Final Report
Sign Language Translation with a Pure Transformer Architecture

Daniela Winter
Computer Vision Master Project - Dr. Ehsan Yaghoubi
Computer Science Department
University of Hamburg, Germany
daniela.winter@studium.uni-hamburg.de
September 11, 2023

***Abstract*—WHO estimates that by 2050 1 in 10 people, or 700 million people will suffer from disabling hearing loss [1]. Tools that enable hard-of-hearing and deaf people to communicate across Sign Languages (SL) and with hearing people can be an important help in everyday challenges. Sign Language Translation (SLT) research tries to tackle this task with the help of different Machine Learning approaches. One of the main challenges in current SLT research includes achieving meaningful encodings of sign videos. The current works done in SLT with Transformers use a CNN as a backbone for feature extraction. Previous research, however, has shown that using a Pure Transformer architecture, that does not rely on the embeddings from a CNN, generally results in better performances. Therefore, in this work, we present several steps of an approach to a Pure Transformer SLT architecture without the need for a CNN. For this, we adapt a standard Transformer architecture for SLT and combine it with the Video Swin Transformer. While not achieving satisfactory performances yet, several obstacles have been tackled in this work and we present the gained insights and follow-up ideas to finalize the successful realization of the Pure SLT Transformer. The code for the different architectures has been made available at https://github.com/daniewin/cv-project.**

I. Introduction

A. Sign Language Translation (SLT) and Glosses

Sign languages (SLs) are visual languages that are commonly used as a form of communication for deaf and hard-of-hearing people. They are official languages with defined signs, grammar, and dialects. Sign language is expressed with manual and non-manual components. Manual components include hand poses, their orientation, and the location of the hands in space; while upper-body movements, lip movement, and other facial expressions constitute the non-manual components [Ananthanarayana et al., 2021].

Since SLs are visual languages, they include specifics that are fundamentally different from spoken languages. Those specifics include, among others, the location of the sign in the signing space, if a sign is repeated, mouthings, facial expressions, and so-called classifiers [Crasborn, 2015]. The location of the sign and classifier signs are used as variables that can store information that can be referred to later. All this information is crucial to understanding the meaning of the signed phrases.

This variety of cues and ways to transfer information makes Sign Language Translation (SLT), which is the translation of sign utterances into spoken language text, an extremely difficult task. Being able to extract the necessary semantic information from the visual data is the most crucial step of the whole SLT process. Once the model is able to create useful embeddings for the visual SL data, the rest of the task of shaping this semantic information into spoken language text output can be realized quite well by Transformer architectures.

In order to guide the learning of this essential step of extracting the needed information from visual SL data, many architectures first train a separate model for recognizing glosses from SL videos. Glosses are a way of representing SLs in written form. This form of transcription is a closer representation of SLs than spoken language text because the latter can not capture all the dimensions of the complexity of signs. Glosses on the other hand capture not only the meaning of the hand pose itself but also other SL specifics that carry additional information. Without this additional information, glosses can not function well as a basis of interpretation for spoken languages. For this reason, the gloss transcription uses so-called diacritics which are

[1] https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss

additional symbols, letters, or numbers as extra notes next to the translated word as can be seen in Figure 1. The first section of Figure 1 includes general rules of the gloss transcription. The reference spoken language word needs to be capitalized and if a word is spelled or a gloss is made up of two signs their transcriptions are split by a ”+”. ”IX” stands for ”index” and refers to locations in the signing space that can be assigned a meaning and referred to later. Repetitions and different pronunciations can also be transcribed. The second section shows how the meaning of a gloss can be changed by negation, used as a location, or used as a classifier for future reference. The last section shows additional information that was included by the usage of other articulators (mouth, face) or for example by additional localization. This enables the gloss transcription to carry more SL-specific information that is needed to interpret the meaning in context.

For SLT, the use of glosses is a great help. The mapping of video frames to spoken language text is a non-monotonic many-to-few mapping since the word order of SL and spoken language can be completely different and multiple video frames correspond to one word meaning. It is possible that several words are needed to describe the meaning of one sign or that one word corresponds to multiple signs. These characteristics make mapping a very hard task. While the mapping of video frames to glosses is still a many-to-one, one sign can always be transcribed with one gloss. Furthermore, the order of the glosses in the transcription is parallel to the order of the signs in a video. Therefore, training a network on recognizing glosses in sign videos can be used as a simplified, intermediate task in the complete pipeline of SLT [Camgöz et al., 2020b]. This process of only recognizing glosses from SL videos instead of translating the videos into spoken language text is called Sign Language Recognition (SLR).

While glosses provide many advantages for SLT, there is a downside to the approach of incorporating gloss representations in the training process. Creating big data sets with gloss annotations is very expensive since the annotations have to be created manually by SL professionals. SL datasets without gloss annotations and just spoken language ground truths on the other hand are very easy to acquire since a signer can translate spoken language to SL in real time. While it is desirable to skip the step of gloss prediction for simplicity, the performance of such pipelines is often worse. This is the case because the task of SLT is too complex to learn for the proposed models, so some form of intermediate supervision helps guide them in the right direction [Camgöz et al., 2020b].

Scheme	Example
gloss in capital letters	WIE-IMMER
finger spelling split by +	A+G+N+E+S
compound glosses split by +	V+LAND
numbers in written form	SIEBEN instead of 7
pointing gestures	IX
extended repetitions	SONNE++
pronunciation variants	TAG#1 TAG#2
classifier signs	cl-KOMMEN
lefthand only signs	lh-SONNE
signs negated by headshake	neg-WIND
signs negated by the alpha rule	negalp-MUESSEN
localization	loc-REGEN
additional mouthing	GLOSS-(mb:hill)
additional facial expression	GLOSS-(mk:strong)
additional localization	GLOSS-(loc:alps)
additional object of sign	GLOSS-(obj:cloud)

Fig. 1: This is an example of the gloss annotations used in the RWTH-PHOENIX-Weather corpus [Forster et al., 2012].

B. Vision Transformers

SL-specific feature extraction is a hard task with or without the use of glosses. Recent research has shown that Pure (Video) Transformer architectures, that do not rely on features extracted by CNNs, can reach a better performance when enough data is available [Arnab et al., 2021] [Liu et al., 2022]. Therefore, it is reasonable to assume that they are better at focusing on and extracting specific visual information from SL videos as well.

Visual data has specific properties that are not necessarily found in text data, such as spatial locality and translation invariance. The encoding of the data needs to preserve those properties to allow meaningful interpretation of the data. To induce the visual bias of spatial locality, Vision Transformer models such as the ViViT (Video Vision Transformer) model [Arnab et al., 2021], similar to 3D CNNs, splits the data into 3D blocks. Since pixels of the video that are close to each other in space or time correlate more with each other than farther pixels, combining them preserves the spatial locality property. Each of those 3D blocks performs as a token for the input of the Transformer encoder. However, these blocks are of fixed size, which ignores the differing sizes that visual elements can have. Moreover, translational invariance is not included in this model.

