Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation

Necati Cihan Camgöz, Oscar Koller, Simon Hadfield and Richard Bowden
CVSSP, University of Surrey, Guildford, UK, Microsoft, Munich, Germany
{n.camgoz, s.hadfield, r.bowden}@surrey.ac.uk, oscar.koller@microsoft.com

Abstract

Prior work on Sign Language Translation has shown that having a mid-level sign gloss representation (effectively recognizing the individual signs) improves the translation performance drastically. In fact, the current state-of-the-art in translation requires gloss level tokenization in order to work. We introduce a novel transformer based architecture that jointly learns Continuous Sign Language Recognition and Translation while being trainable in an end-to-end manner. This is achieved by using a Connectionist Temporal Classification (CTC) loss to bind the recognition and translation problems into a single unified architecture. This joint approach does not require any ground-truth timing information, simultaneously solving two co-dependant sequence-to-sequence learning problems and leads to significant performance gains.

We evaluate the recognition and translation performances of our approaches on the challenging RWTH-PHOENIX-Weather-2014T (PHOENIX14T) dataset. We report state-of-the-art sign language recognition and translation results achieved by our Sign Language Transformers. Our translation networks outperform both sign video to spoken language and gloss to spoken language translation models, in some cases more than doubling the performance (9.58 vs. 21.80 BLEU-4 Score). We also share new baseline translation results using transformer networks for several other text-to-text sign language translation tasks.

1. Introduction

Sign Languages are the native languages of the Deaf and their main medium of communication. As visual languages, they utilize multiple complementary channels (e.g. facial expression, mouth and movement of the head, shoulders and torso) to convey information. This includes manual features, such as hand shape, movement and pose as well as non-manuals features, such as facial expression, mouth and movement of the head, shoulders and torso.

The goal of sign language translation is either to convert written language into a video of sign (production) or to extract an equivalent spoken language sentence from a video of someone performing continuous sign. However, in the field of computer vision, much of this latter work has focused on recognising the sequence of sign glosses (Continuous Sign Language Recognition (CSLR)) rather than the full translation to a spoken language equivalent (Sign Language Translation (SLT)). This distinction is important as the grammar of sign and spoken languages are very different. These differences include (to name a few): different word ordering, multiple channels used to convey concurrent information and the use of direction and space to convey the relationships between objects. Put simply, the mapping between speech and sign is complex and there is no simple word-to-sign mapping.

Generating spoken language sentences given sign language videos is therefore a spatio-temporal machine translation task. Such a translation system requires us to accomplish several sub-tasks, which are currently unsolved:

**Sign Segmentation:** Firstly, the system needs to detect sign sentences, which are commonly formed using topic-comment structures, from continuous sign language videos. This is trivial to achieve for text based machine translation tasks, where the models can use punctuation marks to separate sentences. Speech-based recognition and translation systems, on the other hand, look for pauses, e.g. silent regions, between phonemes to segment spoken language utterances. There have been studies in the literature addressing automatic sign segmentation. However to the best of the authors’ knowledge, there is no study which utilizes sign segmentation for realizing continuous sign language translation.

---

1 Sign glosses are spoken language words that match the meaning of signs and, linguistically, manifest as minimal lexical items.
**Sign Language Recognition and Understanding:** Following successful segmentation, the system needs to understand what information is being conveyed within a sign sentence. Current approaches tackle this by recognizing sign glosses and other linguistic components. Such methods can be grouped under the banner of CSLR [40, 8]. From a computer vision perspective, this is the most challenging task. Considering the input of the system is high dimensional spatio-temporal data, i.e., sign videos, models are required that understand what a signer looks like and how they interact and move within their 3D signing space. Moreover, the model needs to comprehend what these aspects mean in combination. This complex modelling problem is exacerbated by the asynchronous multi-articulatory nature of sign languages [51, 58]. Although there have been promising results towards CSLR, the state-of-the-art [39] can only recognize sign glosses and operate within a limited domain of discourse, namely weather forecasts [26].

