SignNet II: A Transformer-Based Two-Way Sign Language Translation Model

Lipisha Chaudhary, Tejaswini Ananthanarayana, Enjamamul Hoq, and Ifeoma Nwogu, *Senior Member, IEEE*

Abstract—The role of a sign interpreting agent is to bridge the communication gap between the hearing-only and Deaf or Hard of Hearing communities by translating both from sign language to text and from text to sign language. Until now, much of the AI work in automated sign language processing has focused primarily on sign language to text translation, which puts the advantage mainly on the side of hearing individuals. In this article, we describe advances in sign language processing based on transformer networks. Specifically, we introduce SignNet II, a sign language processing architecture, a promising step towards facilitating two-way sign language communication. It is comprised of sign-to-text and text-to-sign networks jointly trained using a dual learning mechanism. Furthermore, by exploiting the notion of sign similarity, a metric embedding learning process is introduced to enhance the text-to-sign translation performance. Using a bank of multi-feature transformers, we analyzed several input feature representations and discovered that keypoint-based pose features consistently performed well, irrespective of the quality of the input videos. We demonstrated that the two jointly trained networks outperformed their singly-trained counterparts, showing noteworthy enhancements in BLEU-1 - BLEU-4 scores when tested on the largest available German Sign Language (GSL) benchmark dataset.

Index Terms—Sign language translations, dual learning, transformer model, metric embedded learning

1 Introduction

According to the World Health Organization, there are approximately 430 million Deaf and Hard of Hearing community (DHH) individuals around the world [26]. Sign language, a visio-spatial natural language, is the primary mode of communication for many DHH individuals. Similarly, according to the World Federation of the Deaf, there are over 200 sign languages, and around 70 million deaf people using them worldwide [39]. Interpreting sign language can be challenging for non-signers, and the inability to freely communicate to a large percentage of the population in their natural language can be challenging for DHH individuals. In addition, the lack of readily available resources to aid general sign understanding makes these issues even harder. AI research on automating Sign Language Translations (SLT) can play a critical role in bridging this communication gap between hearing-only and signing-only individuals.

With the recent successes in neural machine translation (NMT) and video-based activity recognition methods, AI researchers have begun extending these methods to SLT. However, many of these initial works only convert from sign language to text, a relatively easier AI problem to solve. This inadvertently puts the advantage mainly on the side of the hearing individuals who can receive information in their natural modality of speech (readily converted to from text). Such systems do not provide as much of an advantage for the DHH individuals, whose natural mode of communication involves receiving information in the form of signs.

A true sign language interpreting agent should be capable of understanding sign language and translating to text as well as in the reverse direction. To this end, we propose a transformer-based two-way sign language translation model, SignNet II, an initial step towards facilitating two-way sign language communication. The model exploits the notion of the duality of the sign language interpretation problem to learn from both source-to-target and target-to-source translations.

The contributions of this work include (i) the computational justification of pose points for real-life sign language understanding, (ii) the introduction of a metric embedding-based loss function to improve the text-to-sign translation, and (iii) the use of a dual learning approach to enhance both source-to-target and target-to-source translations.

In Section 2 we describe the related works in the progression of automated SLT as well as the role of transformers in this research area. We discuss the challenges of SLT, briefly introduce the standard benchmark dataset for SLT as well as our own real-life, unconstrained American Sign Language (ASL) dataset. We justify our choice of representative features in Section 3. Section 5 presents SignNet II, our proposed two-way SLT model, describing the one-way baseline models as well as their coupled learning process. Section 6 describes the training scheme along with experiments and

- *Lipisha Chaudhary, Enjamamul Hoq, and Ifeoma Nwogu are with the Department of Computer Science and Engineering, University at Buffalo, SUNY, Buffalo, NY 14260 USA. E-mail: {lipishan, inwogu}@buffalo.edu, hoq.enjamamul@gmail.com.*
- *Tejaswini Ananthanarayana is with SONY Electronics, San Diego, CA 92127 USA. E-mail: ta2184@rit.edu.*

Manuscript received 15 February 2022; revised 11 October 2022; accepted 11 December 2022. Date of publication 0 2022; date of current version 0 2022. (Corresponding author: Lipisha Chaudhary.)
Recommended for acceptance by Transformer SI Guest Editors.
Digital Object Identifier no. 10.1109/TPAMI.2022.3232389

results. Lastly, Section 7 presents our conclusions and discusses some of the limitations and next steps in transformer-based SLT research.

2 Sign Language and the Role of Transformers in its Machine Translation

In this section, we discuss the challenges of making the jump from performing NMT on written/spoken languages to involving a rich, complex visual language such as sign language. We discuss the progression of sequence models for SLT until the State of The Art (SoTA) models, including transformers.

2.1 Why is Automated SLT Challenging?

According to [20], a plethora of challenges are encountered when hearing-only individuals attempt to learn a sign language as a second language. These challenges, as described below, are also inherited when we attempt to use NMT systems designed for spoken/text-based languages for visual-spatial languages.

Typically, spoken languages are linearly one-directional, where one word occurs after another. In contrast, sign language is three-dimensional and multi-directional, i.e., two or more signs can be produced simultaneously and can interact with one other at the same time. For example, in a conversation involving a narrative about two people, Jack and Jill, the signer can place Jack in a spatial 3D position located close to the signer, called his signing space [18], on the left-hand side. Similarly, he can place Jill in a similar 3D position on the right side. The signer can thus tell the story by referring to the 3D spatial locations as Jack and Jill interact in the narrative via produced signs. Also, grammatical constructs such as past and future tenses are represented by altering the 3D pose of the signer. For example, a signer may lean forward when signing the same phrase to indicate an event in the future versus one currently occurring.

Sign production is another computational challenge since traditional NMT systems were designed to produce only text. Altering such systems to produce meaningful 3D signs successfully is not trivial and will require a visual component.

The notion of receptive finger-spelling in sign language understanding can also be exacting on the NMT system. Finger-spelling is the process of spelling out words by using handshapes that correspond to the letters in the word. The set of handshapes used to spell words is known as a "manual alphabet". Finger spellings are often used for spelling out the names of people and places or for unusual words for which there is no sign. Another challenge with sign language processing is its differing word order, grammar rules, and structure, from its spoken counterpart.

Gloss is the system of written words, symbols, and other annotations that represent how to produce signs in a given sign language. Gloss is the transcribed form of sign language, which includes various notations to account for the facial and body grammar involved in the signs. Unfortunately, not all signs have a direct meaning in the spoken equivalent. An example of a gloss in American Sign Language (ASL) and its interpretation in spoken/written English [24] is shown below:

ASL Gloss. YESTERDAY PRO-1 INDEX-[at] WORK HAPPEN SOMEONE! MAN CL:1-"walked_past_quickly" I NEVER SEE PRO-3 BEFORE.

Interpretation. Yesterday at work, a stranger (some guy I've never seen before) rushed past me.

It is important to note that DHH individuals do not use gloss in their daily lives. It is only an intermediary reporting and research tool.

2.2 Neural Machines for Continuous Sign Language Translation

Sign Language (SL) analysis often involves working either with isolated sign gestures or with continuous signs. Continuous sign language can therefore be defined as sequential unsegmented sign gestures where the start and end boundaries for each gesture is not clearly annotated. Prior to the advent of neural models, statistical machine translation (SMT) involved translating a sentence S from a source language to a target language sentence T, using statistical models learned over large corpa of examples. SMT therefore aimed to maximize $Pr(S|T)$. Although SMT models out-performed the classical MT systems, their performance was still not optimal due to the narrow focus on sentence-to-sentence translation, where the larger context was not considered. Also, SMT models were often comprised of several small sub-components that needed to be tuned separately.

Neural machine translation (NMT) involves translating sentences in the source language to the corresponding sentence in the target language using neural networks.

2.2.1 *Sequence-to-Sequence NMT Models*

Early NMT models typically consist of encoder-decoder architectures, where the encoder abstracts a sentence from a source language into a fixed-length vector, the embedding, and the decoder uses this to generate the translation in the target language. NMT models, unlike their predecessors, jointly train the encoder and decoder.