To deal with this shortcoming, the Video Swin Transformer uses hierarchical feature maps by merging 3D blocks in later layers [Liu et al., 2022]. In addition to that, the Swin Transformer includes connections across blocks by alternating between two overlapping blocks in consecutive layers. Self-attention is then applied within the blocks. Using this strategy, the Video Swin Transformer captures the biases of visual data better, resulting in an improved performance on image classification tasks.

Generally, both Pure Video Transformers show that their performance exceeds the levels of CNN-based models if enough data is available. Both Pure Video Transformers and CNN-based models directly apply the attention mechanism of transformers directly to the data instead of the embeddings of the frames that were obtained by CNNs. This different way of encoding the data seems to be beneficial for standard Computer Vision tasks such as image classification, semantic segmentation [Liu et al., 2022], and video captioning. Pure Video Transformers are already used successfully in the video captioning domain. Due to its similarity to SLT, this shows promise for the feasibility of this approach in this domain as well. Hence, for this task, Pure Video Transformers were successfully applied using a Video Swin Transformer as an encoder for the input data and a BERT model as a decoder to produce an output sentence [Lin et al., 2022].

C. Motivation

Camgöz et al. have proposed the idea of learning SLT as an end-to-end task without the intermediate gloss prediction step. For above mentioned reasons, it is desirable to find SLT solutions without the need for glosses and this approach would be a step in that direction. In this work, we analyzed if this model could actually solve the task of SLT. We neither use an SLT-specific CNN that was pre-trained on gloss prediction nor gloss predictions as an intermediate step in the SLT pipeline. We adopt the SLT Transformer architecture of Camgöz et al. which is described in detail in Section III and combine it with a Vision Transformer as a feature extractor. The decoder part of the SLT Transformer is kept unchanged while the encoder is partly replaced by the Swin Transformer. This model is a Pure Transformer architecture and we attempted to tackle the complete end-to-end video-to-text SLT as described in their work. However, this joint end-to-end SLT task proved too ambitious for current Machine Learning techniques.

Insights into why certain ideas turn out to not be successful are just as valuable for knowledge gathering and scientific progress as successful outcomes. Therefore, we state the possible reasons why our method of approaching the realization of Pure Transformer SLT architecture did not result in utilizable performances and share insights that were gained through this work.

This paper is structured as follows: In Section II we present the development of SLT model research and the introduction of the standard SLT Transformer that our approach is based on. In Section III we give a detailed overview of this model including its architecture and results from rerunning the implementation on a standard dataset, which is also introduced in this section. In Section IV we present the Pure SLT Transformer architecture and its variants. Moreover, we show the results from training these models qualitatively and quantitatively and state a hypothesis for why these results are unsatisfactory. In Section V we conduct an ablation study to test our hypothesis and compare the performance of holistically and SLT-specific pre-trained backbones for the SLT Transformer. The holistically pre-trained backbone is pre-trained in general image classification, not specific to SL. In addition to that, we compare the feature embeddings by their average Euclidian distances to show why the SLT-specific embeddings might lead to more successful performances. In Section VI we conclude the findings of this paper and give an outlook on future work.

II. Related Work

Machine learning techniques help create applications to simplify communication between languages. Current Neural Machine Translation (NMT) systems using Transformer networks make striking progress. NMT for SLs, on the other hand, still lags behind. Video–to–text Sign Language Translation (SLT) achieves way lower BLEU scores than text-to-text translations of spoken languages [Camgöz et al., 2020b] [Zhu et al., 2020]. While the idea of translating between languages is the same, the task of SLT differs a lot from NMT for spoken languages. As SLs are visual languages, their translation poses challenges from both the NLP and CV domains. NMT of SLs has to deal with visual and textual data and with visuospatial linguistic principles that are unique to SL [Crasborn, 2015].

In video-to-text translation, the encoder part of the SLT Transformer architecture mainly deals with the visual aspects of the language. The effective encoding of visuospatial principles is crucial for the preservation of the passed information - and therefore plays a big role in trying to reach current levels of spoken language NMT applications. To guide the extraction of the general meaning and specific features from SL videos, most approaches divide SLT into two different steps. First, a model is trained to recognize different signs and classify them in glosses (SLR). Then another model uses the proposed gloss sequence and translates it into a spoken language sentence (SLT). The intermediate step over gloss representations helps to encode the visual data in a meaningful way and some works state that the task could be too complex otherwise [Camgöz et al., 2020b]. However, the creation of gloss datasets is extremely expensive and it has been proposed that glosses act as an information bottleneck in the SLT pipeline.

Camgöz et al. proposed an SLT architecture that relies less on the recognition of glosses, while still using them as a guidance for the encoding of the videos. In the following section, this model will be explained in detail, and it will be referred to as the ”baseline SLT Transformer model”. Their model directly predicts spoken language text from SL video embeddings without the direct usage of gloss representations as input and therefore seems to learn the tasks of SLR and SLT jointly. Instead of the gloss predictions themselves, only the hidden representation of the encoder is used as input for the decoder (see Figure 2).

While their approach seems to show very promising results, their work does raise some concerns. When stating their contributions they do not take into account that they use a CNN backbone for feature extraction that was pre-trained on the SLR task with glosses on exactly the same dataset that they train the complete model on. This makes their approach just as dependent on gloss annotations as other works that use separate SLR and SLT. Moreover, the use of a pre-trained CNN backbone that is not incorporated into the learning process contradicts the goal of having an end-to-end SLT model since the biggest part of the SLR learning process has been outsourced.

Keeping this in mind, their work is still a very useful contribution to SLT research. They made their implementation public and it is well structured which makes it easier for other researchers to build on their work. Their model has since been used by other researchers trying to improve the performance with changes that mostly keep the same architecture [Camgöz et al., 2020a] [Voskou et al., 2021]. In this work, we attempted to change the architecture in such manner as to discern if their original claims of an end-to-end joint SLR and SLT model could be realized.

