

APPENDIX

Watch, read and lookup: learning to spot signs from multiple supervisors

Liliane Momeni*, Gül Varol*, Samuel Albanie*,
Triantafyllos Afouras, and Andrew Zisserman

Visual Geometry Group, University of Oxford, UK
{liliane,gul,albanie,afouras,az}@robots.ox.ac.uk

This appendix provides additional qualitative (Sec. A) and experimental results (Sec. B), as well as detailed explanations of the training of our Watch-Read-Lookup framework (Sec. C).

A Qualitative Results

Please watch our video in the project webpage¹ to see qualitative results of our model in action. We illustrate the sign spotting task, as well as the specific applications considered in the main paper: sign variant identification, densification of annotations, and “faux amis” identification between languages.

B Additional Experiments

In this section, we present complementary experimental results to the main paper. We report the variance of the results over multiple random seeds (Sec. B.1), the effect of class-balancing (Sec. B.2), domain-specific layers (Sec. B.3), language-aware negative sampling (Sec. B.4), sliding window stride at test time (Sec. B.5), the mouthing score threshold (Sec. B.6), and the trunk network architecture (Sec. B.7).

B.1 Variance of results

We repeat the experiments in Tables 2 and 3 of the main paper, with multiple random seeds for each model and report means and standard deviations in Tab. A.1 and Tab. A.2 to provide a measure of the variance of the results. We observe that the results are consistent with those reported in the main paper.

B.2 Class-balanced sampling

As described in the main paper, we construct each batch by maximizing the number of negative pairs. To this end, we include one labelled sample per word when

* Equal contribution

¹ <https://www.robots.ox.ac.uk/~vgg/research/bsldict/>

Supervision	Train (1064)		Train (800)	
	Seen (264)		Unseen (264)	
	mAP	R@5	mAP	R@5
Classification	37.13 \pm 0.29	39.68 \pm 0.57	10.33 \pm 0.43	13.33 \pm 1.11
NCE	43.59 \pm 0.76	52.59 \pm 0.75	11.40 \pm 0.42	14.76 \pm 0.40
Watch-Lookup	44.72 \pm 0.85	55.51 \pm 2.17	11.02 \pm 0.27	15.03 \pm 0.45
Watch-Read-Lookup	47.93 \pm 0.20	60.76 \pm 1.45	14.86 \pm 1.29	19.85 \pm 1.94

Table A.1: **Variance of the results with multiple random seeds:** We repeat the Table 2 experiments of the main paper, with three random seeds for each model and report the mean and the standard deviation.

Supervision	Dictionary Vocab	mAP	R@5
Watch-Read-Lookup	800 training vocab	14.86 \pm 1.29	19.85 \pm 1.94
Watch-Read-Lookup	9k full vocab	15.82 \pm 0.48	21.67 \pm 0.72

Table A.2: **Variance of the results with multiple random seeds:** We repeat the Table 3 experiments of the main paper with three random seeds for each model and report the mean and the standard deviation.

Class-balancing	Batch size	mAP	R@5
x	512	41.65	54.73
x	1024	42.07	54.25
x	2048	43.14	54.28
✓	512	43.65	53.03
✓	1024	43.55	54.20

Table A.3: **Class-balancing:** In the main paper, we class-balance our minibatches by including one sample per word from the labelled continuous sequences, thus maximizing the number of negatives within a batch. Here, we investigate removing such class-balancing constraint. In that case, we make sure we do not mark samples with the same labels as negatives, instead we discard them. We experiment with various batch sizes, also going beyond the total number of classes (2048). We observe that the performance is not significantly affected by these changes. (training on the full 1064 vocabulary with Watch-Lookup)

sampling continuous sequences, i.e., class-balancing the minibatches. Thus, all but one of the labelled samples in the batch can be used as negatives for a given dictionary bag corresponding to a labelled sample. Note that this approach limits the batch size to be less than or equal to the number of sign classes. Tab. A.3 experiments with the sampling strategy. We observe that the performance is not significantly different with/without class-balanced sampling for various batch sizes.