One of the earlier NMT works by [19] was comprised of probabilistic continuous sentence-level translation models. [34] used a multilayered Long Short-Term Memory (LSTM) to encode an input sequence to a fixed-length vector and then used another multilayered LSTM to decode the target sequence from the vector. This was one of the first versions of the NMT encoder-decoder architecture. [12] showed that encoder-decoder performance deteriorated with increasing input sequence length, due to the constraints imposed by the fixed-length vector. [6] addressed this limitation by introducing the attention mechanism over the decoder LSTMs. The input sentence was encoded into a sequence of vectors, and the attention mechanism adaptively selected a subset of these vectors when decoding the translation. This allowed the model to handle longer sentences. Similar works using the attention mechanism for the NMT include [25], [40].

One of the earlier works in continuous sign language translation was by [15], who introduced a hierarchical

Notation	Meaning
n	Input sequence length
x	Output sequence length
$f_1, f_2, f_3 .... f_n$	Feature vectors for each frame of the video
$o_1, o_2, o_3 .... o_n$	Output vectors from encoder for each frame of the video
h_{rep}	Latent embedding from encoder
$w_1, w_2, w_3 .... w_x$	Predicted words
$a_1, a_2, a_3 .... a_x$	Encoder-decoder attention vectors for each word
$< s >$	Start of sentence token
$< /s >$	End of sentence token

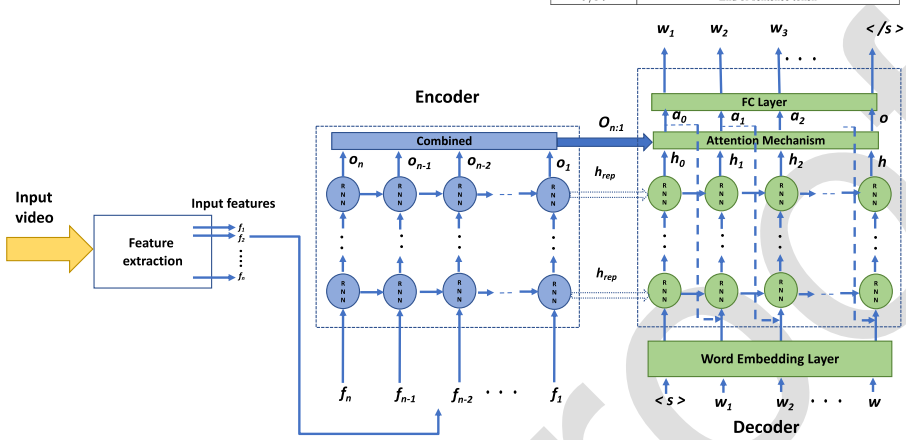

Fig. 1. A sequence-to-sequence encoder-decoder model with attention for sign language to text translation.

bidirectional deep recurrent neural network (HB-RNN) and a probabilistic framework based on Connectionist Temporal Classification (CTC) for word-level and sentence-level ASL recognition[1]. [38] introduced a hybrid model which consisted of a temporal convolution module, a bidirectional gated recurrent unit module, and a fusion module. The model was designed to capture both short-term transitions in sign videos and longer-term context transitions, with the results being fused for better performance. [13] proposed an approach that treated the sign translation problem directly as an NMT task, using sequence-to-sequence RNNs with attention. They also introduced the first continuous SLT dataset, the German RWTH-PHOENIX-Weather2014T, which now serves as the benchmark dataset for continuous sign language understanding. [28] used a 3D residual convolutional network (3D-ResNet) to extract visual features and applied CTC to learn the mapping between the sequential sign features and the gloss of the output text sentence. They also tested their results on the RWTH-PHOENIX-Weather2014T dataset. [43] used a 2-layer LSTM encoder-decoder model for Chinese sign language translation with body, hand, and facial features as the input features. [21] introduced a Korean sign language dataset and developed a multi-layer gated recurrent unit (GRU) encoder and a multi-layer GRU decoder for translating sign language videos into Korean using keypoints extracted from the face, hands, and body parts.

The sequence-to-sequence encoder-decoder model (as shown in Fig. 1) includes an encoder which reads the input sequence of vectors $\mathbf{x} = (x_1, \ldots, x_T)$ so that:

$$h_t = f(x_t, h_{t-1}) \tag{1}$$

and the encoder embedding c is given as:

$$c = q(h_1, \ldots, h_T) \tag{2}$$

1. CTC requires similar word ordering between the source and target languages, hence we refer to the task here as recognition (using Gloss) and not translation (no Gloss and differing word order)

where f and q are recursive neural functions such as LSTM. h_t is the hidden state at time t.

The decoder defines a probability over the translation $\mathbf{y}$ as:

$$p(\mathbf{y}) = \prod_{t=1}^{T} p(y_t | y_1, \ldots, y_{t-1}, c) = \prod_{t=1}^{T} g(y_{t-1}, s_t, c) \tag{3}$$

where s_t is the hidden state of the decoder RNN and g is a nonlinear function whose output is the probability of y_t.

When attention mechanism is introduced into the RNN, the context vector c_i takes all encoder hidden units $(h_1, \ldots, h_T)$ as an input to compute the probability distribution of source language words for every word the decoder wants to generate. By utilizing this mechanism, it is possible for the decoder to capture somewhat global information rather than sole inference based on one hidden state.

The context vector is given as:

$$c_i = \sum_{j=1}^{T} \alpha_{ij} h_j. \tag{4}$$

The weight α_{ij} of each input hidden unit h_j is given by:

$$\alpha_{ij} = \exp(e_{ij}) \sum_{k=1}^{T} \exp(e_{ik}) \tag{5}$$

where $e_{ij} = a(si - 1, h_j$ estimates how well the inputs at position j match with the output at position i.

2.2.2 *Transformer-Based NMT Models*

The transformer models first introduced by [37] extend the encoder-decoder attention mechanism of sequence-to-sequence models without the use of RNNs.

Given a source sequence $\mathbf{x} = x_1, \ldots, x_{|\mathbf{x}|}$ and a target sequence $\mathbf{y} = y_1, \ldots, y_{|\mathbf{y}|}$, the goal of the transformer is to

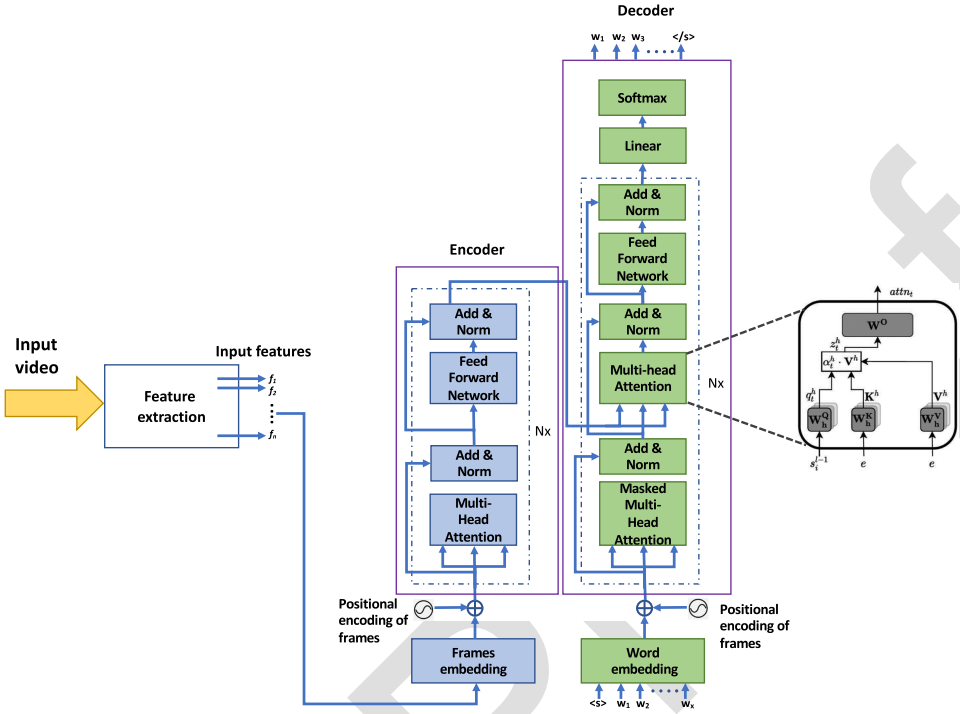

Fig. 2. A basic transformer model for sign to text translation, showing the encoder-decoder components and one of multiple attention modules expanded.

induce alignment such that:

$$P(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{|\mathbf{y}|} P(y_t|y_1, \ldots, y_{t-1}, \mathbf{x}) \tag{6}$$

The transformer consists of a stack of encoder layers, which generate context sequence representations $e = \mathbf{e}_1, \ldots, \mathbf{e}_{|\mathbf{x}|}$ of the source sequence and a stack of decoder layers.

