End to End Sign Language Translation via Multitask Learning

Abstract—**Sign language translation (SLT) is usually seen as a two-step process of continuous sign language recognition (CSLR) and gloss-to-text translation. We propose a novel, Transformer-based architecture to jointly perform CSLR and sign-translation in an end-to-end fashion. We extend the ordinary Transformer decoder with two channels to support multitasking, where each channel is devoted to solving a particular problem. To control the memory footprint of our model, channels are designed to share most of their parameters with each other. However, each channel still has a dedicated set of parameters that is fine-tuned with respect to the channel's task. In order to evaluate the proposed architecture, we focus on translating German signs into English sequences and use the `RWTH-PHOENIX-Weather 2014 T` corpus in our experiments. Evaluation results along with detailed quantitative and qualitative analyses indicate that the mixture of information provided by the multitask decoder was successful and enabled us to achieve superior performance in comparison to other SLT models.**

Index Terms—**Sign Language Translation, Multitasking, End to end learning**

I. Introduction

Sign languages (SLs) are the main medium of communication for people with hearing problems. In such languages, linguistic phenomena are in conjunction with other factors such as body movements, poses, and facial expressions. Accordingly, existing tools designed to process spoken languages are not directly applicable to SLs. It involves translating sign videos to a target language and this makes this task relatively harder compared to traditional Neural Machine Translation (NMT). In this paper, we particularly focus on *translating* these languages and propose a tailored solution to interpret signs from video frames and translate them into text sequences in a target language.

One approach to SLT is to view the process as a combination of three tasks, *viz.* sign segmentation, sign language recognition (SLR), and gloss-to-word translation. In text sequences, punctuation marks and white spaces help segment them into fundamental units. Silent regions namely pause, between phonemes play the same role in speech processing tasks [1]. However, the task of segmentation is not very straightforward when working with SLs and an SL processing task may require some sort of segmentation [2], [3]. The purpose of sign segmentation is to be clear about the input units, and their boundaries, and see how to feed the model. Once the segmentation is completed, the next step would be understanding/recognizing information carried out by signs, which is referred to as SLR in the literature. What SLR generates is a sequence of special tokens known as sign language glosses. The final step, translation, takes glosses and transforms them into words in the target language.

Performing each of these tasks separately requires dedicated models and datasets, which would be quite challenging. [4] proposed a much simpler and more effective solution. They treated the aforementioned three-step pipeline as an end-to-end process of transforming video frames into target-language words and show that their approach can in fact outperform other conventional methods. In their model, SLT is carried out via a single neural network and there is no clear step defined for segmentation or SLR. The network, itself, decides how to set boundaries and use information stored in video frames to accomplish the task.

Our approach to performing SLT is also to develop an end-to-end model. We propose a Transformer [5] model which relies on multitasking. Similar to [4], we do not feed our model with segmented units and let the network decide how to process the video frames. However, on the target side (i.e, on the decoder side), we explicitly force the model to *i)* generate sign glosses and *ii)* transcribe source signs into a target language. This form of training defines a better objective for the network, and it clearly learns what input video frames are processed and how internal representations should be generated in order to serve the target tasks. [4] and other similar models only provide the network with one generic task/objective (to perform SLT), whereas we decompose it into more tangible and detailed goals, and this is the main distinctive feature of our model.

Our aim for using multi-task learning is based upon exploiting the representation bias in the dataset, which helps the model to learn better internal representations that related tasks might prefer. Specifically, our proposed method is based on the hard parameter sharing paradigm for multi-tasking [6], where tasks specific layers are placed after the hidden shared layers. For a fair comparison with our proposed hard parameter sharing-based model, we also train a baseline model ($D_{SEP}++$), which implements the soft parameter sharing paradigm of the multi-task learning framework.

Our main contributions can be summarized as follows:

- Exploiting available gloss sequence at both encoder and decoder side effectively, which performs better than the prior state-of-the-art [4]. In the unavailability of gloss sequence, We use a multitasking objective, where besides decoding the source sign into a fixed target language (i.e, German); we also translate the source sign into a different target language (i.e, English). To train our decoder, we translate target-side German sentences into English via an NMT model. This auxiliary multitasking objective outperforms the baseline Transformer.

- Our proposed approach is task agnostic and similar multitasking objectives can be applied to the other tasks as well.

II. Related Work

The SLT systems were introduced in the early 2000s [7] where language models were used to construct sentences by recognizing the isolated signs [8]. However, there was no sign of directly converting videos into sentences, i.e. end-to-end SLT system until recently. For the SLT system, a large annotated dataset is required but the creation and annotation of sign videos is a laborious task. A few datasets from linguistic sources [9], [10] and broadcast interpretation [11] are available which are either weak (subtitles) or have very few samples to build models which would work on a large domain of discourse.

The CSLR methods [12], [13] (designed to learn from weakly annotated data) were infeasible, as researchers assumed that sign videos and their annotations share the same temporal order. With the creation of SL datasets such as `RWTH-PHOENIX-Weather 2012` [14], `RWTH-PHOENIX-Weather 2014` [15], or `KETI` [16] made it possible for the researchers to directly work on video frames and invent models to interpret signs/meanings residing in them.