Domain-specific layers	mAP	R@5
✓	43.58	53.54
✗	43.65	53.03

Table A.4: **Domain-specific layers:** We experiment with separating the MLP layers to be specific to the continuous and isolated domains. We do not observe any significant difference in performance and therefore adopt a shared MLP for simplicity in all experiments. (Training on the full 1064 vocabulary with Watch-Lookup)

B.3 Domain-specific layers

As noted in the main paper, the videos from the continuous signing and from the dictionaries differ significantly, e.g., continuous signing data is faster than the dictionary signing, and is co-articulated whereas the dictionary has isolated signs. Given such a domain gap, we explore whether it is beneficial to learn domain-specific MLP layers: one for the continuous, and one for the dictionary. Tab. A.4 presents a comparison between domain-specific layers versus shared parameters. We do not observe any gains from such separation. Therefore, we keep a single MLP for both domains for simplicity.

B.4 Language-aware negative sampling

Working with a large vocabulary of words brings the additional challenge of handling synonyms. We consider two types of similarities. First, two different categories in the BSLDICT sign dictionary may belong to the same sign category if the corresponding English words are synonyms. Second, the meta-data we have collected with the BSLDICT dataset provides similarity labels between sign categories, which may be used to group certain signs. In this work, we have largely ignored this issue by associating each sign to a single word. This results in constructing negative pairs for two identical signs such as ‘happy’ and ‘content’. Here, we explore whether it is beneficial to discard such pairs during training, instead of marking them as negatives. Tab. A.5 reports the results. We observe marginal gains with discarding synonyms. However, given the insignificant difference, we do not make such separation in other experiments for simplicity.

B.5 Effect of the sliding window stride

As explained in the main paper, at test time, we extract features from the continuous signing sequence using a sliding window approach with 1 frame as the stride parameter. Our window size is 16 frames, i.e., the number of input frames for the I3D feature extractor. Here, we investigate the effect of the stride parameter. We apply a stride of 8 frames as a comparison. Tab. A.6 shows that a stride of 1 frame is critical to perform precise sign spotting. This can be explained by the fact that sign duration is typically between 7-13 frames (but

Negative sampling	mAP	R@5
Discarding English synonyms	43.27	54.24
Discarding Sign synonyms	45.03	54.19
Keeping all	43.65	53.03

Table A.5: **Language-aware negative sampling:** We explore the use of external knowledge such as English synonyms or the meta-data of the dictionary denoting similar sign categories. We experiment with discarding such similar word pairs, excluding them from both positive and negative pairs. The last row instead marks any pair as negative if their corresponding words are not identical. We observe only marginal gains with the use of external knowledge about the languages. (Training on the full 1064 vocabulary with Watch-Lookup)

Stride	mAP	R@5
8	38.46	47.38
1	43.65	53.03

Table A.6: **Stride parameter of sliding window:** A small stride at test time, when extracting embeddings from the continuous signing video, allows us to temporally localise the signs more precisely. The window size is 16 frames and the typical co-articulated sign duration is 7-13 frames (at 25 fps). (testing 1064-class model trained with Watch-Lookup)

can be shorter) [1] in continuous signing video, and a stride of 8 may skip the most discriminative moment.

B.6 Mouthing confidence threshold at training

The sparse annotations from the BSL-1K dataset are obtained by running a visual keyword spotting method based on mouthing cues. Therefore, the dataset provides a confidence value associated with each label ranging between 0.5 and 1.0. Similar to [2], we experiment with different thresholds to determine the training set. Lower thresholds result in a noisier but larger training set. From Tab. A.7, we conclude that 0.5 mouthing confidence threshold performs the best. This is in accordance with the conclusion from [2].

B.7 Trunk network architecture: S3D vs I3D

As shown in Tab A.8, we compare two popular architectures for computing video representations. We have used I3D [3] in all our experiments. Here, we also train a 1064-way classification with the S3D architecture [4] on BSL-1K as in [2] for sign language recognition. We do not observe improvements with S3D (in practice we found that it overfit the training set to a greater degree); therefore, we use an I3D trunk. Note that the hyperparameters (e.g., learning rate) are tuned for I3D and kept the same for S3D.