At every time step t, the decoder uses the output from the encoder along with the token representations $\mathbf{s}_t^{l-1}$ from l, the previous layer, to compute the probability distribution over the vocabulary of the target language.

The representations of the encoder and decoder are combined in the multi-head attention mechanism (Fig. 2). Encoder embeddings are projected to keys (where the key matrix is given by $\mathbf{K}^h \in \mathbb{R}|\mathbf{x}| \times d_k$) and values (where the key matrix is given by $\mathbf{V}^h \in \mathbb{R}|\mathbf{x}| \times d_v$) in each head of the multi-head attention mechanism; d_k and d_v are the dimensions of the key and values vectors respectively.

For the decoder, the the token representation $\mathbf{s}_t^{l-1}$ is projected to a query vector $\boldsymbol{q}_t^h \in \mathbb{R}_q^d$, and d_q is the dimension of the query vector. The output for each attention head can be computed as:

$$\boldsymbol{z}_t^h = \sum_{j=1}^{|\mathbf{x}|} \boldsymbol{\alpha}_{t,j}^h \boldsymbol{v}_j^h \tag{7}$$

where

$$\boldsymbol{\alpha}_t^h = \text{softmax}\left(\frac{\boldsymbol{q}_t^h \mathbf{K}^\top}{\sqrt{d_k}}\right)$$

At every decoding step t, there is a vector of attention scores $\boldsymbol{\alpha}_t^h$. The attention matrix is the stack of attention scores for every time step. This process occurs concurrently in multiple attention heads and each head computes $\boldsymbol{z}_t^h$, which are eventually all concatenated to obtain the attention at time t. The results from the attention heads can thus be used to calculate final alignments.

Unlike the standard RNN where sequence of inputs are fed one at a time, the transformer takes all the inputs together and the order of inputs are preserved using a positional encoding parameter.

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}}) \tag{8}$$

where pos is the position of the input in the sequence of inputs and i represents embedding dimension. This notion of embedding the order of the input sequence via positional encoding revolutionized the analysis of time series data with neural machines by maintaining the position and order of input sequences (essential for the grammar of any language) and by allowing for variable length inputs.

Other earlier works that extended the standard transformer model include the Generative Pre-Traning (GPT) models - GPT-1 [29], GPT-2 [30], and GPT-3 [8]. The models initially perform unsupervised training with a transformer model using large unlabeled datasets, and then perform supervised learning for text-related tasks such as text classification, sentence similarity, question answering, next-word prediction, and text summarization.

Specifically, in sign language analysis, [9] modified the transformer model for sign language recognition and translation using CTC loss. The input features used in this model were trained on a CNN-LSTM-HMM architecture [22]

Fig. 3. Datasets: the top row shows examples of frames from the RWTH-PHOENIX-Weather2014T GSL video dataset and the bottom row shows examples from the ASLing ASL video dataset.

where gloss labels were incorporated and a hidden Markov model (HMM) was used to align the signs to their glosses. They evaluated their recognition and translation approaches on the PHOENIX14T benchmark dataset. [42] used varying numbers of layers in the transformer to perform a 2-phase sign-to-gloss and gloss-to-text translation. [23] presented a hierarchical feature learning method for input signs by feeding sign segments at multiple scales to a transformer model, thus reducing any errors caused by inaccurate sign segmentations. When evaluated on the PHOENIX14T benchmark dataset, this model outperformed other existing models where gloss was not used as an intermediary step to translation. Motivated by this performance improvements in using multi-scale input features for SLT, as well as the recent successes in multimodal fusion-based learning models ([37] and [5], [36]), [1] introduced a set of fusion-transformers to jointly encode three different scales of the input sign sequences and decode with a standard single transformer decoder. Again, when evaluated on the standard German benchmark dataset, their fusion model yielded new State of The Art (SoTA) performance. [4] presented a detailed survey of the current state of the research on transformer-based continuous sign language translation architectures, detailing the performances with different input features when tested on various sign languages - German, Chinese and American.

Lastly, [31] presented a continuous sign language generation architecture using transformers and mixture density networks. One main contribution of this work was the introduction of a counter decoding method, which allowed for continuous sequence generation and no end of sequence token was required.

3 Datasets

We perform this evaluation on two sign language datasets, (i) the RWTH-PHOENIX-Weather2014T benchmark dataset, consisting of 7096 training, 519 validation, and 642 test samples all annotated with the sign glosses, and (ii) the more realistic, daily life based American sign language dataset (ASLing) introduced by [2]. This dataset consists of 1027 training and 257 testing samples also annotated with the sign glosses. The video samples were collected at 10 frames per seconds and were annotated by 7 signers. We do not use the gloss information in any of the analyses described in this work. Fig. 3 shows examples from the two datasets.

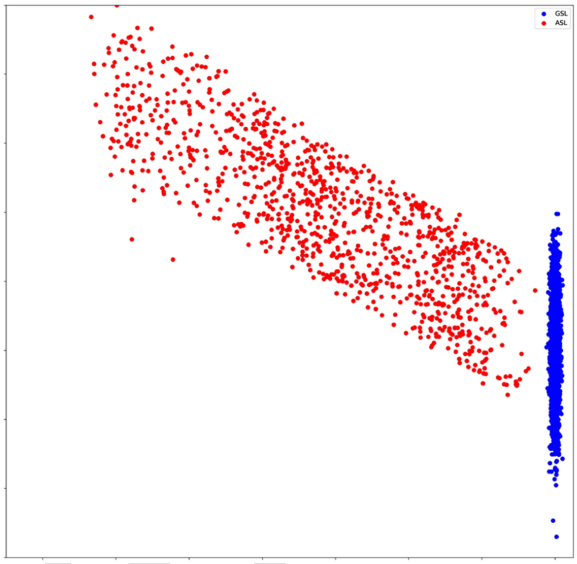

Fig. 4. Two-dimensional PCA projections of the Word2Vec embeddings of the English translations of RWTH-PHOENIX-Weather2014T dataset texts (in blue) plotted alongside the projections of the embeddings of the ASLing dataset texts (in red). (Image best viewed in color).

In the benchmark dataset, the data was collected in a constrained and controlled environment, where the signers were professional weather report interpreters on television. All signers were of the same race and dressed in dark clothing against a uniform light background. In all the signing videos, the camera position was approximately constant, in similar lighting conditions, and set to fully display the upper body including the hands and faces of the signers.

The ASLing dataset was collected from DHH college students on the Rochester Institute of Technology campus. The signers were given a basic set of instructions on how to collect data using their cell phone cameras. They were required to record the sign interpretations of the textual phrases provided in the instructions. No specific instructions were given regarding their clothing, the nature of the background, or environmental lighting conditions. They were only instructed to capture their upper body, including hands and face.

3.1 Analysis

Fig. 3 visually highlights the differences in the two datasets we consider in this evaluation. The ASLing dataset was intentionally collected in less controlled settings to more closely resemble real-life situations, where signing and hearing individuals interact in a myriad of unconstrained environments. For example, the rightmost ASL image in the bottom row of Fig. 3 shows a signer whose shadow is actively moving in the background as she records her video; there is also a lighting source in the image, creating uneven lighting conditions on the recorded video. Lastly, the subject appears closer to the camera than expected.