**Sign Language Translation:** Once the information embedded in the sign sentences is understood by the system, the final step is to generate spoken language sentences. As with any other natural language, sign languages have their own unique linguistic and grammatical structures, which often do not have a one-to-one mapping to their spoken language counterparts. As such, this problem truly represents a machine translation task. Initial studies conducted by computational linguists have used text-to-text statistical machine translation models to learn the mapping between sign glosses and their spoken language translations [45]. However, glosses are simplified representations of sign languages and linguists are yet to come to a consensus on how sign languages should be annotated.

There have been few contributions towards video based continuous SLT, mainly due to the lack of suitable datasets to train such models. More recently, Camgoz et al. [9] released the first publicly available sign language video to spoken language translation dataset, namely PHOENIX14T. In their work, the authors proposed approaching SLT as a Neural Machine Translation (NMT) problem. Using attention-based NMT models [44, 3], they define several SLT tasks and realized the first end-to-end sign language video to spoken language sentence translation model, namely Sign2Text.

One of the main findings of [9] was that using gloss based mid-level representations improved the SLT performance drastically when compared to an end-to-end Sign2Text approach. The resulting Sign2Gloss2Text model first recognized glosses from continuous sign videos using a state-of-the-art CSLR method [41], which worked as a tokenization layer. The recognized sign glosses were then passed to a text-to-text attention-based NMT network [44] to generate spoken language sentences.

We hypothesize that there are two main reasons why Sign2Gloss2Text performs better than Sign2Text (18.13 vs 9.58 BLEU-4 scores). Firstly, the number of sign glosses is much lower than the number of frames in the videos they represent. By using gloss representations instead of the spatial embeddings extracted from the video frames, Sign2Gloss2Text avoids the long-term dependency issues, which Sign2Text suffers from.

We think the second and more critical reason is the lack of direct guidance for understanding sign sentences in Sign2Text training. Given the aforementioned complexity of the task, it might be too difficult for current Neural Sign Language Translation architectures to comprehend sign without any explicit intermediate supervision. In this paper we propose a novel Sign Language Transformer approach, which addresses this issue while avoiding the need for a two-step pipeline, where translation is solely dependent on recognition accuracy. This is achieved by jointly learning sign language recognition and translation from spatial-representations of sign language videos in an end-to-end manner. Exploiting the encoder-decoder based architecture of transformer networks [70], we propose a multi-task formalization of the joint continuous sign language recognition and translation problem.

To help our translation networks with sign language understanding and to achieve CSLR, we introduce a Sign Language Recognition Transformer (SLRT), an encoder transformer model trained using a CTC loss [2], to predict sign gloss sequences. SLRT takes spatial embeddings extracted from sign videos and learns spatio-temporal representations. These representations are then fed to the Sign Language Translation Transformer (SLTT), an autoregressive transformer decoder model, which is trained to predict one word at a time to generate the corresponding spoken language sentence. An overview of the approach can be seen in Figure 1.

The contributions of this paper can be summarized as:
- A novel multi-task formalization of CSLR and SLT which exploits the supervision power of glosses, without limiting the translation to spoken language.
- The first successful application of transformers for CSLR and SLT which achieves state-of-the-art results in both recognition and translation accuracy, vastly outperforming all comparable previous approaches.
- A broad range of new baseline results to guide future research in this field.

The rest of this paper is organized as follows: In Section 2 we survey the previous studies on SLT and the state-of-the-art in the field of NMT. In Section 3 we introduce Sign Language Transformers, a novel joint sign language recognition and translation approach which can be trained in an end-to-end manner. We share our experimental setup in Section 4. We then report quantitative results of the Sign Language Transformers in Section 5 and present new baseline results for the previously defined text-to-text translation tasks [9]. In Section 6, we share translation examples generated by our network to give the reader further qualitative insight of how our approach performs. We conclude the paper in Section 7 by discussing our findings and possible future work.
2. Related Work

Sign languages have been studied by the computer vision community for the last three decades [65, 56]. The end goal of computational sign language research is to build translation and production systems [19], that are capable of translating sign language videos to spoken language sentences and vice versa, to ease the daily lives of the Deaf [15, 6]. However, most of the research to date has mainly focused on Isolated Sign Language Recognition [35, 75, 72, 10, 63, 67], working on application specific datasets [11, 71, 23], thus limiting the applicability of such technologies. More recent work has tackled continuous data [42, 32, 17, 18], but the move from recognition to translation is still in its infancy [9].

