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,afourast,az}@robots.ox.ac.uk

Abstract. The focus of this work is *sign spotting*—given a video of an isolated sign, our task is to identify *whether* and *where* it has been signed in a continuous, co-articulated sign language video. To achieve this sign spotting task, we train a model using multiple types of available supervision by: (1) *watching* existing sparsely labelled footage; (2) *reading* associated subtitles (readily available translations of the signed content) which provide additional *weak-supervision*; (3) *looking up* words (for which no co-articulated labelled examples are available) in visual sign language dictionaries to enable novel sign spotting. These three tasks are integrated into a unified learning framework using the principles of Noise Contrastive Estimation and Multiple Instance Learning. We validate the effectiveness of our approach on low-shot sign spotting benchmarks. In addition, we contribute a machine-readable British Sign Language (BSL) dictionary dataset of isolated signs, BslDict, to facilitate study of this task. The dataset, models and code are available at our project page[1].

1 Introduction

The objective of this work is to develop a *sign spotting* model that can identify and localise instances of signs within sequences of continuous sign language. Sign languages represent the natural means of communication for deaf communities [1] and sign spotting has a broad range of practical applications. Examples include: indexing videos of signing content by keyword to enable content-based search; gathering diverse dictionaries of sign exemplars from unlabelled footage for linguistic study; automatic feedback for language students via an "auto-correct" tool (e.g. "did you mean this sign?"); making voice activated wake word devices accessible to deaf communities; and building sign language datasets by automatically labelling examples of signs.

The recent marriage of large-scale, labelled datasets with deep neural networks has produced considerable progress in audio [2, 3] and visual [4, 5] keyword spotting in *spoken languages*. However, a direct replication of these keyword spotting successes in sign language requires a commensurate quantity of labelled data (note that modern audiovisual spoken keyword spotting datasets contain

* Equal contribution
[1] https://www.robots.ox.ac.uk/~vgg/research/bsldict/

Fig. 1: We consider the task of *sign spotting* in co-articulated, continuous signing. Given a query dictionary video of an isolated sign (e.g. "apple"), we aim to identify *whether* and *where* it appears in videos of continuous signing. The wide domain gap between dictionary examples of *isolated* signs and target sequences of *continuous* signing makes the task extremely challenging.

millions of densely labelled examples [6, 7]). Large-scale corpora of continuous, co-articulated[2] signing from TV broadcast data have recently been built [8], but the labels accompanying this data are: (1) sparse, and (2) cover a limited vocabulary.

It might be thought that a sign language dictionary would offer a relatively straightforward solution to the sign spotting task, particularly to the problem of covering only a limited vocabulary in existing large-scale corpora. But, unfortunately, this is not the case due to the severe *domain differences* between dictionaries and continuous signing in the wild. The challenges are that sign language dictionaries typically: (i) consist of *isolated signs* which differ in appearance from the *co-articulated* sequences of continuous signs (for which we ultimately wish to perform spotting); and (ii) differ in speed (are performed more slowly) relative to co-articulated signing. Furthermore, (iii) dictionaries only possess a few examples of each sign (so learning must be *low shot*); and as one more challenge, (iv) there can be multiple signs corresponding to a single keyword, for example due to regional variations of the sign language [9]. We show through experiments in Sec. 4, that directly training a sign spotter for continuous signing on dictionary examples, obtained from an internet-sourced sign language dictionary, does indeed perform poorly.

To address these challenges, we propose a unified framework in which sign spotting embeddings are learned from the dictionary (to provide broad coverage of the lexicon) in combination with two additional sources of supervision. In aggregate, these multiple types of supervision include: (1) *watching* sign language and learning from existing sparse annotations; (2) exploiting weak-supervision by *reading* the subtitles that accompany the footage and extracting candidates for signs that we expect to be present; (3) *looking up* words (for which we do not have labelled examples) in a sign language dictionary. The recent development of large-scale, subtitled corpora of continuous signing providing sparse annotations [8] allows us to study this problem setting directly. We formulate our approach as a Multiple Instance Learning problem in which positive samples may arise from any of the three sources and employ Noise Contrastive Estimation [10]

[2] *Co-articulation* refers to changes in the appearance of the current sign due to neighbouring signs.

to learn a domain-invariant (valid across both isolated and co-articulated signing) representation of signing content.

We make the following six contributions: (1) We provide a machine readable British Sign Language (BSL) dictionary dataset of isolated signs, BslDict, to facilitate study of the sign spotting task; (2) We propose a unified Multiple Instance Learning framework for learning sign embeddings suitable for spotting from three supervisory sources; (3) We validate the effectiveness of our approach on a co-articulated sign spotting benchmark for which only a small number (low-shot) of isolated signs are provided as labelled training examples, and (4) achieve state-of-the-art performance on the BSL-1K sign spotting benchmark [8] (closed vocabulary). We show qualitatively that the learned embeddings can be used to (5) automatically mine new signing examples, and (6) discover "faux amis" (false friends) between sign languages.

2 Related Work

Our work relates to several themes in the literature: *sign language recognition* (and more specifically *sign spotting*), *sign language datasets*, *multiple instance learning* and *low-shot action localization*. We discuss each of these themes next.

Sign language recognition. The study of automatic sign recognition has a rich history in the computer vision community stretching back over 30 years, with early methods developing carefully engineered features to model trajectories and shape [11, 12, 13, 14]. A series of techniques then emerged which made effective use of hand and body pose cues through robust keypoint estimation encodings [15, 16, 17, 18]. Sign language recognition also has been considered in the context of sequence prediction, with HMMs [11, 13, 19, 20], LSTMs [21, 22, 23, 24], and Transformers [25] proving to be effective mechanisms for this task. Recently, convolutional neural networks have emerged as the dominant approach for appearance modelling [21], and in particular, action recognition models using spatio-temporal convolutions [26] have proven very well-suited for video-based sign recognition [8, 27, 28]. We adopt the I3D architecture [26] as a foundational building block in our studies.

Sign language spotting. The sign language spotting problem—in which the objective is to find performances of a sign (or sign sequence) in a longer sequence of signing—has been studied with Dynamic Time Warping and skin colour histograms [29] and with Hierarchical Sequential Patterns [30]. Different from our work which learns representations from multiple weak supervisory cues, these approaches consider a fully-supervised setting with a single source of supervision and use hand-crafted features to represent signs [31]. Our proposed use of a dictionary is also closely tied to *one-shot/few-shot learning*, in which the learner is assumed to have access to only a handful of annotated examples of the target category. One-shot dictionary learning was studied by [18] – different to their approach, we explicitly account for dialect variations in the dictionary (and validate the improvements brought by doing so in Sec. 4). Textual descriptions from a dictionary of 250 signs were used to study zero-shot learning by [32] – we instead consider the practical setting in which a handful of video examples are

available per-sign (and make this dictionary available). The use of dictionaries to locate signs in subtitled video also shares commonalities with *domain adaptation*, since our method must bridge differences between the dictionary and the target continuous signing distribution. A vast number of techniques have been proposed to tackle distribution shift, including several adversarial feature alignment methods that are specialised for the few-shot setting [33, 34]. In our work, we explore the domain-specific batch normalization (DSBN) method of [35], finding ultimately that simple batch normalization parameter re-initialization is most effective when jointly training on two domains after pre-training on the bigger domain. The concurrent work of [36] also seeks to align representation of isolated and continuous signs. However, our work differs from theirs in several key aspects: (1) rather than assuming access to a large-scale labelled dataset of isolated signs, we consider the setting in which only a handful of dictionary examples may be used to represent a word; (2) we develop a generalised Multiple Instance Learning framework which allows the learning of representations from weakly aligned subtitles whilst exploiting sparse labels and dictionaries (this integrates cues beyond the learning formulation in [36]); (3) we seek to label and improve performance on co-articulated signing (rather than improving recognition performance on isolated signing). Also related to our work, [18] uses a "reservoir" of weakly labelled sign footage to improve the performance of a sign classifier learned from a small number of examples. Different to [18], we propose a multi-instance learning formulation that explicitly accounts for signing variations that are present in the dictionary.