We first translated the ground-truth texts in RWTH-PHOENIX-Weather2014T to English, then obtained the word2vec embeddings for the two datasets and projected

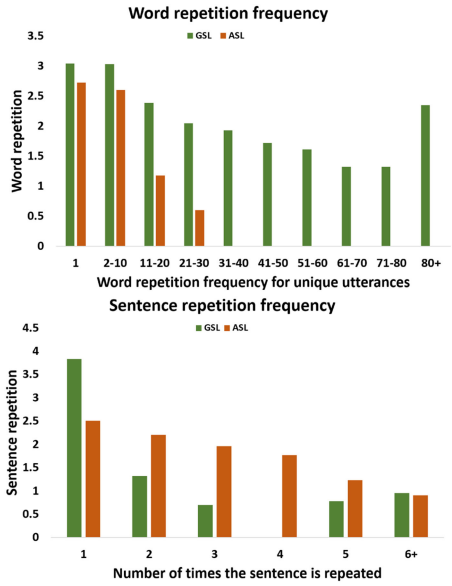

Fig. 5. Top: Word repetition frequency for unique utterances. Bottom: Sentence repetition frequency. Y-axis represents the log10 scale.

them into a 2D space using PCA. The results are shown in Fig. 4.

While the RWTH-PHOENIX-Weather2014T dataset (shown in blue), which focuses on weather-related topics, spans only a narrow region in the word embedding space, the ASLing dataset, which covers a myriad of topics, spans a significantly wider range in the embedding space. The sentence repetition and word repetition are shown in Fig. 5

Both datasets were created to serve different purposes. While the RWTH-PHOENIX-Weather2014T serves as a focused, well-controlled benchmark dataset to qualify and rate newly developed SLT algorithms, ASLing is a more naturalistic dataset, serving as a test-bed for implementations planned for deployment in real-life settings.

4 Cross-Feature Fusion Based Transformer Model

In this section, using a bank of nine cross-modal transformers as shown in Fig. 6, we develop a multimodal encoder system that embeds the interactions between three separate input features.

Three different features are served as inputs to the cross-attention block after adding the positional encoding information. These inputs are passed through a 1D convolutional network before passing to the next stage.

4.1 Methodology

We consider three commonly used visual representations from the input sign videos. The first representation is a 2048-dimensional visual embedding obtained from a CNN ResNet50 [17] architecture pre-trained on ImageNet [14]. Next, using OpenPose [10], we extract two-dimensional (x, y) key points from the input videos - 25 body keypoints, 21 hand keypoints for each hand, and 70 face keypoints resulting in a total of 137 points. Lastly, we compute the dense optical flow [33], [35] from pairs of consecutive frames in the input videos. Similar to the RGB frames, we extract a 2048 dimension vector for each optical flow frame.

We perform this experiment in an attempt to discover the most effective single visual feature useful for sign language understanding. We accomplish this by visualizing our cross-modal transformer attention weights.

If we consider one (of the three) cross-attention blocks in Fig. 6, Equation (9) describes how the attention for the two cross-feature transformer models attending on the CNN

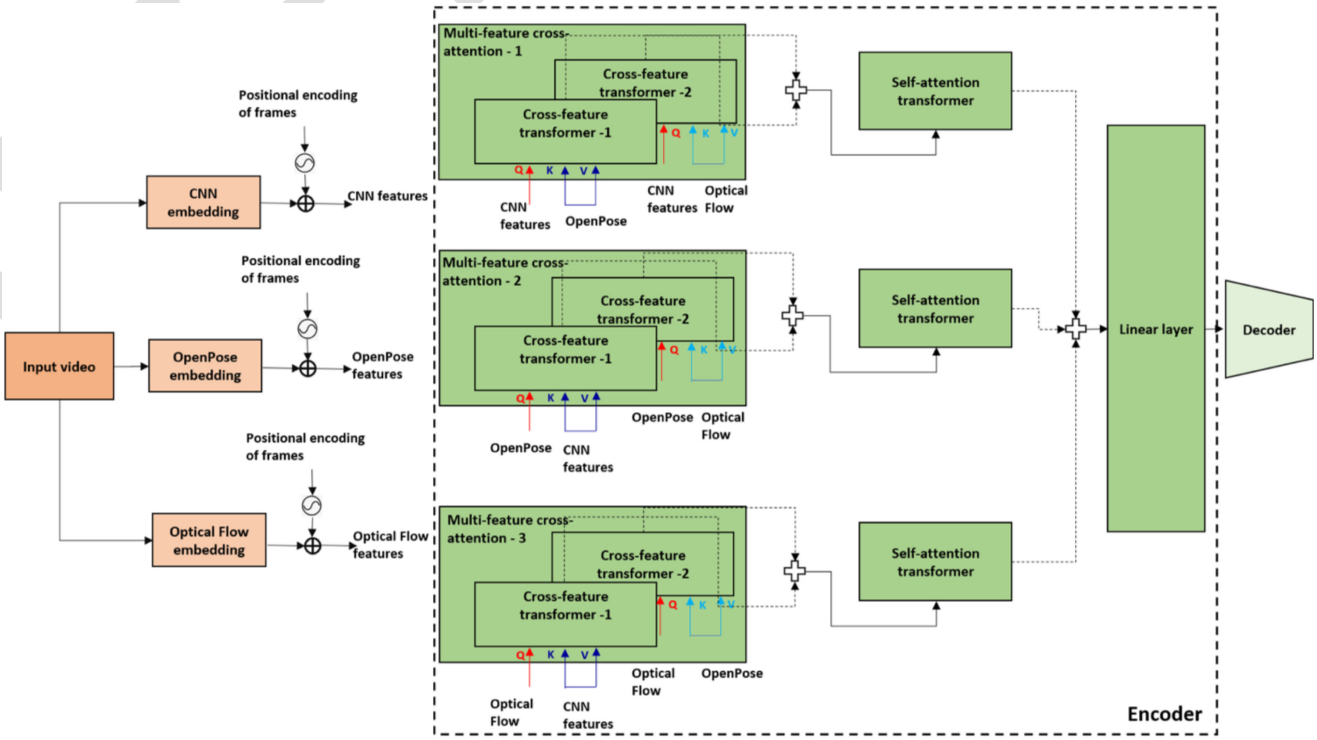

Fig. 6. Cross-modal bank of transformer encoders with single standard decoder. (Image best viewed in color).

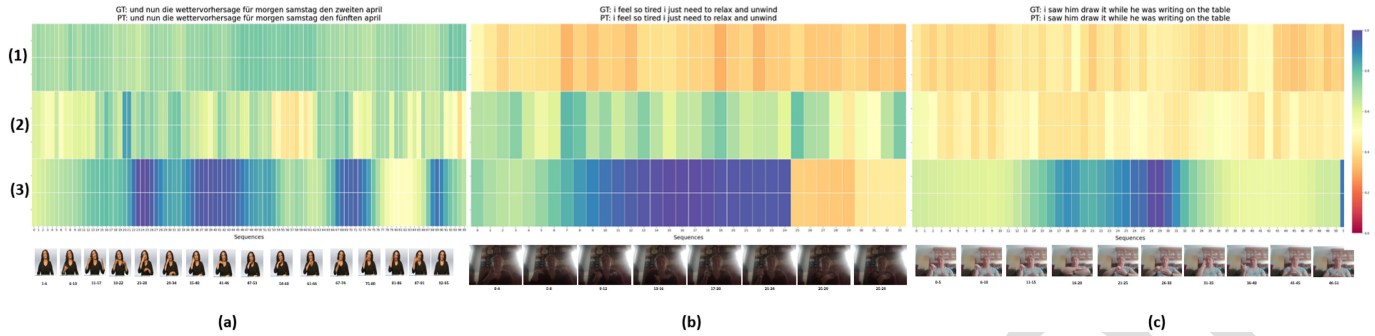

Fig. 7. Attention visualization (best viewed in color). (a) best test sample from the GSL dataset, (b), (c) best test samples from the ASL dataset. (1) ResNet50 based fused features, (2) optical flow based fused features, (3) OpenPose based fused features. BLEU-1 - BLEU-4 scores were close to 100% for the test samples chosen here.

features as the base feature is calculated:

$$Cross\_attention_1 = softmax\left(\frac{Q_{CNN}K_{OP}^T}{\sqrt{d_k}}\right)V_{OP}$$
$$Cross\_attention_2 = softmax\left(\frac{Q_{CNN}K_{OF}^T}{\sqrt{d_k}}\right)V_{OF} \quad (9)$$

where Q_{CNN} (CNN feature modality), K_{OP} (OpenPose feature modality), and V_{OP} (OpenPose feature modality) act as the query, key, and value inputs, respectively, for the first cross-feature transformer and Q_{CNN} (CNN feature modality), K_{OF} (optical flow modality), and V_{OF} (optical flow modality) act, respectively, for the second.

