16.05	6.84	3.72	2.20
noise	17.45	7.47	4.07	2.41

Table 3. Impact of individual augmentations. Performance is measured by BLEU scores on the How2Sign dataset.

Augmentations	B-1	B-2	B-3	B-4
none	17.47	7.19	3.79	2.17
light	15.76	6.23	3.12	1.71
medium	17.27	7.51	4.12	2.46
heavy	16.58	7.10	3.85	2.29

Table 4. Impact of different augmentation protocols. Performance is measured by BLEU scores on the How2Sign dataset.

sults than runs without any interpolation. We hypothesize that the interpolation makes data easier to interpret and additionally gives the model more frames where relevant information is stored.

4.4. Augmentations

We investigate the model’s performance using different types of augmentations. First, we assess the contributions of individual augmentations by applying them with a medium-scale value and evaluating the finetuned models. Based on these individual performances, we select those augmentations that positively impact performance to design a final augmentation protocol with three different scales, as described in Section 3.1.4.

According to Table 3, the overall performance (majority of the BLEU scores) was improved by the shear, rotate elbow, and noise augmentations. In our final augmentation protocols, we used only these three types of augmentation. We tried to analyze the other augmentations and their effect on the inputs. The decrease in performance for the rotate augmentation is probably caused by the fact that rotation is not very common in real-world data examples. Therefore, it does not contribute to the necessary generalization and only makes the training data more difficult. The same is true for the perspective augmentation. Additionally, we argue that augmentation of the shoulder and wrist rotation can be too heavy in the sense that they can easily change the meaning of signs.

The final results of our three augmentation protocols are presented in Table 4.

Based on the results, it seems that the medium augmentation protocol slightly improves the final results. The other two protocols are comparable with the setup without any augmentations. There are two main possible reasons why this phenomenon occurred. First, our training protocol is too short. Based on the analysis of training curves, we do not see any saturation in the results. The lack of saturation, in conjunction with the fact that training with augmentations is generally slower due to the increased complexity of the training set, could result in worse performance after a certain number of iterations. Second, the YouTubeASL dataset is a very complex dataset with a large number of data samples. Therefore, the proposed augmentation may not bring any helpful information into the training. We want to analyze this phenomenon more in our future research.

5. Qualitative Results

In this section we provide qualitative results in form of self- and cross-attention analysis of our T5v1.1-base model. We also analyze translations that are learned on the weakly aligned data from the YouTubeASL dataset.

5.1. Encoder Self-Attention

To analyze the patterns in the encoder attention mechanism during T5 inference, a visualization averaged over all encoder layers (Figure 3) shows that each of the 12 attention heads specializes in identifying a distinct causal pattern within the input signal. Furthermore, each head focuses on a different temporal context surrounding the current frame. These findings stand true for all analyzed data hinting at a learned specialization of each head. More examples with all attention heads visualized can be found in the supplementary material.

5.2. Cross-Attention Behavior

In the cross-attention matrices during inference, we demonstrate a clear causal relationship between encoder and decoder representations. The attention progresses sequentially over time, consistent with the linear advancement of both textual and ASL signals, resulting in an attention distribution that disperses over segmented words, as we present in a selected cross-attention matrix in Figure 4. The other layers' visualization can be found in supplementary material.

Next to this, we have revealed another kind of trend in the cross-attention data. In majority of the analyzed matrices, averaged across heads and layers, there appears to be a spike in intensity in the last few frames towards the end of the clip. Also, in many clips there is an attention spike in several other places across the clip. This behavior suggests that the decoder is placing greater attention on a specific subset of input frames when generating each decoded token. This can be observed in Figure 5a. When the spike appears during the signing we found out that it is usually

Ref: Hello, I'm Rogan and welcome to the first video of 2020.
 Pred: Hello, I'm Rogan and welcome to the first video of 2020.

