SignMusketeers: An Efficient Multi-Stream Approach for Sign Language Translation at Scale

Shester Gueuwou, Xiaodan Du, Greg Shakhnarovich, Karen Livescu
TTI-Chicago
{shesterg,xdu,greg,klivescu}@ttic.edu
http://signmusketeers.pals.ttic.edu

Abstract

A persistent challenge in sign language video processing, including the task of sign to written language translation, is how we learn representations of sign language in an effective and efficient way that preserves the important attributes of these languages, while remaining invariant to irrelevant visual differences. Informed by the nature and linguistics of signed languages, our proposed method focuses on just the most relevant parts in a signing video: the face, hands and body pose of the signer. However, instead of fully relying on pose estimation from off-the-shelf pose tracking models, which have inconsistent performance for hands and faces, we propose to learn a representation of the complex handshapes and facial expressions of sign languages in a self-supervised fashion. Our approach is based on learning from individual frames (rather than video sequences) and is therefore much more efficient than prior work on sign language pre-training. Compared to a recent model that established a new state of the art in sign language translation on the How2Sign dataset, our approach yields similar translation performance, using less than 3% of the compute.

1 Introduction

Recent work on sign language processing spans human-computer interaction (Zafrulla et al., 2010; Bragg et al., 2021), computer vision (Varol et al., 2021; Sandoval-Castaneda et al., 2023), and natural language processing (NLP) (Yin et al., 2021; Müller et al., 2022) research. The nature of signed languages, which involve the use of manual features (handshape, orientation, location, and movement) and non-manual features (facial expressions, head movements, body pose), presents challenges for machine learning models (Bragg et al., 2019). In particular, how to effectively and efficiently represent signed languages while preserving their inherent attributes remains a persistent challenge.

Our focus is on the task of translation from sign language video to a written language (sign language translation, or SLT). This is one of the most practically important tasks, necessary to bridge (part of) the communication gap between Deaf and hard of hearing (DHH) populations and hearing populations (Fox et al., 2023). Recent work (Rust et al., 2024) proposed a self-supervised video pre-training approach to handle sign language translation, which achieved state-of-the-art performance on the How2Sign dataset of American Sign Language (ASL) to English translation (Duarte et al., 2021). The intuition behind this approach is to extend the pre-training of a strong video model (in this case, Hiera (Ryali et al., 2023)) with a large-scale unannotated set of sign language videos (in this case from YouTube-ASL (Uthus et al., 2023)) and use this extended pre-trained model as a feature extractor for the supervised translation task. However, this pre-training is extremely costly: Its longest pre-training run uses 64 A100 80GB GPUs for 14 days, making the approach infeasible for many researchers and practitioners.

The approach in Rust et al. (2024) implicitly treats sign language sequences like any other (long) videos. But signed languages are, first and foremost, languages, and like any other language, they possess linguistic properties that may provide an inductive bias about the more important aspects of the video (Brentari, 1998; Sutton-Spence and Woll, 1999). In this work, we ask the question: *Can we infuse basic linguistic properties of signed languages into self-supervised pre-training to develop a scalable compute-friendly approach?*

In the context of signed language processing, the use of off-the-shelf human pose estimator models (Cao et al., 2017; Lugaresi et al., 2019) has been one of the most common ways of incorporating the linguistic constraints of sign language into models. The intuition of using pose estimation is that it removes irrelevant features that do not

Findings of the Association for Computational Linguistics: ACL 2025, pages 22506–22521

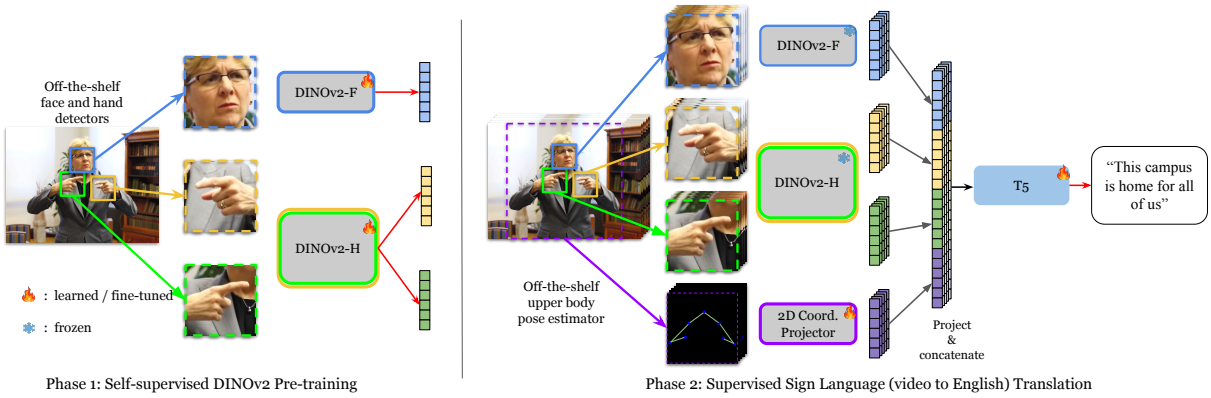

Figure 1: Overview of our approach to sign language translation. We parse every frame of the signing video with off-the-shelf face and hand detectors. (a) In phase 1 (left) we start from pre-trained DINOv2 visual feature extractors and continue training them with a DINO loss on cropped face boxes and hand boxes, producing two separate DINOv2s (DINOv2-F for the face and DINOv2-H for the hands). This stage is purely self-supervised from random video frames; see also Fig. 3 for more detail. (b) In phase 2 (right), fixing the two pre-trained feature extractors, we add a (learned) feature extractor for coarse body pose estimated by an off-the-shelf method (Lugaresi et al., 2019), concatenate and project the features for each frame, and fine-tune a T5 model mapping the resulting sequence of frame features to English text. This stage is supervised by video clips paired with translations.

affect the meaning of signs, such as body shape and visual background, and therefore focuses entirely on the linguistically relevant aspects of hand, face and body pose (De Coster et al., 2023). However, this pose-based approach has several limitations that make it sub-optimal for capturing the details of signed languages, particularly in the representation of hands (Moryossef et al., 2021) and faces (Kuznetsova and Kimmelman, 2024). First, the human pose estimator models used in existing methods (Cao et al., 2017; Lugaresi et al., 2019; Contributors, 2020) are typically trained on everyday handshapes, which are often less complex than the handshapes found in signed languages. Second, human pose estimators are unreliable in capturing crucial non-manual components which are essential for signed languages, such as eye gaze.[1]

Our approach is inspired by the multi-stream/multi-channel property of signed languages (§2.2); that is, the fact that they consist of a combination of actions performed largely independently by multiple body parts (channels). Specifically, our proposed method (§3, see overview in Fig. 1) focuses on the most relevant parts of a signing video—the face, hands, and body pose of the signer (§3.1)—to handle sign language translation at scale (§2.1). Instead of fully relying on off-the-shelf human pose estimators, we propose to learn representations of handshapes and facial expressions directly from signing videos using self-supervised learning (§3.2, 2.3.) Our method captures the intricacies of handshapes and facial expressions for the supervised training stage (§3.3), without the need for extensive pre-training data or computational resources. This allows us to overcome the limitations of human pose estimators and preserve the crucial linguistic information conveyed through handshapes, eye gaze, and facial expressions in signed languages. We name our approach SignMusketeers: Like the heroes of Dumas' books (Dumas, 1894), three image channels (face and two hand boxes) join forces with a fourth companion (pose features) in the quest for glory (accurate sign language translation).