The cross-attentions for the other blocks 2 and 3 are similarly calculated with their respective base features. Then, the outputs from each cross-feature transformer block are fed into its own self-attention block as shown in Fig. 6. Finally, the outputs of all the self-attention blocks are fused and passed to a linear layer to learn their projections.

The decoder, in the standard fashion, takes the word embeddings as input and performs masked-multi head attention by masking the future words. The encoder embedding (from the linear layer) is fed to the multi-head attention block in the decoder, where it learns the encoder-decoder attention and predicts the words after passing through a feed-forward network, linear, and softmax layers.

4.2 Experiments

The Cross-Feature Fusion based transformer model is trained on 1027 ASLing (ASL) samples and tested on 257 held out samples. Adam optimization is used with a learning rate of $1e^{-03}$ and a weight decay rate of $1e^{-03}$. The maximum length of frames in each batch is chosen as the input sequence length for the encoder, and the decoder is fixed at a maximum caption length of 30, based on the average length of the captions. Cross-Feature Fusion based transformer models were trained for 70 - 150 epochs. Other optimal model settings used are encoder-decoder embedding size of 512, along with 3 encoder and decoder layers, and 8 multi-head attention blocks.

We test our cross feature model on the low-resource ASLing (ASL) dataset. Further, we train our Cross-Feature Fusion based transformer model using all three feature inputs. We perform similar experiments on the German Sign Language (GSL) dataset. To measure the performance of the Cross-Feature Fusion model using Bilingual Evaluation Understudy (BLEU) [27]. DUe to the constarints on the number of samples for ASLing dataset, we use the model that was already trained on GSL and fine tuned the ASLing data.

4.3 Attention Visualization

To evaluate the contribution of each of the input features, we selected the test sample that gave the best Bilingual Evaluation Understudy (BLEU) scores for the RWTH-PHOENIX-Weather2014T (GSL) and the ASLing (ASL) datasets. The attention map is shown in Fig. 7. The attention weights from the last layer of each of the three cross-modal encoders are read off, and the attention heatmaps are plotted.

The frames of the video under consideration are shown at the bottom of the heatmaps. The three different heatmaps corresponding to the three feature bases are shown where: (1) is ResNet50 based fused features, (2) is an optical flow-based fused feature, and (3) is OpenPose based fused features.

Observation. The quality of the input frames significantly affects the features selected for processing. Because ResNet50 works directly on the RGB images, when the quality of the input frames is poor, as in many ASLing frames, RESNet50 performs the worst. Irrespective of the quality of the input frames, OpenPose-fused features consistently perform well across both datasets. For the controlled and constrained GSL dataset, ResNet50 CNN features perform well. This is consistent with several of the works reporting state-of-the-art results on the benchmark datasets, such as [23], currently the best performing model on the benchmark RWTH-PHOENIX-Weather2014T dataset.

In summary, going forward, we will use only keypoint-based features for dual learning sign language analysis, as these have been shown to be most versatile and reliable for varying levels of input video quality. We investigate pose-to-text and text-to-pose, where pose is the sign language representation

5 SignNet II Model: Dual Learning Two Transformer-Based Networks

We present a two-way SLT model, SignNet II, learned using a dual learning paradigm [16], [41]. Dual learning for NMT involves the two parallel models, a primal model and a dual model, and is useful for co-learning the parameters of the two models in turn. Although dual learning has been

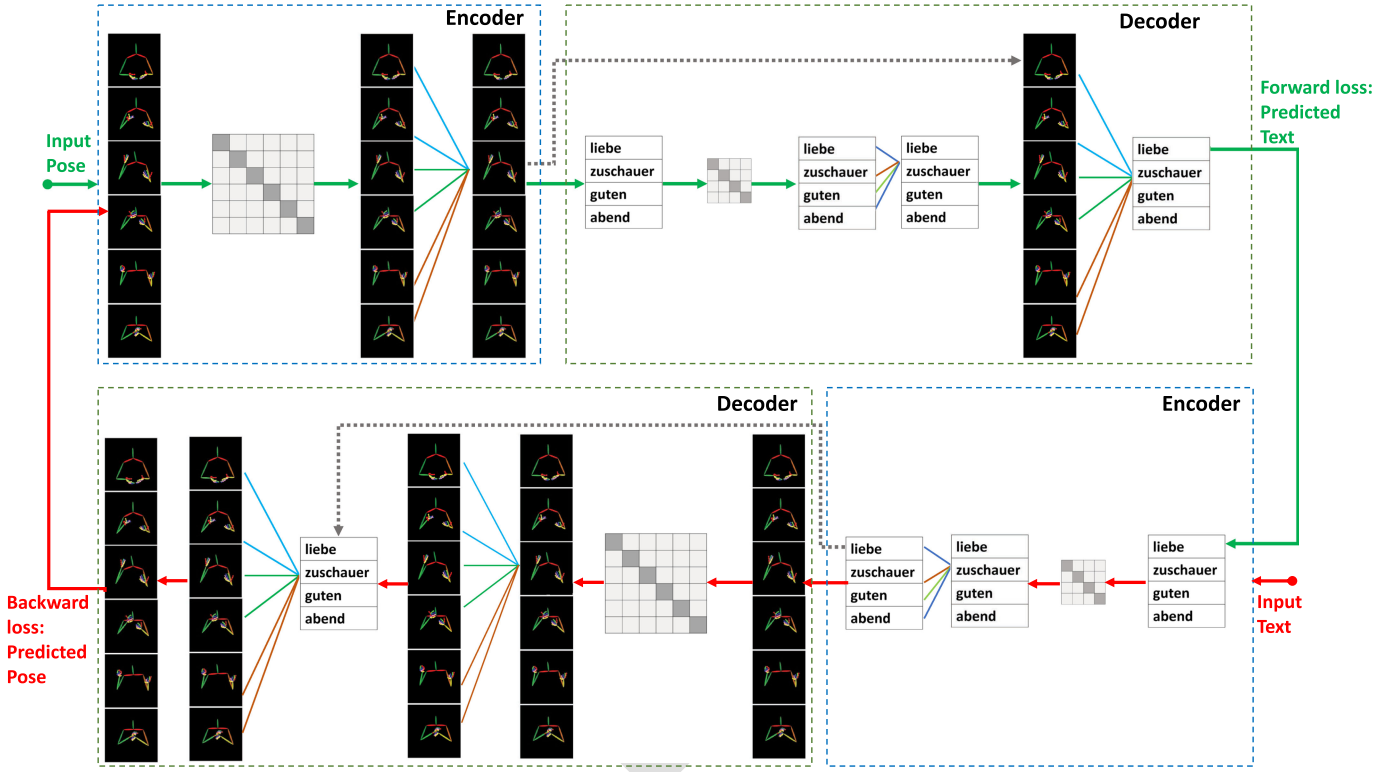

Fig. 8. SignNet II: Two-way Sign Language Translation trained the dual learning paradigm. The top network is the pose-to-text branch, and the bottom network is the text-to-pose branch. The direction of the arrows indicates the forward propagation and the predictions made. The backward propagation of losses will simply be in the reversal direction. Details on how the dual co-learning is accomplished are given in Section 5.3 (Best viewed in color).

useful for learning from unlabeled data, we use it to regularize the learning from our annotated data in a supervised fashion. Our two base models are (1) a sign/pose-to-text model and (2) a text-to-sign/pose model, and dual learning is used to refine them. SignNet II architecture, shown in Fig. 8, depicts the co-learning paradigm.

We briefly describe the two baseline models, especially focusing on their loss structures, and then present the dual learning formulation.

5.1 SignNet II: Pose-to-Text (P2T)

The top network in Fig. 8 performs the pose-to-text translation. We obtain the input 3D pose features by lifting the original 2D OpenPose joint keypoints [11] using an algorithm introduced by [44]. We use the key points from hand, finger, and upper body joints, resulting in fifty 3D joint locations, to give a vector of size 150 for each frame.