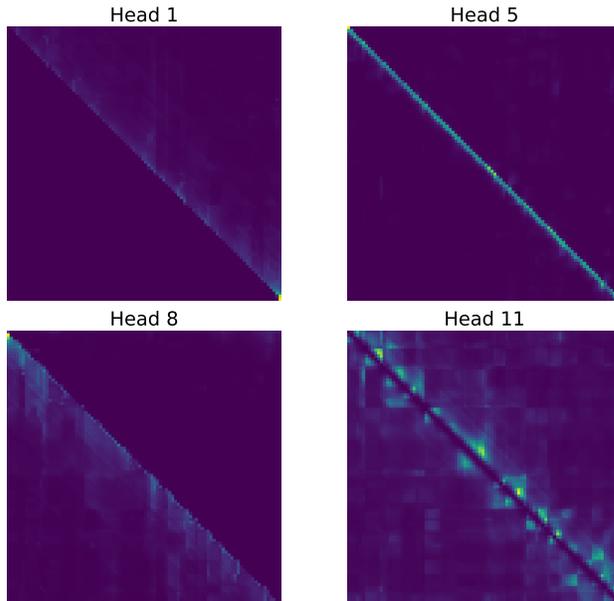


Figure 3. Encoder self-attention averaged over layers per attention head. We observe that while Head 5 strictly focuses on the current token (visible as attention along the diagonal), Heads 1 and 8 specialize in attending to past and future contexts, respectively. Head 11, on the other hand, exhibits a more complex pattern, attending broadly to the surrounding context beyond the immediate diagonal.

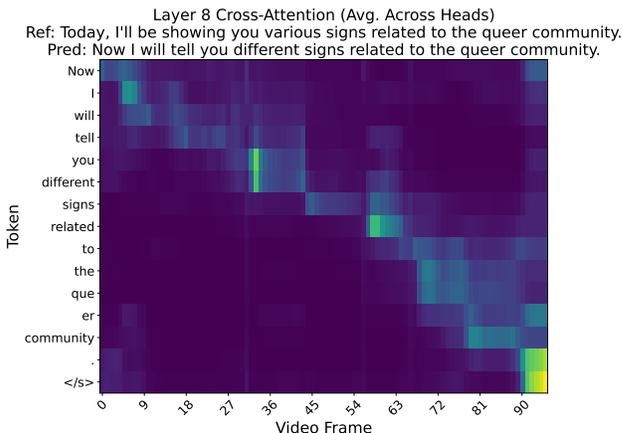
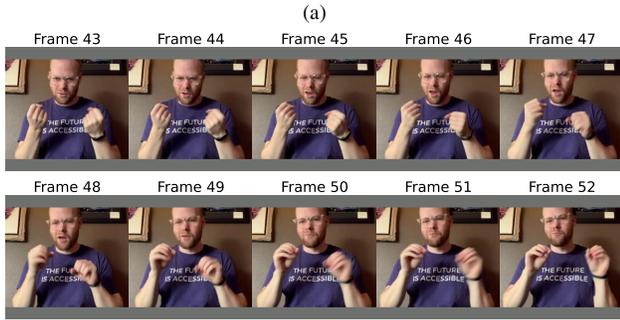
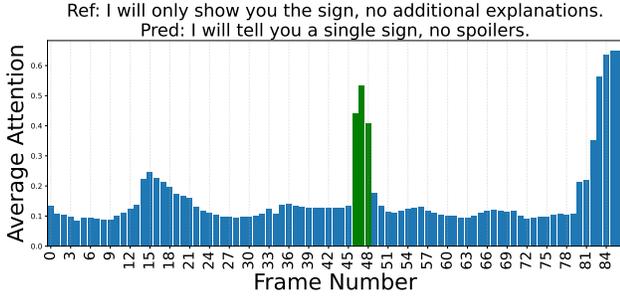


Figure 4. Cross-Attention averaged over all attention heads in a layer, showing temporal progression of tokens attending to frames.

located around a key-sign of the utterance where no transition between signs occurs. This is an expected behavior in the task of SL translation. However, this does not explain the consistent behavior of the high peaks at the end of the utterance observed in almost every clip.