III. Baseline: SLT Transformer Model

A. Architecture

1) SLT-Specific Pre-trained CNN: As a first step, we reran the baseline SLT Transformer code that we are basing our new architecture on. An overview of the baseline architecture can be seen in Figure 2. In addition to this, unlike Figure 2 suggests, the network does not take the SL video frames as input but the already embedded features, and the original dataset is not provided. The authors state that the features are extracted by a CNN, and in a separate subsection, they mention that they compared different pre-trained CNNs for the feature extraction task. They used EfficientNets [Tan and Le, 2019] with different levels of complexity, pre-trained on ImageNet [Deng et al., 2009] and a different model from a previous work [Koller et al., 2019] that they adapted with Batch Normalization and a ReLu layer. As a side note, they mention that this model was pre-trained in a Sign Language Recognition task and don't provide any further details. Checking the original paper of that work shows that this model has learned to recognize glosses from SL videos from exactly the same dataset they are later using for their model. Using a network that already has learned to extract the crucial information out of the frames from the same dataset makes the remaining task for the actual model much easier. Since the comparison of the networks pre-trained on ImageNet and the SLT-specific pre-trained model showed that the latter results in better performances, the authors continued with features extracted by that CNN model.

2) Encoder: The CNN features are then input into a linear spatial embedding layer following the approach of [Camgöz et al., 2018] to project the embeddings from a 1024- to a denser 512-dimensional space. Since Transformer models do not keep information about the position of the input tokens, they use positional encoding as proposed in the original Transformer paper [Vaswani et al., 2017]. After that, the features are forwarded to the encoder part (SLRT: SL recognition Transformer) of the architecture. The encoder consists of three Transformer layers. Each is realized as a combination of a self-attention layer, and a feed-forward layer, both of which include Normalization and residual connections to improve the training process. The resulting hidden representations are then forwarded to the decoder part (SLTT: SL Translation Transformer) of the architecture. At the same time, they are used to compute gloss predictions with a linear layer that creates one-hot encoded feature vectors for each gloss in the dataset vocabulary and a softmax layer. To guide the training of the encoder, Connectionist Temporal Classification (CTC) is used to predict gloss probabilities and compute a loss for the encoder. CTC marginalizes over all possible alignments of the gloss sequence G and sign video V to get the probability of the gloss sequence given the input video [Graves et al., 2006]. CTC accounts for the fact that multiple following frames could predict the same gloss and therefore, gloss sequences with multiple consecutive

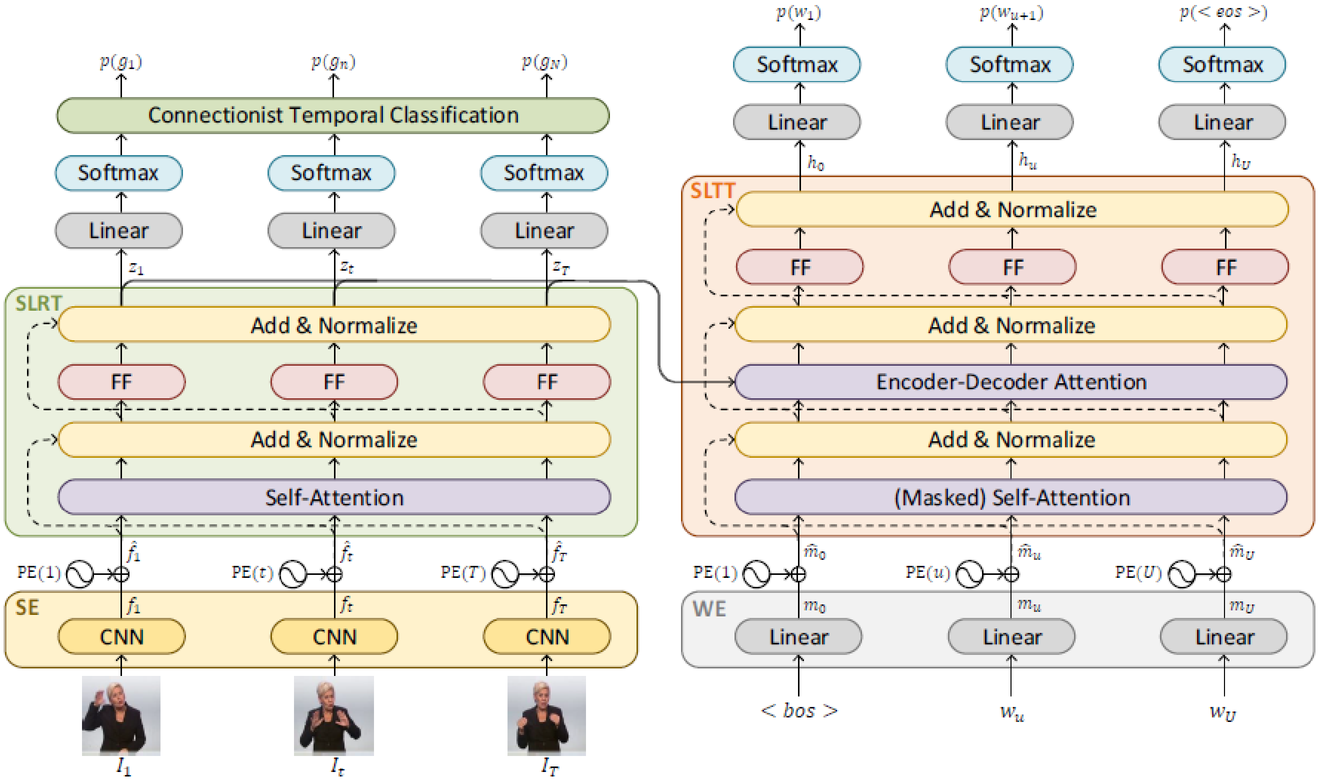

Fig. 2: This is an example of a Transformer Architecture for end-to-end SLT. Self-attention is used in the encoder as well as the decoder and (encoder-decoder) attention in the decoder [Camgöz et al., 2020b].

Legend:

I_t: input frame at timestep t;
z_t: SLRT output at timestep t;
h_u: u^{th} SLTT output
$< bos >$: beginning-of-sentence token;
$< eos >$: end-of-sentence token;
$m_u, \hat{m}_u$: u^{th} word embedding (position-encoded)
SE: Spatial-Embedding;
PE: Positional-Encoding;
SLRT: SLR Transformer;

$f_t, \hat{f}_t$: CNN feature extracted at timestep t (position-encoded);
$p(g_n), p(w_u)$: probability distribution of the n/u^{th} predicted gloss/spoken language word;
w_u: input/output spoken language words
WE: Word-Embedding;
FF: Feed-Forward
SLTT: SLT Transformer

equal glosses can be summarized with only one mentioning of that gloss. π is one possible gloss combination (including repetitions of glosses) and B are all possible combinations that end up in the summarized (without any repetitions) gloss of G:

$$p(G|V) = \sum_{\pi \epsilon B} p(\pi|V)$$

After computing the probabilities for all the gloss sequences G with the given model, the loss is then computed as how well the probability of the ground truth gloss was modeled. Optimally, the probability of the ground truth gloss sequence should be modeled as 1. The higher the probability, the lower the loss of the encoder network. The loss is computed as:

$$L = 1 - p(G^*|V)$$

3) Decoder: Similar to the spatial embedding of the input tokens for the encoder, the word tokens for the decoder network are embedded with a linear layer to project them from a one-hot encoding into a denser space. Following that, a positional encoding is applied following the method of [Vaswani et al., 2017]. The decoder then applies a self-attention layer with masking so that the decoder can not see the next input token which it is supposed to output. This layer includes a residual connection and a normalization step. After that, the encoder and decoder representations are combined and the other decoder transformer layers follow. These layers are built the same way as the encoder transformer layers. The transformer layers are followed by a linear layer and a softmax layer to predict the next words until the end of the sentence token is predicted. The loss for the decoder network is calculated using the cross-entropy loss for each predicted word in the sequence. Both, the encoder and decoder loss are then combined to get a total loss for the complete network.

