APPENDIX
Read and Attend: Temporal Localisation in Sign Language Videos

Gül Varol[1,2*] Liliane Momeni[1*] Samuel Albanie[1*] Triantafyllos Afouras[1*] Andrew Zisserman[1]
[1] Visual Geometry Group, University of Oxford, UK
[2] LIGM, École des Ponts, Univ Gustave Eiffel, CNRS, France
{gul,liliane,albanie,afourast,az}@robots.ox.ac.uk
https://www.robots.ox.ac.uk/~vgg/research/bslattend/

This appendix to the main paper provides a discussion on broader impact (Sec. A), additional qualitative (Sec. B) and quantitative results (Sec. C), as well as implementation details (Sec. D).

A. Broader impact

At present, computer vision technology to usefully assist signers remains in its infancy. In large part, this stems from the high difficulty of achieving robust machine comprehension of sign language, which falls a long way short of human performance [2]. Our work, which focuses specifically on *sign localisation*, takes steps towards enabling several practical applications that may become viable even when a full automatic understanding of sign language remains incomplete. These include: (1) *sign language dictionary construction* to assist students who wish to learn sign language, (2) *index construction* for video corpora, allowing individuals to search videos by the content of their signing, (3) *wake-word spotting* for signing users of smart assistants like Alexa and Siri, (4) *tools for linguists* to assist in the efficient analysis of existing signing data and (5) *automatic large-scale dataset construction* to facilitate future research towards technology that will ultimately be able to provide useful products and services to the Deaf community.

The development of automatic, accurate sign localisation also has risks. Notably, It has the potential to be used for surveillance. Moreover, as with many computer vision methods employing deep neural networks (as ours does), the model is prone to fitting the training distribution closely. As a result, it will be vital that products and services employing this technology ensure that their users are well-represented in the training data to avoid a disparity of performance across groups.

*Equal contribution

B. Qualitative results

We refer to our supplementary video at our project webpage for additional visual results and illustrations. First, we visualise the attention scores for sample test videos, similarly to Fig. 4 of the main paper. To make it easier to assess the localisation quality visually, we show an example dictionary video corresponding to the localised sign. Next, we demonstrate the capability to temporally localise signs in long continuous videos using this attention mechanism on a training sequence. Finally, we present our automatic annotations on the training set (that we obtain through checking against subtitles), which we use for sign language recognition training. When grouping videos corresponding to the same word, we observe a temporal alignment across samples.

C. Additional experimental results

We provide additional quantitative analysis through experimentation with different subtitle preprocessing approaches (Sec. C.1), a detailed breakdown of performance for methods incorporating sparse annotations (Sec. C.2), additional decoding strategies to mine training examples (Sec. C.3), and a recognition architecture study (Sec. C.4).

C.1. Subtitle processing

All experiments in this section are reported on $\text{Test}_{7K}^{\text{Loc}}$ to evaluate the Transformer training for the sequence prediction task.

Stemming. We experiment with stemming versus using the original subtitle words in Tab. A.1. The 11K annotation stems correspond to a vocabulary of 16K words. We train a model by filtering the subtitles to these 16K words without any further processing. We observe no significant differences between the two models. Note that, for a fair comparison, we stem the words at evaluation time.

Vocabulary. In this work, we have used a vocabulary of 11K stems which is determined based on the annotations.

	vocab.	Recall	Prec.	Loc_{GD}	Loc_{TF}
Stemming	11K	16.5	37.2	66.1	44.5
No stemming	16K	16.0	35.5	66.8	52.3

Table A.1. **Stemming subtitles:** We find that stemming might not be necessary for training. Note that for both models, we stem the words at evaluation.