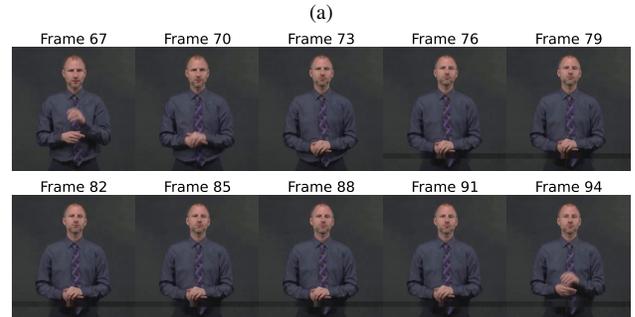
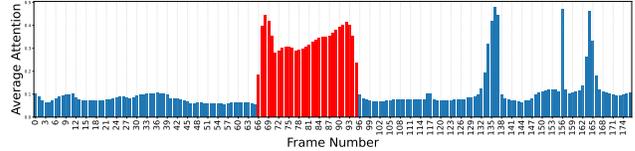
(b)

Figure 5. Histogram (a) visualizes Cross-Attention Distribution over all attention heads and layers, with an intensity spike in frames 46–48, highlighted in green. In (b), the corresponding video frames show the keyframe for the word “SIGN” matches the time of the cross-attention spike.

Upon further examination, we found clips that had high cross-attendances to long segments in various parts of the input, Figure 6a and more in supplementary. When we investigated these clips, we were surprised that the decoder was attending the part of the clip where no signing was performed. This led us to a hypothesis that the T5 model is using these non-informative segments to encode crucial information about the translation. This behavior has been already observed in previous works [3, 8] where they use register buffers as additional tokens to encode such information. In the work [8] the analysis is performed over images where the model is usually encoding important information in patches belonging to the background. This would be analogous to our observations and it might be helpful to use the same principle of adding register buffers to our translation model for better interpretability and generalization.

5.3. Integrated Gradients Analysis

Another standard approach to analyzing the model’s behaviors is an analysis of integrated gradients. In this paper, we utilized Captum library [21] to perform gradient analysis and assign attribution scores to input features. To be more specific, we used the Integrated Gradients tool, which accumulates gradients along a linear path from a baseline (in



(b)

Figure 6. Histogram (a) visualizes Cross-Attention Distribution over all attention heads and layers, with a long intensity spike sequence in frames 66–95, highlighted in red. Video frames (b) show this is a sequence of still, non-informative frames of a transition.

our case, an array of zeros) to the actual input, assigning an attribution score to each frame for the final prediction. These scores reveal which frames positively or negatively influence the model’s translations, among other things also supporting our observations in Sections 5.1 and 5.2. We used well-translated test samples only to clearly correlate positive attributions with high-quality translations. Positive attributions, therefore, indicate that certain frames aid in accurate translations, while negative scores may reflect noise or temporal misalignment; examples are shown below.

- **Reference:** “Today, I’ll be showing you various signs related to the queer community.”
- **Prediction:** “Now, I will tell you different signs related to the queer community.”

and the integrated gradients per output token per input frame are shown in Figure 7.

We observe behavior that is challenging to fully analyze, yet it is noteworthy that it has not been observed for the base (non-finetuned) model. A diagonal trend in integrated gradients is starting to occur. We set an experimental minimal threshold of 0.3 for visualization, see lower Figure in 7. Two clusters emerge for the tokens “signs” (around index 47) and “que” (around index 70). Punctuation marks (dots and commas) show near-zero contributions, suggesting that while the model retains T5’s textual and textual structure understanding, these punctuation marks are not semantically encoded in the input frames. This indicates that frame importance aligns with the temporal occurrence of signing, whereas off-diagonal patches may reflect contextual influences or incomplete model adaptation.

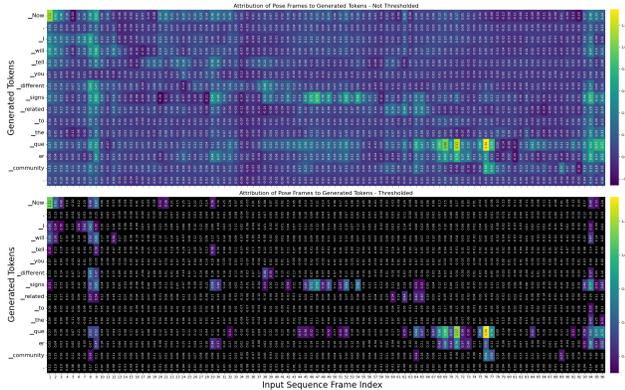


Figure 7. Attribution of Pose Frames to Generated Tokens (top): our finetuned T5 model for SLT translating a chosen phrase, (bottom): the identic model and phrase with minimal threshold of 0.3 to better showcase the diagonal trend.