There have been earlier attempts to realize SLT by computational linguists. However, existing work has solely focused on the text-to-text translation problem and has been very limited in size, averaging around 3000 total words [46, 57, 54]. Using statistical machine translation methods, Stein et al. [57] proposed a weather broadcast translation system from spoken German into German Sign Language - Deutsche Gebärdensprache (DGS) and vice versa, using the RWTH-PHOENIX-Weather-2012 (PHOENIX12) dataset. Another method translated air travel information from spoken English to Irish Sign Language (ISL), spoken German to ISL, spoken English to DGS, and spoken German to DGS [45]. Ebling [22] developed an approach to translate written German train announcements into Swiss German Sign Language - Deutschschweizer Gebärdensprache (DGS). While non-manual information has not been included in most previous systems, Ebling & Huenerfauth [24] proposed a sequence classification based model to schedule the automatic generation of non-manual features after the core machine translation step.

Conceptual video based SLT systems were introduced in the early 2000s [7]. There have been studies, such as [12], which propose recognizing signs in isolation and then constructing sentences using a language model. However, end-to-end SLT from video has not been realized until recently.

The most important obstacle to vision based SLT research has been the availability of suitable datasets. Curating and annotating continuous sign language videos with spoken language translations is a laborious task. There are datasets available from linguistic sources [53, 31] and sign language interpretations from broadcasts [14]. However, the available annotations are either weak (subtitles) or too few to build models which would work on a large domain of discourse. In addition, such datasets lack the human pose information which legacy Sign Language Recognition (SLR) methods heavily relied on.

The relationship between sign sentences and their spoken language translations are non-monotonic, as they have different ordering. Also, sign glosses and linguistic constructs do not necessarily have a one-to-one mapping with their spoken language counterparts. This made the use of available CSLR methods [42, 41] (that were designed to learn from weakly annotated data) infeasible, as they are build on the assumption that sign language videos and corresponding annotations share the same temporal order.

To address these issues, Camgoz et al. [9] released the first publicly available SLT dataset, PHOENIX14T, which is an extension of the popular RWTH-PHOENIX-Weather-2014 (PHOENIX14) CSLR dataset. The authors approached the task as a spatio-temporal neural machine translation problem, which they term ‘Neural Sign Language Translation’. They proposed a system using Convolutional Neural Networks (CNNs) in combination with attention-based NMT methods [44, 3], to realize the first end-to-end SLT models. Following this, Ko et al. proposed a similar approach but used body key-point coordinates as input for their translation networks, and evaluated their method on a Korean Sign Language dataset [33].

Concurrently, there have been several advancements in the field of NMT, one of the most important being the introduction of transformer networks [70]. Transformers drastically improved the translation performance over legacy attention based encoder-decoder approaches. Also due to the fully-connected nature of the architecture, transformers are fast and easy to parallelize, which has enabled them to become the new go to architecture for many machine translation tasks. In addition to NMT, transformers have achieved success in various other challenging tasks, such as language modelling [19, 77], learning sentence representations [21], multi-modal language understanding [68], activity [73] and speech recognition [34]. Inspired by their recent wide-spread success, in this work we propose a novel architecture where multiple co-dependent transformer networks are simultaneously trained to jointly solve related tasks. We then apply this architecture to the problem of simultaneous recognition and translation where joint training provides significant benefits.