% of subtitle vocabulary	train vocab.	test vocab.	#test subtitles	Recall	Prec.	Loc_{GD}	Loc_{TF}
100%	26K	26K	7588	14.9	35.6	67.9	49.9
		11K(11K)	7497	15.9	36.0	68.0	50.2
75%	19K	19K	7567	15.2	36.0	67.7	38.9
		11K(10K)	7497	16.2	36.3	67.8	39.6
50%	13K	13K	7516	15.9	36.8	66.6	52.4
		11K(9K)	7497	16.7	36.9	66.7	52.3
25%	6K	6K	7271	17.0	40.0	66.3	52.0
		11K(6K)	7497	17.0	39.0	66.6	51.8
Annot. vocab.	11K	11K	7497	16.5	37.2	66.1	44.5

Table A.2. **Vocabulary size:** We systematically change the training vocabulary of stems by taking subsets of the full subtitle vocabulary. We take the top 25%, 50%, 75%, 100% of stems according to their frequencies in the subtitles. Each trained model is tested twice (two rows per model): with (a) the same vocabulary used for training, (b) the comparable 11K vocabulary used in the rest of the experiments. Note that in (b), there might not be a full overlap between the train and test; the numbers in parenthesis represent the intersection.

	vocab.	Recall	Prec.	Loc_{GD}	Loc_{TF}
Without stop words	11K	16.5	37.2	66.1	44.5
With stop words	11K	13.9	25.9	69.5	52.5

Table A.3. **Removing stop words:** We train a model by including the stop words (although these rarely have corresponding signs), and obtain lower performance (13.9% recall).

In Tab. A.2, we train additional Transformer models by using vocabularies determined by the subtitles. We sort the stems appearing in all subtitles based on their frequencies. We train with top 25%, 50%, 75%, 100% of all stems. We observe that the models are not very sensitive to the choice of training vocabulary. Note that in all cases, we filter out the stop words which do not have sign correspondences.

Stop words. In Tab. A.3, we train one model by keeping the stop words and compare against our model. Note that we determine the list of stop words according to English stop words in the `nltk.corpus`. Qualitatively, we observe frequent occurrence of the words "and" and "to" in the predictions. The precision and recall metrics reflect the reduced quality of the outputs as well. Therefore, we filter out the stop words in all other models.

Naive translation. Tab. A.4 reports results for training a model with a large vocabulary, without filtering and without stemming. We again stem the words at evaluation time. We observe poor performance and highlight the difficulty of the translation problem on in-the-wild sign language data. A few qualitative predictions are provided below. We note that while some examples have overlap between ground truth and prediction (#1, #2, #3), many examples repeat the same prediction (#4, #5), or output frequent words (#6). As argued in the discussion section of the main paper (Sec. 4.6), we believe that the video-text alignment and large-vocabulary sign recognition problems should become more advanced to achieve in-the-wild translation.

train vocab.	test vocab.	#test subtitles	Recall	Prec.	Loc_{GD}	Loc_{TF}
40K	40K	7413	9.5	7.5	70.3	27.1
40K	16K	7299	10.3	7.5	70.3	27.0

Table A.4. **Naive translation with 40K vocabulary:** We report the results of training a model without stemming and without vocabulary filtering (except we filter to the 260K English vocabulary of Transformer-XL to remove the noise in the subtitles, due to OCR mistakes etc.). We test the model on (a) the same 40K vocabulary used for training, and (b) the 16K subset covering the annotations. Overall, we observe poor precision and recall.

```
Example #1
Reference:  through your own admission your last
time in the competition
Hypothesis: the competition is a competition

Example #2
Reference:  lots of water to help digest such a meal
Hypothesis: and then the water is the water

Example #3
Reference:  people talk you see
Hypothesis: i think the people were a good

Example #4
Reference:  and just tease out the dead growth
Hypothesis: and the whole thing is a little bit

Example #5
Reference:  and how little we knew about the species
Hypothesis: and the whole thing

Example #6
Reference:  wrong here again going to give it how many
Hypothesis: i think that is a good
```

C.2. Incorporating sparse annotations

As in Sec. C.1, all experiments in this section are reported on $\text{Test}_{7K}^{\text{Loc}}$.

Alignment loss on sparse annotations. Tab. A.5 presents detailed results on the incorporation of the alignment loss as described in Sec. 4.3 of the main paper. Although minor improvements are observed with the addition of such a loss term, we do not use it in the final model for simplicity.