Mouthing confidence	Training size	mAP	R@5
0.9	10K	37.55	47.54
0.8	21K	39.49	48.84
0.7	33K	41.87	51.15
0.6	49K	42.44	52.42
0.5	78K	43.65	53.03

Table A.7: **Mouthing confidence threshold:** The results suggest that lower confidence automatic annotations of BSL-1K provide better training, by increasing the amount of data (training on the full 1064 vocabulary with Watch-Lookup).

Training data	per-instance		per-class	
	top-1	top-5	top-1	top-5
S3D	64.76	81.88	46.27	63.71
I3D [2]	75.51	88.83	52.76	72.14

Table A.8: **Trunk network architecture:** We compare I3D [3] with the S3D [4] architecture for the task of sign language recognition, in a comparable setup to [2]. We use the last 20 frames before the mouthing annotations with confidence above 0.5. We do not obtain gains with the S3D architecture; therefore, we use I3D in all the experiments to compute video features.

C Training Details

In this section, we cover architectural details (Sec. C.1), a detailed formulation of our positive/negative bag sampling strategy (Sec. C.2) and a brief description of the infrastructure used to perform the experiments in the main paper (Sec. C.3).

C.1 Architectural details

As explained in the main paper, our sign embeddings correspond to the output of a two-stage architecture: (i) an I3D trunk, and (ii) a three-layer MLP. We first train the I3D on both labelled continuous video clips and the dictionary videos jointly. We then freeze the I3D trunk and use it as a feature extractor. We only train the MLP layers with our loss formulation in the Watch-Read-Lookup framework.

I3D trunk. We first train the I3D parameters only with the BSL-1K annotated clips that have mouthing confidences more than 0.5. For 1064-class training, we use the model from [2] provided by the authors; for 800-class training, we perform our own training, also first pretraining with pose distillation.

We then *re-initialise the batch normalization layers* (as noted in Sec. 2 of the main paper). We fine-tune the model jointly on BSL-1K annotated clips (the

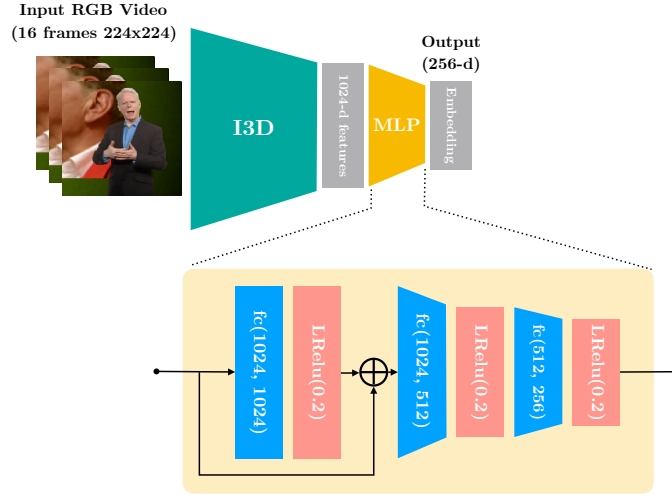


Fig. A.1: **MLP architecture:** We detail the layers of our embedding architecture. We freeze the I3D trunk and use it as a feature extractor. We only train the MLP layers with our loss formulation in the proposed framework. The same layers (and parameters) are used both for the dictionary video inputs and the continuous signing video inputs.

ones with mouthing confidence more than 0.8) and BSLDICT samples. The sampling frequency for the two data sources are balanced. In the I3D classification pretraining phase, we treat each dictionary video independently with its corresponding label. We observe that the 1064-way classification performance on the *training* dictionary videos remain at 48.09% per-instance top-1 accuracy without the batch normalization re-initialization, as opposed to 78.94%. We also experimented with domain-specific batch normalization layers [5], but the training accuracy for the dictionary videos was still low (62.73%).

As detailed in Sec. 3.2 of the main paper, we subsample the dictionary videos to roughly match their speed to the continuous signing videos. This subsampling includes a *random shift* and a *random fps*. We observe a decrease of 6.68% in the training dictionary classification accuracy if we instead sample 16 consecutive frames from the original temporal resolution, which is not sufficient to capture the full extent of a sign because one dictionary video is 56 frames on average.