We conduct experiments (§4) on How2Sign and find that our approach achieves competitive performance while using a smaller model (in terms of number of parameters), with 41x less pre-training data and 160x fewer pre-training iterations (§4.1) , using roughly 3% of the compute resources of the previous state-of-the-art approach (Rust et al., 2024). We also provide ablation experiments showing the value of individual design decisions (§A).

2 Related Work

2.1 Sign Language Translation at Scale

Until very recently, the lack of large-scale datasets has been a major obstacle in advancing sign language translation. Most research has been con-

[1]For instance, in British Sign Language, the main difference between the signs for "God" and "Boss" lies in the eye gaze, which existing human pose estimators do not capture (Sutton-Spence and Woll, 1999).

ducted on small datasets such as PHOENIX14T (Camgoz et al., 2018), with only 9 hours of content and a modest vocabulary size of 3,000. While fairly high BLEU[2] scores (>20) have been reported on this dataset, translation of more realistic video is far more challenging. Recent efforts to create larger datasets, such as BOBSL (∼1,500 hours) (Albanie et al., 2021), OpenASL (∼300 hours) (Shi et al., 2022), JWSign (∼2,500 hours) (Gueuwou et al., 2023), and the SRF corpus (∼400 hours) (Müller et al., 2023), have revealed the difficulty of the task, with BLEU scores only around 2-7.

In this study, we do not include common benchmark datasets like PHOENIX14T (Camgoz et al., 2018) and CSL-Daily (Zhou et al., 2021a) due to two main reasons: (i) The goal of our study is to investigate sign language translation at *scale*. We focus on American Sign Language (ASL) because it has easily accessible large-scale datasets for pre-training and smaller datasets for fine-tuning. At the time of this study, there were no easily accessible large-scale datasets for German Sign Language or Chinese Sign Language that would support our approach. (ii) Using YouTube-ASL (Uthus et al., 2023) for pre-training and How2Sign (Duarte et al., 2021) for fine-tuning follows established precedent in prior work (mentioned below), enabling direct comparisons with our method.

Uthus et al. (2023) used the YouTube-ASL dataset for large-scale training of ASL translation models. By fine-tuning a T5 (Raffel et al., 2020) model on YouTube-ASL, then fine-tuning it on the smaller dataset How2Sign (Duarte et al., 2021), the authors achieved a BLEU score of 12.39. The input to the T5 model consisted of selected human poses obtained from the off-the-shelf human pose estimator MediaPipe (Lugaresi et al., 2019).

Building upon this paradigm, Rust et al. (2024) further improved performance on How2Sign by training a video encoder on YouTube-ASL initialized from a self-supervised video model (Hiera-Base) pre-trained as a masked autoencoder. However, this approach is computationally prohibitive, requiring 64 A100 80GB GPUs for 14 days for a single training run. The authors found that the good results indeed depend on these large compute requirements: Significantly reducing the number of video frames ingested by the encoder from 128, or the number of pre-training iterations from 800 epochs, greatly reduced performance. One key feature of this method is its privacy-awareness through face blurring.[3] However, the face is an important non-manual cue that helps disambiguate some statements. We also note that face blurring may not be sufficient for preserving privacy, especially with large-scale datasets (Oh et al., 2016).

We propose an alternative approach that focuses on embedding fine-grained handshapes and facial expressions using an image encoder with a smaller ViT backbone. Our encoder takes just a single frame at a time and requires much lower training time and compute resources.

2.2 Multi-Channel Sign Language Processing

Signed languages are inherently multi-channel, employing a combination of manual features (handshapes, orientation, location, and movement) and non-manual features (facial expressions, head movements, and (upper) body pose) to convey meaning (Sandler and Lillo-Martin, 2006; Pfau et al., 2010; Brentari, 2019). Multi-channel processing aims to capture and integrate these diverse sources of information for various tasks, such as sign language recognition, translation, and generation. The concept of tackling (American) sign language processing through a multi-channel approach was first introduced in the early 2000s (Vogler and Metaxas, 2001), inspired by linguistic evidence that American Sign Language can be modeled, at least partially, as a combination of independent channels (Liddell and Johnson, 1989). Over the years, several other approaches have used multi-channel ideas for sign language recognition (Holden et al., 2005; Pu et al., 2016) and later translation (Camgoz et al., 2020; Zhou et al., 2021b; Shi et al., 2022) and production (Saunders et al., 2020; Tornay et al., 2020) tasks, although the specific channels and how they are used varies. One common characteristic in these approaches is that the feature extractors for the different components were not learned specifically for sign languages. This is an example of a general issue in sign language research that the methods are not sufficiently adapted to the needs of these languages (Fox et al., 2023; Desai et al., 2024).

[2]Unless specified otherwise, BLEU means BLEU-4 scores, computed with `sacrebleu` (Post, 2018) `version : BLEU+c.mixed+#.1+s.exp+tok.13a+v.1.4.1.`

[3]The details of the blurring approach in (Rust et al., 2024), that uses an internal software tool, have not been published, making it irreproducible.

2.3 Face, Hands, and Body Pose Representation Learning

To achieve our goal of learning semantically meaningful multi-channel features independently of irrelevant visual details, we draw inspiration from prior work related to facial expression representation learning, hand pose (or shape) estimation, and body pose learning.

Facial expression. One approach for face analysis is da Silva et al. (2020), which investigates the recognition of affective and grammatical facial expressions in Brazilian Sign Language (Libras). The authors utilize a combination of geometric features, such as facial landmarks, and appearance features to represent facial expressions. Another approach is MARLIN (Cai et al., 2023), a masked autoencoder for facial video representation learning, which is effective on various facial analysis tasks, including facial expression recognition. Gao and Patras (2024) propose a self-supervised learning approach for facial representation learning with facial region awareness. This method leverages the structure of the human face by dividing it into regions, such as eyes, nose, and mouth. The authors utilize BYOL (Grill et al., 2020), a popular self-supervised learning framework based on instance discrimination. Our approach shares some ideas with the work of Gao and Patras (2024); however, instead of using BYOL, we employ DINOv2 (Oquab et al., 2023), a state-of-the-art visual self-supervised learning framework that builds upon a similar principle of instance discrimination as BYOL.