Sign language datasets. A number of sign language datasets have been proposed for studying Finnish [29], German [37, 38], American [27, 28, 39, 40] and Chinese [22, 41] sign recognition. For British Sign Language (BSL), [42] gathered a corpus labelled with sparse, but fine-grained linguistic annotations, and more recently [8] collected BSL-1K, a large-scale dataset of BSL signs that were obtained using a mouthing-based keyword spotting model. In this work, we contribute BslDict, a dictionary-style dataset that is complementary to the datasets of [8, 42] – it contains only a handful of instances of each sign, but achieves a comprehensive coverage of the BSL lexicon with a 9K vocabulary (vs a 1K vocabulary in [8]). As we show in the sequel, this dataset enables a number of sign spotting applications.

Multiple instance learning. Motivated by the readily available sign language footage that is accompanied by subtitles, a number of methods have been proposed for learning the association between signs and words that occur in the subtitle text [15, 18, 43, 44]. In this work, we adopt the framework of Multiple Instance Learning (MIL) [45] to tackle this problem, previously explored by [15, 46]. Our work differs from these works through the incorporation of a dictionary, and a principled mechanism for explicitly handling sign variants, to guide the learning process. Furthermore, we generalise the MIL framework so that it can learn to further exploit sparse labels. We also conduct experiments at significantly greater scale to make use of the full potential of MIL, considering more than two orders of magnitude more weakly supervised data than [15, 46].

Low-shot action localization. This theme investigates semantic video localization: given one or more query videos the objective is to localize the segment in

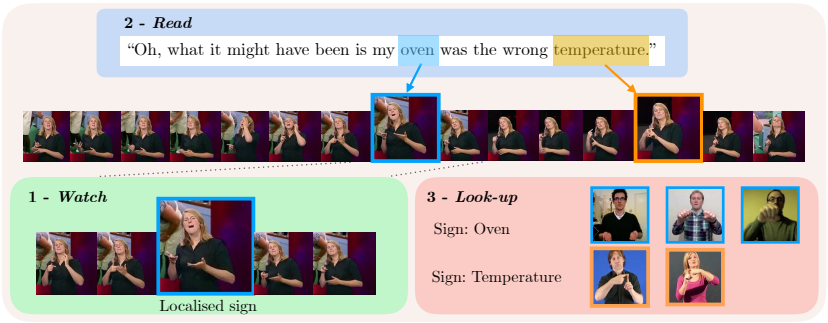


Fig. 2: **The proposed *Watch, Read and Lookup* framework** trains sign spotting embeddings with three cues: (1) *watching* videos and learning from sparse annotation in the form of localised signs (lower-left); (2) *reading* subtitles to find candidate signs that may appear in the source footage (top); (3) *looking up* corresponding visual examples in a sign language dictionary and aligning the representation against the embedded source segment (lower-right).

an untrimmed video that corresponds semantically to the query video [47, 48, 49]. Semantic matching is too general for the sign-spotting considered in this paper. However, we build on the temporal ordering ideas explored in this theme.

3 Learning Sign Spotting Embeddings from Multiple Supervisors

In this section, we describe the task of *sign spotting* and the three forms of supervision we assume access to. Let $\mathcal{X}_{\mathfrak{L}}$ denote the space of RGB video segments containing a frontal-facing individual communicating in sign language $\mathfrak{L}$ and denote by $\mathcal{X}_{\mathfrak{L}}^{\text{single}}$ its restriction to the set of segments containing a single sign. Further, let $\mathcal{T}$ denote the space of subtitle sentences and $\mathcal{V}_{\mathfrak{L}} = \{1, \dots, V\}$ denote the *vocabulary*—an index set corresponding to an enumeration of written words that are equivalent to signs that can be performed in $\mathfrak{L}$[3].

Our objective, illustrated in Fig. 1, is to discover all occurrences of a given keyword in a collection of continuous signing sequences. To do so, we assume access to: (i) a subtitled collection of videos containing continuous signing, $\mathcal{S} = \{(x_i, s_i) : i \in \{1, \dots, I\}, x_i \in \mathcal{X}_{\mathfrak{L}}, s_i \in \mathcal{T}\}$; (ii) a sparse collection of temporal sub-segments of these videos that have been annotated with their corresponding word, $\mathcal{M} = \{(x_k, v_k) : k \in \{1, \dots, K\}, v_k \in \mathcal{V}_{\mathfrak{L}}, x_k \in \mathcal{X}_{\mathfrak{L}}^{\text{single}}, \exists (x_i, s_i) \in \mathcal{S}\, s.t.\, x_k \subseteq x_i\}$; (iii) a curated *dictionary* of signing instances $\mathcal{D} = \{(x_j, v_j) : j \in \{1, \dots, J\}, x_j \in \mathcal{X}_{\mathfrak{L}}^{\text{single}}, v_j \in \mathcal{V}_{\mathfrak{L}}\}$. To address the sign spotting task, we propose to learn a *data representation* $f : \mathcal{X}_{\mathfrak{L}} \to \mathbb{R}^d$ that maps video segments to vectors

[3] Sign language dictionaries provide a word-level or phrase-level correspondence (between sign language and spoken language) for many signs but no universally accepted *glossing* scheme exists for transcribing languages such as BSL [1].

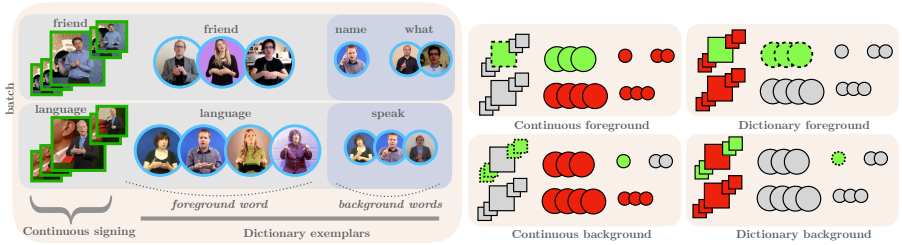


Fig. 3: **Batch sampling and positive/negative pairs:** We illustrate the formation of a batch when jointly training on continuous signing video (squares) and dictionaries of isolated signing (circles). **Left:** For each continuous video, we sample the dictionaries corresponding to the labelled word (foreground), as well as to the rest of the subtitles (background). **Right:** We construct positive/negative pairs by anchoring at 4 different portions of a batch item: continuous foreground/background and dictionary foreground/background. Positives and negatives (defined across continuous and dictionary domains) are green and red, respectively; anchors have a dashed border (see Appendix C.2 for details).

such that they are *discriminative* for sign spotting and *invariant* to other factors of variation. Formally, for any labelled pair of video segments $(x, v), (x', v')$ with $x, x' \in \mathcal{X}_{\mathfrak{L}}$ and $v, v' \in \mathcal{V}_{\mathfrak{L}}$, we seek a data representation, f, that satisfies the constraint $\delta_{f(x)f(x')} = \delta_{vv'}$, where δ represents the Kronecker delta.

3.1 Integrating Cues through Multiple Instance Learning

To learn f, we must address several challenges. First, as noted in Sec. 1, there may be a considerable distribution shift between the dictionary videos of isolated signs in $\mathcal{D}$ and the co-articulated signing videos in $\mathcal{S}$. Second, sign languages often contain multiple sign variants for a single written word (resulting from regional dialects and synonyms). Third, since the subtitles in $\mathcal{S}$ are only weakly aligned with the sign sequence, we must learn to associate signs and words from a noisy signal that lacks temporal localisation. Fourth, the localised annotations provided by $\mathcal{M}$ are sparse, and therefore we must make good use of the remaining segments of subtitled videos in $\mathcal{S}$ if we are to learn an effective representation.

