Sign Stitching: A Novel Approach to Sign Language Production

Harry Walsh
harry.walsh@surrey.ac.uk
Ben Saunders
b.saunders@surrey.ac.uk
Richard Bowden
r.bowden@surrey.ac.uk

CVSSP
University of Surrey
Guildford, UK

Abstract

Sign Language Production (SLP) is a challenging task, given the limited resources available and the inherent diversity within sign data. As a result, previous works have suffered from the problem of regression to the mean, leading to under-articulated and incomprehensible signing. In this paper, we propose using dictionary examples to create expressive sign language sequences. However, simply concatenating the signs would create robotic and unnatural sequences. Therefore, we present a 7-step approach to effectively stitch the signs together. First, by normalising each sign into a canonical pose, cropping and stitching we create a continuous sequence. Then by applying filtering in the frequency domain and resampling each sign we create cohesive natural sequences, that mimic the prosody found in the original data. We leverage the SignGAN model to map the output to a photo-realistic signer and present a complete Text-to-Sign (T2S) SLP pipeline. Our evaluation demonstrates the effectiveness of this approach, showcasing state-of-the-art performance across all datasets.

1 Introduction

Sign Language Production (SLP) is an essential step in facilitating two-way communication between the Deaf and Hearing communities. Sign language is inherently multi-channelled, with channels performed asynchronously and categorised into manual (hands and body) and non-manual (facial, rhythm, stress and intonation) features. For sign language to be truly understandable, both manual and non-manual features must be present. Analogous to the tone and rhythm used in spoken language, signed language exhibits prosody. The natural rhythm, stress and intonation that signed languages use to convey information [39].

Sign language corpora containing linguistic annotation are limited due to the cost and time required to create such annotations [15]. Previous works have attempted to directly regress a sequence of poses from the spoken language or representations such as gloss [10, 11, 24, 28, 31, 33, 40, 41]. However, given that sign language is a low-resource language and the complexity is under-represented in small datasets, previous approaches have suffered from regression to the mean, resulting in under-articulated and incomprehensible

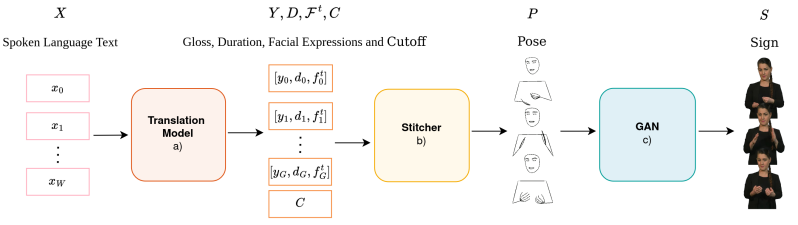

Figure 1: Overview of our approach. a) Spoken language to gloss, duration, facial expression and cutoff translation. b) Pose sequence generation. c) Photorealistic signer production.

signing. Additionally, previous works have implicitly modelled prosody, but due to the limited resources, it is often lost in production.

In this paper, we propose a novel approach to SLP that effectively stitches together dictionary examples to create a meaningful, continuous sequence of sign language. By using isolated signs, we ensure the sequence remains expressive, overcoming previous shortcomings related to regression to the mean. However, each example lacks non-manual features, so we propose a Noise Substitution Vector Quantization (NSVQ) transformer architecture to learn a dictionary of facial expressions that can be added to each sign to create a realistic sequence. To the best of our knowledge, we are the first to explicitly model aspects of signed prosody in the context of SLP. By training a translation model to predict glosses, alongside a duration, facial expression and a cutoff value, we can modify the sequence to eliminate robotic and unnatural movements. Resampling each sign to the predicted duration allows us to alter the velocity associated with signing stress and rhythm [38]. Furthermore, by applying filtering in the frequency domain, we can adjust the trajectory of each sign to create softer signing, akin to how signers modify a sign to convey sentiment [2, 9, 23]. Our approach demonstrates it is capable of modifying the stitched sequence to emulate aspects of prosody seen in the original continuous data. Evaluation of the produced sequence with back translation showcases state-of-the-art performance on all three datasets.