Hand shape and orientation. DeepHand (Koller et al., 2016) is a convolutional neural network (CNN) approach for hand shape classification in continuous sign language video streams. It addresses the challenge of weakly labeled data by proposing a training strategy that exploits the temporal coherence of hand shapes within a sign. Zimmermann et al. (2021) propose a contrastive representation learning approach for hand shape estimation, which learns hand shape representations by contrasting positive and negative pairs of hand images in a self-supervised manner, using a novel loss function that encourages invariance to changes in viewpoint, articulation, and lighting conditions. FineHand (Santhalingam et al., 2020) is a deep learning approach specifically designed for American Sign Language (ASL) recognition. Our approach draws inspiration from these studies and aims to learn hand shape representations in a self-supervised manner using DINOv2, which enables us to capture the fine-grained details of hand shapes and orientation without relying on explicit annotations. We note also that Wong et al. (2024) used DINOv2 as a feature extractor backbone for sign language translation.

Upper body pose. Compared to analysis of the fine-grained gestures of the face and hands, techniques for general (global) human pose estimation (Hachiuma et al., 2023; Yan et al., 2023) are more mature and robust. In our approach, we therefore simply utilize an off-the-shelf human pose estimation model, MediaPipe (Lugaresi et al., 2019), to represent the body pose of a signer.

3 Method

Below we first describe the data preprocessing (frame parsing) procedure, then the self-supervised pre-training of feature extractors, and finally the supervised learning of the ASL video to English translation system.

3.1 Frame Parsing

Our frame parsing pipeline extracts and normalizes the relevant regions of interest (ROIs) from the sign language video frames, focusing on the face, left hand, right hand, and upper body pose. Each of these four components is mapped to a feature vector (channel); concatenating and projecting the four channels for each frame yields the frame vector, which is then fed to a sequence model for translation. We use the MediaPipe Holistic framework (Lugaresi et al., 2019) to extract face, hand, and pose landmarks from the video frames.

Face cropping To extract the face ROI, we first determine the smallest square bounding box that can fit all the face landmarks while preserving the aspect ratio of the face. This initial bounding box is then scaled up by a factor of 1.2 in each dimension to compensate for any parts of the face that might have been missed.

In cases where face landmarks are not detected, we estimate the face region using the upper body pose landmarks (indices 0 to 10). The bounding box is adjusted to ensure it fits within the frame boundaries.

Hand cropping For the hand ROIs, we follow a similar approach to the face ROI extraction when

hand landmarks are available: We determine the smallest square bounding box that can fit all the hand landmarks while preserving the aspect ratio of the hand and scale it by a factor of 1.2.

In cases where hand landmarks are not detected, we estimate the hand regions using the few finger pose landmarks (indices 17, 19, 21 for the left hand fingers and 18, 20, 22 for the right hand fingers. See Figure 4 in Appendix D for an ilustrative diagram). Although these are often inaccurate, they provide a good estimate of the hand location(s). We then create a square bounding box of the same size as the face bounding box with its center at the mean of the relevant pose landmarks.

To handle the occasional cases where the MediaPipe hand landmark detector returns erroneous values when the hand is not in the frame, we adjust the bounding boxes to maintain temporal consistency across the channels. Specifically, we use two strategies: shifting the bounding box inward to keep the hand within the frame or using the last previously detected hand ROI before it went out of the frame.

The extracted face and hand ROIs are then resized to a fixed 224x224 pixels using bicubic interpolation while preserving the aspect ratio. This resizing step ensures consistent input dimensions for the self-supervised learning models in the next stage.

Upper body pose normalization We extract body poses from the relevant upper body landmarks, and normalize them to encourage invariance to position and scale differences. Specifically, we extract MediaPipe human poses for the nose (index 0), left shoulder (index 11), right shoulder (index 12), left elbow (index 13), right elbow (index 14), left wrist (index 15), and right wrist (index 16). We assume that these seven landmarks are enough to capture the essential components of the upper body pose needed to recognize movements and spatial positions of the hands and face with respect to each other, leaving the finer-grained hand pose and facial expression to the other channels (face and hand ROIs).

Next, we define a normalized signing space based on the signer's body proportions, similarly to Boháček and Hrúz (2022). We define the head unit as the distance between the left and right shoulders divided by 2. The signing space width is set to 6 times the head unit, and the signing space height is set to 7 times the head unit. The signing space bounding box is determined using the left eye and nose landmarks as reference points.

Finally, we normalize the seven pose coordinates by scaling the bounding box of the signing space to unit width/height with center at (0.5,0.5). The normalized coordinates are then flattened into a (14-dimensional) vector.

To handle cases where the pose landmarks are not detected in a frame, we employ a strategy similar to the one used for the hands: If the pose landmarks are not detected and there is a pose available from a previous frame, we use the previous pose for the current frame. If there is no previous pose available, we create a placeholder array of negative values to indicate missing data (Uthus et al., 2023).

3.2 Self-Supervised Sign Components Pre-training

Our method has two training phases. The first stage is self-supervised, and aims to produce encoders that specialize in sign language facial expressions and hand gestures. Using the DINOv2-Small architecture (Oquab et al., 2023; Darcet et al., 2024), we pre-train the two encoders separately. We initialize ViT-small *student* and *teacher* backbones (due to computational constraints, we could not use a ViT-Base backbone as in Rust et al. (2024)) with the teacher weights from the original dinov2_vits14_reg checkpoint, while the linear heads are randomly initialized.

We largely follow the training protocols of the DINOv2 paper with the recommended 4 registers (Darcet et al., 2024). The input face/hand images are randomly transformed into 2 global and 8 local views, using a scale of 0.5 to 1.0, and 0.25 to 0.5, respectively. Then, all views (both global and local) are passed to the student network. The student features are normalized with a Softmax to obtain the score vectors p_s. The teacher network, on the other hand, only accepts global views as input. We use the Sinkhorn-Knopp centering algorithm (Caron et al., 2020) on the features from the teacher network to obtain p_t. We compute cross-entropy loss between p_s and p_t and use it to update the student network. The gradient backpropagation is disabled for the teacher network. The weights of the teacher network are updated with an exponential moving average (ema) of the student network weights. In addition, DINOv2 also masks out random patches of the input views to the student and computes a patch-level cross-entropy loss (iBOT loss (Zhou et al., 2022)) between the masked student tokens

Figure 2: Comparison of data and computation usage between SignMusketeers (Ours) and Rust et al. (2024). Horizontal axis: GPU-Hours for the entire training schedule, including self-supervised training and supervised training. Vertical axis: BLEU score. Bubble size: number of frames (in millions) used during the pre-training stage. Labels: the first line is the pre-training protocol and the second line is the supervised training protocol. The number in parentheses is the number of pre-training epochs. YT: YouTube-ASL, H2S: How2Sign; X→Y means train on X then fine-tune on Y; X+Y means train on the union X∪Y. Note: GPU-Hours for Rust et al. (2024) is computed based on Section C.3 of Rust et al. (2024).