B. Experiments

1) Dataset: The data in the PHOENIX-Weather 2014T dataset [Camgöz et al., 2018] consists of videos from the weather forecast of the Phoenix TV channel in German SL [Forster et al., 2012] and their spoken language and gloss annotations. Since the videos are all from weather forecasts, the topic and the SL vocabulary are quite restricted. The resolution of the videos is 210 × 260px with a frame rate of 25fps. Nine different people are used as signers and in total 2889 different words and 1078 different glosses were used. With the proposed splitting of the dataset, the subset for training contains 7096 videos, the validation subset 519, and the test subset 642. The big advantage of this dataset is, that it has a controlled environment which makes it possible to easily detect the signer in front of a constant, homogeneous, undetailed background (see Figure 3). The dataset has been the state-of-the-art continuous SLT dataset with both gloss and spoken language annotations.

Fig. 3: Samples from the PHOENIX-Weather 2014T dataset [Ananthanarayana et al., 2021]. The dataset shows the signers in front of an un-detailed background. The background is grey and the signers wear dark clothing to ensure contrast.

2) Implementation Details: The implementation of the baseline SLT Transformer by Camgöz et al. is based on the code of JoeyNMT [Kreutzer et al., 2019] and implemented mostly in PyTorch with only separate parts in TensorFlow. The encoder and decoder transformers each have 8 attention-heads per layer and 3 transformer layers. The hidden states have a dimension size of 512 and a dropout rate of 0.1 is used to reduce overfitting. The model is trained using the Adam optimizer [Kingma and Ba, 2014] and we used a batch size of 1 due to hardware restrictions. The learning rate was 0.001. To evaluate the model's text prediction performance, the BLEU score metric [Papineni et al., 2002] is used. The training process was run on an RTX 3090 GPU with a VRAM of 24 GB.

3) Results: When rerunning the training of their model with the available code and data, we reached an impressive result after only a few training epochs.

Text Reference	und nun die wettervorhersage für morgen samstag den zehnten april .
Text Hypothesis	und nun die wettervorhersage für morgen samstag den neunten juni .
Text Reference	vor allem im nordwesten ist es sehr windig .
Text Hypothesis	im nordwesten und nordwesten auch frischer wind .

TABLE I: The qualitative results showed grammatically and semantically almost correct output sentences already after only 5 training epochs.
English translations: First example: Text Reference: and now the weather forecast for tomorrow Saturday the tenth of April. Text Hypothesis: and now the weather forecast for tomorrow Saturday the ninth of June.
Second example: Text Reference: Especially in the northwest it is very windy. Text Hypothesis: in the northwest and the northwest also cool wind.

The qualitative results showed grammatically and semantically almost correct output sentences already after only 5 training epochs. Table I shows some representative examples of predicted versus ground truth sentences after 5 training epochs. The main concept of the sentence is contained in the prediction and the grammar is correct. Mostly the way of expressing the content differs slightly between prediction and ground-truth. It has to be noted that there can be SL dialect differences between the signers and annotators. This can lead to differences in the signs and their annotations for months and days of the week [2]. This

[2] https://www.deutschland.de/de/topic/leben/deutschland-sprache-deutsche-gebaerdensprache

	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Baseline Paper	46.61	33.73	26.19	21.32
Our rerun 14 epochs	43.24	29.76	22.50	18.10
Our rerun 59 epochs	44.21	30.80	23.20	18.50

TABLE II: BLEU Scores of the baseline SLT Transformer from the baseline paper and when rerunning the implementation.

could be the reason for the confusion of April and June in the first example.

The quantitative results show that the model did not need much training to reach comparable results with the baseline paper performance. After only 14 epochs it reaches a BLEU-4 score of 18.10 and 45 epochs later it only increased to 18.50 as can be seen in Table II. The training ended at this point as there was no increase in performance afterwards. These results are very impressive for SLT usually being such a complicated task and hint towards the fact that the biggest obstacle is already tackled by the pre-trained CNN Backbone. There is a slight difference between the performance of the baseline paper and our training performance but this could be due to the fact of different random initializations that converge towards different local optima.

IV. Our Approach

Camgöz et al. claimed that it is possible to train an end-to-end video-to-text SLT model by only using glosses for the supervision but not training it directly on gloss prediction. In our paper, we investigated if these claims hold. We designed an architecture that is not dependent on a CNN as the backbone for feature extraction that was pre-trained on gloss recognition. We also kept the general approach of the baseline model from Camgöz et al., our decoder part of the model does not take glosses as input - the ground truth glosses are only used for supervising the learning of the encoder. Both these facts make this approach take a step in the direction of less gloss dependency of SLT models. Moreover, our model takes the SLT videos directly as input. Therefore, it actually functions as an end-to-end SLT model.

To tackle the very hard task of extracting SL-specific information from videos, we used a Swin Transformer instead of a CNN. Vision Transformers have proven to perform better in a wide range of Computer Vision tasks. Since feature extraction is the most crucial part of the SLT pipeline, using a Transformer for this task could be a promising approach. For this reason, we propose a Pure Transformer architecture for SLT without the use of any CNN.

A. Architecture

To create a Pure Transformer architecture, we modified the architecture of the SLT from Camgöz et al. by replacing the CNN backbone with a Swin Transformer. The decoder part of their architecture and the loss computations have been kept the same. We experimented with several versions of this approach. The first version of our adaptation is explained here in detail with the following versions only including small variances. The Swin Transformers have been both trained from scratch or with weights that were pre-trained for an image classification task on ImageNet1k. However, for all different versions, no significant performance differences could be observed.