3. Sign Language Transformers

In this section we introduce Sign Language Transformers which jointly learn to recognize and translate sign video sequences into sign glosses and spoken language sentences in an end-to-end manner. Our objective is to learn the conditional probabilities $p(G|V)$ and $p(S|V)$ of generating a sign gloss sequence $G = (g_1, ..., g_N)$ with $N$ glosses and a spoken language sentence $S = (w_1, ..., w_U)$ with $U$ words given a sign video $V = (I_1, ..., I_T)$ with $T$ frames.

Modelling these conditional probabilities is a sequence-to-sequence task, and poses several challenges. In both cases, the number of tokens in the source domain is much larger than the corresponding target sequence lengths (i.e. $T \gg N$ and $T \gg U$). Furthermore, the mapping between sign language videos, $V$, and spoken language sentences, $S$, is non-monotonic, as both languages have different vocabularies, grammatical rules and orderings.

Previous sequence-to-sequence based literature on SLT can be categorized into two groups: The first group break down the problem in two stages. They consider CSLR as an
initial process and then try to solve the problem as a text-to-
text translation task \[12, 9\]. Camgoz et al. utilized a state-
of-the-art CSLR method \[41\] to obtain sign glosses, and
then used an attention-based text-to-text NMT model \[44\] to
learn the sign gloss to spoken language sentence translation,
p(S|G) \[9\]. However, in doing so, this approach introduces
an information bottleneck in the mid-level gloss representation.
This limits the network’s ability to understand sign
language as the translation model can only be as good as the
sign gloss annotations it was trained from. There is also an
inherent loss of information as a sign gloss is an incomplete
annotation intended only for linguistic study and it therefore
neglects many crucial details and information present in the
original sign language video.

The second group of methods focus on translation from
the sign video representations to spoken language with no
intermediate representation \[9, 38\]. These approaches at-
tempt to learn p(S|V) directly. Given enough data and a
sufficiently sophisticated network architecture, such mod-
els could theoretically realize end-to-end SLT with no need
for a human-interpretable information that act as a bottle-
neck. However, due to the lack of direct supervision guid-
ing sign language understanding, such methods have signif-
ificantly lower performance than their counterparts on cur-
cently available datasets \[9\].

To address this, we propose to jointly learn p(G|V) and
p(S|V), in an end-to-end manner. We build upon trans-
former networks \[70\] to create a unified model, which we
call Sign Language Transformers (See Figure 2). We
train our networks to generate spoken language sentences
from sign language video representations. During training,
we inject intermediate gloss supervision in the form of a
CTC loss into the Sign Language Recognition Transformer
(SLRT) encoder. This helps our networks learn more mean-

![Figure 2: A detailed overview of a single layered Sign Language Transformer. (SE: Spatial Embedding, WE: Word Embedding, PE: Positional Encoding, FF: Feed Forward)](image)

3.1. Spatial and Word Embeddings

Following the classic NMT pipeline, we start by embed-
ding our source and target tokens, namely sign language
video frames and spoken language words. As word embed-
ding we use a linear layer, which is initialized from scratch
during training, to project a one-hot-vector representation
of the words into a denser space. To embed video frames, we
use the SpatialEmbedding approach \[9\], and propagate each
image through CNNs. We formulate these operations as:

\[
m_u = \text{WordEmbedding}(w_u) \\
 f_t = \text{SpatialEmbedding}(I_t) 
\]

where \(m_u\) is the embedded representation of the spoken
language word \(w_u\) and \(f_t\) corresponds to the non-linear
frame level spatial representation obtained from a CNN.

Unlike other sequence-to-sequence models \[61, 27\],
transformer networks do not employ recurrence or con-
volutions, thus lacking the positional information within se-
quencies. To address this issue we follow the positional en-
coding method proposed in \[70\] and add temporal ordering
information to our embedded representations as:

\[
f_t = f_t + \text{PositionalEncoding}(t) \\
 m_u = m_u + \text{PositionalEncoding}(u) 
\]

where PositionalEncoding is a predefined function which
produces a unique vector in the form of a phase shifted sine
wave for each time step.
3.2. Sign Language Recognition Transformers

The aim of SLRT is to recognize glosses from continuous sign language videos while learning meaningful spatio-temporal representations for the end goal of sign language translation. Using the positionally encoded spatial embeddings, \( \hat{f}_{1:T} \), we train a transformer encoder model [70].