Given full supervision, we could simply adopt a pairwise metric learning approach to align segments from the videos in $\mathcal{S}$ with dictionary videos from $\mathcal{D}$ by requiring that f maps a pair of isolated and co-articulated signing segments to the same point in the embedding space if they correspond to the same sign (*positive* pairs) and apart if they do not (*negative* pairs). As noted above, in practice we do not have access to positive pairs because: (1) for any annotated segment $(x_k, v_k) \in \mathcal{M}$, 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 In-

stance Learning [45] (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 pairing between a segment from a video in $\mathcal{S}$ and a dictionary video from $\mathcal{D}$ to contain the same sign, and negative bags of pairs, $\mathcal{N}^{\text{bags}}$, in which we expect no (video segment, dictionary video) pair 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 (due to space constraints, a formal mathematical description of the positive and negative bags described below is deferred to Appendix C.2).

Watch and Lookup: using sparse annotations and dictionaries. Here, we describe a baseline where we assume no subtitles are available. To learn f from $\mathcal{M}$ and $\mathcal{D}$, we define each positive bag as the set of possible pairs between a *labelled* (*foreground*) temporal segment of a continuous video from $\mathcal{M}$ and the examples of the corresponding sign in the dictionary (green regions in Fig A.2). The key assumption here is that each labelled sign segment from $\mathcal{M}$ matches *at least one* sign variation in the dictionary. Negative bags are constructed by (i) anchoring on a continuous foreground segment and selecting dictionary examples corresponding to different words from other batch items; (ii) anchoring on a dictionary foreground set and selecting continuous foreground segments from other batch items (red regions in Fig A.2). To maximize the number of negatives within one minibatch, we sample a different word per batch item.

Watch, Read and Lookup: using sparse annotations, subtitles and dictionaries. Using just the labelled sign segments from $\mathcal{M}$ to construct bags has a significant limitation: f is not encouraged to represent signs beyond the initial vocabulary represented in $\mathcal{M}$. We therefore look at the subtitles (which contain words beyond $\mathcal{M}$) to construct additional bags. We determine more positive bags between the set of *unlabelled (background)* segments in the continuous footage and the set of dictionaries corresponding to the background words in the subtitle (green regions in Fig. 3, right-bottom). Negatives (red regions in Fig. 3) are formed as the complements to these sets by (i) pairing continuous background segments with dictionary samples that can be excluded as matches (through subtitles) and (ii) pairing background dictionary entries with the foreground continuous segment. In both cases, we also define negatives from other batch items by selecting pairs where the word(s) have no overlap, e.g., in Fig. 3, the dictionary examples for the background word 'speak' from the second batch item are negatives for the background continuous segments from the first batch item, corresponding to the unlabelled words 'name' and 'what' in the subtitle.

To assess the similarity of two embedded video segments, we employ a similarity function $\psi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ whose value increases as its arguments become more similar (in this work, we use cosine similarity). For notational convenience below, we write ψ_{ij} as shorthand for $\psi(f(x_i), f(x_j))$. To learn f, we consider a generalization of the InfoNCE loss [50, 51] (a non-parametric softmax loss formulation of Noise Contrastive Estimation [10]) recently proposed by [52]:

$$\mathcal{L}_{\text{MIL-NCE}} = -\mathbb{E}_i \left[\log \frac{\sum_{(j,k)\in\mathcal{P}(i)} \exp(\psi_{jk}/\tau)}{\sum_{(j,k)\in\mathcal{P}(i)} \exp(\psi_{jk}/\tau) + \sum_{(l,m)\in\mathcal{N}(i)} \exp(\psi_{lm}/\tau)} \right], \quad (1)$$

where $\mathcal{P}(i) \in \mathcal{P}^{\text{bags}}$, $\mathcal{N}(i) \in \mathcal{N}^{\text{bags}}$, τ, often referred to as the *temperature*, is set as a hyperparameter (we explore the effect of its value in Sec. 4).

3.2 Implementation details

In this section, we provide details for the learning framework covering the embedding architecture, sampling protocol and optimization procedure.

Embedding architecture. The architecture comprises an I3D spatio-temporal trunk network [26] to which we attach an MLP consisting of three linear layers separated by leaky ReLU activations (with negative slope 0.2) and a skip connection. The trunk network takes as input 16 frames from a 224×224 resolution video clip and produces 1024-dimensional embeddings which are then projected to 256-dimensional sign spotting embeddings by the MLP. More details about the embedding architecture can be found in Appendix C.1.

Joint pretraining. The I3D trunk parameters are initialised by pretraining for sign classification jointly over the sparse annotations $\mathcal{M}$ of a continuous signing dataset (BSL-1K [8]) and examples from a sign dictionary dataset (BSLDICT) which fall within their common vocabulary. Since we find that dictionary videos of isolated signs tend to be performed more slowly, we uniformly sample 16 frames from each dictionary video with a random shift and random frame rate n times, where n is proportional to the length of the video, and pass these clips through the I3D trunk then average the resulting vectors before they are processed by the MLP to produce the final dictionary embeddings. We find that this form of random sampling performs better than sampling 16 consecutive frames from the isolated signing videos (see Appendix C.1 for more details). During pretraining, minibatches of size 4 are used; and colour, scale and horizontal flip augmentations are applied to the input video, following the procedure described in [8]. The trunk parameters are then frozen and the MLP outputs are used as embeddings. Both datasets are described in detail in Sec. 4.1.

Minibatch sampling. To train the MLP given the pretrained I3D features, we sample data by first iterating over the set of labelled segments comprising the sparse annotations, $\mathcal{M}$, that accompany the dataset of continuous, subtitled sampling to form minibatches. For each continuous video, we sample 16 consecutive frames around the annotated timestamp (more precisely a random offset within 20 frames before, 5 frames after, following the timing study in [8]). We randomly sample 10 additional 16-frame clips from this video outside of the labelled window, i.e., continuous background segments. For each subtitled sequence, we sample the dictionary entries for all subtitle words that appear in $\mathcal{V}_{\mathfrak{L}}$ (see Fig. 3 for a sample batch formation).

Our minibatch comprises 128 sequences of continuous signing and their corresponding dictionary entries (we investigate the impact of batch size in Sec. 4.3). The embeddings are then trained by minimising the loss defined in Eqn.(1) in conjunction with positive bags, $\mathcal{P}^{\text{bags}}$, and negative bags, $\mathcal{N}^{\text{bags}}$, which are constructed on-the-fly for each minibatch (see Fig. 3).

Optimization. We use a SGD optimizer with an initial learning rate of 10^{-2} to train the embedding architecture. The learning rate is decayed twice by a factor

Dataset	#Videos	Vocab.	#Signers
BSL-1K[8]	273K	1,064	40
BslDict	14,210	9,283	148

Table 1: **Datasets:** We provide (i) the number of individual sign videos, (ii) the vocabulary size of the annotated signs, and (iii) the number of signers for BSL-1K and BslDict. BSL-1K is large in the number of annotated signs whereas BslDict is large in the vocabulary size. Note that we use a different partition of BSL-1K with longer sequences around the annotations as described in Sec. 4.1.

of 10 (at epoch 40 and 45). We train all models, including baselines and ablation studies, for 50 epochs at which point we find that learning has always converged.
Test time. To perform spotting, we obtain the embeddings learned with the MLP. For the dictionary, we have a single embedding averaged over the video. Continuous video embeddings are obtained with sliding window (stride 1) on the entire sequence. We calculate the cosine similarity score between the continuous signing sequence embeddings and the embedding for a given dictionary video. We determine the location with the maximum similarity as the location of the queried sign. We maintain embedding sets of all variants of dictionary videos for a given word and choose the best match as the one with the highest similarity.