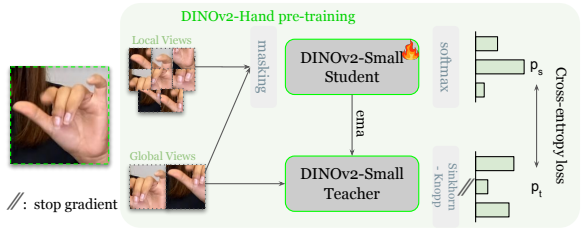

Figure 3: Self-supervised pre-training of DINOv2 on hand crops (stage 1 of our approach), yielding the hand-specific DINOv2 feature extractor. We pool the right and left hand boxes. We repeat this step separately for face boxes, yielding the face-specific feature extractor.

and the corresponding visible teacher tokens. Fig. 3 illustrates this setup.

We do this self-supervised pre-training using 1 million face crops and hand crops (obtained as described in Section (§3.1)) independently. We use a base learning rate of 2×10^{-4} and batch size per GPU of 128 on 8 A6000Ada GPUs (i.e., an effective batch size of 1024). To account for our relatively small dataset, we follow the recommendation of Roth et al. (2024) and adjust the number of iterations per pseudo-epoch and the number of pseudo-epochs, resulting in 5 effective epochs in total. In the KoLeoLoss we change the hyperparameter ϵ from 10^{-8} to 10^{-4} to avoid infinite loss values.

The resulting teacher networks are used as our face encoder (DINOv2-F) and hand encoder (DINOv2-H) for the next stage, which is supervised training. We note that for the face, we found that blurring the whole face directly hurts performance. However, a modified approach of first greying areas except the eyes and mouth, and then blurring the remaining image, performs similarly to training on the whole face crop. We therefore use this strategy, which aims to balance between keeping important facial attributes while also having some privacy-awareness.

3.3 Supervised Sign Language Translation

Given a video of T frames with an associated written language translation, we first obtain channel crops as described in (§3.1). Each crop is passed to the relevant frozen encoder, which is obtained as described in (§3.2). This results in 3 $T \times 384$ matrices for the face, left hand and right hand. These matrices are projected to $T \times 256$ via stream-specific linear layers. The normalized body pose vectors are transformed to a higher dimensionality, from $T \times 14$ to $T \times 128$ (via a linear layer trained from scratch). All four feature streams (face, left hand, right hand, and body pose) are then temporally

concatenated and projected to $T \times 768$ (the input size of the T5 model) via another linear layer, also trained from scratch.

To summarize: In stage 1, we independently pre-train two (face- and hand-specific) image-to-vector feature extractors. In stage 2, we jointly train a human pose feature linear transformation layer and a single linear layer transformation for the concatenated four-stream features, and we fine-tune the T5 model for translation. Both stages are shown in Figure 1.

4 Experiments

As in other work (Uthus et al., 2023; Rust et al., 2024), we use the YouTube-ASL and How2Sign (Duarte et al., 2021) datasets. YouTube-ASL contains roughly 600,000 clips, or roughly 700 hours, of ASL video with weakly aligned English text translations.[4] How2Sign consists of 31,128 / 1,741 / 2,322 clips for the training / validation / test sets.

For the self-supervised training of DINOv2-Hand and DINOv2-Face, due to computational constraints we limit ourselves to 1 million random face crops and 1 million random hand crops from YouTube-ASL. We note that the state-of-the-art method proposed by Rust et al. (2024) sees about 50 million frames during pre-training.

In line with previous studies, we employ the following training schedules for the supervised (translation) stage, using a stride of 2 for every video clip:

H2S: Supervised training exclusively on the How2Sign dataset, without using the YouTube-ASL dataset.

YT: Supervised training solely on YouTube-ASL, and evaluation on How2Sign in a zero-shot setting.

YT→H2S: Supervised training on YouTube-ASL, followed by supervised fine-tuning on How2Sign.

Note that Uthus et al. (2023) and Rust et al. (2024) include an additional training schedule, YT + H2S, which involves training on a mixture of YT and H2S.

For the supervised training stage, similarly to other work (Uthus et al., 2023; Rust et al., 2024), we initiliaze our T5 from a T5.1.1-Base pre-trained checkpoint. We use a batch size of 128 (16 per GPU running on 8 GPUs), with other hyperparameters identical to Rust et al. (2024). Additionally, when further fine-tuning on How2Sign after training on YouTube-ASL, we perform an extra 5,000 steps of fine-tuning.

4.1 Comparison to prior work

Table 1 compares our models to prior results on the How2Sign ASL-English translation task, including two approaches that train on a combination of YouTube-ASL and How2Sign. First, we observe that our method consistently outperforms the approach of Uthus et al. (2023) across various metrics. For example, we improve BLEU by 1.9 points when using the YT→H2S training schedule. This improvement provides evidence for the benefits of using learned features over pose estimator features/coordinates. When using the H2S-only supervised training schedule, our scores are 1.2 BLEU scores above the ones of Uthus et al. (2023) on this same training schedule (1.2 BLEU vs. 2.4 BLEU).

Comparing with the SSVP-SLT approach of Rust et al. (2024), in the most restrictive H2S-only schedule, our method is far behind (by 9 BLEU points). However, in the best-performing schedule (YT→H2S), the gap between the two approaches reduces significantly, and our performance is just 0.4 BLEU points below that of Rust et al. (2024).[5] We note that the same trend holds for the method of Uthus et al. (2023) when compared to SSVP-SLT. On H2S alone, there is a substantial gap between the two methods (+10.5 BLEU points), but on YT→H2S, the gap reduces to +2.4 BLEU points.

Human pose estimation, as used in the method of Uthus et al. (2023), is inherently multi-stream since it returns vectors (coordinates) describing multiple body parts, which are later concatenated. We suspect that both of the multi-stream approaches do not perform well with small datasets in supervised training. We hypothesize that this might be because the features they return are entirely frame-level features and do not *yet* contain any information about how these frame-level features relate to each other. In contrast, SSVP-SLT pre-trained features already

[4] Although Uthus et al. (2023) report a total of 610,193 clips in YouTube-ASL, we were only able to retrieve 601,995 clips, presumably because some clips have been deleted between the time of the dataset's creation and our retrieval.

[5] It is important to note that the authors of Rust et al. (2024) report an even higher BLEU score of 15.5 for a method that trains an additional CLIP (Radford et al., 2021) model on English text, on the union of YouTube-ASL and How2Sign. We include this result in Figure 2 (top right). Such techniques might also improve our performance, but we are unable to do the experiment due to computation constraints.