The inputs to SLRT are first modelled by a Self-Attention layer which learns the contextual relationship between the frame representations of a video. Outputs of the self-attention are then passed through a non-linear point-wise feed forward layer. All the operations are followed by residual connections and normalization to help training. We formulate this encoding process as:

\[
z_t = \text{SLRT}(\hat{f}_t | \hat{f}_{1:T})
\]  

(2)

where \( z_t \) denotes the spatio-temporal representation of the frame \( I_t \), which is generated by SLRT at time step \( t \), given the spatial representations of all of the video frames, \( \hat{f}_{1:T} \).

We inject intermediate supervision to help our networks understand sign and to guide them to learn a meaningful sign representation which helps with the main task of translation. We train the SLRT to model \( p(G|V) \) and predict sign glosses. Due to the spatio-temporal nature of the signs, glosses have a one-to-many mapping to video frames but share the same ordering.

One way to train the SLRT would be using cross-entropy loss [29] with frame level annotations. However, sign gloss annotations with such precision are rare. An alternative form of weaker supervision is to use a sequence-to-sequence learning loss functions, such as CTC [30].

Given spatio-temporal representations, \( z_{1:T} \), we obtain frame level gloss probabilities, \( p(g_t|V) \), using a linear projection layer followed by a softmax activation. We then use CTC to compute \( p(G|V) \) by marginalizing over all possible \( V \) to \( G \) alignments as:

\[
p(G|V) = \sum_{\pi \in B} p(\pi | V)
\]  

(3)

where \( \pi \) is a path and \( B \) is the set of all viable paths that correspond to \( G \). We then use the \( p(G|V) \) to calculate the CSLR loss as:

\[
L_R = 1 - p(G^*|V)
\]  

(4)

where \( G^* \) is the ground truth gloss sequence.

3.3. Sign Language Translation Transformers

The end goal of our approach is to generate spoken language sentences from sign video representations. We propose training an autoregressive transformer decoder model, named SLTT, which exploits the spatio-temporal representations learned by the SLRT. We start by prefixing the target spoken language sentence \( S \) with the special beginning of sentence token, \( < \text{bos} > \). We then extract the positionally encoded word embeddings. These embeddings are passed to a masked self-attention layer. Although the main idea behind self-attention is the same as in SLRT, the SLTT utilizes a mask over the self-attention layer inputs. This ensures that each token may only use its predecessors while extracting contextual information. This masking operation is necessary, as at inference time the SLTT won’t have access to the output tokens which would follow the token currently being decoded.

Representations extracted from both SLRT and SLTT self-attention layers are combined and given to an encoder-decoder attention module which learns the mapping between source and target sequences. Outputs of the encoder-decoder attention are then passed through a non-linear point-wise feed forward layer. Similar to SLRT, all the operations are followed by residual connections and normalization. We formulate this decoding process as:

\[
h_{u+1} = \text{SLTT}(\hat{m}_u|\hat{m}_{1,u-1}, z_{1:T}).
\]  

(5)

SLTT learns to generate one word at a time until it produces the special end of sentence token, \( < \text{eos} > \). It is trained by decomposing the sequence level conditional probability \( p(S|V) \) into ordered conditional probabilities

\[
p(S|V) = \prod_{u=1}^{U} p(w_u|h_u)
\]  

(6)

which are used to calculate the cross-entropy loss for each word as:

\[
L_T = 1 - \prod_{u=1}^{U} \sum_{d=1}^{D} p(\hat{w}_u^d)p(w_u^d|h_u)
\]  

(7)

where \( p(\hat{w}_u^d) \) represents the ground truth probability of word \( w_u^d \) at decoding step \( u \) and \( D \) is the target language vocabulary size.