MLP. Fig. A.1 illustrates the layers considered for our MLP architecture. It consists of 3 fully connected layers with LeakyRelu activations between them. The first linear layer also has a residual connection on the 1024-dimensional input features. We then reduce the dimensionality gradually to 512 and 256 for efficient training and testing.

C.2 Positive/Negative bag sampling formulations

In the main paper, we described two approaches for sampling positive/negative MIL bags in Sec. 3.1. Due to space constraints, the sampling mechanisms were described at a high-level. Here, we provide more precise definitions of each bag. In addition to the set notation below, we include in the supplementary material, the loss implementation as a PyTorch [6] function in `code/loss.py`, together with a sample input (`code/sample_inputs.pkl`) comprising embedding outputs from the MLP for continuous and dictionary videos.

As noted in the main paper, we do not have access to positive pairs because: (1) for the segments of videos in \mathcal{S} that are annotated (i.e. $(x_k, v_k) \in \mathcal{M}$), we have a set of potential sign variations represented in the dictionary (annotated with the common label v_k), rather than a single unique sign; (2) since \mathcal{S} provides only weak supervision, even when a word is mentioned in the subtitles we do not know where it appears in the continuous signing sequence (if it appears at all). These ambiguities motivate a Multiple Instance Learning [7] (MIL) objective. Rather than forming positive and negative pairs, we instead form positive *bags* of pairs, $\mathcal{P}^{\text{bags}}$, in which we expect at least one segment from a video from \mathcal{S} (or a video from \mathcal{M} when labels are available) and a video \mathcal{D} to contain the same sign, and negative bags of pairs, $\mathcal{N}^{\text{bags}}$, in which we expect no pair of video segments from \mathcal{S} (or \mathcal{M}) and \mathcal{D} to contain the same sign. To incorporate the available sources of supervision into this formulation, we consider two categories of positive and negative bag formations, described next. Each bag is formulated as a set of paired indices—the first value indexes into the collections of continuous signing videos (either \mathcal{S} or \mathcal{M} , depending on context) and the second value indexes into the set of dictionary videos contained in \mathcal{D} .

Watch and Lookup: using sparse annotations and dictionaries. In the first formulation, *Watch-Lookup*, we only make use of \mathcal{D} and \mathcal{M} (and not \mathcal{S}) to learn the data representation f . We define positive bags in two ways: (1) by anchoring on the labelled segment

$$\mathcal{P}_{\text{watch,lookup}}^{\text{bags(seg)}} = \{\{i\} \times B_i : (x_i^{\mathcal{M}}, v_i^{\mathcal{M}}) \in \mathcal{M}, (x_j^{\mathcal{D}}, v_j^{\mathcal{D}}) \in \mathcal{D}, B_i = \{j : v_j^{\mathcal{D}} = v_i^{\mathcal{M}}\}\} \quad (1)$$

i.e. each bag consists of a labelled temporal segment and the set of sign variations of the corresponding word in the dictionary (illustrated in Fig. A.2 (i), top row), or by (2) anchoring on the dictionary samples that correspond to the labelled segment, to define a second set $\mathcal{P}_{\text{watch,lookup}}^{\text{bags(dict)}}$, which takes a mathematically identical form to $\mathcal{P}_{\text{watch,lookup}}^{\text{bags(seg)}}$ (i.e. each bag consists of the set of sign variations of the word in the dictionary that corresponds to a given labelled temporal segment, illustrated in Fig. A.2 (ii), top row). The key assumption in both cases is that each labelled segment matches *at least one* sign variation in the dictionary. Negative bags can be constructed by (1) anchoring on labelled segments and selecting dictionary examples corresponding to different words (Fig. A.2 (i), top-left and bottom-right); (2) anchoring on the dictionary set for a given word