The first model version can be seen in Figure 4 and uses a 3D Swin Transformer as an encoder which takes whole videos as input and processes all frames together. This enables the 3D Swin Transformer to also find connections between frames. The outputs of the Swin Transformer are upsampled to the original frame number since the features of the 3D Swin Transformer are only half as many as the frame number. This is needed for the consistency of the model since the decoder expects one predicted feature vector per frame.

In the following, the three parts of the architecture, encoder, gloss prediction, and decoder are explained in detail.

1) Encoder: Since we follow the architecture of the Video Swin Transformer for our encoder network, the raw input video frames will be used as direct input for the encoder. The Video Swin Transformer [Liu et al., 2022] takes the input video of the size $T \times H \times W \times 3$ (T: number of frames) and splits it into 3D blocks of size $2 \times 4 \times 4 \times 3$ that are flattened. Each of them counts as an input token. The number of tokens hence depends on the size of the video. Each 3D token is a 96-dimensional vector. However, a linear embedding layer allows transferring the input to a variable dimension size C. The Video Swin Transformer consists of several stages, each stage representing the hierarchy of the processing scale. In each stage, the patch dimension is changed in the patch merging layer by two times spatial downsampling. Spatially neighboring patches are grouped by 2×2 and the concatenated features are then downsampled to half their dimension size. This results in a final dimension size of 8C after the last block of the encoder network (see

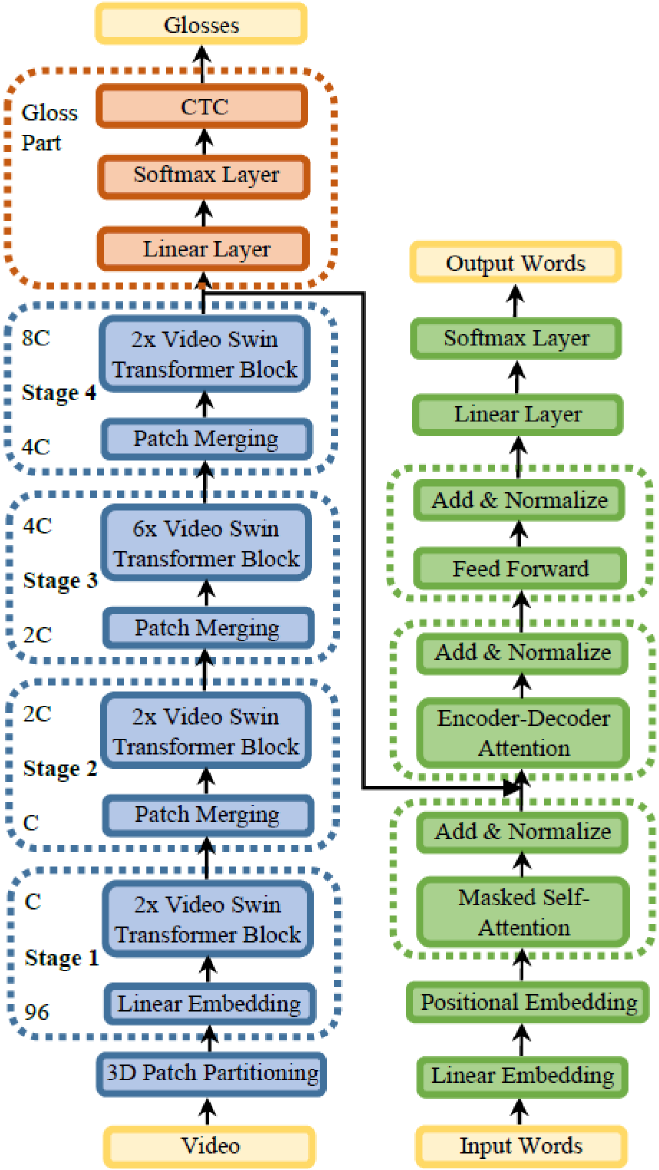

Fig. 4: This is our proposed model in its first version. We follow the architecture of the Video Swin Transformer [Liu et al., 2022] and the SLT Transformer [Camgöz et al., 2020b]. The blue blocks depict the Video Swin Transformer encoder, the red blocks the gloss computation, the green blocks the Transformer decoder, and the yellow blocks the inputs/outputs. The numbers and letters to the left in the blue blocks of the encoder refer to the input and output dimensions of the features in different stages.

Figure 4). In each stage Video Swin Transformer blocks are applied multiple times (see Figure 4) [Liu et al., 2022]. This module includes multi-head self-attention that incorporates the 3D shifted window approach of the Swin Transformer to enable cross-patch connections [Liu et al., 2021]. The outputs of the encoder network are spatiotemporal encodings that are forwarded to the decoder network.

2) Gloss Prediction: The above-mentioned encodings can optionally also be used to predict glosses. A linear layer followed by a softmax layer predicts possible words for the gloss transcription. Using those predictions, the CTC layer predicts alignments of the gloss transcription sequence to the parts of the video. The CTC layer tackles the alignment difficulty of having a long input sequence (video frames) and a relatively short output sequence (gloss transcriptions). It produces different possible gloss alignments with a probability assigned to them. The loss of the CTC layer is calculated as the sum of the possible gloss alignment probabilities [Graves et al., 2006]. Additionally, it can be used to guide the training of the encoder, not only with the loss from the output of the decoder. With the gloss supervision, the training of the encoder is focused on learning representations that are closer to glosses. This can help steer the training in the right direction. However, it should be mentioned that glosses lack important information that is contained in the sign videos [Camgöz et al., 2020b]. Therefore, we will analyze how the performance of the network will be without the usage of gloss supervision for training.

3) Decoder: For our decoder, we follow the implementation used for the SLT Transformer by Camgöz et al. [Camgöz et al., 2020b]. Before inputting them into the decoder network, the one-hot encoded input words are converted into a denser representation by the word embedding layer. Positional encoding is then added to keep track of the order of the words in the sentence. After that, we apply a self-attention layer that masks the inputs during training so that only previous inputs but not later ones are known by the network when trying to predict the next word. The resulting representation of the spoken language context and the sign video representations yielded from the encoder are then combined for the encoder-decoder attention layer. We then use the outputs from that attention layer to pass through a non-linear feed-forward layer. Each block of the decoder has residual connections and normalization applied after the layers. Finally, a linear layer and a softmax layer are used to predict the probability distributions of the output words over the vocabulary. This is done word by word until the end-of-sentence token is reached.

For the training, we compute a cross-entropy loss from the predicted words. The complete network can be trained with either a combination of gloss and word prediction loss or with only the word prediction loss.

The presented first version of our model has been adapted into two more versions. The second model version is very similar to the first version. Instead of a 3D Swin Transformer, it uses a 2D Swin Transformer that treats the frames of a video as a batch of images and then extracts features per frame. This approach is closer to the baseline model since Camgöz et al. extracted features per frame with a 2D CNN.