We train our networks by minimizing the joint loss term \( L \), which is a weighted sum of the recognition loss \( L_R \) and the translation loss \( L_T \) as:

\[
L = \lambda_R L_R + \lambda_T L_T
\]  

(8)

where \( \lambda_R \) and \( \lambda_T \) are hyper parameters which decides the importance of each loss function during training and are evaluated in Section 5.

4. Dataset and Translation Protocols

We evaluate our approach on the recently released PHOENIX14T dataset [2], which is a large vocabulary, continuous SLT corpus. PHOENIX14T is a translation focused extension of the PHOENIX14 corpus, which has become the primary benchmark for CSLR in recent years.

PHOENIX14T contains parallel sign language videos, gloss annotations and their translations, which makes it the only available dataset suitable for training and evaluating joint SLR and SLT techniques. The corpus includes unconstrained continuous sign language from 9 different signers with a vocabulary of 1066 different signs. Translations for these videos are provided in German spoken language with a vocabulary of 2887 different words.
The evaluation protocols on the PHOENIX14T dataset, as laid down by [9], are as follows:

Sign2Text is the end goal of SLT, where the objective is to translate directly from continuous sign videos to spoken language sentences without going via any intermediary representation, such as glosses.

Gloss2Text is a text-to-text translation problem, where the objective is to translate ground truth sign gloss sequences to German spoken language sentences. The results of these experiments act as a virtual upper bound for the available NMT translation technology. This assumption is based on the fact that perfect sign language recognition/understanding is simulated by using the ground truth gloss annotation. However, as mentioned earlier, one needs to bear in mind that gloss representations are imprecise. As glosses are textual representations of multi-channel temporal signals, they represent an information bottleneck for any translation system. This means that under ideal conditions, a Sign2Text system could and should outperform Gloss2Text. However, more sophisticated network architectures and data are needed to achieve this and hence such a goal remains a longer term objective beyond the scope of this manuscript.

Sign2Gloss2Text is the current state-of-the-art in SLT. This approach utilizes CSLR models to extract gloss sequences from sign language videos which are then used to solve the translation task as a text-to-text problem by training a Gloss2Text network using the CSLR predictions. Sign2Gloss $\rightarrow$ Gloss2Text is similar to Sign2Gloss2Text and also uses CSLR models to extract gloss sequences. However, instead of training text-to-text translation networks from scratch, Sign2Gloss $\rightarrow$ Gloss2Text models use the best performing Gloss2Text network, which has been trained with ground truth gloss annotations, to generate spoken language sentences from intermediate sign gloss sequences from the output of the CSLR models.

In addition to evaluating our networks in the context of the above protocols, we additionally introduce two new protocols which follow the same naming convention. Sign2Gloss is a protocol which essentially performs CSLR, while Sign2(Gloss+Text) requires joint learning of continuous sign language recognition and translation.

5. Quantitative Results

In this section we share our sign language recognition and translation experimental setups and report quantitative results. We first go over the implementation details and introduce the evaluation metrics we will be using to measure the performance of our models. We start our experiments by applying transformer networks to the text-to-text based SLT tasks, namely Gloss2Text, Sign2Gloss2Text, Sign2Gloss $\rightarrow$ Gloss2Text and report improved performance over using Recurrent Neural Network (RNN) based models. We share our Sign2Gloss experiments, in which we explore the effects of different types of spatial embeddings and network structures on the performance of CSLR. We then train Sign2Text and Sign2(Gloss+Text) models using the best performing Sign2Gloss configuration and investigate the effect of different recognition loss weights on the joint recognition and translation performance. Finally, we compare our best performing models against other approaches and report state-of-the-art results.

5.1. Implementation and Evaluation Details

Framework: We used a modified version of JoeyNMT [43] to implement our Sign Language Transformers. All components of our network were built using the PyTorch framework [50], except the CTC beam search decoding, for which we utilized the TensorFlow implementation [11].