4 Experiments

In this section, we first present the datasets used in this work (including the contributed BslDict dataset) in Sec. 4.1, followed by the evaluation protocol in Sec. 4.2. We illustrate the benefits of the *Watch, Read and Lookup* learning framework for sign spotting against several baselines with a comprehensive ablation study that validates our design choices (Sec. 4.3). Finally, we investigate three applications of our method in Sec. 4.4, showing that it can be used to (i) not only spot signs, but also identify the specific sign variant that was used, (ii) label sign instances in continuous signing footage given the associated subtitles, and (iii) discover "faux amis" between different sign languages.

4.1 Datasets

Although our method is conceptually applicable to a number of sign languages, in this work we focus primarily on BSL, the sign language of British deaf communities. We use BSL-1K [8], a large-scale, subtitled and sparsely annotated dataset of more than 1000 hours of continuous signing which offers an ideal setting in which to evaluate the effectiveness of the *Watch, Read and Lookup* sign spotting framework. To provide dictionary data for the *lookup* component of our approach, we also contribute BslDict, a diverse visual dictionary of signs. These two datasets are summarised in Table 1 and described in more detail below.
BSL-1K [8] comprises a vocabulary of 1,064 signs which are sparsely annotated over 1,000 hours of video of continuous sign language. The videos are accompanied by subtitles. The dataset consists of 273K localised sign annotations,

automatically generated from sign-language-interpreted BBC television broadcasts, by leveraging weakly aligned subtitles and applying keyword spotting to signer *mouthings*. Please refer to [8] for more details on the automatic annotation pipeline. In this work, we process this data to extract long videos with subtitles. In particular, we pad +/-2 seconds around the subtitle timestamps and we add the corresponding video to our training set if there is a sparse annotation word falling within this time window, assuming that the signing is reasonably well-aligned with its subtitles in these cases. We further consider only the videos whose subtitle duration is longer than 2 seconds. For testing, we use the automatic test set (corresponding to mouthing locations with confidences above 0.9). Thus we obtain 78K training and 3K test videos, each of which has a subtitle of 8 words on average and 1 sparse mouthing annotation.

BslDict. BSL dictionary videos are collected from a BSL sign aggregation platform `signbsl.com` [53], giving us a total of 14,210 video clips for a vocabulary of 9,283 signs. Each sign is typically performed several times by different signers, often in different ways. The dictionary videos are downloaded from 28 known website sources and each source has at least 1 signer. We used face embeddings computed with SENet-50 [54] (trained on VGGFace2 [55]) to cluster signer identities and manually verified that there are a total of 148 different signers. The dictionary videos are of isolated signs (as opposed to co-articulated in BSL-1K): this means (i) the start and end of the video clips usually consist of a still signer pausing, and (ii) the sign is performed at a much slower rate for clarity. We first trim the sign dictionary videos, using body keypoints estimated with OpenPose [56] which indicate the start and end of wrist motion, to discard frames where the signer is still. With this process, the average number of frames per video drops from 78 to 56 (still significantly larger than co-articulated signs). To the best of our knowledge, BslDict is the first curated, BSL sign dictionary dataset for computer vision research, which will be made available. For the experiments in which BslDict is filtered to the 1,064 vocabulary of BSL-1K (see below), we have a total of 2,992 videos. Within this subset, each sign has between 1 and 10 examples (average of 3).

4.2 Evaluation Protocols

Protocols. We define two settings: (i) training with the entire 1064 vocabulary of annotations in BSL-1K; and (ii) training on a subset with 800 signs. The latter is needed to assess the performance on novel signs, for which we do not have access to co-articulated labels at training. We thus use the remaining 264 words for testing. This test set is therefore common to both training settings, it is either 'seen' or 'unseen' at training. However, we do not limit the vocabulary of the dictionary as a practical assumption, for which we show benefits.
Metrics. The performance is evaluated based on ranking metrics. For every sign s_i in the test vocabulary, we first select the BSL-1K test set clips which have a mouthing annotation of s_i and then record the percentage of dictionary clips of s_i that appear in the first 5 retrieved results, this is the 'Recall at 5' (R@5). This is motivated by the fact that different English words can correspond to the same

		Train (1064)		Train (800)	
		Seen (264)		Unseen (264)	
Embedding arch.	Supervision	mAP	R@5	mAP	R@5
I3D$^{\text{BslDict}}$	Classification	2.68	3.57	1.21	1.29
I3D$^{\text{BSL-1K}}$ [8]	Classification	13.09	17.25	6.74	8.94
I3D$^{\text{BSL-1K,BslDict}}$	Classification	19.81	25.57	4.81	6.89
I3D$^{\text{BSL-1K,BslDict}}$+MLP	Classification	36.75	40.15	10.28	14.19
I3D$^{\text{BSL-1K,BslDict}}$+MLP	InfoNCE	42.52	53.54	10.88	14.23
I3D$^{\text{BSL-1K,BslDict}}$+MLP	Watch-Lookup	43.65	53.03	11.05	14.62
I3D$^{\text{BSL-1K,BslDict}}$+MLP	Watch-Read-Lookup	**48.11**	**58.71**	**13.69**	**17.79**

Table 2: **The effect of the loss formulation:** Embeddings learned with the classification loss are suboptimal since they are not trained for matching the two domains. Contrastive-based loss formulations (NCE) significantly improve, particularly when we adopt the multiple-instance variant introduced as our Watch-Read-Lookup framework of multiple supervisory signals.

sign, and vice versa. We also report mean average precision (mAP). For each video pair, the match is considered correct if (i) the dictionary clip corresponds to s_i and the BSL-1K video clip has a mouthing annotation of s_i, and (ii) if the predicted location of the sign in the BSL-1K video clip, i.e. the time frame where the maximum similarity occurs, lies within certain frames around the ground truth mouthing timing. In particular, we determine the correct interval to be defined between 20 frames before and 5 frames after the labelled time (based on the study in [8]). Finally, because BSL-1K test is class-unbalanced, we report performances averaged over the test classes.

4.3 Ablation Study

In this section, we evaluate different components of our approach. We first compare our contrastive learning approach with classification baselines. Then, we investigate the effect of our multiple-instance loss formulation. We provide ablations for the hyperparameters, such as the batch size and the temperature, and report performance on a sign spotting benchmark.

I3D baselines. We start by evaluating baseline I3D models trained with classification on the task of spotting, using the embeddings before the classification layer. We have three variants in Tab. 2: (i) I3D$^{\text{BSL-1K}}$ provided by [8] which is trained only on the BSL-1K dataset, and we also train (ii) I3D$^{\text{BslDict}}$ and (iii) I3D$^{\text{BSL-1K,BslDict}}$. Training only on BslDict (I3D$^{\text{BslDict}}$) performs significantly worse due to the few examples available per class and the domain gap that must be bridged to spot co-articulated signs, suggesting that dictionary samples alone do not suffice to solve the task. We observe improvements with fine-tuning I3D$^{\text{BSL-1K}}$ jointly on the two datasets (I3D$^{\text{BSL-1K,BslDict}}$), which becomes our base feature extractor for the remaining experiments to train a shallow MLP.

Loss formulation. We first train the MLP parameters on top of the frozen I3D trunk with classification to establish a baseline in a comparable setup. Note that, this shallow architecture can be trained with larger batches than I3D. Next, we

Supervision	Dictionary Vocab	mAP	R@5
Watch-Read-Lookup	800 training vocab	13.69	17.79
Watch-Read-Lookup	9k full vocab	**15.39**	**20.87**

Table 3: **Extending the dictionary vocabulary:** We show the benefits of sampling dictionary videos outside of the sparse annotations, using subtitles. Extending the lookup to the dictionary from the subtitles to the full vocabulary of BslDict brings significant improvements for novel signs (the training uses sparse annotations for the 800 words, and the remaining 264 for test).