The third model version uses a 2D Swin Transformer in combination with the three transformer layers of the baseline SLT model. This approach is even closer to the baseline model in that it also first uses a CNN for feature extraction and then uses three transformer layers in the encoder. This approach could allow the encoder to adapt more specifically to the task. While the Swin Transformer is extracting the visual information from the video, the extra transformer layers could help to use this information to create more language-specific representations.

In all versions, following the process of the baseline model, the Swin Transformer is followed by a spatial embedding linear layer which projects the output to a denser 512-dimensional space. We experimented with weights that were pre-trained on image classification and weights initialized randomly.

B. Experiments

We used the same PHOENIX-Weather 2014T dataset for all experiments. However, due to hardware limitations, we had to decrease the video sizes to 100 frames and the training set size to 1500 training videos for the experiments on all models other than the baseline SLT Transformer. This results in a vocabulary size of 1765 different words and 715 different glosses. That is still more than half the vocabulary size of the original dataset.

1) Implementation Details: Since our models are harder to train, we increased the initial learning rate to 0.005. For the remaining implementation details see Section III-B2.

2) Results: Even after a high number of epochs, it can be seen that the predicted sentences have no more than random semantic overlap with the ground truth sentences as can be seen in Table III. The same exemplary predictions are also shown in Figure 5 including some sampled input frames from the SL video and the corresponding ground truth glosses. That the predicted sentences do not semantically coincide with the ground truth sentences is the case for all our model versions - with or without pre-trained weights for the Swin Transformer.

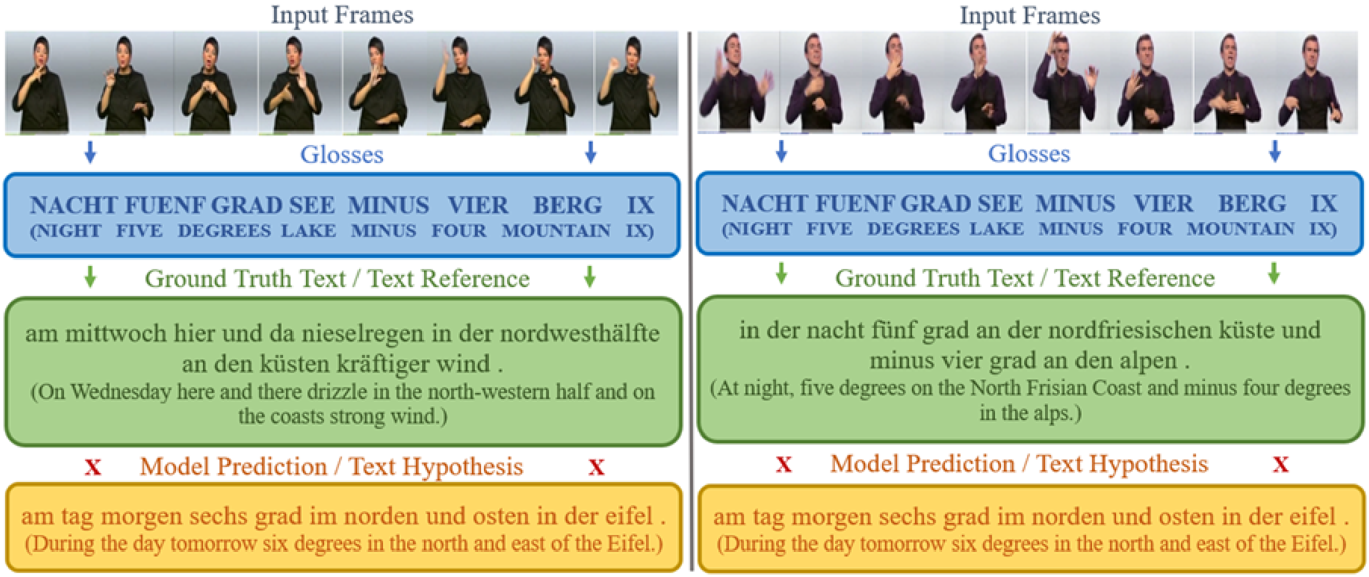
Input Frames
Glosses
NACHT FUENF GRAD SEE MINUS VIER BERG IX
(NIGHT FIVE DEGREES LAKE MINUS FOUR MOUNTAIN IX)
Ground Truth Text / Text Reference
am mittwoch hier und da nieselregen in der nordwesthälfte an den küsten kräftiger wind .
(On Wednesday here and there drizzle in the north-western half and on the coasts strong wind.)
X Model Prediction / Text Hypothesis X
am tag morgen sechs grad im norden und osten in der eifel .
(During the day tomorrow six degrees in the north and east of the Eifel.)
Input Frames
Glosses
NACHT FUENF GRAD SEE MINUS VIER BERG IX
(NIGHT FIVE DEGREES LAKE MINUS FOUR MOUNTAIN IX)
Ground Truth Text / Text Reference
in der nacht fünf grad an der nordfriesischen küste und minus vier grad an den alpen .
(At night, five degrees on the North Frisian Coast and minus four degrees in the alps.)
X Model Prediction / Text Hypothesis X
am tag morgen sechs grad im norden und osten in der eifel .
(During the day tomorrow six degrees in the north and east of the Eifel.)

Fig. 5: The figure shows some input frames from the SL video, the ground truth gloss predictions, and the corresponding text. In addition to that, it shows the predicted sentences of our model (version 2) after 150 training epochs. It is representative for the results of the other model versions as well. It shows that there is no semantic correlation between the model prediction and the ground truth text. Moreover, independently of the input video, it always predicts the same output sentence per learning step.

Epoch 150	
Text Reference	am mittwoch hier und da nieselregen in der nordwesthälfte an den küsten kräftiger wind .
	(On Wednesday here and there drizzle in the north-western half and on the coasts strong wind.)
Text Hypothesis	am tag morgen sechs grad im norden und osten in der eifel .
	(During the day tomorrow six degrees in the north and east of the Eifel.)
Text Reference	in der nacht fünf grad an der nordfriesischen küste und minus vier grad an den alpen .
	(At night, five degrees on the North Frisian Coast and minus four degrees in the alps.)
Text Hypothesis	am tag morgen sechs grad im norden und osten in der eifel einige regentropfen .
	(During the day tomorrow six degrees in the north and east of the Eifel.)
Epoch 151	
Text Reference	später setzt im nordosten verbreitet regen ein im nordwesten bleibt es trocken .
	(Later, rain starts in the northeast and in the northwest, it stays dry.)
Text Hypothesis	am tag dann wird es im westen und süden zum teil noch kalt .
	(During the day then it is still partly getting cold in the west and south.)
Text Reference	und nun die wettervorhersage für morgen dienstag den dreißigsten märz .
	(And now the weather forecast for tomorrow, Tuesday, the thirtieth of March.)
Text Hypothesis	am tag dann wird es im westen und süden zum teil noch kalt .
	(During the day then it is still partly getting cold in the west and south.)