Curriculum learning with sparse annotations. Tab. A.6 reports results with and without the curriculum strategy described in Sec. 4.3 of the main paper. We obtain minor improvements with pretraining the Transformer on shorter temporal segments containing only 1 annotated sign, finetuning the model later on 2 and 3 signs. Note that this model uses 1 layer in both encoder and decoder unlike other experiments which use 2 layers (we note from our Transformer

$\lambda_{\mathcal{L}_{align}}$	L	Recall	Prec.	Loc. Acc. (GD) layer 1/2	[avg]	Loc. Acc. (TF) layer 1/2	[avg]
0	-	16.5	37.2	63.9/57.8	[66.1]	51.1/37.6	[44.5]
10	avg	**16.8**	37.5	64.7/59.2	[66.0]	51.4/36.1	[45.2]
100	avg	16.4	37.2	67.4/60.7	[68.0]	52.8/42.3	[48.0]
1000	avg	14.4	34.5	68.9/59.0	[67.3]	52.5/55.8	[56.6]
10	1	**16.8**	38.3	62.9/63.8	[66.4]	51.2/40.7	[43.1]
100	1	16.7	37.3	67.5/63.6	[66.7]	52.9/38.5	[42.1]
1000	1	15.7	33.5	59.4/69.8	[69.6]	57.0/35.6	[48.7]
10	2	**16.8**	37.4	65.8/59.2	[67.3]	51.7/36.8	[47.1]
100	2	16.2	37.7	**68.5**/57.2	[68.0]	46.0/50.4	[47.5]
1000	2	14.8	35.5	67.1/59.2	[66.3]	42.5/**58.7**	[53.6]

Table A.5. **Alignment loss on sparse annotations:** We experiment with different weighting terms for the alignment loss ($\lambda_{\mathcal{L}_{align}}$) in addition to the classification loss during subtitle training. We define the loss on various attention layers (L) of a 2-layer architecture. We observe minor improvements.

Training schedule	Recall	Prec.	Loc_{GD}	Loc_{TF}
No curriculum: Subtitle	15.8	36.4	65.9	44.8
With curriculum: 1→2→3→Subtitle	16.0	37.1	66.6	44.3

Table A.6. **Curriculum learning with sparse annotations:** We observe minor improvements by incorporating curriculum learning, which gradually extends the temporal window of the input video. Note that a 1-layer encoder-decoder architecture is used for this experiment.

Training subtitles	Recall	Prec.	Loc_{GD}	Loc_{TF}
662K not aligned	6.8	16.3	67.1	27.0
183K not aligned (subset)	6.2	15.0	65.4	25.5
230K aligned	15.4	38.3	66.7	51.5
301K coarse (pad ±2-sec)	13.9	45.2	67.6	48.3
183K coarse	16.5	37.2	66.1	44.5

Table A.7. **Subtitle-video alignment:** Our coarse alignment, which uses the assumption that the subtitles that have at least one annotation within the subtitle timestamp is aligned with its video, obtains the best performance over other alignment variants or using no alignment.

layer ablations reported the main paper that this does not dramatically affect localisation performance).

Video-subtitle alignment. Tab. A.7 details our experiments which highlight the importance of video-subtitle alignment. When using all subtitles without considering whether they contain an annotation or not (662K subtitles), we obtain poor recall on the test set where there is at least one high-confidence ($\geq$0.9) annotation. To keep the number of training subtitles same as our final model, we also experiment with taking a random subset of 183K subtitles, and observe a similar outcome of poor performance. When using active signer detection and sparse annotations to apply a simple algorithm to align the subtitles, we get to 230K training subtitles that have at least 1 annotation; however, this model does not impact the results significantly. To take the uncertainty into account, we also experiment with padding ±2 seconds at the start and end of the subtitle times to input more video features to the model. However, this model also reduces the recall. Our simple coarse alignment strategy of using subtitles that have at least 1 annotation results in the best performance.

Spotting mode	#subtitles unannot.	#ann. 11K	#ann. 1K	top-1 per-inst	top-1 per-cls
TF ($\geq$ .05)	457K	2.3M	754K	38.7	14.4
BS (10 best)	109K	329K	166K	49.6	22.7
BS (1 best)	109K	316K	161K	50.7	23.3
TF prediction	57K	195K	110K	51.3	22.5
GD	53K	188K	107K	53.9	24.7

Table A.8. **Other decoding strategies:** TF: teacher-forced decoding, filtering with a threshold on attention scores; BS (10 best): beam search decoding with beam size 10 - all returned hypotheses are used when looking for new instances; BS (1 best): the same beam search is performed (with beam size 10) but only the hypothesis with the highest recall is used; TF prediction: teacher-forced decoding, using the hypothesis predicted by the model and checking against subtitle; GD: greedy decoding. The strategies shown in bold font refer to experiments not included in the main paper and are described in more detail in Sec. C.3. For the rest, we refer to Sec. 4.3 of the main paper.