SLR models utilized convolutional modules to encode the video frames and recurrent mechanisms to capture temporal structures and dependencies in between frames [12], [17]. SLT models also benefited from similar technologies for translating information into actual sentences [18], [19]. Researchers customized this pipeline based on their own needs, e.g. [16] augmented network inputs with key points extracted from human faces, hands, and body parts. [20] proposed the connectionist temporal classification (CTC) loss which is useful when working with weakly annotated datasets. Due to its success, CTC quickly turned into a mainstream loss function in sequence-to-sequence applications. [4] embedded the CTC loss into Transformers [5] to learn the continuous sign language recognition and translation. Though recent methods like [21] propose reasonable breakthroughs for SLT, they either do not perform recognition and translation jointly or are very complex to be applicable to other similar tasks.

III. Methodology

Current state-of-the-art for SLT [4] relies on a Transformer-based architecture [1] in which the encoder is fed with sign video frames and the decoder produces translations conditioned on encoder's representations. In this framework, the encoder is trained to act as a gloss generator and this makes it possible to perform SLR and SLT simultaneously. Our model also follows a similar process but via a different and better architecture.

While our best-performing model implements the same encoding process as in [4], our decoder is equipped with a multitasking strategy where SLT is decomposed into two tasks of *i*) sign-to-spoken language conversion where source (German in our case) signs are converted to the source tokens. Then we have *ii*) gloss sequence prediction that provides additional annotations to facilitate the SLT process. In case of the unavailability of gloss annotations, a complementary second task is proposed, where we translate source signs into a target language. Figure 1 illustrates the high-level design of our architecture.

As the figure shows, the decoder has three channels, namely D_{ts}, D_g, and D_{tr} for transcribing input frames and generating gloss tokens and translation, respectively. Each of the channels is structurally the same as a Transformer decoder layer. All these channels share parameters of their first n blocks with each other. This feature helps us control the memory footprint of our model. Moreover, exchanging information between channels yields richer internal representations. In addition to those n blocks, each of D_g and D_{tr} has one additional block whose parameters are *not* shared. Therefore, both D_g and D_{tr} have $n+1$ and D_{ts} has n blocks. Dedicated blocks are designed to reach better performance and mitigate the complexity of multitasking. *It is to be noted that the best-performing architecture does not train D_g and D_{tr} simultaneously. Also, we only train D_{tr} to facilitate our complementary translation task, when we can not train D_g due to the unavailability of gloss sequences.* The following sections describe the encoding and decoding process of our proposed model.

A. Encoding Sign Videos

The encoder takes a sign-video V as its input. We segment V into frames $[f_1,\ f_2,...,f_F]$, then each frame is spatially embedded using a particular Inception network [22] which is pre-trained and fine-tuned convolutional model for the SLR purposes [23]. Intermediate embeddings generated by the convolutional module are then passed through *batch normalization* and *rectified linear units* [24] in order to enrich internal representations. The impact of these units and how they boost the test-time performance are comprehensively discussed in [4].

Transformers are non-recurrent networks, so in order to maintain the temporal order of frames we augment embeddings with position information, as shown in Equation 1:

$$\begin{aligned} I_t &= \text{CNN}(f_t) \\ \hat{I}_t &= I_t + \text{PosEmb}(t) \end{aligned} \tag{1}$$

where CNN(.) refers to the convolutional model and PosEmb(t) is the embedding correlates with the *t*-th time step. This process is identical to positional encoding proposed by [5]. $\hat{I}_t$ is an intermediate representation that consists of intra-frame spacial and inter-frame positional information. Each processed frame $\hat{I}_t$ is passed through multiple encoder blocks and is transformed into an output vector z_t, as shown in Equation 2:

$$z_t = \text{Encoder}(\hat{I}_t) \tag{2}$$

[1] We assume that the reader is familiar with Transformers so we skip related details.