TABLE III: The table shows the spoken language text predictions of the model (version 2). It is representative for the results of the other model versions as well. After 150 training epochs and more, it still does not show a semantic correlation between the text hypothesis and the text reference. Moreover, independently of the input video, it always predicts the same output sentence per learning step.

Moreover, per validation step, all predicted sentences are the same, independent of the input video. It seems that the features extraction part of the model fails to find information that is relevant for SL. This is the case for the Swin Transformer trained from scratch and also for the model with pre-trained weights. The pre-training on a holistic image classification task that is not specific to SL does not seem to give a good enough starting point for SL-specific Transfer Learning. Using the pre-trained Swin Transformer to classify the frames of the SL videos, its class prediction output only focuses on the clothing of the signers (e.g. class ”trench coat”). It focuses on completely different content of the image that is independent of the hand poses of the signers. With that focus being so far away from the visual information that is necessary for the understanding of the signs, the pre-training task is not related enough to the SLT task.

	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Our model 150 epochs	21.98	8.04	3.19	1.15
Our model 151 epochs	19.72	6.05	2.16	0.00

TABLE IV: BLEU Scores of our Pure SLT Transformer model version 2 after 150 and 151 epochs.

However, the model has learned German grammar structures during the training process. It seems to be the best approach for predicting close enough sentences to the ground truth when the input does not contain useful information. Independently of which input vector is given, the network arranges itself to output grammatically correct sentences. This leads to the model by chance sometimes achieving 4 words in a row correctly. Therefore, some results show a BLEU-4 score higher than 0 (see Table IV). However, since it is so close to 0, this is still due to chance.

All in all, it can be said that without any SL-specific pre-training and no intermediate gloss prediction step, the end-to-end SLT training that Camgöz et al. were envisioning, does not work with the current state of research in SLT. The task of learning to recognize signs and translate sign sequences into spoken language text jointly in one step is too hard to learn even for a Pure Transformer architecture. The point where the architecture fails is highly likely the feature extraction from the video frames which is the hardest step in the SLT pipeline.

V. Ablation Studies

A. *Holistically vs. SLT-specific Pre-trained Backbones*

Since training our encoder for feature extraction from SL videos was not feasible and too difficult a task, we wanted to examine the impact of model pre-training on SL-specific feature extraction in comparison to holistic feature extraction and how that differs between CNNs and transformers. To realize this, we removed the CNN feature inputs to the SLT Transformer encoder of the baseline model by Camgöz et al. with other backbone embeddings. The rest of the model was kept the same. We ran the experiment once with the EfficientNet B0 and once with a 2D Swin Transformer; both pre-trained

Backbone Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4
2D Swin Transformer	17.41	6.02	3.14	1.91
EfficientNet B0	15.34	6.07	4.07	3.17
SLT CNN	44.21	30.80	23.20	18.50

TABLE V: BLEU Scores of the baseline SLT Transformer with an SLT-specific pre-trained backbone compared with the scores of the model when using the 2D Swin Transformer and the EfficientNet B0 (both pre-trained on ImageNet1k) as a backbone.

on image classification for ImageNet1k. The implementation details were the same as in previous experiments.

1) Results: When training the model with feature embeddings from the pre-trained 2D Swin Transformer or the EfficientNet B0, the same patterns appear as when the whole model was trained. Independently of the input sentence, the output sentences stay the same for the same training step. The sentences are random and neither literally nor semantically correlate with the ground truth sentences. The best BLEU-4 score that was reached with the model that uses the pre-trained 2D Swin Transformer as Backbone was 1.91 and with the EfficientNet B0 3.17 (see Table V). That is not even comparable to the results that were reached after 14 epochs with the SL-specific pre-trained CNN Backbone of a BLEU-4 score of 18.10. Since this BLEU-4 score is so low and does not increase for longer training periods, it is most likely achieved due to random overlap of common phrases and not due to successful learning. Since such a low BLEU-4 score does not show the actual quality of the output but indicates random overlaps, with these results it can not be stated that the EfficientNet shows better results than the 2D Swin Transformer.

B. Euclidian Distances

We assume that the holistically pre-trained networks extract information from the SL video frames that does not change during the video. In the videos, the biggest difference between the frames is that the signer is moving their hands and face. Since the holistically pre-trained networks do not focus on such information but rather on general object appearances, the embeddings should be very similar for all frames. Contrary to that, the embeddings used by the authors of the baseline model should differ more from each other, since the SL-specific information changes between frames.

To verify this assumption we computed the Euclidian distance between two consecutive frame embeddings of the different backbone models and averaged over 1000 distances in total. The Euclidian distance shows how far two vectors are from each other in an n-dimensional feature space. Different backbone models extract varying amounts of information contained in the video frames. The amount of variation in the embeddings can be captured by computing the Euclidian distance. The farther two embedding vectors are from each other in the embedding space, the more their information content differs. If all embeddings are very close to each other, the information is too similar to contain all necessary SL cues. Due to the natures of the different backbones, the dimensions of the embeddings can be different for each model. We compared this distance metric for the CNN used by Camgöz et al. (1024-dimensional embeddings), the 2D Swin Transformer (768-dimensional embeddings), the ViT (768-dimensional embeddings), the EfficientNet B7 (924160-dimensional embeddings), and the EfficientNet B0 (62720-dimensional embeddings). Apart from the CNN by Camgöz et al., they were all pre-trained on ImageNet1K for image classification. The results can be seen in Table VI. For randomly distributed vectors, the Euclidian Distance would increase for higher dimensions. This phenomenon is called the ”curse of dimensionality” and complicates the comparison of distances of different dimensionalities. However, in our case, it is still possible to draw conclusions, since the higher dimensional vectors do not always show higher distances. The SLT-specific pre-trained CNN shows much higher distances while having a lower dimensionality than the CNNs pre-trained on ImageNet1k. This strengthens our assumption that using a holistically pre-trained CNN does not extract enough SL-specific cues from the video frames and is not suitable for this SLT task. Interestingly, the Transformer models, while also pre-trained on a holistic image classification task, show a much better differentiation between the frames than the EfficientNet CNNs while having a much lower dimensionality. Although they

Backbone Model	Feature Size	Average Euclidian Distance
SLT CNN	1024	56.96
2D Swin Transformer	768	8.80
ViT	768	9.12
EfficientNet B7	924160	3.97
EfficientNet B0	62720	0.23

TABLE VI: Average Euclidian distances for the embedded feature vectors of the different backbone models. For the comparison, the SLT-specific pre-trained CNN and other models pre-trained on ImageNet1k are used. The dimensions of the feature vectors can be seen in the second column.

are still not comparable to the SLT-specific pre-trained network, this outcome still supports our general idea of using transformers instead of CNNs for the feature extraction of SLT tasks since they could be able to manage this complicated task better. To make the transformer models comparable to the SLT-specific CNN, it would be necessary to pre-train them on a sign recognition task as well. This approach would be a very interesting next step towards a Pure Transformer SLT architecture to pursue the unexplored potential of Transformers for this specific task.