Furthermore, to conduct a realistic user evaluation we use SignGAN, a Generative Adversarial Network (GAN) capable of generating photo-realistic sign language videos from a sequence of poses [25]. Thus, we present a full Text-to-Sign (T2S) SLP pipeline that contains both manual and non-manual features. The user evaluation agrees that the approach outperforms the baseline method [26] and improves the realism of the signed sequence. An overview of the approach can be seen in Fig. 1.

2 Related Work

Sign language Translation: For the last 30 years Computational Sign Language Translation (SLT) has been an active area of research [32]. Initially focusing on isolated Sign Language Recognition (SLR) where a single instance of a sign was classified using a Convolutional Neural Network (CNN) [16]. Subsequent works extended to Continuous Sign Language Recognition (CSLR), which requires both the segmentation of a video into its constituent signs and their respective classification [14]. Later Camgoz et al. introduced the task of Sign-to-Text (S2T) [4] using neural networks, an extension of CSLR that requires the additional task of translation to spoken language. S2T performance was later improved

using a Transformer [36]. Although there has been a lot of work since, the architecture has since become the standard when computing back-translation performance [24, 26, 28].

Sign Language Production: SLP is the reverse task to SLT, which aims to translate spoken sentences into continuous sign language. Early approaches to SLP used an animated avatar driven with either motion capture data or parameterised glosses [1, 6, 7, 8, 35, 42]. These works all required expensive annotation systems, such as the Hamburg Notation System (HamNoSys) [21] or SigML [13] and have shown to be unpopular with the Deaf community due to the robotic motion and under articulated signing [22]. None of these approaches attempt to join the isolated signs effectively. Instead opting to play each sign in the sequence, with unnatural transitions in between.

Early deep learning SLP approaches used Neural Machine Translation (NMT) and broke the task down into three steps, Text-to-Gloss (T2G), Gloss-to-Pose (G2P) and Pose-to-Sign (P2S) [31]. Saunders et al. introduced the Progressive Transformer (PT) [26], a transformer architecture that synthesises poses directly from text. Although better results were achieved using gloss as an intermediate representation, the approach suffered from regression to the mean, caused by the lack of training data and the diversity of lexical variants. To reduce the problem, adversarial training and Mixture Density Network (MDN) were applied [24, 28] and since then a range of approaches have been proposed [10, 11, 27, 33, 40, 41]. However, visual inspection of the results shows that the approaches still suffer from regression to the mean, and as a result, they fail to effectively convey the translation. Here we propose a method to effectively join isolated signs, which means the produced sequences are guaranteed to be expressive and do not suffer from regression to the mean.

Furthermore, preliminary experiments reveal that each sign language sequence contains a distinct range of frequencies correlated to the signer's style. Fast motions contain high frequencies, while soft, slow signing involves lower frequencies, typically within the range of 1 to 25 Hz. To emulate this characteristic we filter the produced sequences in the frequency domain using a low-pass Butterworth filter [3]. This ensures the movements are stylistically cohesive. In addition, by adjusting the duration of each sign, we recreate the prosody observed in the original data.