The input representation can be denoted as $POSE = \{[(x_{0_1}, y_{0_1}, z_{0_1}), \ldots, (x_{149_1}, y_{149_1}, z_{149_1})], \ldots, [(x_{0_N}, y_{0_N}, z_{0_N}), \ldots, (x_{149_N}, y_{149_N}, z_{149_N})]\}$,where $(x_{i_j}, y_{i_j}, z_{i_j})$ represents the i^{th} 3D joint in frame j and N is the number of frames in the input sequences.

To retain the input ordering information, we follow a similar pattern [37] and implement positional encoding for our input joints representation. We learn temporal dependencies across the entire sequence by correlating (or attending-to) all the frames with respect to a single frame, continuously, for all frames. This way, we learn the context alignment between the source and target.

To this end, we perform context learning between frames by initially computing the dot product of one frame (Q) with all other frames (K) of the video under consideration. To avoid exploding values after taking the dot product, we scale by $\sqrt{d}$ [[37]]. Finally, to retain the context information relevant to each frame, softmax activation $(softmax(\frac{QK^T}{\sqrt{d}})V)$ is applied on the frames (V). The resulting embeddings are then passed through a linear layer for enhanced features.

We follow a similar pattern on the decoder side, initially obtaining word embeddings for each word, adding positional information, and then learning the context between the words. Additionally, we learn the mapping between the frames and the words by taking the context information from the encoder and performing a scaled dot product with word-based attention. These embeddings are then passed onto a linear layer and softmax to predict continuous text.

5.1.1 *Translation Loss (L_{P2T})*

The primary task of the P2T branch of the 2-way interpretation model is to generate a written/spoken language sentence $S = (w_1, \ldots, w_U)$ given a sign video V, as defined previously. The translation process discussed here aims to learn $p(S|V)$. Going from pose to sentences, in the decoding phase, we have:

$$p(S|V) = \prod_{i=1}^{U} p(w_i|w_{i-1}) = \prod_{i=1}^{U} \mathbf{Z}_{i,s_i} \quad (10)$$

where U is the length of the sentence and $\mathbf{Z} = (Z_{j,k}) = [z_1, \ldots, z_U]^\top$ is the probability distribution of the sentence when translated. $Z_{j,k}$ is the probability of word w_j having a word label k, given w_{j-1}.

$$\mathcal{L}_{P2T} = 1 - p(S^T|V) \tag{11}$$

where S^T is the ground truth sentence corresponding to video V, comprised of the aggregation of the ground truth probability of words during the decoding phase.

5.2 SignNet II: Text-to-Pose (T2P)

The bottom part of Fig. 8 performs the text-to-pose translation. The workings of this block are similar to the P2T network explained above. Here, input 3D pose points are fed to the network. The encoder learns the context between different words of the input phrase, and the decoder learns the context between frames individually and between frames and words. The output of this network is the sequence of predicted poses.

5.2.1 Metric Embedded Learning for Pose Similarity

We are interested in ensuring that the predicted pose-based signs in the T2P arm of the 2-way SLT architecture predict continuous poses that are as similar as possible to the ground-truth signs and as distant as possible to other signs in the same batch.

To accomplish this, we have:

$$\underbrace{\|f(B) - f(T)\|^2}_{d(B,T)} - \underbrace{\|f(B) - f(S)\|^2}_{d(B,S)} \leq 0 \tag{12}$$

where B is a baseline sign, T is a true sign required to be as similar to B as possible, S is a false sign (not as similar to the baseline), and d(.) is the distance function.

To avoid the trivial solution where our function $f(.)$ will produce zero or one where $f(B) = f(T)$, we introduce a margin similar to [32] to impose a stronger constraint. The resulting distance function $d(B, T, S)$ is given in Equation (13):

$$d(B, T, S) = \max(\ (d(B, T) - d(B, S) + \alpha), 0\ ) \tag{13}$$

We refer to the loss derived based on this distance as the *pose similarity metric-based loss function*, given in Equation (14), which is useful for enhancing the performance of the T2P branch of the 2-way SLT training mechanism.

Choosing the similarity metric samples: While any random choice can readily satisfy $d(B, T) + \alpha \leq d(B, S)$, the underlying neural network will simply not learn if it gets it right too many times. If the choice of samples is made such that $d(B, T) \approx d(B, S)$, the network is forced to work hard to learn the differences. This seemingly simple choice significantly increases the efficiency of the learning algorithm. We, therefore, select our samples in the following manner:

Consider a batch $\|B\| = 4$, where we are interested in calculating the similarity loss for the first sample $i = 1$. The baseline here is the ground-truth sign which we will refer to as $B^{(i)}$. The truth $T^{(i)}$ is the network prediction for sample i. Lastly, the false value $S^{(i)}$ is the ground-truth for any other sample $j \neq i \in \{B\}$, where j is randomly selected.

5.2.2 Sign Similarity Metric-Based Loss (L_{T2P})

L_{T2P_1} is calculated in the lower branch of the 2-way mechanism shown in Fig. 8. This loss was introduced to reduce the risk of the network confusing similar signs. While training for efficient pose generation we use Dynamic Time Warping (DTW) [7] as our evaluation metric and optimize the network for the lowest DTW score.

The sign similarity based loss over all samples can be given as:

$$\mathcal{L}_{T2P_1} = \sum_{i}^{M} d(B^{(i)}, T^{(i)}, S^{(i)}) \tag{14}$$

Justification: There is only a finite number of valid poses that make up a valid sign, meaning there is often significant overlap between signs in the same batch. Without the strong constraint to separate truth and false examples, the network tends to readily confuse signs when predicting them from input text.

5.2.3 L_2 Regression Loss (L_{T2P_2})

The objective here is to learn the probability $p(V|S)$ of producing a sequence of sign-poses $V = (s_1, \ldots, s_T)$ over T time steps, given a spoken/written language sentence $S = (w_1, \ldots, w_U)$ having U words. Similar to the translation loss L_{P2T} described previously, this L_2 regression loss is also computed from the output of the decoder, although this occurs in the T2P branch of the model. Given the text sentence S as the inputs, the completed decoder output sequence of pose-signs can be expressed as $\hat{s}_{1:T} = \hat{s}_1, \ldots, \hat{s}_T$. The Mean Squared Error (MSE) loss between the predicted sequence, $\hat{s}_{1:T}$, and the ground truth,$s^T_{1:T}$ is given as:

$$\mathcal{L}_{P2T} = \mathcal{L}_{LMSE} = \frac{1}{T}\sum_{i=1}^{t}(s^T_{1:T} - \hat{s}_{1:T})^2 \tag{15}$$

5.3 Dual Learning

We use a dual learning mechanism motivated by [16] to jointly refine the model parameters for both the networks in SigNet II. We perform supervised dual learning because we have annotated data for both the P2T and T2P branches. The goal is to better utilize our two sets of annotated training data by enhancing probabilistic correlations within the two models.

Let us define a sign phrase as $\mathbf{x}$ and its textual translation as $\mathbf{y}$. For a bilingual sign-text sentence pair $(\mathbf{x}, \mathbf{y})$, ideally $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{x}|\mathbf{y}) = p(\mathbf{y})p(\mathbf{y}|\mathbf{x})$. If the two models are only trained apart, it becomes challenging to satisfy $p(\mathbf{x})p(\mathbf{x}|\mathbf{y}) = p(\mathbf{y})p(\mathbf{y}|\mathbf{x})$; hence a joint training of the two models can be performed as:

$$\begin{aligned}\mathcal{L}_{DL} = &(\log \hat{p}(\mathbf{x}) + \log \hat{p}(\mathbf{y}|\mathbf{x}; \theta_{x\to y}) \\ &- \log \hat{p}(\mathbf{y}) - \log \hat{p}(\mathbf{x}|\mathbf{y}; \theta_{y\to x})\end{aligned} \tag{16}$$

where we $\hat{p}(\mathbf{x})$ and $\hat{p}(\mathbf{y})$ can be viewed as empirical statistics of the data. During our implementation, we approximate these statistics as the data induced scores r_A and r_B defined

below. Details of the dual learning implementation of are given in Algorithm 1.