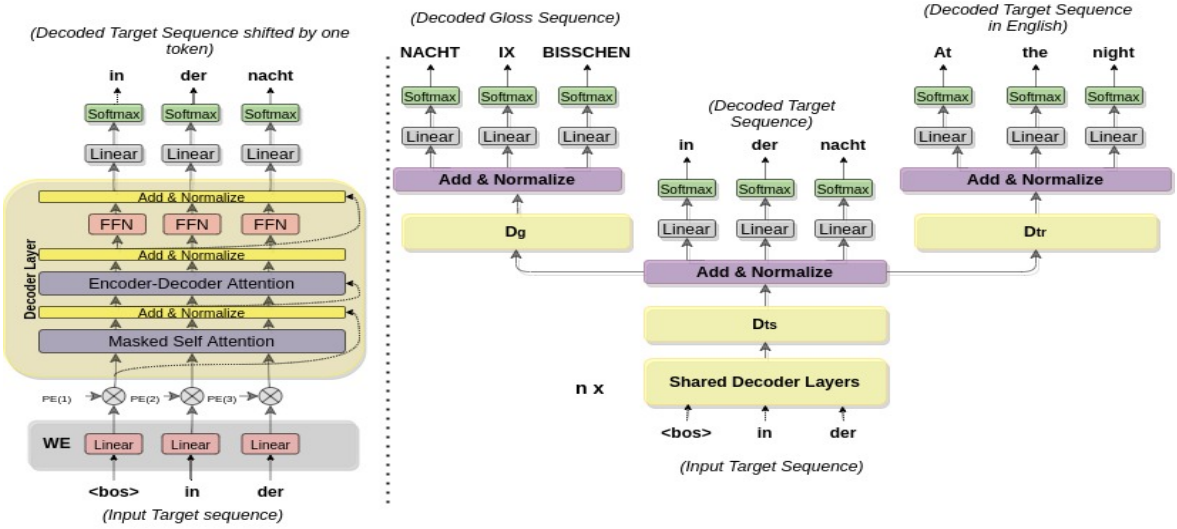


Fig. 1. **Left:** The architecture of an ordinary transformer decoder. **Right:** The architecture of the proposed model relies on a triple-channel decoder. D_{tr} and D_g denote two dedicated decoder blocks for translating input sequences into the target language (English) and gloss sequences, respectively. Aside from these two channels, there is a third one, namely D_{ts}, which transcribes the input and generates real German words. The backbone of D_{tr} and D_g channels are shared and they only differ in the last block, i.e. the first **n** blocks but the last dedicated ones are shared in between channels. Therefore, each of D_{tr} and D_g have n shared and 1 dedicated blocks. D_{ts} has only n blocks with no additional, dedicated block and all its n blocks share parameters with other channels. **The encoder part follows the same architecture as [4].**

1) Enriching Encoder Representations: Our Encoder serves as a strong, multi-channel decoder so it is supposed to provide as rich information as possible. In our experiments, we realized that only encoding sign videos is not sufficient enough and we need a more explicit way of teaching the encoder about its role and the form of representations it should learn. To this end, we tried to inject gloss-level information by forcing the encoder to generate gloss labels in addition to its main task. In other words, we treat the encoder as a sequence labeller to solve the $P(G|V)$ problem, with G being a sequence of glosses. The encoder consumes video frames and it generates which glosses are related to those frames. This is an ordinary sequence-to-sequence problem which can be solved via an ordinary loss function such as cross-entropy. However, framing the problem that way requires an accurately-labelled dataset, which is not practical in our setting. Instead, we use the CTC loss which provides weaker supervision but satisfies our needs.

The log-likelihood of a gloss sequence given the input frames can be computed as shown in Equation 3:

$$\log p_\theta(G|V) = \log \sum_{a \in \beta^{-1}(G)} p_\theta(a|V) \tag{3}$$

where θ is a set of all encoder parameters and $\beta(G)$ returns all the possible alignments. For more details about the fundamentals of CTC and gloss-frame alignments, see [20] and [4], respectively. Computing $p_\theta(G|V)$ is intractable, and so the summation in the equation can be simplified as in Equation 4:

$$p_\theta(a|V) = \prod_i p(a_i|V;\theta) \tag{4}$$

where frame-level gloss probabilities are directly obtained from the encoder, which is connected to a *Softmax* function through a projection layer in our architecture.

B. Multi-Channel Decoding

Our decoder is essentially a Transformer-based sequence generator and follows the same structure as other ordinary decoders [5]. Therefore, it is a stack of Transformer blocks with all the positional encoding, masking, self-attention, encoder-decoder attention, position-wise feed-forward, and layer-norm components. We are also faithful to the original configuration of these components.

Although the main skeleton of our decoder relies on Transformers, ours has multiple output channels instead of one. The first channel D_{ts} transforms the video information to source-side words and can be used as a transcriber. Essentially, D_{ts} is used to generate German sentences corresponding to source sign videos. Finally, the second channel denoted by D_g decodes the gloss sequences. These channels exchange information among each other through shared parameters and this helps the decoder be aware of the target language, source language, and auxiliary annotations about the input frames at the same time, and we show empirically this is the main origin of our model's superiority. A natural question arises if the gloss sequences are unavailable, our proposed model is essentially a transformer architecture which cannot exploit gloss sequences both on the encoder and decoder sides. In that case, the second channel of the decoder, D_g is useless. To alleviate this issue, we use a separate channel D_{tr} in place of D_g which is to be used for the generation of target tokens corresponding to the input video frames in another language other than the language in which D_{ts} is trained on. (for our dataset, we generate sentences in English via D_{tr}, which are machine translated from the available German sentences).

We follow the structure as shown in Figure 1 to implement our decoder. Each channel of the decoder is trained by computing the cross-entropy loss of its generated tokens, as shown in Equation 5:

$$\begin{aligned} \mathcal{L}_{CH} &= 1 - \prod_{t=1}^{T} \sum_{l=1}^{L_{CH}} p(\hat{w}_t^l) p(w_t^l | s_t) \\ CH &= \{D_{tr}, D_{ts}, D_g\} \end{aligned} \tag{5}$$

where w_t^l denotes the probability distribution of the *l*-th target token at time step *t* whose ground-truth label is provided by $\hat{w_t^l}$. Each channel generates a different token, e.g. w is a target-language token for D_{tr}, whereas D_g works with glosses. L_{CH} shows the length of the vocabulary side that each channel works with. s_t is the internal state of the decoder which is computed as shown in Equation 6:

$$s_t = \text{Decoder}(w_{t-1} | w_{1:t-1}, z_{1:F}) \tag{6}$$

As the equation shows, the generation of each token at a particular time step is conditioned on all the previously generated target words ($w_{1:t-1}$) as well as the encoder's outputs ($z_{1:F}$) for the input video segment.