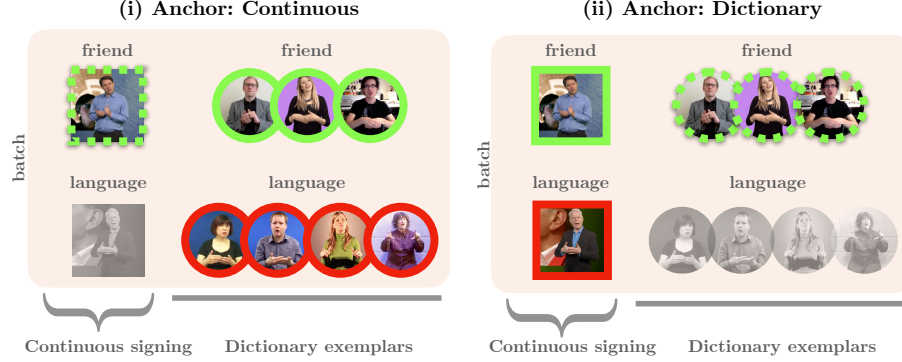


Fig.A.2: **Watch-Lookup**: We illustrate the batch formation and positive/negative sampling for the simplified version of our framework which is not using the subtitles, but only performing Watch-Lookup. We define two sets of positive/negative pairs, anchoring at a different position in each case. Anchor is denoted with dashed lines, positive samples with solid green, negative samples with solid red lines. Gray samples are discarded. (i) anchors at a labelled continuous video, making the dictionary samples for the labelled word a positive bag, and all other dictionary samples in the batch a negative bag. (ii) anchors at a bag of dictionary samples, making the corresponding continuous labelled video positive, and all others in the batch negatives. We refer to Fig A.3 for the illustration of our Watch-Read-Lookup extension.

and selecting labelled segments of a different word (Fig. A.2 (ii), top-right and bottom-left). These sets manifest as

$$\mathcal{N}_{\text{watch,lookup}}^{\text{bags(seg)}} = \{\{i\} \times B_i : (x_i^{\mathcal{M}}, v_i^{\mathcal{M}}) \in \mathcal{M}, (x_j^{\mathcal{D}}, v_j^{\mathcal{D}}) \in \mathcal{D}, B_i = \{j : v_j^{\mathcal{D}} \neq v_i^{\mathcal{M}}\}\} \quad (2)$$

for the former and as

$$\mathcal{N}_{\text{watch,lookup}}^{\text{bags(dict)}} = \{A_i \times B_i : A_i = \{l : x_l, x_i \subseteq x_k, (x_k, s_k) \in \mathcal{S}, x_l \cap x_i = \emptyset\} \quad (3)$$

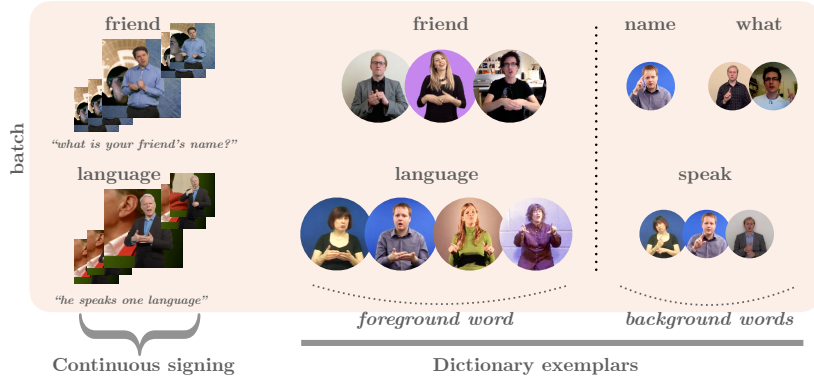
$$B_i = \{j : v_j^{\mathcal{D}} \neq v_i^{\mathcal{M}}\}, (x_i^{\mathcal{M}}, v_i^{\mathcal{M}}) \in \mathcal{M}, (x_j^{\mathcal{D}}, v_j^{\mathcal{D}}) \in \mathcal{D}\}.$$

for the latter. The complete set of positive and negative bags is formed via the unions of these collections:

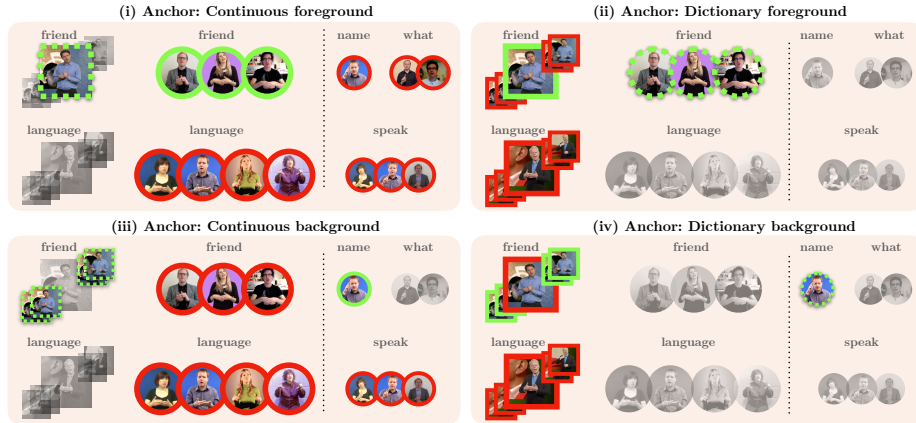
$$\mathcal{P}_{\text{watch,lookup}}^{\text{bags}} \triangleq \mathcal{P}_{\text{watch,lookup}}^{\text{bags(seg)}} \cup \mathcal{P}_{\text{watch,lookup}}^{\text{bags(dict)}} \quad (4)$$

and

$$\mathcal{N}_{\text{watch,lookup}}^{\text{bags}} \triangleq \mathcal{N}_{\text{watch,lookup}}^{\text{bags(seg)}} \cup \mathcal{N}_{\text{watch,lookup}}^{\text{bags(dict)}}. \quad (5)$$



(a) **Input:** We illustrate an example minibatch formation for our Watch-Read-Lookup framework. We sample continuous videos with only one labelled segment, which we refer to as the ‘foreground’ word (e.g., *friend*, *language*). Each continuous video has a subtitle, which we use to sample additional words for which we do not have continuous signing labels, (‘background’ words), e.g. *name* and *what* for “*what is your friend's name?*”. We sample all the dictionary videos corresponding to these words. Each word has multiple dictionary instances grouped into overlapping circles.



(b) **Sampling positive/negative pairs:** We anchor at 4 different positions within the batch to determine the pairs. Anchors are denoted with dashed lines, positive samples with solid green, negative samples with solid red lines. Gray samples are discarded. For example, (iii) anchoring at the continuous background marks the dictionary video for *name* positive, because it appears in the subtitle, but it is not within the annotated temporal window. All other dictionary samples *friend*, *language*, *speak* become negative to this anchor. We repeat this for each dictionary background, i.e., marking *what* as positive. See text for detailed explanations on each case. We also provide a video animation at our project page to show all possible positive/negative pairs for cases (i) to (iv).

Fig. A.3: Watch-Read-Lookup in detail.

Watch, Read and Lookup. The *Watch-Lookup* bag formulation defined above has a significant limitation: the data representation, f , is not encouraged to represent signs beyond the initial vocabulary represented in \mathcal{M} . We therefore look at the subtitles present in \mathcal{S} (which contain words beyond \mathcal{M}) in addition to \mathcal{M} to construct bags. To do so, we introduce an additional piece of terminology—when considering a subtitled video for which only one segment is labelled, we use the term “foreground” to refer to the subtitle word that corresponds to the label, and “background” for words which do not possess labelled segments in the video. Similarly to *Watch-Lookup*, we can construct positive bags, $\mathcal{P}_{\text{watch,lookup}}^{\text{bags}}$ (Fig. A.3 (i) and (ii), top rows) which correspond to the use of foreground subtitle words. However, these can now be extended by (a) anchoring on a background segment in the continuous footage and find candidate matches in the dictionary among all possible matches for the subtitle words (Fig. A.3 (iii), top row) and (b) anchoring on dictionary entries for background subtitle words (Fig. A.3 (iv), top row). Formally, let $\text{Tokenize}(\cdot) : \mathcal{S} \rightarrow \mathcal{V}_{\mathcal{L}}$ denote the function which extracts words from the subtitle that are present in the vocabulary: $\text{Tokenize}(s) \triangleq \{w \in s : w \in \mathcal{V}_{\mathcal{L}}\}$. Then define background segment-anchored positive bags as:

$$\mathcal{P}_{\text{watch,read,lookup}}^{\text{bags(seg-back)}} = \{\{i\} \times B_i : \exists(x_k, s_k) \in \mathcal{S} \text{ s.t. } x_i \subseteq x_k, (x_j^{\mathcal{D}}, v_j^{\mathcal{D}}) \in \mathcal{D}, \quad (6)$$