C.3. Decoding strategies

In Sec. 4.3 of the main paper, we have described different decoding mechanisms to mine new training annotations by applying the Transformer model. Here, we provide two more strategies to complement Tab. 3 (a) **1 best:** choosing the hypothesis with the highest recall when applying beam search with size 10, (b) **TF prediction:** decoding with teacher forcing and forming a hypothesis using the model's prediction at every step, then filtering the hypothesis by only keeping the tokens that are also present in the corresponding subtitle (same as with GD); this is an alternative form of the TF baseline – here we also use the attentions of the first layer only. The new results are denoted with bold font in Tab. A.8 for $\text{Test}^{\text{Rec}}_{37K}$. The best result is achieved by simple greedy decoding (GD) which has smaller but more noise-free sign localisations.

Fig. A.1 shows several training plots corresponding to different decoding mechanisms. The curves suggest that mining more examples with higher noise results in low training performance. The plotted metric is top-1 per instance accuracy over 30 training epochs.

C.4. Recognition architecture study

We present an experimental study for the architecture design of our MLP which is used for recognition. Tab. A.9 summarises the results on $\text{Test}^{\text{Rec}}_{37K}$ for training with all M+D+A annotations, i.e., our best model in Tab. 4 of the main paper. While the results are not significantly different, we observe minor improvements with increased capacity, which quickly saturates when adding more layers.

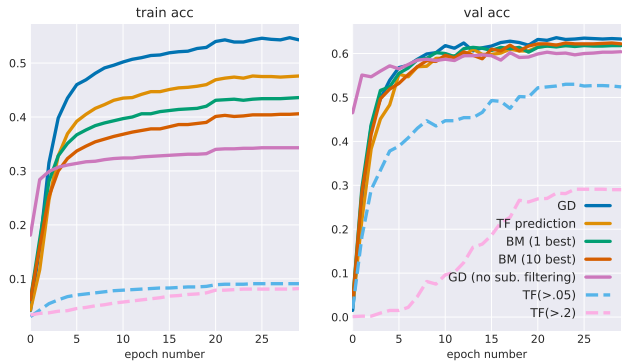

Figure A.1. **Training recognition with attention spottings:** We plot the training (left) and validation (right) accuracy curves against the epoch number for different MLP models, corresponding to different decoding strategies to mine training examples. The legend corresponds to descriptions in Tab. A.8 and Tab. 3 of the main paper. See Sec. C.3 and Sec. 4.3 of the main paper for details. We conclude that increased noise in teacher forcing mechanism (dashed), despite its large yield, makes learning difficult.

Architecture	per-instance top-1	per-instance top-5	per-class top-1	per-class top-5
1024→res(1024)→512→256	65.1	82.6	38.0	56.4
1024→512→256	64.6	82.3	36.5	55.1
1024→256	63.8	82.0	35.2	53.9
1024→res(1024)→512→128	64.9	82.6	37.4	55.9
1024→res(1024)→512→512	65.3	82.6	38.4	57.0
1024→res(1024)→512→1024	**65.5**	**82.9**	**39.0**	**57.4**
1024→res(1024)→512	65.3	**82.9**	37.9	56.6
1024→res(1024)→512→512→1024	**65.5**	82.7	**39.0**	**57.4**
1024→res(1024)→512→512→512→1024	65.0	82.4	38.3	57.1

Table A.9. **Architecture study for recognition:** We experiment with different number of layers for the sign recognition model. The input dimensionality is 1024, which is a temporally-averaged I3D embedding over 16 frames. The output is 1064-dimensional class probabilities. The top row is what is reported in the main paper (corresponding to Fig. A.2). We observe minor improvements by increasing the network capacity.