According to Equation 5, each channel has a dedicated loss. We also define an auxiliary loss for the encoder ($\mathcal{L}_{enc}$). Therefore, the final loss for training our model is a composition of four loss terms, as shown in Equation 7:

$$\mathcal{L} = \lambda_{tr}\mathcal{L}_{D_{tr}} + \lambda_{ts}\mathcal{L}_{D_{ts}} + \lambda_g\mathcal{L}_{D_g} + \lambda_{enc}\mathcal{L}_{enc} \tag{7}$$

λ assigned to each loss is a weight to control the contribution of each loss to the translation process.

IV. Experimental Study

A. Datasets

To train our models and in the interest of fair comparisons, we selected the `RWTH-PHOENIX-Weather 2014 T` dataset[2] [25]. It contains the sign language videos along with their gloss annotations and translations in German.

To train our proposed model in the unavailability of the gloss sequences, we extend their train set by translating German spoken language sentences into English. For translation, we make use of the NMT system developed as a *WMT-19* submission by [26][3]. We provide an example from our training set in Table I.

Firstly, we normalize punctuation & tokenize our target side of the dataset. Following tokenization, we use Byte Pair Encoding Scheme (BPE) [27], as currently used by almost all state-of-the-art NMT systems, to pre-process the target side of our dataset. This solves the problem of out-of-vocabulary (OOV) words in the test set as BPE encodes unknown words as a sequence of sub-words.

B. Experimental Setup

With a specific set of model hyper-parameters, we perform all the experiments. We did not use any specific hyper-parameter optimizer for finding the optimal set of hyper-parameters. The following set of hyper-parameters is chosen.

We use $batch\_size = 32$, $num\_enc = 3$, $num\_dec = 3$, $\lambda_{enc} = 5.0$, $\lambda_g = 0.7$ and $num\_attention\_heads = 8$) to train and test our models. For all the experiments, we set $\lambda_{ts} = 1.0$. Note that the best-performing model setup assigns $\lambda_{tr} = 0$ in the availability of gloss sequence (ref. Table III). Adam [28] is used as the optimizer to train the models with an initial learning rate of 10^{-3} (β1=0.9, β2=0.998) and a weight decay of 10^{-3}. We use *plateau learning rate scheduler* which tracks the development set performance. We evaluate our model on the development set after every 200 iteration of training steps and if the BLEU-4 score ([29]) does not increase for 15 evaluation steps, the learning rate is reduced by a factor of 0.7 until it reaches 10^{-7}, after which the training is stopped. While testing our proposed model, we use beam search to decode the target tokens with a fixed beam width of 5.

Our model has a performance score of 22.4 ± 0.2 BLEU-4 over a range of choices for λ_g (c.f. Table V; with fixed no of the encoder and decoder layers of 3, $\lambda_{enc} = 5.0$, $\lambda_{tr} = 0$, $\lambda_{ts} = 1.0$). Though we report only the best performance score of 22.59 BLEU-4 score, the lowest performance of 22.2 is still better than the state-of-the-art score proposed in [4], which is 21.32.

C. Baseline Models

We design two baseline models for our experiments. The design decision is based on the premise that we do not use any gloss-level supervision while training the baseline models. This entails having a fair comparison with our proposed architecture which uses gloss-level annotations for training.

1) Ordinary Transformer Network: We train an ordinary transformer model by setting hyper-parameters associated with the joint loss term (see equation 7) λ_{enc}, λ_g, λ_{tr} to zero. It alleviates any gloss-level supervision and our triple-channel decoder works as a single decoder which directly decodes German spoken language sentences from the sign language videos. This model has the poorest performance of a 20.52 BLEU-4 score.

2) Separate Decoder Networks (D_{SEP} and D_{SEP++}): Instead of our proposed model, which is equipped with multitasking by exploiting a shared decoder representation via D_g, baseline model D_{SEP} has two separate decoders which do not share any information with each other. In Figure 2, Dec_T and Dec_G refer to two separate decoders which use the same encoder representation to predict the target sequence and gloss sequence from the input sequence, respectively. Dec_T and Dec_G are respectively complementary to that of D_{ts} and D_g in our proposed model. As the decoders in this architecture (D_{SEP}) do not share any previous decoder layers as our proposed architecture does, this baseline model suffers from weak supervision of gloss annotations and thus performs somewhat poorly (BLEU-4 score of 20.90) compared to our

[2] Link: `RWTH-PHOENIX-Weather 2014 T`

[3] WMT19 Fairseq

TABLE I
AN EXAMPLE FROM THE RWTH-PHOENIX-WEATHER 2014 T DATASET USED FOR TRAINING.