METHOD	BLEU-1	BLEU-2	BLEU-3	BLEU	GPU-Hrs	PT % of Frames
Supervised training Schedule: H2S						
Lin et al. (2023)	14.9	7.3	3.9	2.2	—	—
Tarrés et al. (2023)	34.0	19.3	12.2	8.0	—	—
Uthus et al. (2023)	15.0	5.1	2.3	1.2	—	—
SSVP-SLT $^{\text{YT}(50)}_{800}$ (Rust et al., 2024)	38.1	23.7	16.3	11.7	18535*	50
SignMusketeers $^{\text{YT}(1.2)}_{5}$ (Ours)	18.8	8.1	4.2	2.4	592	1.2
Supervised training Schedule: YT						
Uthus et al. (2023)	20.9	10.4	6.1	4.0	—	—
SSVP-SLT $^{\text{YT}(50)}_{800}$ (Rust et al., 2024)	29.2	16.6	10.7	7.1	18729*	50
SignMusketeers $^{\text{YT}(1.2)}_{5}$ (Ours)	26.3	13.8	8.2	5.2	864	1.2
***Supervised training Schedule: YT + H2S**						
Uthus et al. (2023)	36.3	23.0	16.1	11.9	—	—
SSVP-SLT $^{\text{YT}(50)}_{800}$ (Rust et al., 2024)	41.6	27.2	19.3	14.3	18768*	50
SSVP-SLT $^{\text{YT+H2S}(50)}_{100}$ (Rust et al., 2024)	—	—	—	12.5	2754*	50
SSVP-SLT-LSP $^{\text{YT+H2S}}_{600\rightarrow200}$ (Rust et al., 2024)	43.2	28.8	20.8	15.5	32912*	50
Supervised training Schedule: YT → H2S						
Uthus et al. (2023)	37.8	24.1	16.9	12.4	—	—
SSVP-SLT $^{\text{YT}(50)}_{800}$ (Rust et al., 2024)	41.9	27.7	19.8	14.7	18768*	—
SignMusketeers $^{\text{YT}(1.2)}_{5}$ (Ours)	41.5	27.2	19.3	14.3	880	1.2

Table 1: Quantitative results on How2Sign. GPU-Hrs = GPU hours used during the entire training stage. PT % of frames = Percentage of YouTube-ASL frames used in the self-supervised pre-training stage. *Adjusted by throughput ratio reported at https://lambdalabs.com/gpu-benchmarks. *We did not include YT + H2S experiments as this setting does not reflect the common paradigm of pre-training on a large dataset and fine-tuning on a different smaller dataset. The YT → H2S paradigm better represents adaptability to real-world use cases.

have some information about how frames relate to each other.

Nevertheless, with a larger dataset during the supervised training stage, the lead of SSVP-SLT diminishes drastically compared to multi-stream approaches like Uthus et al. (2023) and our SignMusketeers method. This suggests that as the amount of labeled training data increases, the advantage of pre-trained features that capture some temporal relationships becomes less pronounced, and multi-stream approaches can bridge the performance gap.

We suspect that the 0.4 BLEU difference between our result and that of SSVP-SLT may be attributable not only to the model architecture, but also to other significant factors that vary between the two approaches. One key difference is the amount of data used during pre-training. Our method uses a sample of only 1.2% of the YouTube-ASL frames, while SSVP-SLT uses 50% of the YouTube-ASL frames (every video at a stride of 2). Additionally, due to computational constraints, we pre-train our model for only 5 epochs, whereas SSVP-SLT pre-trains for 800 epochs. As another comparison, at epoch 100, SSVP-SLT achieves a BLEU score of 12.5 (see Table 1 SSVP-SLT $^{\text{YT+H2S}(50)}_{100}$ and Figure 2). In addition, we note that SSVP-SLT is pre-trained on both YouTube-ASL and How2Sign—the target domain dataset—while our model is pre-trained only on YouTube-ASL. Fig. 2 shows the performance-resources tradeoff for multiple models, showing that our approach (top left, in blue) surpasses SSVP-SLT by 1.7 BLEU points while training for 20 times fewer pre-training epochs and without utilizing the target domain dataset during pre-training.

These observations suggest that our method, despite using significantly less pre-training data and fewer pre-training epochs, can achieve competitive performance compared to the state of the art. Further investigation into the impact of pre-training data size and the number of pre-training epochs on the final performance could provide valuable insights into the efficiency and scalability of sign language translation models.

We perform ablation studies to investigate three key components of the model design, as shown in Appendix A. First, pre-training DINOv2 specifically on hand and face crops proves beneficial, improving performance. Second, the multi-stream approach demonstrates significant advantages over

using just the original frames, significantly boosting the scores. Finally, adding raw frames as a fifth stream alongside the existing features (perhaps suprisingly) degrades performance. See Appendix A for more details.

5 Conclusion

SignMusketeers is a data- and compute-efficient method for sign language translation at scale. It uses a multi-stream encoding scheme that focuses on important parts of a signing video (facial expressions, hands, and body pose) and requires only individual frames during self-supervised pre-training, in contrast to prior work on pre-training that requires long duration videos. SignMusketeers achieves competitive performance with roughly $40\times$ less data and $50\times$ less computation than prior work.

Limitations This research has several key limitations. The most significant is that despite matching or exceeding previous studies' results, the overall performance remains inadequate. Therefore, machine translation cannot yet reliably replace human sign language interpreters across diverse contexts. Another limitation stems from the much smaller training datasets compared to those typically used for developing self-supervised models in the speech and text domains. Additionally, since this study focuses solely on American Sign Language, we cannot determine whether our approach would work effectively for other sign languages, even though different sign languages share common features and communication methods. Also, in this work we focus only on the sign language translation task from video to text. It will be important to adapt this approach to the opposite direction (text to video) and other sign language processing tasks such as isolated sign language recognition, continuous sign language recognition, and fingerspelling detection. To overcome these challenges, future research should focus on building larger datasets and incorporating multiple sign languages into the training process.

Acknowledgment We are grateful to Anand Bhattad, Ankita Pasad, Ju-Chieh Chou, and Chung-Ming Chien for their valuable suggestions throughout this project.

References

Samuel Albanie, Gül Varol, Liliane Momeni, Hannah Bull, Triantafyllos Afouras, Himel Chowdhury, Neil Fox, Bencie Woll, Rob Cooper, Andrew McParland, et al. 2021. BBC-Oxford British Sign Language dataset. *arXiv preprint arXiv:2111.03635*.

Matyáš Boháček and Marek Hrúz. 2022. Sign pose-based transformer for word-level sign language recognition. In *Proc. Winter Conference on Applications of Computer Vision (WACV)*.

Danielle Bragg, Naomi Caselli, John W Gallagher, Miriam Goldberg, Courtney J Oka, and William Thies. 2021. ASL sea battle: gamifying sign language data collection. In *Proc. CHI*.

Danielle Bragg, Oscar Koller, Mary Bellard, Larwan Berke, Patrick Boudreault, Annelies Braffort, Naomi Caselli, Matt Huenerfauth, Hernisa Kacorri, Tessa Verhoef, et al. 2019. Sign language recognition, generation, and translation: An interdisciplinary perspective. In *Proc. SIGACCESS*.