VI. Discussion, Conclusion and Outlook

The baseline SLT Transformer model is a valuable contribution to SLT research because of its clear code structure and its potential as a basis for further research. However, the authors seemed to overstate its contributions towards gloss independence and actual end-to-end joint SLR and SLT learning from scratch.

The authors claim that the model is an end-to-end video-to-text solution, which is contestable because the CNN part of the model is not trained with the rest of the model. Therefore, the hardest learning task is not included in this model. It uses CNN embeddings from a pre-trained SLR model but lacks adequate documentation of how these embeddings are obtained since they do not provide any implementation or access to any pre-trained model checkpoint. In particular, the lack of explicit mention of reliance on the SLR-specific pre-trained model could give a misleading impression of progress in the field of SLT. Additionally, the authors mention that their main contribution is to reduce reliance on gloss monitoring, however, the authors' approach relies on pre-trained CNN embeddings trained for gloss detection. This reliance on SLR-specific embeddings essentially maintains the reliance on gloss annotations. The combination of these two misleading claims might lead one to believe that SLT is more advanced than it actually is.

This work has demonstrated that there is more research needed and an end-to-end video-to-text SLT is not realistic yet. We have shown that the step of extracting SLT-specific features from the SL videos is the most crucial step of SLT learning. Without any pre-training on SLR, the SLT task is too complicated to learn from scratch and it is not possible to learn the whole task in an end-to-end manner - even with a Pure Transformer architecture. An alternative is to use SLR-specific pre-trained models as feature embedders. With such a backbone, an architecture like the baseline SLT Transformer can create useful spoken language translations. However, gloss transcriptions are still needed to train that backbone. Holistically pre-trained backbones do not lead to satisfactory outputs. This leads to our conclusion that, with current models, it is not possible to achieve acceptable results without using glosses in the training process.

Our ablation studies have shown, however, that Transformers might be a better alternative for such a feature embedder than a CNN. Future research could include pre-training a 2D or 3D Swin Transformer model on gloss predictions. Using this pre-trained model as a backbone in the baseline SLT Transformer model could help increase the BLEU-4 score. It would be interesting to see, how such a backbone model would perform when it is pre-trained on a smaller dataset with a smaller gloss vocabulary and then used when training a model that is less dependent on glosses on a bigger dataset. If that leads to acceptable performances, this approach could be a meaningful step towards less gloss dependency.

Instead of keeping the feature extraction part out of the training loop, the same idea could also be explored using Transfer Learning. A Swin Transformer would be pre-trained on a smaller dataset for gloss recognition and then used in an end-to-end training process to adapt to a bigger dataset. If the feature extractor is included in the training, instead of used as a backbone, the model might be able to adapt to the bigger dataset and improve the information content of the extracted embeddings. Less annotations for glosses would be required, and the decoder would not use them directly as input which would make the whole model less dependent on the glosses. This approach seems promising with the insights of this work since it is a Pure Transformer architecture and if successful, it would realize the training in the desired end-to-end manner.

References

[Ananthanarayana et al., 2021] Ananthanarayana, T., Srivastava, P., Chintha, A., Santha, A., Landy, B., Panaro, J., Webster, A., Kotecha, N., Sah, S., Sarchet, T., et al. (2021). Deep learning methods for sign language translation. *ACM Transactions on Accessible Computing*, 14(4):1–30.

[Arnab et al., 2021] Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., and Schmid, C. (2021). Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6836–6846.

[Camgöz et al., 2018] Camgöz, N. C., Hadfield, S., Koller, O., Ney, H., and Bowden, R. (2018). Neural sign language translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7784–7793.

[Camgöz et al., 2020a] Camgöz, N. C., Koller, O., Hadfield, S., and Bowden, R. (2020a). Multi-channel transformers for multi-articulatory sign language translation. In *European Conference on Computer Vision*, pages 301–319. Springer.

[Camgöz et al., 2020b] Camgöz, N. C., Koller, O., Hadfield, S., and Bowden, R. (2020b). Sign language transformers: Joint end-to-end sign language recognition and translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[Crasborn, 2015] Crasborn, O. A. (2015). Transcription and notation methods. *Research Methods in Sign Language Studies: A Practical Guide*, pages 74–88.

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

[Forster et al., 2012] Forster, J., Schmidt, C., Hoyoux, T., Koller, O., Zelle, U., Piater, J., and Ney, H. (2012). RWTH-PHOENIX-weather: A large vocabulary sign language recognition and translation corpus. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 3785–3789.

[Graves et al., 2006] Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Koller et al., 2019] Koller, O., Camgoz, N. C., Ney, H., and Bowden, R. (2019). Weakly supervised learning with multi-stream cnn-lstm-hmms to discover sequential parallelism in sign language videos. *IEEE transactions on pattern analysis and machine intelligence*, 42(9):2306–2320.

[Kreutzer et al., 2019] Kreutzer, J., Bastings, J., and Riezler, S. (2019). Joey nmt: A minimalist nmt toolkit for novices. *arXiv preprint arXiv:1907.12484*.

[Lin et al., 2022] Lin, K., Li, L., Lin, C.-C., Ahmed, F., Gan, Z., Liu, Z., Lu, Y., and Wang, L. (2022). SwinBERT: End-to-end transformers with sparse attention for video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17949–17958.

[Liu et al., 2021] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022.

[Liu et al., 2022] Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., and Hu, H. (2022). Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3202–3211.

[Papineni et al., 2002] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

[Tan and Le, 2019] Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[Voskou et al., 2021] Voskou, A., Panousis, K. P., Kosmopoulos, D., Metaxas, D. N., and Chatzis, S. (2021). Stochastic transformer networks with linear competing units: Application to end-to-end SL translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11946–11955.

[Zhu et al., 2020] Zhu, J., Xia, Y., Wu, L., He, D., Qin, T., Zhou, W., Li, H., and Liu, T.-Y. (2020). Incorporating BERT into neural machine translation. In *Proceedings of the 8th International Conference on Learning Representations*.