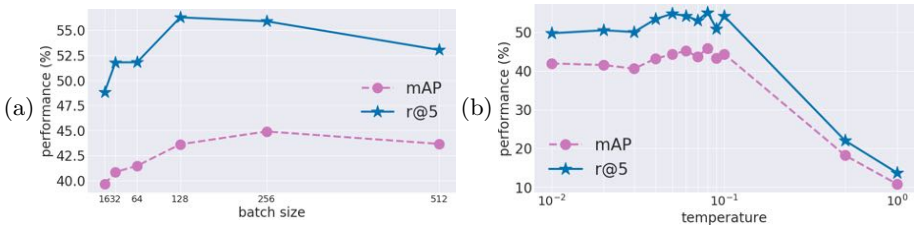


Fig. 4: The effect of (a) the **batch size** that determines the number of negatives across sign classes and (b) the **temperature** hyper-parameter for the MIL-NCE loss in Watch-Lookup against mAP and R@5 (trained on the full 1064 vocab.)

investigate variants of our loss to learn a joint sign embedding between BSL-1K and BslDict video domains: (i) standard single-instance InfoNCE [50, 51] loss which pairs each BSL-1K video clip with *one* positive BslDict clip of the same sign, (ii) Watch-Lookup which considers multiple positive dictionary candidates, but does not consider subtitles (therefore limited to the annotated video clips). Table 2 summarizes the results. Our Watch-Read-Lookup formulation which effectively combines multiple sources of supervision in a multiple-instance framework outperforms the other baselines in both *seen* and *unseen* protocols.

Extending the vocabulary. The results presented so far were using the same vocabulary for both continuous and dictionary datasets. In reality, one can assume access to the entire vocabulary in the dictionary, but obtaining annotations for the continuous videos is prohibitive. Table 3 investigates removing the vocabulary limit on the dictionary side, but keeping the continuous annotations vocabulary at 800 signs. We show that using the full 9k vocabulary from BslDict significantly improves the results on the unseen setting.

Batch size. Next, we investigate the effect of increasing the number of negative pairs by increasing the batch size when training with Watch-Lookup on 1064 categories. We observe in Figure 4(a) an improvement in performance with greater numbers of negatives before saturating. Our final Watch-Read-Lookup model has high memory requirements, for which we use 128 batch size. Note that the effective size of the batch with our sampling is larger due to sampling extra video clips corresponding to subtitles.

Temperature. Finally, we analyze the impact of the temperature hyperparameter τ on the performance of Watch-Lookup. We observe a major decrease in

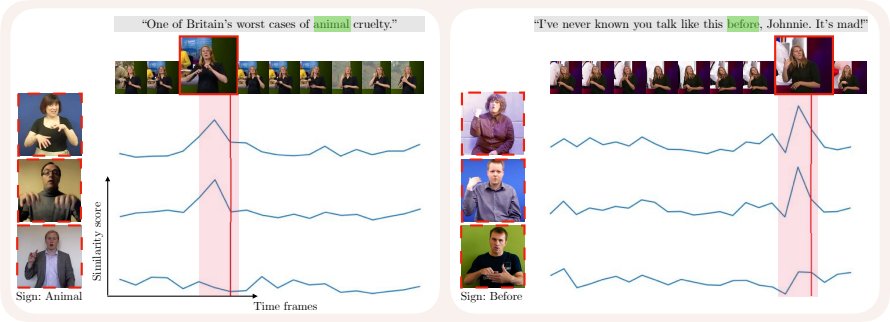


Fig. 5: **Sign variant identification:** We plot the similarity scores between BSL-1K test clips and BslDict variants of the sign "animal" (left) and "before" (right) over time. The labelled mouthing times are shown by red vertical lines and the sign proposal regions are shaded. A high similarity occurs for the first two rows, where the BslDict examples match the variant used in BSL-1K.

performance when τ approaches 1. We choose $\tau = 0.07$ used in [51, 57] for all other experiments. Additional ablations are provided in Appendix B.

BSL-1K Sign spotting benchmark. Although our learning framework primarily targets good performance on unseen continuous signs, it can also be naively applied to the (closed-vocabulary) sign spotting benchmark proposed by [8]. We evaluate the performance of our Watch-Read-Lookup model and achieve a score of 0.170 mAP, outperforming the previous state-of-the-art performance of 0.160 mAP [8].

4.4 Applications

In this section, we investigate three applications of our sign spotting method.

Sign variant identification. We show the ability of our model to spot specifically which variant of the sign was used. In Fig. 5, we observe high similarity scores when the variant of the sign matches in both BSL-1K and BslDict videos. Identifying such sign variations allows a better understanding of regional differences and can potentially help standardisation efforts of BSL.

Dense annotations. We demonstrate the potential of our model to obtain dense annotations on continuous sign language video data. Sign spotting through the use of sign dictionaries is not limited to mouthings as in [8] and therefore is of great importance to scale up datasets for learning more robust sign language models. In Fig. 6, we show qualitative examples of localising multiple signs in a given sentence in BSL-1K, where we only query the words that occur in the subtitles, reducing the search space. In fact, if we assume the word to be known, we obtain 83.08% sign localisation accuracy on BSL-1K with our best model. This is defined as the number of times the maximum similarity occurs within -20/+5 frames of the end label time provided by [8].

"Faux Amis". There are works investigating lexical similarities between sign languages manually [58, 59]. We show qualitatively the potential of our model to discover similarities, as well as "faux-amis" between different sign languages,

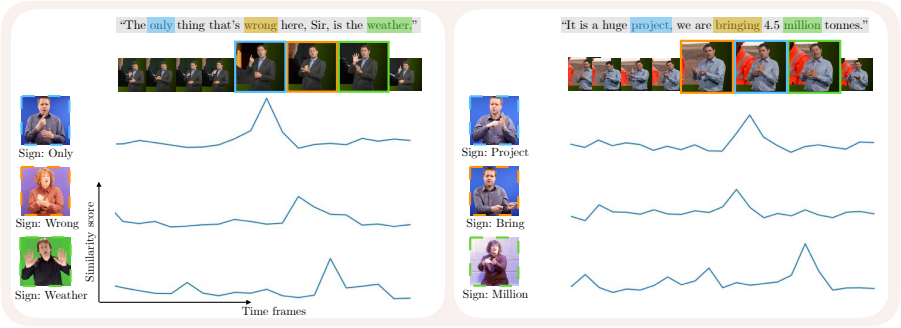


Fig. 6: **Densification:** We plot the similarity scores between BSL-1K test clips and BslDict examples over time, by querying only the words in the subtitle. The predicted locations of the signs correspond to the peak similarity scores.

Fig. 7: **"Faux amis" in BSL/ASL:** Same/similar manual features for different English words (left), as well as for the same English words (right), are identified between BslDict and WLASL isolated sign language datasets.

in particular between British (BSL) and American (ASL) Sign Languages. We retrieve nearest neighbors according to visual embedding similarities between BslDict which has a 9K vocabulary and WLASL [28], an ASL isolated sign language dataset, with a 2K vocabulary. We provide some examples in Fig. 7.

5 Conclusions

We have presented an approach to spot signs in continuous sign language videos using visual sign dictionary videos, and have shown the benefits of leveraging multiple supervisory signals available in a realistic setting: (i) sparse annotations in continuous signing, (ii) accompanied with subtitles, and (iii) a few dictionary samples per word from a large vocabulary. We employ multiple-instance contrastive learning to incorporate these signals into a unified framework. Our analysis suggests the potential of sign spotting in several applications, which we think will help in scaling up the automatic annotation of sign language datasets.

Acknowledgements. This work was supported by EPSRC grant ExTol. The authors would to like thank A. Sophia Koepke, Andrew Brown, Necati Cihan Camgöz, and Bencie Woll for their help. S.A would like to acknowledge the generous support of S. Carlson in enabling his contribution, and his son David, who bravely waited until after the submission deadline to enter this world.