3 Methodology

SLP aims to facilitate the continuous translation from spoken to signed languages by converting a source spoken language sequence, $X = (x_1, x_2, ..., x_W)$ with W words into a video of photo-realistic sign, denoted as $V = (v_1, v_2, \ldots, v_U)$ with U frames. To accomplish this the approach uses two intermediate representations, following Fig. 1 from left to right. First, the spoken language is translated to a sequence of glosses, $Y = (y_1, y_2, \ldots, y_G)$, face tokens, $\mathcal{F}^t = (f_1^t, f_2^t, \ldots, f_G^t)$ and duration's, $D = (d_1, d_2, \ldots, d_G)$, all with length G. Additionally, for each sequence, we predict a low pass cutoff, $\mathcal{C}$, which we use to filter the movements (Fig. 1.a). Each gloss and facial expression is stitched together using these parameters, to produce a continuous sequence of poses, denoted as $P = (p_1, p_2, ..., p_U)$ with U frames (Fig. 1.b). Finally, we use the pose sequence to condition the SignGAN module allowing us to produce a photo-realistic signer. Next, we provide a detailed explanation of each step in our pipeline, following the order illustrated in Fig. 1 from left to right. We then elaborate on the process of generating the cutoff frequencies and the dictionary of facial expressions.

3.1 Translation Model

Given a spoken language sequence $X = (x_1, x_2, ..., x_W)$, our goal is to generate a corresponding sequence of glosses $Y = (y_1, y_2, ..., y_G)$. We design the transformer with four output layers, enabling the model to predict the corresponding duration (in frames) and facial expression for each gloss, plus a low-pass filter cutoff in Hz for each sequence. Thus the model learns the conditional probability $p(Y, D, \mathcal{F}^t, \mathcal{C}|X)$.

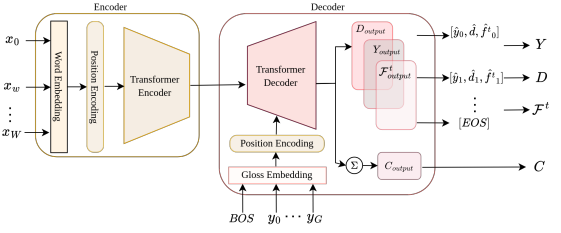

Figure 2: An overview of the Translation module.

The model is an encoder-decoder transformer with Multi-Headed Attention (MHA). The spoken language and gloss sequences are tokenized at the word level, and the embedding for a sequence is generated using a token embedding layer. Following the embedding layer, we add sine and cosine positional encoding.

The encoder learns to generate a contextualized representation for each token in the sequence. This representation is then fed into the decoder, which consists of multiple layers of self and cross MHA along with feedforward layers, and residual connections. The gloss, facial expression and duration predictions are obtained by passing the decoder output through their respective output layers. To obtain the cutoff prediction, we pool the decoder embedding across each time step and pass the output through a linear layer. The model is trained end-to-end with the following loss function;

$$L_{total} = \lambda_y \sum_{i=1}^{Y} \hat{y}_i \log(y_i) + \lambda_f \sum_{i=1}^{\mathcal{F}} \hat{F}_i \log(F_i) + \lambda_d \frac{1}{n} \sum_{i=1}^{n} (d_i - \hat{d}_i)^2 + \lambda_C (C - \hat{C})^2 \quad (1)$$

Each component is scaled by a factor, λ_y, λ_d, λ_f and λ_C before being combined to give the total loss, L_{total}. The predictions from this model are passed to the stitching module to generate a pose sequence.

3.2 Stitching

For each dataset, we collect an isolated instance of each gloss in our target vocabulary. For each sign, we extract Mediapipe skeletons [18] and run an additional optimization to uplift the 2D skeletons to 3D [17]. The optimisation uses forward kinematics and a neural network to solve for joint angles, J_a. We choose to store our dictionary as joint angles, as this allows us to apply a canonical skeleton. This ensures the stitched sequence is consistent even if the original signers have different bone lengths. We define a dictionary of, N_s, signs as $DS = [S_1, S_2, ..., S_{N_s}]$ where each sign in the dictionary consists of a sequence of angles, such that $S_i = (a_1, a_2, ..., a_{U_s})$ and $a_i \in \mathbb{R}^{J_a}$, where U_s is the duration in frames. In addition we define a learnt dictionary of, N_f, facial expressions as $DF = [F_1, F_2, ..., F_{N_f}]$, where $F_i \in \mathbb{R}^{U_f \times J \times D}$. Fig. 3 illustrates our seven-step stitching pipeline, we now detail each step.