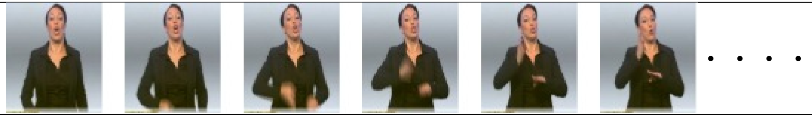
Gloss	NORDWEST HEUTE NACHT TROCKEN BLEIBEN SUEDWEST KOENNEN REGEN ORT GEWITTER DAZU
Text	im nordwesten bleibt es heute nacht meist trocken sonst muss mit teilweise kräftigen schauern gerechnet werden örtlich mit blitz und donner
Signer	Signer08
Name	train/11August_2010_Wednesday_tagesschau-5
Sign Video	
English Translation	In the northwest, it will remain mostly dry tonight, with some heavy showers expected with thunder and lightning

TABLE II
COMPARISON BETWEEN BASELINE MODELS. *For more about our baseline models, see section IV-C*

Tasks	**DEV**				**TEST**			
Sign to Text w/o gloss supervision	BLEU-1	BLEU-2	BLEU-3	BLEU-4	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Sign2Text [4]	45.54	32.60	25.30	20.69	45.34	32.31	24.83	20.17
Sign2Text (with label smoothing)	45.43	32.67	25.38	20.74	45.45	32.68	25.24	20.52
D_{SEP}	44.52	31.96	25.00	20.58	45.17	32.82	25.45	20.90
$D_{SEP}++$	**46.55**	**34.08**	**26.50**	**21.63**	**46.56**	**34.04**	**26.39**	**21.59**

TABLE III
COMPARISON BETWEEN STATE-OF-THE-ART AND OUR MODEL. HERE, $T^{\lambda_{enc}}_{\lambda_g,\lambda_{tr}}$ DENOTES OUR PROPOSED ARCHITECTURE. FOR DIFFERENT VALUES OF λ_{enc}, λ_g AND λ_{tr}, WE TABULATE THEIR EFFECTS ON TEST BLEU-4 SCORE. WE SHOW THREE OF OUR BEST RESULTS AND TABULATE THEM ACCORDINGLY. *For all experiments, we set* $\lambda_{ts} = 1.0$. $T^{5.0}_{0.0,0.0}$ REFERS TO OUR RE-IMPLEMENTATION OF STATE-OF-THE-ART ARCHITECTURE [4] WITH THE SAME TRAINING SETTING DESCRIBED IN [4]

Tasks	**DEV**				**TEST**			
Sign to Gloss to Text	BLEU-1	BLEU-2	BLEU-3	BLEU-4	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Sign2Gloss2Text [4]	47.73	34.82	27.11	22.11	48.47	35.35	27.57	22.45
Sign2Gloss → Gloss2Text [4]	47.84	34.65	26.88	21.84	47.74	34.37	26.55	21.59
End-to-End Sign to (Gloss+Text)	BLEU-1	BLEU-2	BLEU-3	BLEU-4	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Recog. Sign2(Gloss+Text) [4]	46.56	34.03	26.83	22.12	47.20	34.46	26.75	21.80
Trans. Sign2(Gloss+Text) [4]	47.26	34.40	27.05	22.38	46.61	33.73	26.19	21.32
$T^{5.0}_{0.0,0.0}$ **[4]**	45.03	32.31	24.92	20.26	46.66	33.20	25.81	21.07
$T^{5.0}_{0.7,0.0}$ **(Proposed Model)**	**48.01**	**35.46**	**27.94**	**23.05**	**47.59**	**35.16**	**27.60**	**22.59**

proposed model with $\lambda_{enc} = 0$ (BLEU-4 score of 21.08). When equipped with encoder side gloss sequence decoding, by setting $\lambda_{enc} = 5.0$, the performance of D_{SEP} is increased to 21.59. We call this enhanced D_{SEP} as D_{SEP++}

D. Results & Comparisons

Sign to Text tasks with gloss supervision can be divided into two parts, namely ***Sign2Gloss2Text*** and ***Sign2 (Gloss+Text)***. We discuss these briefly and compare them with our proposed method.

1) Models with mid-level gloss supervision: Sign2Gloss2Text uses intermediate gloss level representation. It is a two-step process. The first step uses a CSLR (Continuous sign language recognition) model to generate the gloss sequences corresponding to a sign video. In the second step, the generated glosses are fed to train an NMT model which acts as a *Gloss2Text* translator, translating gloss sequences into a sequence of spoken language words. A variation of *Sign2Gloss2Text* is known as **Sign2Gloss → Gloss2Text**. This is similar to *Sign2Gloss2Text*, but instead uses the best performing *Gloss2Text* network instead of training it from the scratch. For both of these architectures, we list the state-of-the-art scores in Table III.

2) End-to-End models: The second category of tasks (*Sign2(Gloss+Text)*) essentially refers to learning both the gloss sequences and textual representations jointly, as done in [4]. Our model is an extension of the approach used in [4]. Table III shows that our model with best-performing setup obtains a BLEU-4 score of 22.59, which is 0.79 absolute increase from the score of 21.80 obtained by [4] for *Sign2(Gloss+Text)* tasks. The improvement was found to be statistically significant over the prior state-of-the-art using bootstrap hypothesis testing [4] to test the Null Hypothesis (H_0) that the same system generated the two hypothesis translations, using the technique utilized in [4] and our proposed method. *At 95% confidence level, P-Value comes out as* 0.029. *This entails that* H_0 *can be rejected, subsequently firming the claim that our method is better than the existing state-of-the-art.*

[4] Bootstrap Hypothesis Testing

TABLE IV

SOME EXAMPLES FROM OUR BEST PERFORMING MODEL ($T^{5.0}_{0.7,0.0}$) AND ITS COMPARISON WITH OUR RE-IMPLEMENTATION OF THE STATE-OF-THE-ART MODEL ($T^{5.0}_{0.0,0.0}$) FROM [4]. *Reference* SHOWS THE GROUND TRUTH REFERENCE TRANSLATIONS.