Diane Brentari. 1998. *A Prosodic Model of Sign Language Phonology*. MIT Press.

Diane Brentari. 2019. *Sign Language Phonology*. Cambridge University Press.

Zhixi Cai, Shreya Ghosh, Kalin Stefanov, Abhinav Dhall, Jianfei Cai, Hamid Rezatofighi, Reza Haffari, and Munawar Hayat. 2023. Marlin: Masked autoencoder for facial video representation learning. In *Proc. CVPR*.

Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. 2018. Neural sign language translation. In *Proc. CVPR*.

Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. 2020. Multi-channel transformers for multi-articulatory sign language translation. In *ECCV Workshops*.

Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *Proc. CVPR*.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 2020. Unsupervised learning of visual features by contrasting cluster assignments. *Proc. NeurIPS*.

MMPose Contributors. 2020. Openmmlab pose estimation toolbox and benchmark. https://github.com/open-mmlab/mmpose.

Emely Pujólli da Silva, Paula Dornhofer Paro Costa, Kate Mamhy Oliveira Kumada, José Mario De Martino, and Gabriela Araújo Florentino. 2020. Recognition of affective and grammatical facial expressions: A study for brazilian sign language. In *ECCV Workshops*.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. 2024. Vision transformers need registers. In *The Twelfth International Conference on Learning Representations*.

Mathieu De Coster, Ellen Rushe, Ruth Holmes, Anthony Ventresque, and Joni Dambre. 2023. Towards the extraction of robust sign embeddings for low resource sign language recognition. *arXiv preprint arXiv:2306.17558*.

Aashaka Desai, Maartje De Meulder, Julie A Hochgesang, Annemarie Kocab, and Alex X Lu. 2024. Systemic biases in sign language ai research: A deaf-led call to reevaluate research agendas. *arXiv preprint arXiv:2403.02563*.

Amanda Duarte, Shruti Palaskar, Lucas Ventura, Deepti Ghadiyaram, Kenneth DeHaan, Florian Metze, Jordi Torres, and Xavier Giro-i Nieto. 2021. How2Sign: A large-scale multimodal dataset for continuous American Sign Language. In *Proc. CVPR*.

Alexandre Dumas. 1894. *The Works of Alexandre Dumas: Three Musketeers*. Estes and Lauriat.

Neil Fox, Bencie Woll, and Kearsy Cormier. 2023. Best practices for sign language technology research. *Universal Access in the Information Society*.

Zheng Gao and Ioannis Patras. 2024. Self-supervised facial representation learning with facial region awareness. *arXiv preprint arXiv:2403.02138*.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. 2020. Bootstrap your own latent: A new approach to self-supervised learning. *Proc. NeurIPS*.

Shester Gueuwou, Sophie Siake, Colin Leong, and Mathias Müller. 2023. JWSign: A highly multilingual corpus of bible translations for more diversity in sign language processing. In *Findings of the Association for Computational Linguistics: EMNLP*.

Ryo Hachiuma, Fumiaki Sato, and Taiki Sekii. 2023. Unified keypoint-based action recognition framework via structured keypoint pooling. In *Proc. CVPR*.

Eun-Jung Holden, Gareth Lee, and Robyn Owens. 2005. Australian sign language recognition. *Machine Vision and Applications*.

Oscar Koller, Hermann Ney, and Richard Bowden. 2016. Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled. In *Proc. CVPR*.

Anna Kuznetsova and Vadim Kimmelman. 2024. Testing MediaPipe Holistic for linguistic analysis of nonmanual markers in sign languages. *arXiv preprint arXiv:2403.10367*.

Scott K Liddell and Robert E Johnson. 1989. American Sign Language: The phonological base. *Sign Language Studies*.

Kezhou Lin, Xiaohan Wang, Linchao Zhu, Ke Sun, Bang Zhang, and Yi Yang. 2023. Gloss-free end-to-end sign language translation. In *Proc. ACL*.

Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. 2019. Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*.

Amit Moryossef, Ioannis Tsochantaridis, Joe Dinn, Necati Cihan Camgoz, Richard Bowden, Tao Jiang, Annette Rios, Mathias Müller, and Sarah Ebling. 2021. Evaluating the immediate applicability of pose estimation for sign language recognition. In *Proc. CVPR*.

Mathias Müller, Malihe Alikhani, Eleftherios Avramidis, Richard Bowden, Annelies Braffort, Necati Cihan Camgoz, Sarah Ebling, Cristina España-Bonet, Anne Göhring, Roman Grundkiewicz, et al. 2023. Findings of the second wmt shared task on sign language translation (wmt-slt23). In *Proc. Conference on Machine Translation (WMT)*.

Mathias Müller, Sarah Ebling, Eleftherios Avramidis, Alessia Battisti, Michèle Berger, Richard Bowden, Annelies Braffort, Necati Cihan Camgoz, Cristina España-Bonet, Roman Grundkiewicz, et al. 2022. Findings of the first wmt shared task on sign language translation (wmt-slt22). In *Proc. Conference on Machine Translation (WMT)*.

Seong Joon Oh, Rodrigo Benenson, Mario Fritz, and Bernt Schiele. 2016. Faceless person recognition: Privacy implications in social media. In *Proc. ECCV*.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. 2023. DINOv2: Learning robust visual features without supervision. *Trans. Machine Learning Research*.

Roland Pfau, Josep Quer, et al. 2010. *Nonmanuals: Their Grammatical and Prosodic Roles*. Cambridge University Press.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proc. Conference on Machine Translation (WMT)*.

Junfu Pu, Wengang Zhou, and Houqiang Li. 2016. Sign language recognition with multi-modal features. In *Advances in Multimedia Information Processing–PCM 2016: 17th Pacific-Rim Conference on Multimedia*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark,

et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*.

Benedikt Roth, Valentin Koch, Sophia J Wagner, Julia A Schnabel, Carsten Marr, and Tingying Peng. 2024. Low-resource finetuning of foundation models beats state-of-the-art in histopathology.

Phillip Rust, Bowen Shi, Skyler Wang, Necati Cihan Camgoz, and Jean Maillard. 2024. Towards privacy-aware sign language translation at scale. In *Proc. ACL*.

Chaitanya Ryali, Yuan-Ting Hu, Daniel Bolya, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy Hoffman, et al. 2023. Hiera: A hierarchical vision transformer without the bells-and-whistles. In *Proc. ICML*.

Wendy Sandler and Diane Carolyn Lillo-Martin. 2006. *Sign Language and Linguistic Universals*. Cambridge University Press.

Marcelo Sandoval-Castaneda, Yanhong Li, Diane Brentari, Karen Livescu, and Gregory Shakhnarovich. 2023. Self-supervised video transformers for isolated sign language recognition. *arXiv preprint arXiv:2309.02450*.

Panneer Selvam Santhalingam, Parth Pathak, Huzefa Rangwala, Jana Košecká, et al. 2020. FineHand: Learning hand shapes for American Sign Language recognition. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG)*.

Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. 2020. Adversarial training for multi-channel sign language production. *arXiv preprint arXiv:2008.12405*.

Bowen Shi, Diane Brentari, Gregory Shakhnarovich, and Karen Livescu. 2022. Open-domain sign language translation learned from online video. In *Proc. EMNLP*.

Rachel Sutton-Spence and Bencie Woll. 1999. *The Linguistics of British Sign Language: An Introduction*. Cambridge University Press.

Laia Tarrés, Gerard I Gállego, Amanda Duarte, Jordi Torres, and Xavier Giró-i Nieto. 2023. Sign language translation from instructional videos. In *Proc. CVPR*.

Sandrine Tornay, Necati Cihan Camgoz, Richard Bowden, and Mathew Magimai Doss. 2020. A phonology-based approach for isolated sign production assessment in sign language. In *Companion Publication of the 2020 International Conference on Multimodal Interaction*.

Dave Uthus, Garrett Tanzer, and Manfred Georg. 2023. YouTube-ASL: A large-scale, open-domain American Sign Language-English parallel corpus. In *Proc. NeurIPS*.

Gul Varol, Liliane Momeni, Samuel Albanie, Triantafyllos Afouras, and Andrew Zisserman. 2021. Read and attend: Temporal localisation in sign language videos. In *Proc. CVPR*.

Christian Vogler and Dimitris Metaxas. 2001. A framework for recognizing the simultaneous aspects of American Sign Language. *Computer Vision and Image Understanding*.

Ryan Wong, Necati Cihan Camgoz, and Richard Bowden. 2024. Sign2GPT: Leveraging large language models for gloss-free sign language translation. In *Proc. ICLR*.

Hong Yan, Yang Liu, Yushen Wei, Zhen Li, Guanbin Li, and Liang Lin. 2023. Skeletonmae: Graph-based masked autoencoder for skeleton sequence pre-training. In *Proc. CVPR*.

Kayo Yin, Amit Moryossef, Julie Hochgesang, Yoav Goldberg, and Malihe Alikhani. 2021. Including signed languages in natural language processing. In *Proc. ACL-IJCNLP*.

Zahoor Zafrulla, Helene Brashear, Pei Yin, Peter Presti, Thad Starner, and Harley Hamilton. 2010. American Sign Language phrase verification in an educational game for deaf children. In *Proc. ICPR*.

Hao Zhou, Wengang Zhou, Weizhen Qi, Junfu Pu, and Houqiang Li. 2021a. Improving sign language translation with monolingual data by sign back-translation. In *Proc. CVPR*.

Hao Zhou, Wengang Zhou, Yun Zhou, and Houqiang Li. 2021b. Spatial-temporal multi-cue network for sign language recognition and translation. *IEEE Trans. Multimedia*, 24:768–779.

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. 2022. iBOT: Image BERT pre-training with online tokenizer. *Proc. ICLR*.

Christian Zimmermann, Max Argus, and Thomas Brox. 2021. Contrastive representation learning for hand shape estimation. In *DAGM German Conference on Pattern Recognition*.

A Ablations

We conduct ablation studies to assess the effectiveness of pre-training DINOv2 on hand and face crops, the benefits of the multi-stream approach, and the potential of incorporating raw frames as an additional input stream.

Do we benefit from pre-training face/hand specific DINOv2 feature extractors? As shown in

	BLEU-1	BLEU-2	BLEU-3	BLEU
Uncrop + orig.	23.5	12.4	7.6	4.9
Crop + orig.	36.5	23.1	15.9	11.3
Crop + pre-tr. + global frame	36.7	23.8	16.7	12.2
Crop + pre-tr. (Ours)	**41.5**	**27.2**	**19.3**	**14.3**

Table 2: Ablation studies of our design choices. All models are pre-trained on YT in stage1 and fine-tuned with YT → H2S schedule for stage2 supervised training.

Table 2, using the original DINOv2 features without further pre-training on hand and face crops achieves a BLEU score of 11.3. When we pre-train DINOv2 on hand and face crops and utilize the learned features, the BLEU score improves to 14.3, a substantial increase of 3.0 points. This indicates that while DINOv2 features are indeed robust, continued pre-training on domain-specific data (i.e., hand and face crops) enhances the model's ability to capture relevant information for sign language translation.

Is the multi-stream approach beneficial compared to just using the original frames? To evaluate the effectiveness of the multi-stream approach, we compare the performance of using only the original frames (uncropped) with that of the multi-stream model. As shown in Table 2, the model trained on uncropped frames achieves a BLEU score of 4.9, while the multi-stream model (using the same original DINOv2 checkpoint model) obtains a significantly higher BLEU score of 11.4. This substantial improvement demonstrates the benefit of the multi-stream approach (given the same model as feature extractor) in capturing fine-grained details and relevant information from different body parts for sign language translation.

Does a 5^{th} stream containing the raw frames help the model? We investigate the potential of incorporating an additional stream containing the raw frames alongside the upper body pose features and cropped hand and face features. As shown in Table 2, adding the global frame as an extra stream to the multi-stream model results in a BLEU score of 12.2, which is lower than the 14.3 BLEU score achieved by the model without the global frame. This suggests that the global frame may add noise rather than improve the model.

Pre-training Frames	H2S	YT → H2S
0.1M	1.5	11.6
0.4M	1.5	-
0.8M	2.1	-
1.0M	2.4	14.3

Table 3: Analysis of model performance (BLEU scores) with varying amounts of pre-training data.

Streams Used	BLEU
Face only	0.6
Hands only	0.2
Upper body only	0.8
Hands + Upper Body	2.1
All streams (full model)	2.4

Table 4: Impact of different information streams on translation performance (BLEU scores) on How2Sign.

B Additional Analysis and Discussion

B.1 Scaling Analysis

To better understand the scaling properties of our approach, we conduct additional experiments varying the amount of pre-training data. Table 3 shows how model performance changes as we increase the number of frames used during pre-training from 0.1M to 1.0M frames on both How2Sign and YouTube-ASL datasets.

The results demonstrate consistent improvement in translation quality as we increase the amount of pre-training data, with gains of 0.9 and 2.7 BLEU points on How2Sign and YouTube-ASL respectively when scaling from 0.1M to 1.0M frames. This suggests that our method can effectively leverage additional pre-training data, though the gains may be bounded by the model capacity of the ViT-Small architecture used in our experiments.

B.2 Multi-Stream Contribution Analysis

To validate the importance of each information stream in our multi-stream architecture, we conduct ablation experiments using different combinations of streams. Table 4 presents the results on the How2Sign dataset.

These results empirically demonstrate that each stream contributes meaningful information to the translation task. While the upper body stream provides the strongest individual signal (0.8 BLEU), the combination of streams yields substantially better performance (2.4 BLEU), supporting our hy-

Backbone	BLEU
MAE (whole frame pre-trained)	0.3
DINOv2 (whole frame pre-trained)	1.0
DINOv2 (original)	1.1

Table 5: Comparison of visual backbones on How2Sign using a whole-frame approach.