$$B_i = \{j : v_j^{\mathcal{D}} \in \text{Tokenize}(s_k)\}, (x_i, v_i) \notin \mathcal{M}\}$$

i.e. each bag contains a background segment from the continuous signing which is paired with all dictionary segments whose labels match any token from the corresponding subtitle sentence (visualised as the top row of Fig. A.3 (iii)). Next, we define dictionary-anchored positive background bags as follows:

$$\mathcal{P}_{\text{watch,read,lookup}}^{\text{bags(dict-back)}} = \{A_i \times B_i : (x_i^{\mathcal{D}}, v_i^{\mathcal{D}}) \in \mathcal{D}, A_i = \{j : v_j^{\mathcal{D}} \in \text{Tokenize}(s_k), \quad (7)$$

$$(x_k, s_k) \in \mathcal{S}, x_j \subseteq x_k, (x_j, v_j) \notin \mathcal{M}\}, B_i = \{l : v_l^{\mathcal{D}} = v_i^{\mathcal{D}}\}\}$$

i.e. the bags contain all pairwise combinations of dictionary entries for a given word and segments in continuous signing whose subtitle contains that background word (visualised as top row of Fig. A.3 (iv)). We combine these bags with the *Watch-Lookup* positive bags to maximally exploit the available supervisory signal for positives:

$$\mathcal{P}_{\text{watch,read,lookup}}^{\text{bags}} = \mathcal{P}_{\text{watch,lookup}}^{\text{bags}} \cup \mathcal{P}_{\text{watch,read,lookup}}^{\text{bags(seg-back)}} \cup \mathcal{P}_{\text{watch,read,lookup}}^{\text{bags(dict-back)}}. \quad (8)$$

To counterbalance the positives, we use \mathcal{S} in combination with \mathcal{M} and \mathcal{D} to create four kinds of negative bags. Differently to positive sampling, negatives can be constructed across the full minibatch rather than solely from the current (subtitled video, dictionary) pairing. We first anchor negative bags on foreground segments:

$$\mathcal{N}_{\text{watch,read,lookup}}^{\text{bags(seg-fore)}} = \{\{i\} \times B_i : (x_i^{\mathcal{M}}, v_i^{\mathcal{M}}) \in \mathcal{M}, (x_j^{\mathcal{D}}, v_j^{\mathcal{D}}) \in \mathcal{D}, \quad (9)$$

$$B_i = \{j : v_j^{\mathcal{D}} \neq v_i^{\mathcal{M}}\}\}$$

so that they contain pairs between a given foreground segment and all available dictionary videos whose label does not match the segment (visualised in Fig. A.3 (i), both rows). We next anchor on the foreground dictionary videos:

$$\begin{aligned} \mathcal{N}_{\text{watch,read,lookup}}^{\text{bags(dict-fore)}} &= \{A_i \times B_i : (x_i^{\mathcal{D}}, v_i^{\mathcal{D}}) \in \mathcal{D}, A_i = \{j : v_j^{\mathcal{D}} \in \text{Tokenize}(s_k), \\ &\quad (x_k, s_k) \in \mathcal{S}, x_j \subseteq x_k, (x_j, v_j) \notin \mathcal{M}\} \cup \{(x_m, v_m) \in \mathcal{M}, v_m \neq v_i\}, \\ &\quad B_i = \{l : v_l^{\mathcal{D}} = v_i^{\mathcal{D}}\} \} \end{aligned} \quad (10)$$

comprising of pairings between the dictionary foreground set and segments within the minibatch that are either labelled with a different word, or can be excluded as a potential match through the subtitles (Fig. A.3 (ii), both rows). Next, we anchor on the background continuous segments:

$$\begin{aligned} \mathcal{N}_{\text{watch,read,lookup}}^{\text{bags(seg-back)}} &= \{\{i\} \times B_i : \exists (x_k, s_k) \in \mathcal{S}, x_i \subseteq x_k, (x_j^{\mathcal{D}}, v_j^{\mathcal{D}}) \in \mathcal{D}, \\ &\quad B_i = \{j : v_j^{\mathcal{D}} \notin \text{Tokenize}(s_k)\} \} \end{aligned} \quad (11)$$