Network Details: Our transformers are built using 512 hidden units and 8 heads in each layer. We use Xavier initialization [28] and train all of our networks from scratch. We also utilize dropout with 0.1 drop rate on transformer layers and word embeddings to mitigate over-fitting.

Performance Metrics: We use Word Error Rate (WER) for assessing our recognition models, as it is the prevalent metric for evaluating CSLR performance [40]. To measure the translation performance of our networks, we utilized BLEU [49] score (n-grams ranging from 1 to 4), which is the most common metric for machine translation.

Training: We used the Adam [27] optimizer to train our networks using a batch size of 32 with a learning rate of $10^{-3}$ ($\beta_1=0.9$, $\beta_2=0.998$) and a weight decay of $10^{-3}$. We utilize plateau learning rate scheduling which tracks the development set performance. We evaluate our network every 100 iterations. If the development score does not decrease for 8 evaluation steps, we decrease the learning rate by a factor of 0.7. This continues until the learning rate drops below $10^{-6}$.

Decoding: During the training and validation steps we employ a greedy search to decode both gloss sequences and spoken language sentences. At inference time, we utilize beam search decoding with widths ranging from 0 to 10. We also implement a length penalty [74] with $\alpha$ values ranging from 0 to 2. We find the best performing combination of beam width and $\alpha$ on the development set and use these values for the test set evaluation.

5.2. Text-to-Text Sign Language Translation

In our first set of experiments, we adapt the transformer backbone of our technique, for text-to-text sign language translation. We then evaluate the performance gain achieved over the RNN-based attention architectures.

As can be seen in Table 1 utilizing transformers for text-to-text sign language translation improved the performance across all tasks, reaching an impressive 25.35/24.54 BLEU-4 score on the development and test sets. We believe this performance gain is due to the more sophisticated attention architectures, namely self-attention modules, which learn the contextual information within both source and target sequences.

https://github.com/neccam/slt
Table 1: (Top) New baseline results for text-to-text tasks on Phoenix2014T [9] using transformer networks and (Bottom) Our best performing Sign Language Transformers compared against the state-of-the-art.

5.3. Sign2Gloss

To tackle the Sign2Gloss task, we utilize our SLRT networks. Any CNN architecture can be used as spatial embedding layers to learn the sign language video frame representation while training SLRT in an end-to-end manner. However, due to hardware limitations (graphics card memory) we utilize pretrained CNNs as our spatial embeddings. We extract frame level representations from sign videos and train our sign language transformers to learn CSLR and SLT jointly in an end-to-end manner.

In our first set of experiments, we investigate which CNN we should be using to represent our sign videos. We utilize state-of-the-art EfficientNets [66], namely B0, B4 and B7, which were trained on ImageNet [20]. We also use an Inception [64] network which was pretrained for learning sign language recognition in a CNN+LSTM+HMM setup [59]. In this set of experiments we employed a two layered transformer encoder model.

Table 2 shows that as the spatial embedding layer becomes more advanced, i.e. B0 vs B7, the recognition performance increases. However, our networks benefited more when we used pretrained features, as these networks had seen sign videos before and learned kernels which can embed more meaningful representations in the latent space. We then tried utilizing Batch Normalization [33] followed by a ReLU [47] to normalize our inputs and allow our networks to learn more abstract non-linear representations. This improved our results drastically, giving us a boost of nearly 7% and 6% of absolute WER reduction on the development and test sets, respectively. Considering these findings, the rest of our experiments used the batch normalized pretrained CNN features of [39] followed by ReLU.

Next, we investigated the effects of having different numbers of transformer layers. Although having a larger number of layers would allow our networks to learn more abstract representations, it also makes them prone to overfitting. To this end, we built our SLRT networks using one to six layers and evaluate their CSLR performance.

Our recognition performance initially improves with additional layers (See Table 3). However, as we continue adding more layers, our networks started to over-fit on the training data, causing performance degradation. In the light of this, for the rest of our experiments, we constructed our sign language transformers using three layers.