D. Implementation details

D.1. Application of M [1] and D [3]

As explained in Sec. 4.1, we apply the method of [1] to localise signs through mouthing cues on a large vocabulary of words beyond 1K (which is used in the original work). In particular, we query 36K words, and out of these, a vocabulary of 15K words are localised with confidence above 0.7. When applying the method of [3] to localise signs through similarity matching with dictionary videos, we query 9K signs from the full BSLDict dataset with search windows of ± 4 seconds padding around the subtitle timestamps. The resulting sign localisations with confidence above 0.7 cover a vocabulary of 4K words. The combination of these two methods gives us a total vocabulary of 16K words, which results in the 11K stems used for our Transformer training.

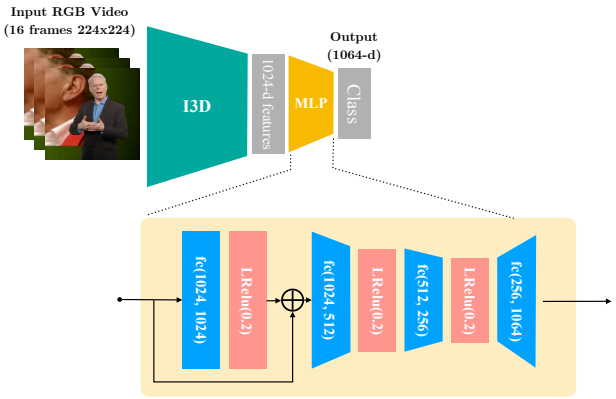

Figure A.2. **MLP architecture for recognition:** We follow [4], and use one residual connection, followed by 3 more fully connected layers on top of the I3D pre-extracted features.

D.2. Architecture and training details

Our Transformer model consists of 2 attention layers for both encoding and decoding. The input 1024-dimensional video feature is mapped to 512 dimensions with a linear layer. Then 512-d embeddings are used both for output words and input videos. We use 2 heads in each attention layer.

When reporting localisation accuracy, we average the encoder-decoder attention scores over the 2 heads. We take the first layer attention for teacher forcing (TF) and the average over two layers for greedy decoding (GD). We mark a correct localisation if the maximum location over the input video is within ± 2 feature frames from the annotation time. This is because one sign approximately lasts for 7-13 frames (at 25fps) [4] and our features are extracted with a stride of 4 frames, making our valid window duration ± 8 video frames. This also accounts for some uncertainty in the 'ground-truth' annotation times which are obtained automatically.

We detail our MLP architecture in Fig A.2. We use a design similar to [4]. The architecture study in Sec. C.4 reports variations of this model. We train it for 30 epochs, with an initial learning rate of $1e^{-2}$ reduced by a factor 10 at the 20th and 25th epoch.

D.3. Infrastructure

We use Nvidia M40 graphics cards for our experiments. The video-subtitle Transformer model trains in 10 hours on a single GPU. The annotation mining time is roughly 30 minutes to obtain 107K annotations, i.e., Transformer forward pass runtime over 1302h of training videos (duration of 685K subtitles padded with ± 2 seconds) on a single GPU. The final best MLP (M+D+A) for sign recognition trains in 7 hours on a single GPU. The M+D I3D backbone is trained with 4 GPUs over a duration of 1 week.

References

[1] S. Albanie, G. Varol, L. Momeni, T. Afouras, J. S. Chung, N. Fox, and A. Zisserman. BSL-1K: Scaling up co-articulated sign language recognition using mouthing cues. In *ECCV*, 2020. 4

[2] D. Bragg, O. Koller, M. Bellard, L. Berke, P. Boudreault, A. Braffort, N. Caselli, M. Huenerfauth, H. Kacorri, T. Verhoef, C. Vogler, and M. Ringel Morris. Sign language recognition, generation, and translation: An interdisciplinary perspective. In *ACM SIGACCESS*, 2019. 1

[3] L. Momeni, T. Afouras, T. Stafylakis, S. Albanie, and A. Zisserman. Seeing wake words: Audio-visual keyword spotting. In *BMVC*, 2020. 4

[4] L. Momeni, G. Varol, S. Albanie, T. Afouras, and A. Zisserman. Watch, read and lookup: learning to spot signs from multiple supervisors. In *ACCV*, 2020. 4