Ours	**die gewitter fällt dann nur im westen und nordwesten können sich die mitunter unwetterartig ausfallen können .** *(The thunderstorm then only falls in the west and northwest, which can sometimes turn out like a storm.)*
SOTA	**sonst kaum noch schauer und gewitter nur im westen auch mal längere zeit klar .** *(otherwise hardly any showers and thunderstorms only clear in the west for a long time.)*
Reference	**zunächst ist das gewitterrisiko nur im westen erhöht von tag zu tag steigt es aber auch richtung osten .** *(Initially, the risk of thunderstorms is only increased in the west, but it also increases in the east from day to day.)*
Ours	**orgen vormittag wird es dann immer noch schauer im osten und auch kräftig regnen in brandenburg gibt es noch einzelne schauer .** *(Tomorrow morning there will still be showers in the east and there will also be heavy rain in Brandenburg.)*
SOTA	**morgen vormittag regnet es dann zwischen dreizehn im osten dagegen nur dreizehn und südosten haben wir noch kräftig sein .** *(Tomorrow morning it will rain between thirteen in the east on the other hand only thirteen and in the southeast we still have to be strong.)*
Reference	**morgen vormittag bleibt von dem ganzen starken regen noch hier im osten einige gewitter hängen in brandenburg starker regen in vorpommern und an der ostsee .** *(Tomorrow morning from all the heavy rain there will still be some thunderstorms hanging here in the east in Brandenburg, heavy rain in Western Pomerania and on the Baltic Sea.)*
Ours	**hoher luftdruck bestimmt unser wetter .** *(high air pressure determines our weather.)*
SOTA	**im norden machen sich währenddessen tiefausläufer bemerkbar der in der neuen woche .** *(In the north, meanwhile, low levels are noticeable in the new week.)*
Reference	**das hoch das sich richtung osteuropa verlagert bestimmt auch in den kommenden tagen unser wetter .** *(The high that is moving towards Eastern Europe will definitely change our weather in the coming days.)*
Ours	**der wind weht schwach bis mäßig .** *(the wind blows weak to moderate.)*
SOTA	**auch sonst weht der wind schwach bis mäßig .** *(otherwise the wind blows weak to moderate.)*
Reference	**in deutschland gibt es nur schwache luftdruckunterschiede .** *(in germany there are only slight differences in air pressure.)*
Ours	**am freitag wechselhaftes wetter .** *(changeable weather on friday.)*
SOTA	**am freitag hier und da etwas regen .** *(A little rain here and there on Friday.)*
Reference	**am freitag wechselhaftes schauerwetter .** *(changeable rainy weather on friday.)*

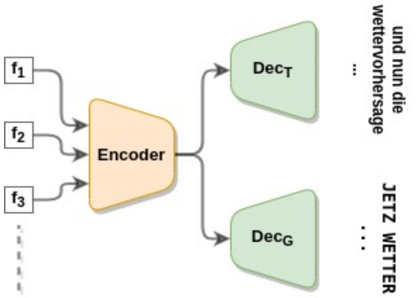


Fig. 2. Architecture of our baseline model with one encoder and two separate decoders. Here, f_1, f_2,......,f_n are the spatial representation of the video frames obtained from the pre-trained CNN. Dec_T and Dec_G denote two separate decoders for decoding text and gloss sequence, respectively. For Sign2Text experiments, we drop Dec_G.

TABLE V

COMPARISON BETWEEN PROPOSED MODELS WITH DIFFERENT LOSS WEIGHTS. $T^{\lambda_{enc}}_{\lambda_g,\lambda_{tr}}$ DENOTES OUR PROPOSED ARCHITECTURE.

Models	**METRICS**	
Models for Sign2(Gloss+Text)	BLEU-4	ROUGE
$T^{0.0}_{0.0,0.0}$	20.52	45.92
$T^{0.0}_{0.0,0.5}$	20.79	47.03
D_{SEP}	20.90	46.41
$T^{5.0}_{0.0,0.0}$	21.07	46.00
$T^{0.0}_{0.7,0.0}$	21.08	46.06
$D_{SEP}++$	21.59	47.69
$T^{5.0}_{0.7,0.2}$	22.05	48.25
$T^{5.0}_{0.7,0.0}$	**22.59**	**48.82**

E. Ablation experiments

The performance of our proposed architecture depends on the choice of the weights (λ_{enc}, λ_{tr}, λ_g) associated with the loss term (7) used to train our model. We perform an ablation study to show the effect of hyper-parameter variations. Firstly, we consider our baseline models and consider how their performance changes if the gloss-level supervision at the encoder side (c.f III-A1) is added. Secondly, we consider our proposed model and compare it with their baseline counterparts.

We can conclude the following based on Table V. The model without any gloss-level supervision ($T^{0.0}_{0.0,0.0}$) has the lowest BLEU-4 score of 20.52. Gloss-level supervision using a separate decoder network (D_{SEP}) boosts the baseline accuracy from 20.52 to 20.90. Training $D_{SEP}++$ which uses the architecture from D_{SEP} along with an added objective of enriching encoder representation (refer to Section III-A1) could subsequently increase the performance of D_{SEP} from 20.90 to 21.59. Following this increasing trend of performance we hypothesize that adding gloss-level supervision, both at the encoder and decoder side, is the most useful multitasking approach to follow.