Bibliography

[1] Sutton-Spence, R., Woll, B.: The Linguistics of British Sign Language: An Introduction. Cambridge University Press (1999) 1, 5

[2] Coucke, A., Chlieh, M., Gisselbrecht, T., Leroy, D., Poumeyrol, M., Lavril, T.: Efficient keyword spotting using dilated convolutions and gating. In: ICASSP. (2019) 1

[3] Véniat, T., Schwander, O., Denoyer, L.: Stochastic adaptive neural architecture search for keyword spotting. In: ICASSP. (2019) 1

[4] Momeni, L., Afouras, T., Stafylakis, T., Albanie, S., Zisserman, A.: Seeing wake words: Audio-visual keyword spotting. In: BMVC. (2020) 1

[5] Stafylakis, T., Tzimiropoulos, G.: Zero-shot keyword spotting for visual speech recognition in-the-wild. In: ECCV. (2018) 1

[6] Chung, J.S., Senior, A., Vinyals, O., Zisserman, A.: Lip reading sentences in the wild. In: CVPR. (2017) 2

[7] Afouras, T., Chung, J.S., Zisserman, A.: LRS3-TED: a large-scale dataset for visual speech recognition. arXiv preprint arXiv:1809.00496 (2018) 2

[8] 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) 2, 3, 4, 8, 9, 10, 11, 13, 22, 23

[9] Schembri, A., Fenlon, J., Rentelis, R., Cormier, K.: British Sign Language Corpus Project: A corpus of digital video data and annotations of British Sign Language 2008-2017 (Third Edition) (2017) 2

[10] Gutmann, M., Hyvärinen, A.: Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. (2010) 297–304 2, 7

[11] Kadir, T., Bowden, R., Ong, E.J., Zisserman, A.: Minimal training, large lexicon, unconstrained sign language recognition. In: Proc. BMVC. (2004) 3

[12] Tamura, S., Kawasaki, S.: Recognition of sign language motion images. Pattern Recognition **21** (1988) 343 – 353 3

[13] Starner, T.: Visual recognition of american sign language using hidden markov models. Master's thesis, Massachusetts Institute of Technology (1995) 3

[14] Fillbrandt, H., Akyol, S., Kraiss, K..: Extraction of 3D hand shape and posture from image sequences for sign language recognition. In: IEEE International SOI Conference. (2003) 3

[15] Buehler, P., Everingham, M., Zisserman, A.: Learning sign language by watching TV (using weakly aligned subtitles). In: Proc. CVPR. (2009) 3, 4

[16] Cooper, H., Pugeault, N., Bowden, R.: Reading the signs: A video based sign dictionary. In: ICCVW. (2011) 3

[17] Ong, E., Cooper, H., Pugeault, N., Bowden, R.: Sign language recognition using sequential pattern trees. In: CVPR. (2012) 3

[18] Pfister, T., Charles, J., Zisserman, A.: Domain-adaptive discriminative one-shot learning of gestures. In: Proc. ECCV. (2014) 3, 4
[19] Agris, U., Zieren, J., Canzler, U., Bauer, B., Kraiss, K.F.: Recent developments in visual sign language recognition. Universal Access in the Information Society **6** (2008) 323–362 3
[20] Forster, J., Oberdörfer, C., Koller, O., Ney, H.: Modality combination techniques for continuous sign language recognition. In: Pattern Recognition and Image Analysis. (2013) 3
[21] Camgoz, N.C., Hadfield, S., Koller, O., Bowden, R.: SubUNets: End-to-end hand shape and continuous sign language recognition. In: ICCV. (2017) 3
[22] Huang, J., Zhou, W., Zhang, Q., Li, H., Li, W.: Video-based sign language recognition without temporal segmentation. In: AAAI. (2018) 3, 4
[23] Ye, Y., Tian, Y., Huenerfauth, M., Liu, J.: Recognizing american sign language gestures from within continuous videos. In: CVPRW. (2018) 3
[24] Zhou, H., Zhou, W., Zhou, Y., Li, H.: Spatial-temporal multi-cue network for continuous sign language recognition. CoRR **abs/2002.03187** (2020) 3
[25] Camgoz, N.C., Koller, O., Hadfield, S., Bowden, R.: Sign language transformers: Joint end-to-end sign language recognition and translation. In: CVPR. (2020) 3
[26] Carreira, J., Zisserman, A.: Quo vadis, action recognition? A new model and the Kinetics dataset. In: CVPR. (2017) 3, 8, 23
[27] Joze, H.R.V., Koller, O.: MS-ASL: A large-scale data set and benchmark for understanding american sign language. In: BMVC. (2019) 3, 4
[28] Li, D., Opazo, C.R., Yu, X., Li, H.: Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In: WACV. (2019) 3, 4, 14
[29] Viitaniemi, V., Jantunen, T., Savolainen, L., Karppa, M., Laaksonen, J.: S-pot – a benchmark in spotting signs within continuous signing. In: LREC. (2014) 3, 4
[30] Eng-Jon Ong, Koller, O., Pugeault, N., Bowden, R.: Sign spotting using hierarchical sequential patterns with temporal intervals. In: CVPR. (2014) 3
[31] Farhadi, A., Forsyth, D.A., White, R.: Transfer learning in sign language. In: CVPR. (2007) 3
[32] Bilge, Y.C., Ikizler, N., Cinbis, R.: Zero-shot sign language recognition: Can textual data uncover sign languages? In: BMVC. (2019) 3
[33] Motiian, S., Jones, Q., Iranmanesh, S.M., Doretto, G.: Few-shot adversarial domain adaptation. In: NeurIPS. (2017) 4
[34] Zhang, J., Chen, Z., Huang, J., Lin, L., Zhang, D.: Few-shot structured domain adaptation for virtual-to-real scene parsing. In: ICCVW. (2019) 4
[35] Chang, W.G., You, T., Seo, S., Kwak, S., Han, B.: Domain-specific batch normalization for unsupervised domain adaptation. In: CVPR. (2019) 4, 24
[36] Li, D., Yu, X., Xu, C., Petersson, L., Li, H.: Transferring cross-domain knowledge for video sign language recognition. In: CVPR. (2020) 4

[37] Koller, O., Forster, J., Ney, H.: Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. Computer Vision and Image Understanding **141** (2015) 108–125 4
[38] von Agris, U., Knorr, M., Kraiss, K.: The significance of facial features for automatic sign language recognition. In: 2008 8th IEEE International Conference on Automatic Face Gesture Recognition. (2008) 4
[39] Athitsos, V., Neidle, C., Sclaroff, S., Nash, J., Stefan, A., Quan Yuan, Thangali, A.: The american sign language lexicon video dataset. In: CVPRW. (2008) 4
[40] Wilbur, R.B., Kak, A.C.: Purdue RVL-SLLL American sign language database. School of Electrical and Computer Engineering Technical Report, TR-06-12, Purdue University, W. Lafayette, IN 47906. (2006) 4
[41] Chai, X., Wang, H., Chen, X.: The devisign large vocabulary of chinese sign language database and baseline evaluations. Technical report VIPL-TR-14-SLR-001. Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS (2014) 4
[42] Schembri, A., Fenlon, J., Rentelis, R., Reynolds, S., Cormier, K.: Building the British sign language corpus. Language Documentation & Conservation **7** (2013) 136–154 4
[43] Cooper, H., Bowden, R.: Learning signs from subtitles: A weakly supervised approach to sign language recognition. In: CVPR. (2009) 4
[44] Chung, J.S., Zisserman, A.: Signs in time: Encoding human motion as a temporal image. In: Workshop on Brave New Ideas for Motion Representations, ECCV. (2016) 4
[45] Dietterich, T.G., Lathrop, R.H., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artificial intelligence **89** (1997) 31–71 4, 7, 25
[46] Pfister, T., Charles, J., Zisserman, A.: Large-scale learning of sign language by watching tv (using co-occurrences). In: BMVC. (2013) 4, 22
[47] Feng, Y., Ma, L., Liu, W., Zhang, T., Luo, J.: Video re-localization. In: ECCV. (2018) 5
[48] Yang, H., He, X., Porikli, F.: One-shot action localization by learning sequence matching network. In: CVPR. (2018) 5
[49] Cao, K., Ji, J., Cao, Z., Chang, C.Y., Niebles, J.C.: Few-shot video classification via temporal alignment. In: CVPR. (2020) 5
[50] Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018) 7, 12
[51] Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: CVPR. (2018) 7, 12, 13
[52] Miech, A., Alayrac, J.B., Smaira, L., Laptev, I., Sivic, J., Zisserman, A.: End-to-end learning of visual representations from uncurated instructional videos. In: CVPR. (2020) 7
[53] https://www.signbsl.com/: (British sign language dictionary) 10
[54] Hu, J., Shen, L., Albanie, S., Sun, G., Wu, E.: Squeeze-and-excitation networks. IEEE Transactions on Pattern Analysis and Machine Intelligence (2019) 10