which amounts to the pairings between each background segment and the set of dictionary videos which do not correspond to any of the words in the background subtitles (Fig. A.3 (iii), both rows). The fourth negative bag set construction anchors on the background dictionaries:

$$\begin{aligned} \mathcal{N}_{\text{watch,read,lookup}}^{\text{bags(dict-back)}} &= \{A_i \times B_i : (x_i^{\mathcal{D}}, v_i^{\mathcal{D}}) \in \mathcal{D}, A_i = \{j : v_j^{\mathcal{D}} \notin \text{Tokenize}(s_k), \\ &\quad (x_k, s_k) \in \mathcal{S}, x_j \subseteq x_k, (x_j, v_j) \notin \mathcal{M}\} \cup \{(x_m, v_m) \in \mathcal{M}, v_m \neq v_i\}, \\ &\quad B_i = \{l : v_l^{\mathcal{D}} = v_i^{\mathcal{D}}\} \} \end{aligned} \quad (12)$$

and thus the pairings arise between dictionary examples for a background segment and its corresponding foreground segment, as well all segments from other batch elements (Fig. A.3 (iv), both rows). These four sets of bags are combined to form the full negative bag set:

$$\begin{aligned} \mathcal{N}_{\text{watch,read,lookup}}^{\text{bags}} &= \mathcal{N}_{\text{watch,read,lookup}}^{\text{bags(seg-fore)}} \cup \mathcal{N}_{\text{watch,read,lookup}}^{\text{bags(seg-dict)}} \\ &\quad \cup \mathcal{N}_{\text{watch,read,lookup}}^{\text{bags(seg-back)}} \cup \mathcal{N}_{\text{watch,read,lookup}}^{\text{bags(dict-back)}}. \end{aligned} \quad (13)$$

In the main paper, these bag formulations are used through Eqn. (1) (the MIL-NCE loss function) to guide learning. Concretely, the *Watch-Lookup* framework defines positive and negative bags via $\mathcal{P}^{\text{bags}} = \mathcal{P}_{\text{watch,lookup}}^{\text{bags}}$, $\mathcal{N}^{\text{bags}} = \mathcal{N}_{\text{watch,lookup}}^{\text{bags}}$ and the *Watch-Read-Lookup* formulation instead defines the positive and negative bags via $\mathcal{P}^{\text{bags}} = \mathcal{P}_{\text{watch,read,lookup}}^{\text{bags}}$, $\mathcal{N}^{\text{bags}} = \mathcal{N}_{\text{watch,read,lookup}}^{\text{bags}}$.

C.3 Infrastructure

The I3D trunk BSL-1K pretraining experiments were performed with four Nvidia M40 graphics cards and took 2-3 days to complete. After freezing the I3D trunk, training the parameters of the MLP with the *Watch-Read-Lookup* framework took approximately two hours on a single Nvidia M40 graphics card.

References

1. Pfister, T., Charles, J., Zisserman, A.: Large-scale learning of sign language by watching tv (using co-occurrences). In: BMVC. (2013) [4](#)
2. Albanie, S., Varol, G., Momeni, L., Afouras, T., Chung, J.S., Fox, N., Zisserman, A.: BSL-1K: Scaling up co-articulated sign recognition using mouthing cues. In: ECCV. (2020) [4](#), [5](#)
3. Carreira, J., Zisserman, A.: Quo vadis, action recognition? A new model and the Kinetics dataset. In: CVPR. (2017) [4](#), [5](#)
4. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning for video understanding. In: ECCV. (2018) [4](#), [5](#)
5. Chang, W.G., You, T., Seo, S., Kwak, S., Han, B.: Domain-specific batch normalization for unsupervised domain adaptation. In: CVPR. (2019) [6](#)
6. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché Buc, F., Fox, E., Garnett, R., eds.: Advances in Neural Information Processing Systems 32. Curran Associates, Inc. (2019) 8024–8035 [7](#)
7. Dietterich, T.G., Lathrop, R.H., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artificial intelligence **89** (1997) 31–71 [7](#)