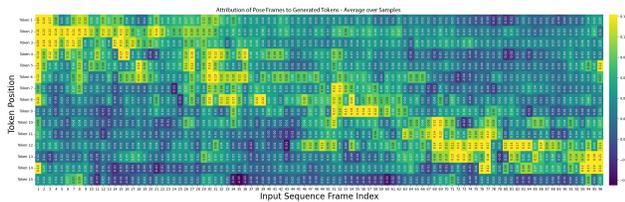


Figure 8. Attribution of Pose Frames to Generated Tokens - filtered average over multiple data samples

As a following step in the gradients analysis, we performed integrated gradients average over multiple relatively better translated data samples. These were chosen with the rule of a minimal BLEU-1 translation score of 10. In Figure 8 we observe not just a clear diagonal trend with some integrated gradient clustering tendencies (that aligns with the observation from a SignAttention study [1]), but also attributions of multiple last frames to some of the predicted token positions as already discussed in Section 5.2 and also seen from a single data sample analysis, Figure 7.

5.4. Analysis of Generalization Capabilities

In some cases we have observed that the predicted translations have surprisingly surpassed the reference ones. As YouTubeASL is a weakly-aligned dataset, not all translation labels (taken from video captions) are always correct. For example, the model correctly recognized and translated fingerspelling (Figure 9a) and the signs for numerals (Figure 9b), which were labeled incorrectly and not even present in the reference translation. The reference pushes the gradients in a wrong direction while the model is being optimized. It might be helpful to automatically re-label some dataset samples using machine translated pseudo-labels. Similar ideas were presented in many fields, for SLT notably in [46]. A mechanism that would be able to detect rel-

evant samples and decide which pseudo-labels to use would need to be implemented and will be the subject of our future work.

Ref: You can abbreviate it to NB, but you must have spelled it out first, then you can do NB.
 Pred: It can be abbreviated N-B, but it must be fingerspelled NB-I-N-A-R-Y and then fingerspelled.



(a)

Ref: to be tempted by the devil.
 Pred: So, at the same time, 40 days!



(b)

Figure 9. Example video frame sequences where the model has overcome wrong labels and correctly recognized (a) fingerspelling and (b) numerals.

6. Conclusion

This study systematically explored the impact of pose-based preprocessing techniques on Sign Language Translation while using a T5-based model. In extensive ablation studies, we demonstrated the importance of normalization, interpolation, and augmentation techniques. These techniques can significantly impact model robustness, mitigating signer variability and spatial inconsistencies. The ablation studies highlight the effectiveness of normalization based on signing space, interpolation of missing key-points, and suitable augmentation protocol. Moreover, attention analysis revealed valuable insights into model behavior, suggesting that register tokens could further enhance SLT performance.

In our future work, we would like to focus on incorporating register tokens and evaluating their influence on SLT accuracy. Furthermore, we would like to explore the possibility of using appearance-based features, such as MAE or DINO features, as additional input into the model.

Acknowledgment

The work has been supported by the grant of the University of West Bohemia, project No. SGS-2025-011. Computational resources were provided by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth and Sports of the Czech Republic.