We follow the previous experiments using our proposed model. Our baseline D_{SEP} uses extra supervision from gloss sequences employing two separate decoders, implementing a soft parameter-sharing paradigm for multitasking. For a fair comparison with D_{SEP}, we run our proposed model which implements a hard parameter sharing paradigm of multi-

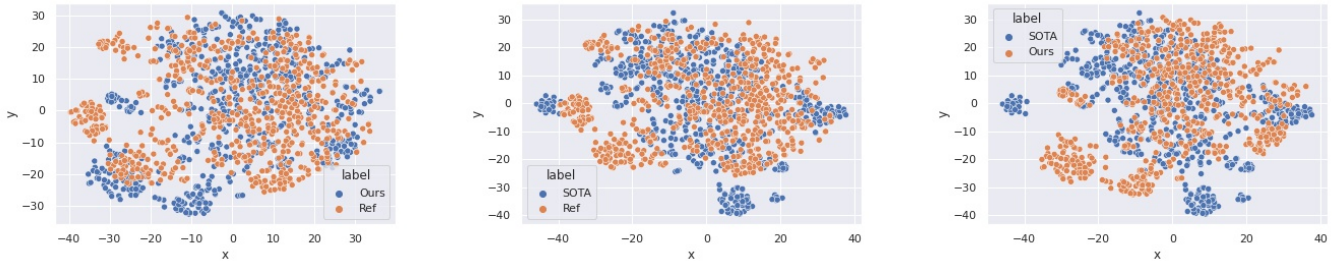


Fig. 3. T-Sne Visualization of the embedding of the generated translations using our model and state-of-the-art model and their comparison with ground truth reference text. Note that each dot represents a sentence as a two-dimensional projection. For visualization, we embedded candidate sentences using Multilingual Universal Sentence Encoder. 2-dimensional projection is done using T-SNE. For a proper understanding of what each of the sub-figures refers to, visit Section V

tasking ($T^{0.0}_{0.7,0.0}$). This uses a shared backbone of n layers of the decoder and 2 task-specific decoder layers. It boosts up the performance of D_{SEP} from 20.90 to 21.08, subsequently showing that *using representation from shared layers could boost multitasking performance when compared to separately obtained representations.* $T^{5.0}_{0.7,0.0}$ denotes our proposed model with an added objective of training the encoder with an auxiliary loss $\mathcal{L}_{enc}$, thereby setting $\lambda_{enc} = 5.0$. It gives a huge boost in terms of the BLEU-4 score. This achieves the new state-of-the-art score of 22.59, with an impressive ROUGE score of 48.82.

Note that our re-implementation of the state-of-the-art [4] ($T^{5.0}_{0.0,0.0}$) and our proposed model with a decoder-only multi-tasking ($T^{0.0}_{0.7,0.0}$) have the similar performance, thereby firming our belief that exploiting gloss sequence in the target side is as useful as it is for the source side. Though our dual channel decoder has a dedicated channel (D_{tr}) for German to English translation, training it with D_g and D_{ts} harms the overall performance (by setting $\lambda_{tr} = 0.2$)[5]. When gloss annotations are unavailable, we can use German-to-English translation as a proxy task to improve the baseline performance. It is facilitated by only training two channels, D_{tr} and D_{ts}. $T^{0.0}_{0.0,0.5}$ surpasses the performance of the baseline *Sign2Text* model (+0.27 and +1.11 improvement in BLEU-4 & ROUGE scores).

The marginal improvement could be attributed to noisy machine-translated data used to train D_{tr}.

F. Human Evaluation

We appointed two in-house annotators to manually score 100 randomly chosen translations from our proposed model and SOTA model [4] across both Fluency and Adequacy (both rated between 1 or 2 or 3). Fluency refers to how grammatically accurate the generated sentences are and Adequacy refers to the meaning preservation of the generated sentence with respect to the reference sentence. We obtain the fluency of the proposed model's output to be 2.66 which is higher than the fluency of SOTA at 2.5. Similarly, our proposed model also beats SOTA in terms of adequacy. We obtain an average adequacy score of 2.16 for our proposed model, higher than the 1.83 adequacy score obtained by the SOTA model. Note that these statistics are averaged across annotations given by the two annotators across the randomly chosen 100 sentences.

V. T-SNE Visualization

To visualize how our proposed method improves over proposed state-of-the-art by [4], we perform a t-sne [30] Visualization of the embeddings obtained by projecting translated sentences via Multilingual Universal Sentence Encoder (MUSE) [31]. In the leftmost part of Figure 3, we illustrate the overlap between the translation embedding obtained from our model to that of the reference translation embedding. In the middle part, we show an overlap between translation embedding obtained from the State-of-the-art method by [4] to that of the reference. In the rightmost part, we show an overlap between translation embedding obtained from our proposed method by translation embedding.

We observe that embeddings obtained from our model translations are highly dispersed and fit the reference translation embeddings well compared to the state-of-the-art. Thus, visually, we can infer the better quality of translations obtained from our model.