5.4. Sign2Text and Sign2(Gloss+Text)

In our next set of experiments we examine the performance gain achieved by unifying the recognition and translation tasks into a single model. As a baseline, we trained a Sign2Text network by setting our recognition loss weight to zero. We then jointly train our sign language transformers, for recognition and translation, with various weightings between the losses.

As can be seen in Table 4, jointly learning recognition and translation with equal weighting (\(\lambda_R = \lambda_T = 1.0\)) improves the translation performance, while degrading the recognition performance compared to task specific networks. We believe this is due to scale differences of the CTC and word-level cross entropy losses. Increasing the recognition loss weight improved both the recognition and
the translation performance, demonstrating the value of sharing training between these related tasks.

Compared to previously published methods, our Sign Language Transformers surpass both their recognition and translation performance (See Table 1). We report a decrease of 2% WER over [8] on the test set in both Sign2Gloss and Sign2(Gloss+Text) setups. More impressively, both our Sign2Text and Sign2(Gloss+Text) networks doubled the previous state-of-the-art translation results (9.58 vs. 20.17 and 21.32 BLEU-4, respectively). Furthermore, our best performing translation Sign2(Gloss+Text) outperforms Camgoz et al.’s text-to-text based Gloss2Text translation performance (19.26 vs 21.32 BLEU-4), which was previously proposed as a pseudo upper bound on performance in [9]. This supports our claim that given more sophisticated network architectures, one would and should achieve better performance translating directly from video representations rather than doing text-to-text translation through a limited gloss representation.

### 6. Qualitative Results

In this section we report our qualitative results. We share the spoken language translations generated by our best performing Sign2(Gloss+Text) model given sign video representations (See Table 5). As the annotations in the PHOENIX14T dataset are in German, we share both the produced sentences and their translations in English.

Overall, the quality of the translations is good, and even where the exact wording differs, it conveys the same information. The most difficult translations seem to be named entities such as locations which occur in limited contexts in the training data. Specific numbers are also challenging as there is no grammatical context to distinguish one from another. Despite this, the sentences produced follow standard grammar with surprisingly few exceptions.

### 7. Conclusion and Future Work

Sign language recognition and understanding is an essential part of the sign language translation task. Previous translation approaches relied heavily on recognition as the initial step of their system. In this paper we proposed Sign Language Transformers, a novel transformer based architecture to jointly learn sign language recognition and trans-

---

**Table 4:** Training Sign Language Transformers to jointly recognize and translate with different weight on recognition loss.

<table>
<thead>
<tr>
<th>Loss Weights</th>
<th>DEV WER</th>
<th>BLEU-4</th>
<th>TEST WER</th>
<th>BLEU-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_1$</td>
<td></td>
<td></td>
<td>$\lambda_2$</td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>24.88</td>
<td>0.0</td>
<td>24.59</td>
</tr>
<tr>
<td>0.0</td>
<td>1.0</td>
<td>-</td>
<td>20.69</td>
<td>-</td>
</tr>
<tr>
<td>1.0</td>
<td>1.0</td>
<td>35.13</td>
<td>21.73</td>
<td>33.75</td>
</tr>
<tr>
<td>2.5</td>
<td>1.0</td>
<td>26.99</td>
<td>22.11</td>
<td>27.55</td>
</tr>
<tr>
<td>5.0</td>
<td>1.0</td>
<td>24.61</td>
<td>22.12</td>
<td>24.49</td>
</tr>
<tr>
<td>10.0</td>
<td>1.0</td>
<td>24.98</td>
<td>22.38</td>
<td>26.16</td>
</tr>
<tr>
<td>20.0</td>
<td>1.0</td>
<td>25.87</td>
<td>20.90</td>
<td>25.73</td>
</tr>
</tbody>
</table>

---

**Table 5:** Generated spoken language translations by our Sign Language Transformers.

---

4Visit our code repository for further qualitative examples.
References