[55] Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A.: VGGFace2: A dataset for recognising faces across pose and age. In: Proc. Int. Conf. Autom. Face and Gesture Recog. (2018) 10

[56] Cao, Z., Hidalgo, G., Simon, T., Wei, S.E., Sheikh, Y.: OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. In: arXiv preprint arXiv:1812.08008. (2018) 10

[57] He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR. (2020) 13

[58] SignumMcKee, D., Kennedy, G.: Lexical comparison of signs from American, Australian, British and New Zealand sign languages. The signs of language revisited: An anthology to honor Ursula Bellugi and Edward Klima (2000) 13

[59] Aldersson, R., McEntee-Atalianis, L.: A lexical comparison of Icelandic sign language and Danish sign language. Birkbeck Studies in Applied Linguistics **2** (2007) 13

[60] Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning for video understanding. In: ECCV. (2018) 23

[61] 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 25

APPENDIX

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 webpage[4] 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.

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

[4] https://www.robots.ox.ac.uk/~vgg/research/bsldict/

Supervision	Dictionary Vocab	mAP	R@5
Watch-Read-Lookup	800 training vocab	14.86 ± 1.29	19.85 ± 1.94
Watch-Read-Lookup	9k full vocab	**15.82** ± 0.48	**21.67** ± 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
✗	512	41.65	54.73
✗	1024	42.07	54.25
✗	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 mini-batches 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)

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 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.

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

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)

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)

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

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)

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 can be shorter) [46] 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 [8], 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 [8].

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).

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 [26] in all our experiments. Here, we also train a 1064-way classification with the S3D architecture [60] on BSL-1K as in [8] 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.

	per-instance		per-class	
Training data	top-1	top-5	top-1	top-5
S3D	64.76	81.88	46.27	63.71
I3D [8]	**75.51**	**88.83**	**52.76**	**72.14**

Table A.8: **Trunk network architecture:** We compare I3D [26] with the S3D [60] architecture for the task of sign language recognition, in a comparable setup to [8]. 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 [8] 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 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

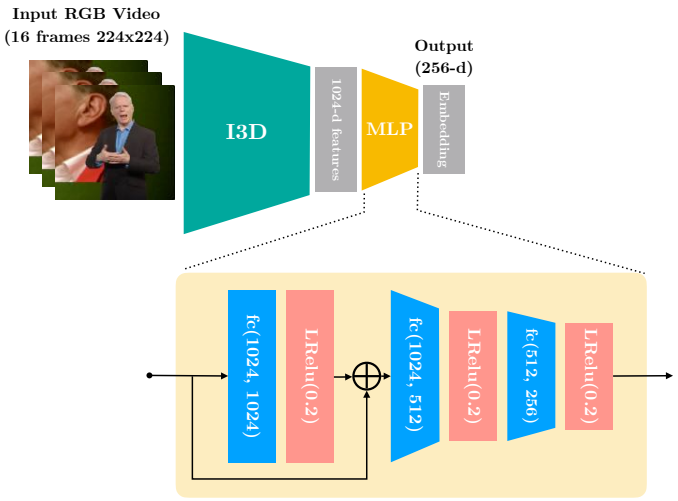


Fig. A.1: **MLP architecture:** We detail the layers of our embedding architecture. We freeze the I3D trunk and use it as a feature extractor. We only train the MLP layers with our loss formulation in the proposed framework. The same layers (and parameters) are used both for the dictionary video inputs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

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 [35], 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 [61] 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 [45] (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

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

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 $\mathcal{P}^{\text{bags(dict)}}_{\text{watch,lookup}}$, which takes a mathematically identical form to $\mathcal{P}^{\text{bags(seg)}}_{\text{watch,lookup}}$ (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), red examples); (2) anchoring on the dictionary set for a given word and selecting

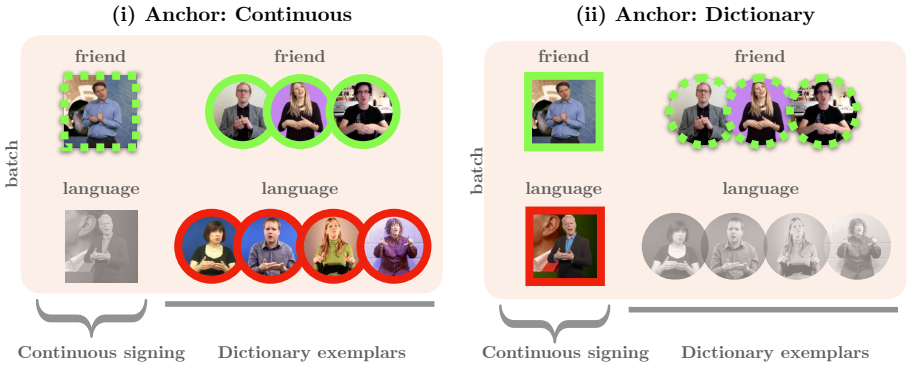


Fig. A.2: **Watch-Lookup:** We illustrate the batch formation and positive/negative sampling for the simplified version of our framework which is not using the subtitles, but only performing Watch-Lookup. We define two sets of positive/negative pairs, anchoring at a different position in each case. Anchor 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.

labelled segments of a different word (Fig. A.2 (ii), red example). These sets manifest as

$$\mathcal{N}^{\text{bags(seg)}}_{\text{watch,lookup}} = \{\{i\} \times B_i : (x_i^{\mathcal{M}}, v_i^{\mathcal{M}}) \in \mathcal{M}, (x_j^{\mathcal{D}}, v_j^{\mathcal{D}}) \in \mathcal{D}, B_i = \{j : v_j^{\mathcal{D}} \neq v_i^{\mathcal{M}}\}\} \tag{3}$$

for the former and as

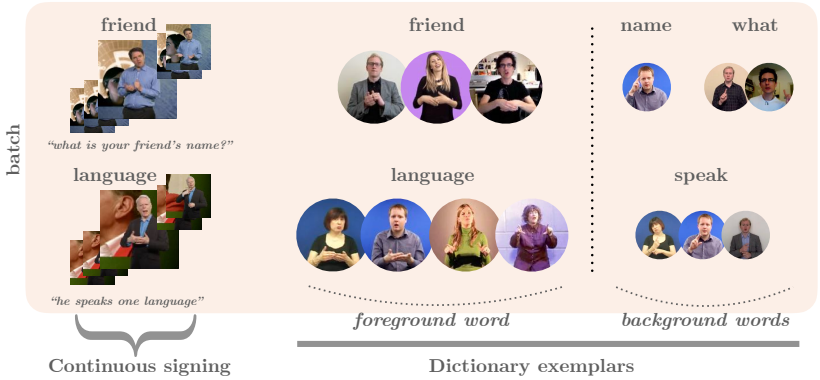
$$\mathcal{N}^{\text{bags(dict)}}_{\text{watch,lookup}} = \{A_i \times B_i : A_i = \{l : x_l, x_i \subseteq x_k, (x_k, s_k) \in \mathcal{S}, x_l \cap x_i = \emptyset\} \\ B_i = \{j : v_j^{\mathcal{D}} \neq v_i^{\mathcal{M}}\}, (x_i^{\mathcal{M}}, v_i^{\mathcal{M}}) \in \mathcal{M}, (x_j^{\mathcal{D}}, v_j^{\mathcal{D}}) \in \mathcal{D}\}. \tag{4}$$