VI. Conclusion

In this paper, we have proposed a transformer-based novel architecture to perform the task of CSLR and SLT in an end-to-end fashion. The findings of this research can be summarized below:

- Exploiting intermediate sequences in an end-to-end fashion (e.g. gloss sequences) can be an effective approach to training the SLT models.
- If the gloss sequences are available, we can use related tasks as a proxy for improving the performance of the baseline model and we hypothesize that the task design is important.

As our approach is both model and task agnostic, extending our approach to other language understanding (NLU) tasks using various deep learning architectures is a promising research direction and in future, we would like to explore that direction.

Acknowledgement

This paper is an outcome of the project entitled **"NLTM: VIDYAPAATI"** sponsored by Electronics and IT, Ministry of

[5] For comparison, see BLEU-4 score of $T^{5.0}_{0.7,0.2}$ and $T^{5.0}_{0.7,0.0}$ in Table V

Electronics and Information Technology (MeitY), Government of India. We acknowledge anonymous reviewers for their useful feedback. We would also like to thank Peyman Passban, who helped in the development of the methodology presented in this paper.

References

[1] J. P. van Hemert, "Automatic segmentation of speech," *IEEE Transactions on Signal Processing*, vol. 39, no. 4, pp. 1008–1012, 1991.

[2] P. Santemiz, O. Aran, M. Saraclar, and L. Akarun, "Automatic sign segmentation from continuous signing via multiple sequence alignment," in *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pp. 2001–2008, IEEE, 2009.

[3] S. Khan, D. G. Bailey, and G. S. Gupta, "Pause detection in continuous sign language," *International journal of computer applications in technology*, vol. 50, no. 1-2, pp. 75–83, 2014.

[4] N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden, "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*, pp. 10023–10033, 2020.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[6] R. Caruana, "Multitask learning: A knowledge-based source of inductive bias," in *Proceedings of the Tenth International Conference on Machine Learning*, pp. 41–48, Morgan Kaufmann, 1993.

[7] J. Bungeroth and H. Ney, "Statistical sign language translation," in *Workshop on representation and processing of sign languages, LREC*, vol. 4, pp. 105–108, Citeseer, 2004.

[8] X. Chai, G. Li, Y. Lin, Z. Xu, Y. Tang, X. Chen, and M. Zhou, "Sign language recognition and translation with kinect," in *IEEE Conf. on AFGR*, vol. 655, p. 4, 2013.

[9] T. Hanke, L. König, S. Wagner, and S. Matthes, "Dgs corpus & dicta-sign: The hamburg studio setup," in *4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies (CSLT 2010), Valletta, Malta*, pp. 106–110, 2010.

[10] A. Schembri, J. Fenlon, R. Rentelis, S. Reynolds, and K. Cormier, "Building the british sign language corpus," *Language Documentation & Conservation*, vol. 7, pp. 136–154, 2013.

[11] H. Cooper and R. Bowden, "Learning signs from subtitles: A weakly supervised approach to sign language recognition," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2568–2574, IEEE, 2009.

[12] O. Koller, S. Zargaran, and H. Ney, "Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent cnn-hmms," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4297–4305, 2017.

[13] O. Koller, H. Ney, and R. Bowden, "Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3793–3802, 2016.

[14] J. Forster, C. Schmidt, T. Hoyoux, O. Koller, U. Zelle, J. H. Piater, and H. Ney, "Rwth-phoenix-weather: A large vocabulary sign language recognition and translation corpus.," in *LREC*, vol. 9, pp. 3785–3789, 2012.

[15] J. Forster, C. Schmidt, O. Koller, M. Bellgardt, and H. Ney, "Extensions of the sign language recognition and translation corpus rwth-phoenix-weather.," in *LREC*, pp. 1911–1916, 2014.

[16] S.-K. Ko, C. J. Kim, H. Jung, and C. Cho, "Neural sign language translation based on human keypoint estimation," *Applied Sciences*, vol. 9, no. 13, p. 2683, 2019.

[17] N. C. Camgoz, S. Hadfield, O. Koller, and R. Bowden, "Subunets: End-to-end hand shape and continuous sign language recognition," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 3075–3084, IEEE, 2017.

[18] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," *arXiv preprint arXiv:1705.03122*, 2017.

[19] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

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

[21] Y. Chen, R. Zuo, F. Wei, Y. Wu, S. Liu, and B. Mak, "Two-stream network for sign language recognition and translation," 2023.

[22] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," 2016.

[23] O. Koller, C. Camgoz, H. Ney, and R. Bowden, "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*, 2019.

[24] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines.," in *ICML* (J. Fürnkranz and T. Joachims, eds.), pp. 807–814, Omnipress, 2010.

[25] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Neural sign language translation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[26] N. Ng, K. Yee, A. Baevski, M. Ott, M. Auli, and S. Edunov, "Facebook FAIR's WMT19 news translation task submission," in *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, (Florence, Italy), pp. 314–319, Association for Computational Linguistics, Aug. 2019.

[27] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Berlin, Germany), pp. 1715–1725, Association for Computational Linguistics, Aug. 2016.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

[29] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, (Philadelphia, Pennsylvania, USA), pp. 311–318, Association for Computational Linguistics, July 2002.

[30] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.

[31] G. Lample, A. Conneau, L. Denoyer, and M. Ranzato, "Unsupervised machine translation using monolingual corpora only," *arXiv preprint arXiv:1711.00043*, 2017.