Before the first iteration starts, the two baseline translation models, P2T and T2P, are pretrained by back-propagating the losses described previously. In the first iteration, the P2T network acts as a forward translation step and T2P as the backward translation step whose inputs are text predictions from P2T. The losses back-propagated in backward network are also weighted by the effects of the forward translation performance.

In the next iteration, the models are reversed so that T2P becomes the forward translation while P2T becomes the backward one. Similarly, the forward pose predictions now act as inputs to the backward P2T network, and weighted losses are back-propagated. This completes one full loop of the dual learning mechanism. The process is repeated until both sets of losses converge.

Algorithm 1. SigNet II Dual Learning

Data: 3D OpenPose points P, Pose translations T, initial two translation model T_A for Pose-to-Text & T_B for Text-to-Pose, hyperparameter α

repeat
Create samples S with poses s_A from P and s_B from T
for *each sample* $S_A \in \mathcal{S}$ **do**
  Set $M_A = T_A$, $M_B = T_B$
  Set $fw\_loss = \mathcal{L}_{P2T} = 1 - p(S^T|V)$
  Set $bk\_loss = \mathcal{L}_{T2P} = \sum_i^M d(B^{(i)}, T^{(i)}, S^{(i)})$
  Predict N phrases from Pose-to-Text translation model M_A Set the Pose-to-text reward:
  $r_A = \frac{(fw\_loss - \frac{1}{N}\sum(fw\_loss))}{\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(fw\_loss_i - \bar{fw\_loss})^2}}$
  for *Each* n *in* N **do**
    Predict n poses using Text-to-Pose translation model M_B
    Set the Text-to-Pose reward:
    $r_B = \frac{(bw\_loss - \frac{1}{N}\sum(bw\_loss))}{\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(bw\_loss_i - \bar{bw\_loss})^2}}$
    Compute the final reward, $r_n = \alpha * r_A + r_B * (1 - \alpha)$
  end
  Compute the stochastic gradient for M_A:
  $\theta_A = \frac{1}{N}\sum(fw\_loos * r_{P2T})$
  Compute the stochastic gradient for M_B: $\theta_B = \frac{1}{N}\sum(bw\_loos * (1 - \alpha))$
  Update models M_A & M_B
Set $M_A = T_B$, $M_B = T_A$
end

We sampled the training data from both the sets of inputs P and T. We denote Pose-to-Text model as model T_A and Text-to-Pose model as T_B. We assume that we have two well-trained T_A and T_B, meaning each gives a different predicted output which is the input of the other. For the first pass, we consider T_A to be model 'A', M_A or as the anchor model, for each of the pose samples model M_A predicts the corresponding phrases. We compute the loss considering this operation to be the forward translation step using Section 5.1.1. For the backward translation, we compute the loss using the outputs from model T_B and leverage the total loss Section 5.2.2 function.

$$r_A = \frac{(fw\_losses - \frac{1}{N}\sum(fw\_losses))}{\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(fw\_losses_i - \bar{fw\_losses})^2}} \quad (17)$$

$$r_B = \frac{(bw\_losses - \frac{1}{N}\sum(bw\_losses))}{\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(bw\_losses_i - \bar{bw\_losses})^2}} \quad (18)$$

The predicted text are considered to be the initial translation output for which we compute the reward variable r_A shown in Equation (17) (line 8 of 20). We predict the final output of the first pass and calculate it's immediate reward function r_B using the formulation shown in Equation (18) (line 12 of 20). We mathematically compute both the rewards by achieving the forward and the backward losses obtained from the first pass. We calculate the final reward value which is linear summation of both the forward r_A and backward r_B rewards, $r_n = \alpha * r_{P2T} + r_{P2T} * (1 - \alpha)$. We then compute the gradients of the rewards with reference to both the models in the network. The model parameters are updated based on the computed gradients, and the second pass reverses the roles of the models.

6 Experiments and Results

6.1 Metrics

We evaluate our SigNet II model using Bilingual Evaluation Understudy [27]. This metric is especially used to evaluate automatic machine translation mechanisms. BLEU-4 evaluates the performance of 4-gram words i.e., four consecutive words, while BLEU-1 evaluates the individual word-based performance i.e., 1-gram. We tested our dual learned P2T and T2P models on the benchmark German Sign Language (GSL) dataset Section 3 and used BLEU as the evaluation metric to gauge their final performance.

6.2 Training Schedule

The SignNet II model was trained on the benchmark German Sign Language dataset. For training, we selected an initial learning rate of 1e-7 with a weight decay rate of 1e-3. The encoder embedding dimension of 512 was used with 2 layers of encoder and decoder each and 4 multi-head attention for pose-to-text and 2 multi-head attention blocks for text-to-pose. A grid search was done for both the branches (pose-to-text & text-to-pose) to select the optimal hyperparameter values. The SignNet II pose-to-text model has approximately 2.67 million parameters and text-to-pose model has 17.04 million parameters. The model were developed using Pytorch framework. We utilized Nvidia RTX 3090 Ti processor with 24 GB memory for training.

6.3 Discussion

Both the P2T and T2P models were trained individually, while the SigNet II jointly trained both the models. We used only the GSL dataset to train and test our model because unlike ASLing, GSL has a significantly larger number of samples. We evaluated the P2T arm by passing the test set through the top branch of the SigNet II model shown in Fig. 8. Table 3 shows the pose-to-text translation results after applying the dual learning algorithm. We compared our results with [1], [13], [23] as these make use of single feature inputs to evaluate their models. We refer to single feature

TABLE 1
Statistics on the Benchmark GSL and Our Collected ASL Datasets

	GSL (benchmark)	ASLing (ours)
# training samples	7096	1017
# testing samples	693	254
# signers	9	7
Average # signs per signer	859	183
Same user in train and test	Yes	Yes

TABLE 2
Cross-Feature Fusion Based Transformer Model on Both ASL and GSL Datasets

Experiment type	Test			
	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Cross-Feature Fusion based transformer model (ASLing (ASL))	22.39	15.96	13.56	12.25
Cross-Feature Fusion based transformer model (GSL)	27.33	18.18	13.26	10.46

TABLE 3
Translation Performance on Predicted Text Using Dual Learning

Experiment type	Test			
	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Conv2d-RNN [13]	27.10	15.61	10.82	8.35
Conv2d-RNN [13] + Luong Attention [25]	29.86	17.52	11.96	9.00
Conv2d-RNN [13] + Bahdanau Attention [12]	32.24	19.03	12.83	9.58
Feature scaling (OP_8) [1]	21.83	13.85	10.34	8.28
TSPNet-Single 8 [23]	30.29	17.75	12.35	9.41
TSPNet-Single 12 [23]	29.02	17.03	12.08	9.39
TSPNet-Single 16 [23]	32.52	20.33	14.75	11.61
Baseline P2T (ours)	38.87	23.67	16.11	11.67
SigNet II: Dual Learning P2T for T2P (ours)	**39.17**	**24.57**	**16.94**	**12.34**

P2T - Pose-to-Text.

input as using only one feature, in this case, keypoint-based pose feature. On the other hand, Section 4 describes the use of multi-features (CNN, OpenPose, optical flow features) to perform the same task of pose-to-text. We achieved a score of 39.17 for BLEU-1 and 12.34 for BLEU-4, which is a significant increase in performance when compared to other models.

For text-to-pose translation using the dual learning algorithm, we show the results in Table 4. Saunders et al. [31] trained their text-to-pose model using Mean Squared Error (MSE) loss to compare the ground truth sequences with the predicted poses. But SignNet II used a metric-based embedding loss in addition to the MSE loss and improved the BLEU score performance.

Out of the total 8000+ data samples available, we could only use about 10% of the samples to train our model. This was due to the fact that while training the dual learning algorithm, we simultaneously worked with two heavily parameterized models, which turned out to be computationally expensive. This 10% was randomly selected and was repeated multiple times with different batches. However, based on the result trends we observed, we believe that given a sufficient amount of resources we will continue along this performance trajectory.