References

- [1] Pedro Alejandro Dal Bianco, Oscar Agust n Stanchi, Facundo Manuel Quiroga, Franco Ronchetti, and Enzo Ferrante. Signattention: On the interpretability of transformer models for sign language translation. *arXiv preprint arXiv:2410.14506*, 2024. 2, 8
- [2] Maty s Boh cek and Marek Hr z. Sign pose-based transformer for word-level sign language recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, pages 182–191, 2022. 2, 3
- [3] Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. Quantizable transformers: Removing outliers by helping attention heads do nothing. *Advances in Neural Information Processing Systems*, 36:75067–75096, 2023. 7
- [4] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. Neural sign language translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1
- [5] Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. Multi-channel transformers for multi-articulatory sign language translation. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 301–319. Springer, 2020. 2
- [6] Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. Sign language transformers: Joint end-to-end sign language recognition and translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10023–10033, 2020. 1, 2
- [7] Yutong Chen, Fangyun Wei, Xiao Sun, Zhirong Wu, and Stephen Lin. A simple multi-modality transfer learning baseline for sign language translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5120–5130, 2022. 2
- [8] Timoth e Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers, 2024. 7
- [9] Aashaka Desai, Lauren Berger, Fyodor Minakov, Nessa Milano, Chinmay Singh, Kriston Pumphrey, Richard Ladner, Hal Daum  III, Alex X Lu, Naomi Caselli, et al. Asl citizen: a community-sourced dataset for advancing isolated sign language recognition. *Advances in Neural Information Processing Systems*, 36:76893–76907, 2023. 1
- [10] Amanda Duarte, Shruti Palaskar, Lucas Ventura, Deepti Ghadiyaram, Kenneth DeHaan, Florian Metze, Jordi Torres, and Xavier Giro i Nieto. How2sign: A large-scale multi-modal dataset for continuous american sign language, 2021. 1, 2
- [11] Sen Fang, Lei Wang, Ce Zheng, Yapeng Tian, and Chen Chen. Signllm: Sign languages production large language models. *arXiv preprint arXiv:2405.10718*, 2024. 2
- [12] Jia Gong, Lin Geng Foo, Yixuan He, Hossein Rahmani, and Jun Liu. Llm are good sign language translators, 2024. 2
- [13] Mo Guan, Yan Wang, Guangkun Ma, Jiarui Liu, and Mingzu Sun. Multi-stream keypoint attention network for sign language recognition and translation. *arXiv preprint arXiv:2405.05672*, 2024. 2
- [14] Dan Guo, Wengang Zhou, Houqiang Li, and Meng Wang. Hierarchical lstm for sign language translation. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 1
- [15] Hezhen Hu, Weichao Zhao, Wengang Zhou, Yuechen Wang, and Houqiang Li. Signbert: pre-training of hand-model-aware representation for sign language recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11087–11096, 2021. 1
- [16] Hezhen Hu, Weichao Zhao, Wengang Zhou, and Houqiang Li. Signbert+: Hand-model-aware self-supervised pre-training for sign language understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):11221–11239, 2023. 2
- [17] Eui Jun Hwang, Sukmin Cho, Huije Lee, Youngwoo Yoon, and Jong C Park. Universal gloss-level representation for gloss-free sign language translation and production. *arXiv preprint arXiv:2407.02854*, 2024. 2
- [18] Eui Jun Hwang, Huije Lee, and Jong C Park. A gloss-free sign language production with discrete representation. In *2024 IEEE 18th International Conference on Automatic Face and Gesture Recognition (FG)*, pages 1–6. IEEE, 2024. 2
- [19] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLO, 2023. 3
- [20] Rawal Khirodkar, Timur Bagautdinov, Julieta Martinez, Su Zhaoen, Austin James, Peter Selednik, Stuart Anderson, and Shunsuke Saito. Sapiens: Foundation for human vision models. In *European Conference on Computer Vision*, pages 206–228. Springer, 2024. 2
- [21] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, et al. Captum: A unified and generic model interpretability library for pytorch. *arXiv preprint arXiv:2009.07896*, 2020. 7