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

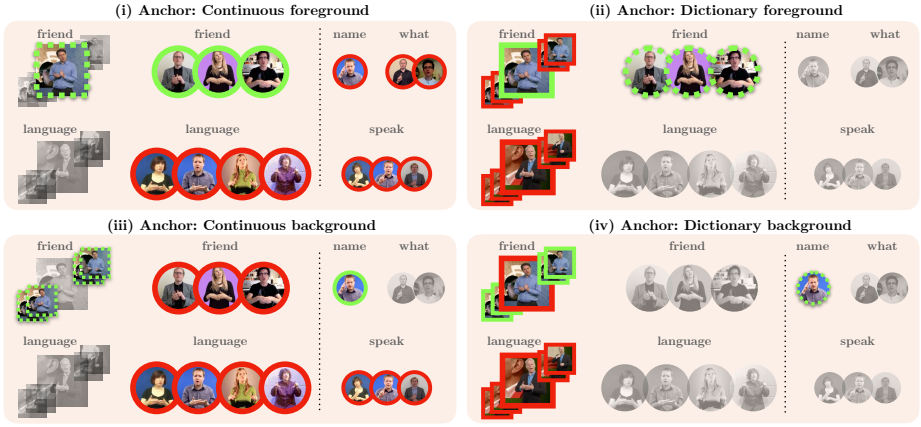
$$\mathcal{P}^{\text{bags}}_{\text{watch,lookup}} \triangleq \mathcal{P}^{\text{bags(seg)}}_{\text{watch,lookup}} \cup \mathcal{P}^{\text{bags(dict)}}_{\text{watch,lookup}} \tag{5}$$

and

$$\mathcal{N}^{\text{bags}}_{\text{watch,lookup}} \triangleq \mathcal{N}^{\text{bags(seg)}}_{\text{watch,lookup}} \cup \mathcal{N}^{\text{bags(dict)}}_{\text{watch,lookup}}. \tag{6}$$



(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, $\mathcal{P}^{\text{bags}}_{\text{watch,lookup}}$ (Fig. A.3 (i) and (ii), top rows) which correspond to the use of foreground subtitle words. However, these can now by 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 subtitles 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 $\text{Tokenize}(\cdot) : \mathcal{S} \rightarrow \mathcal{V}_{\mathfrak{L}}$ denote the function which extracts words from the subtitle that are present in the vocabulary: $\text{Tokenize}(s) \triangleq \{w \in s : w \in \mathcal{V}_{\mathfrak{L}}\}$. Then define background segment-anchored positive bags as:

$$\mathcal{P}^{\text{bags(seg-back)}}_{\text{watch,read,lookup}} = \{\{i\} \times B_i : \exists (x_k, s_k) \in \mathcal{S} \text{ s.t } x_i \subseteq x_k, (x_j^{\mathcal{D}}, v_j^{\mathcal{D}}) \in \mathcal{D}, \quad (7)$$
$$B_i = \{j : v_j^{\mathcal{D}} \in \text{Tokenize}(s_k)\}, (x_i, v_i) \notin \mathcal{M}\}$$

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:

$$\mathcal{P}^{\text{bags(dict-back)}}_{\text{watch,read,lookup}} = \{A_i \times B_i : (x_i^{\mathcal{D}}, v_i^{\mathcal{D}}) \in \mathcal{D}, A_i = \{j : v_i^{\mathcal{D}} \in \text{Tokenize}(s_k), \quad (8)$$
$$(x_k, s_k) \in \mathcal{S}, x_j \subseteq x_k, (x_j, v_j) \notin \mathcal{M}\}, B_i = \{l : v_l^{\mathcal{D}} = v_i^{\mathcal{D}}\}\}$$

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:

$$\mathcal{P}^{\text{bags}}_{\text{watch,read,lookup}} = \mathcal{P}^{\text{bags}}_{\text{watch,lookup}} \cup \mathcal{P}^{\text{bags(seg-back)}}_{\text{watch,read,lookup}} \cup \mathcal{P}^{\text{bags(dict-back)}}_{\text{watch,read,lookup}}. \quad (9)$$

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 negatives bags on foreground segments:

$$\mathcal{N}^{\text{bags(seg-fore)}}_{\text{watch,read,lookup}} = \{\{i\} \times B_i : (x_i^{\mathcal{M}}, v_i^{\mathcal{M}}) \in \mathcal{M}, (x_j^{\mathcal{D}}, v_j^{\mathcal{D}}) \in \mathcal{D}, \quad (10)$$
$$B_i = \{j : v_j^{\mathcal{D}} \neq v_i^{\mathcal{M}}\}\}$$

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:

$$\mathcal{N}^{\text{bags(dict-fore)}}_{\text{watch,read,lookup}} = \{A_i \times B_i : (x_i^{\mathcal{D}}, v_i^{\mathcal{D}}) \in \mathcal{D}, A_i = \{j : v_i^{\mathcal{D}} \in \text{Tokenize}(s_k), \\ (x_k, s_k) \in \mathcal{S}, x_j \subseteq x_k, (x_j, v_j) \notin \mathcal{M}\} \cup \{(x_m, v_m) \in \mathcal{M}, v_m \neq v_i\}, \\ B_i = \{l : v_l^{\mathcal{D}} = v_i^{\mathcal{D}}\}\} \tag{11}$$

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:

$$\mathcal{N}^{\text{bags(seg-back)}}_{\text{watch,read,lookup}} = \{\{i\} \times B_i : \exists (x_k, s_k) \in \mathcal{S}, x_i \subseteq x_k, (x_j^{\mathcal{D}}, v_j^{\mathcal{D}}) \in \mathcal{D}, \\ B_i = \{j : v_j^{\mathcal{D}} \notin \text{Tokenize}(s_k)\}\} \tag{12}$$

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:

$$\mathcal{N}^{\text{bags(dict-back)}}_{\text{watch,read,lookup}} = \{A_i \times B_i : (x_i^{\mathcal{D}}, v_i^{\mathcal{D}}) \in \mathcal{D}, A_i = \{j : v_i^{\mathcal{D}} \notin \text{Tokenize}(s_k), \\ (x_k, s_k) \in \mathcal{S}, x_j \subseteq x_k, (x_j, v_j) \notin \mathcal{M}\} \cup \{(x_m, v_m) \in \mathcal{M}, v_m \neq v_i\}, \\ B_i = \{l : v_l^{\mathcal{D}} = v_i^{\mathcal{D}}\}\} \tag{13}$$

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:

$$\mathcal{N}^{\text{bags}}_{\text{watch,read,lookup}} = \mathcal{N}^{\text{bags(seg-fore)}}_{\text{watch,read,lookup}} \cup \mathcal{N}^{\text{bags(seg-dict)}}_{\text{watch,read,lookup}} \\ \cup \mathcal{N}^{\text{bags(seg-back)}}_{\text{watch,read,lookup}} \cup \mathcal{N}^{\text{bags(dict-back)}}_{\text{watch,read,lookup}}. \tag{14}$$

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 $\mathcal{P}^{\text{bags}} = \mathcal{P}^{\text{bags}}_{\text{watch,lookup}}$, $\mathcal{N}^{\text{bags}} = \mathcal{N}^{\text{bags}}_{\text{watch,lookup}}$ and the *Watch-Read-Lookup* formulation instead defines the positive and negative bags via $\mathcal{P}^{\text{bags}} = \mathcal{P}^{\text{bags}}_{\text{watch,read,lookup}}$, $\mathcal{N}^{\text{bags}} = \mathcal{N}^{\text{bags}}_{\text{watch,read,lookup}}$.

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.