The evaluation in this paper does not provide evidence that these results are immediately useful to sign language users. Furthermore, according to [3] in their experiment "Human as an oracle", they compared how well a human expert signer could translate a sign language video versus translating the corresponding animation, created using OpenPose skeletons. Their results on 340 ASL videos showed very poor translation, with a BLEU-4 score of nearly zero. This suggests that although sign dynamics are

TABLE 4
Translation Performance on Predicted Poses Using Dual Learning

Experiment type	Test			
	BLEU-1	BLEU-2	BLEU-3	BLEU-4
PT G2P (MSE only) [31]	31.8	19.19	13.51	10.43
PT T2P (MSE only) [31]	31.36	19.04	13.54	10.51
SignNet T2P (ours w/o metric-based loss)	36.19	20.77	13.27	8.8
Singly trained T2P (ours - MSE + metric loss)	36.79	21.79	14.77	**10.66**
dual learning T2P (ours - MSE + metric loss)	**38.26**	**22.48**	**14.80**	10.21

T2P - Text-to-Pose.

successfully captured with pose data, additional embodiment of a signing agent will be needed for improved sign language understanding.

7 Conclusion

In this article, we introduced a novel transformer based dual learning algorithm, SignNet II, a promising step towards facilitating 2-way sign language interpretation. We presented a multi-feature cross-attention transformer-based architecture, and its output attention maps indicated that keypoint-based pose features were the most versatile and reliable for analyzing different input videos of varying quality. Hence, this became our single feature of choice for dual learning.

Our results showed that using dual learning for complex tasks such as sign language translations can be useful in boosting the performances of the models. When compared to the SoTA models, SignNet II showed improvements in the BLEU scores for 2-way sign language translations. We have taken a step towards constructing a model that jointly trains complex translation models, but given adequate resources, SignNet II has the potential to continue to improve model performance.

References

[1] T. Ananthanarayana, L. Chaudhary, and I. Nwogu, "Effects of feature scaling and fusion on sign language translation," in *Proc. 22nd Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 2292–2296.

[2] T. Ananthanarayana, N. Kotecha, P. Srivastava, L. Chaudhary, N. Wilkins, and I. Nwogu, "Dynamic cross-feature fusion for american sign language translation," in *Proc. IEEE 16th Int. Conf. Autom. Face Gesture Recognit.*, 2021, pp. 1–8.

[3] T. Ananthanarayana, P. Raymond, R. Majid, L. Alexander, and S. Andreas, "A comprehensive approach to automated sign language translation," PhD thesis, 2021.

[4] T. Ananthanarayana et al., "Deep learning methods for sign language translation," *ACM Trans. Accessible Comput.*, vol. 14, no. 4, pp. 22:1–22:30, 2021.

[5] A. B. Zadeh, P. P. Liang, S. Poria, E. Cambria, and L.-P. Morency, "Multimodal language analysis in the wild: CMU-MOSEI dataset and interpretable dynamic fusion graph," in *Proc. 56th Annu. Meeting ACL*, 2018, pp. 2236–2246.

[6] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015.

[7] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *Proc. 3rd Int. Conf. Knowl. Discov. Data Mining*, 1994, pp. 359–370.

[8] T. B. Brown et al., "Language models are few-shot learners," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, Art. no. 159.

[9] N. C. Camgöz, O. Koller, S. Hadfield, and R. Bowden, "Sign language transformers: Joint end-to-end sign language recognition and translation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10020–10030.

[10] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 172–186.

[11] Z. Cao, G. H. Martinez, T. Simon, S. Wei, and Y. A. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 172–186, Jan. 2021.

[12] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," in *Proc. 8th Workshop Syntax Semantics Struct. Statist. Transl.*, 2014, pp. 103–111.

[13] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Neural sign language translation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7784–7793.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[15] B. Fang, J. Co, and M. Zhang, "DeepASL: Enabling ubiquitous and non-intrusive word and sentence-level sign language translation," in *Proc. 15th ACM Conf. Embedded Netw. Sensor Syst.*, 2017, pp. 1–13.

[16] D. He et al., "Dual learning for machine translation," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 820–828.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[18] M. Huenerfauth and P. Lu, "Effect of spatial reference and verb inflection on the usability of sign language animations," *Universal Access Inf. Soc.*, vol. 11, no. 2, pp. 169–184, 2012.

[19] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 1700–1709.

[20] M. Kemp, "Why is learning american sign language a challenge?," *Amer. Ann. Deaf*, vol. 143, pp. 255–259, 1998.

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

[22] O. Koller, N. 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 Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 9, pp. 2306–2320, Sep. 2020.

[23] D. Li et al., "TSPNet: Hierarchical feature learning via temporal semantic pyramid for sign language translation," 2020, *arXiv: 2010.05468*.

[24] Lifetime.com, "Lifetime.com," [Online]. Available: htpp://www.Lifetime.com

[25] M. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," 2015, *arXiv:1508.04025*.

[26] R. E. Mitchell, "How many deaf people are there in the United States? Estimates from the survey of income and program participation," *J. Deaf Stud. Deaf Edu.*, vol. 11, no. 1, pp. 112–119, Sep. 2005.

[27] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*, 2002, pp. 311–318.

[28] J. Pu, W. Zhou, and H. Li, "Dilated convolutional network with iterative optimization for continuous sign language recognition," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 885–891.

[29] A. Radford, "Improving language understanding by generative pre-training," 2018.

[30] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, 2019, Art. no. 9.

[31] B. Saunders, N. C. Camgöz, and R. Bowden, "Progressive transformers for end-to-end sign language production," 2020, *arXiv: 2004.14874*.

[32] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 815–823.

[33] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 568–576.

[34] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[35] J. Sánchez Pérez, E. Meinhardt-Llopis, and G. Facciolo, "TV-L1 optical flow estimation," *Image Process. On Line*, vol. 3, pp. 137–150, 2013.

[36] Y.-H. H. Tsai, S. Bai, P. P. Liang, J. Z. Kolter, L. Morency, and R. Salakhutdinov, "Multimodal transformer for unaligned multimodal language sequences," in *Proc. 57th Conf. Assoc. Comput. Linguistics*, 2019, Art. no. 6558.

[37] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[38] S. Wang, D. Guo, W.-G. Zhou, Z.-J. Zha, and M. Wang, "Connectionist temporal fusion for sign language translation," in *Proc. 26th ACM Int. Conf. Multimedia*, 2018, pp. 1483–1491.

[39] WFD, "World federation of deaf," [Online]. Available: http://wfdeaf.org/our-work/

[40] Y. Wu et al., "Google's neural machine translation system: Bridging the gap between human and machine translation," 2016, *arXiv:1609.08144*.

[41] Y. Xia, T. Qin, W. Chen, J. Bian, N. Yu, and T.-Y. Liu, "Dual supervised learning," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3789–3798.
[42] K. Yin, "Sign language translation with transformers," 2020, *arXiv: 2004.00588*.
[43] T. Yuan et al., "Large scale sign language interpretation," in *Proc. IEEE 14th Int. Conf. Autom. Face Gesture Recognit.*, 2019, pp. 1–5.
[44] J. Zelinka and J. Kanis, "Neural sign language synthesis: Words are our glosses," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2020, pp. 3384–3392.

Lipisha Chaudhary received the BEng degree from the University of Mumbai, India, in 2016 with distinction, and the master's degree in computer science from the Rochester Institute of Technology, NY, in 2021. She is currently working toward the PhD degree with the Human Behavior Modeling Lab, University at Buffalo, SUNY. Her research interests include machine learning for social good.

Tejaswini Ananthanarayana received the BEng degree from the University of Mumbai, India, and the PhD degree in engineering from the Rochester Institute of Technology, NY, in 2021. She currently works as a senior algorithm engineer with Sony Electronics, USA. Her research interests include deep learning for sign language understanding.

Enjamamul Hoq received the BSc degree from the Khulna University of Engineering & Technology, Bangladesh, in 2016, and the master's degree in mechanical engineering from the University of Massachusetts Dartmouth, MA, in 2021. He is currently working toward the PhD degree with the Human Behavior Modeling Lab, University at Buffalo, SUNY. His research interests include the causal inference for human behavioral modeling.

Ifeoma Nwogu (Senior Member, IEEE) received the BSc degree from the University of Lagos, Nigeria, the master's degree in computing and information sciences from the University of Pennsylvania, and the PhD degree in computer science and engineering from the University at Buffalo, in 2009, where she currently works as an associate professor. Her research interests involve computational models for learning human behaviors.