- [22] 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 *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1459–1469, 2020. 1
- [23] Dongxu Li, Xin Yu, Chenchen Xu, Lars Petersson, and Hongdong Li. Transferring cross-domain knowledge for video sign language recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6205–6214, 2020. 1
- [24] Han Liang, Chengyu Huang, Yuecheng Xu, Cheng Tang, Weicai Ye, Juze Zhang, Xin Chen, Jingyi Yu, and Lan Xu. Llava-slt: Visual language tuning for sign language translation, 2024. 2
- [25] Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742, 2020. 2
- [26] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuoling Chang, Ming Yong, Juhyun Lee, et al. 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. 3
- [27] JiHwan Moon, Jihoon Park, Jungeun Kim, Jongseong Bae, Hyeongwoo Jeon, and Ha Young Kim. Diffslr: Enhancing diversity in sign language translation via diffusion model, 2024. 2
- [28] JiHwan Moon, Jihoon Park, Jungeun Kim, Jongseong Bae, Hyeongwoo Jeon, and Ha Young Kim. Diffslr: Enhancing diversity in sign language translation via diffusion model. *arXiv preprint arXiv:2411.17248*, 2024. 2
- [29] Marina Perea-Trigo, Enrique J López-Ortiz, Luis M Soria-Morillo, Juan A Álvarez-García, and JJ Vegas-Olmos. Impact of face swapping and data augmentation on sign language recognition. *Universal Access in the Information Society*, pages 1–12, 2024. 2
- [30] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. 2, 4
- [31] Phillip Rust, Bowen Shi, Skyler Wang, Necati Cihan Camgöz, and Jean Maillard. Towards privacy-aware sign language translation at scale, 2024. 2
- [32] Bowen Shi, Diane Brentari, Greg Shakhnarovich, and Karen Livescu. Open-domain sign language translation learned from online video. In *EMNLP*, 2022. 1
- [33] Neil Song and Yu Xiang. Sigtformer: An attention-based approach to sign language recognition. *arXiv preprint arXiv:2212.10746*, 2022. 2
- [34] Garrett Tanzer and Biao Zhang. Youtube-sl-25: A large-scale, open-domain multilingual sign language parallel corpus, 2024. 1
- [35] David Uthus, Garrett Tanzer, and Manfred Georg. Youtube-asl: A large-scale, open-domain american sign language-english parallel corpus, 2023. 1, 2, 3
- [36] Ryan Wong, Necati Cihan Camgoz, and Richard Bowden. Sign2gpt: Leveraging large language models for gloss-free sign language translation, 2024. 2
- [37] Luke T Woods and Zeeshan A Rana. Modelling sign language with encoder-only transformers and human pose estimation keypoint data. *Mathematics*, 11(9):2129, 2023. 2
- [38] Chihiro Yano, Akihiko Fukuchi, Shoko Fukasawa, Hideyuki Tachibana, and Yotaro Watanabe. Multilingual sentence-t5: Scalable sentence encoders for multilingual applications. *arXiv preprint arXiv:2403.17528*, 2024. 2
- [39] Jinhui Ye, Xing Wang, Wenxiang Jiao, Junwei Liang, and Hui Xiong. Improving gloss-free sign language translation by reducing representation density. *arXiv preprint arXiv:2405.14312*, 2024. 2
- [40] Aoxiong Yin, Tianyun Zhong, Li Tang, Weike Jin, Tao Jin, and Zhou Zhao. Gloss attention for gloss-free sign language translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2551–2562, 2023. 2
- [41] Biao Zhang, Mathias Müller, and Rico Sennrich. Sltunet: A simple unified model for sign language translation. *arXiv preprint arXiv:2305.01778*, 2023. 2
- [42] Biao Zhang, Garrett Tanzer, and Orhan Firat. Scaling sign language translation, 2024. 2
- [43] Jian Zhao, Weizhen Qi, Wengang Zhou, Nan Duan, Ming Zhou, and Houqiang Li. Conditional sentence generation and cross-modal reranking for sign language translation. *IEEE Transactions on Multimedia*, 24:2662–2672, 2021. 2
- [44] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 23141–23150, 2023. 2
- [45] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20871–20881, 2023. 2
- [46] Hao Zhou, Wengang Zhou, and Houqiang Li. Dynamic pseudo label decoding for continuous sign language recognition. In *2019 IEEE International conference on multimedia and expo (ICME)*, pages 1282–1287. IEEE, 2019. 8
- [47] Hao Zhou, Wengang Zhou, Weizhen Qi, Junfu Pu, and Houqiang Li. Improving sign language translation with monolingual data by sign back-translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1316–1325, 2021. 2