Pre-training data	Frames	BLEU
10%	0.1M	1.5
100%	1.0M	2.4

Table 6: Impact of pre-training data volume on translation performance on How2Sign.

pothesis that different visual cues complement each other. The relatively low performance of individual streams also highlights the importance of our multi-stream approach in capturing the full range of linguistic information present in sign language videos.

B.3 Choice of Visual Backbone

We investigate different visual backbones for feature extraction. While prior work has established DINOv2's superiority over ResNet architectures for sign language tasks, we conducted additional experiments comparing DINOv2 with MAE on the How2Sign dataset.

The results, shown in Table 5, demonstrate DINOv2's effectiveness as a feature extractor for sign language translation. Notably, the performance difference between pre-trained and original DINOv2 is minimal when using whole frames, which motivated our exploration of the multi-stream approach.

B.4 Data Efficiency and Scaling

To understand the relationship between pre-training data volume and model performance, we evaluate our model at different stages of pre-training, as shown in Table 6.

The improvement from 1.5 to 2.4 BLEU with increased pre-training data suggests that our approach can effectively utilize additional data. However, we note that these gains are achieved while using only 1.2% of the available YouTube-ASL frames for pre-training, highlighting the efficiency of our method.

C Hyperparameter Settings

The hyperparameter values used in our experiments are shown in Table 7 and Table 8.

D MediaPipe Pose Landmark Indices

The MediaPipe pose landmark indices are defined in Figure 4.

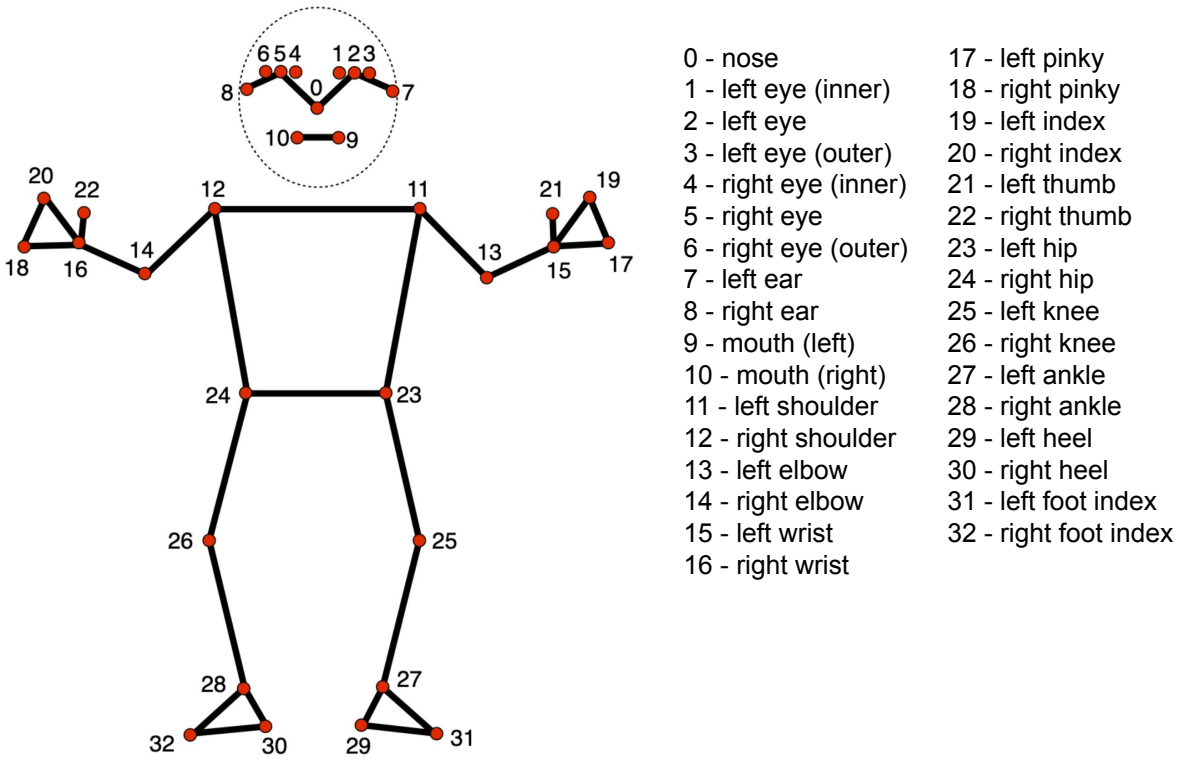

Figure 4: MediaPipe pose landmark indices. Source: Google AI MediaPipe Documentation.

PARAMETER	VALUE
dino:	
loss_weight	1.0
head_nlayers	3
head_hidden_dim	2048
koleo_loss_weight	0.1
ibot:	
loss_weight	1.0
mask_sample_probability	0.5
head_bottleneck_dim	256
head_nlayers	3
head_hidden_dim	2048
train:	
batch_size_per_gpu	128
saveckp_freq	20
seed	0
num_workers	10
OFFICIAL_EPOCH_LENGTH	100
cache_dataset	true
centering	sinkhorn_knopp
student:	
arch	vit_small
patch_size	14
drop_path_rate	0.4
layerscale	1.0e-05
drop_path_uniform	false
pretrained_weights	null
ffn_layer	mlp
block_chunks	0
qkv_bias	true
proj_bias	true
ffn_bias	true
num_register_tokens	4
interpolate_antialias	false
interpolate_offset	0.1
teacher:	
momentum_teacher	0.994
final_momentum_teacher	1
warmup_teacher_temp	0.04
teacher_temp	0.07
warmup_teacher_temp_epochs	30
epochs	100
weight_decay	0.04
weight_decay_end	0.2
base_lr	0.0002
lr	0.0
warmup_epochs	10
min_lr	1.0e-06
clip_grad	3.0
freeze_last_layer_epochs	1
scaling_rule	sqrt_wrt_1024
patch_embed_lr_mult	0.2
layerwise_decay	0.9
adamw_beta1	0.9
adamw_beta2	0.999
crops:	
global_crops_scale	(0.5,1.0)
local_crops_number	8
local_crops_scale	(0.25,0.5)
global_crops_size	224
local_crops_size	98

Table 7: SignMusketeers pre-training settings.

PARAMETER	VALUE
model_name	t5-1.1
target_language	English
expand_time	false
freeze_model	false
freeze_adapter	false
force_target_language	false
batch_size_per_gpu	16
learning_rate	0.001
train_steps	5000
fp16	false
test_every	500
save_every	500
log_every	500
num_workers	4
optimizer	adamw_torch
schedule	cosine
weight_decay	1e-1
label_smoothing_factor	0.2
generation_max_length	128
generation_num_beams	5
grad_clipping	1.0

Table 8: Fine-tuning settings for How2Sign (H2S).