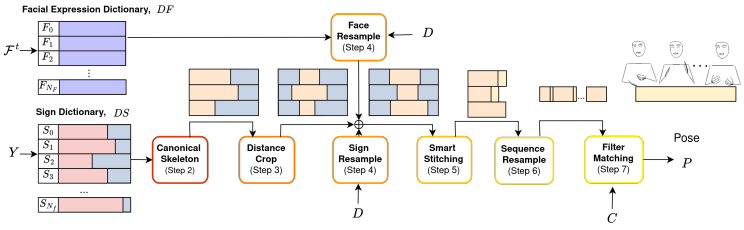

Figure 3: An overview of the stitching module.

Step 1) Given a list of glosses, Y, we select the corresponding signs in our dictionary. If a gloss is absent from the dictionary, we initially lemmatize and format the gloss. If still, we are unable to find a match in the dictionary, we apply a word embedding model and compute the cosine similarity with all words in the dictionary. We then select the closest sign as the substitute. Such that;

$$j_{sub} = \arg\max_j \left(\frac{\sum_{j=1}^{N_s} \varepsilon(y_q) \cdot \varepsilon(DS_j^y)}{\sqrt{\varepsilon(y_q)^2} \cdot \sqrt{\sum_{j=1}^{N_s} \varepsilon(DS_j^y)^2}} \right) \qquad S = DS[j_{sub}] \tag{2}$$

Here $\varepsilon()$ represents the word embedding model, y_q is the query gloss and DS^y is the dictionary's corresponding gloss tags. We find word embeddings capture the meaning of words, enabling substitutions such as replacing RUHRGEBIET (RUHR AREA) with LANDSCHAFT (LANDSCAPE). Simultaneously, in this step, we select the corresponding facial expressions, F, from the dictionary, DF, given the predicted face tokens, $\mathcal{F}^t$.

Step 2) The selected signs are converted from angles into a 3D canonical pose. We normalise the rotation of the signer, such that the midpoint of the hips is located at the origin and the shoulders are fixed on the y plane. This ensures the skeleton is consistent across all the signs. Consequently, we convert from a sequence of angles $S_n \in \mathbb{R}^{U_S \times J_a}$ to a sequence of poses, $P = (p_1, p_2, ..., p_{U_s})$ with the same number of frames, U_s. Each pose, p_u, is represented in D-dimensional space and consists of J joints, denoted as $p_u \in \mathbb{R}^{J \times D}$.

Step 3) The dictionary signs often start and end from a rest pose. Therefore, to avoid unnatural transitions we cropped the beginning and end of each sign. For this, we track the keypoint T corresponding to the wrist of the signer's dominant hand and measure the distance travelled. Thus, for each dictionary sign we create a sequence, $P^\Delta = (p_2^\Delta, p_3^\Delta, \ldots, p_{U_s}^\Delta)$ $n \in 2, 3, \ldots, U_s$, representing the distance travelled for a dictionary sign. We remove the beginning frames once the sign has moved by a threshold, α_{crop}. The crop index is given by:

$$\text{index}_{\text{start}} = \arg\max_u \left(\sum_{u=1}^{U_s} P_u^\Delta - \alpha_{crop} \cdot \sum_{u=1}^{\max(U_s)} P_u^\Delta \right) \tag{3}$$

To crop the end, we reverse the order of frames and repeat the process.

Step 4) As detailed above, we predict the duration of each gloss. Here, we utilize the duration to resample the length of each sign and facial expression, emulating the natural rhythm in the original data. This process involves upsampling or downsampling the sign using linear interpolation. Once the facial expression and sign are resampled to the same length, we shift and rotate the face onto the body creating the complete skeleton.

Step 5) Having created a list of signs in a canonical pose, cropped, and resampled to match the original data, the next step involves joining these signs into a single sequence using the smart stitching module. The objective is to achieve a natural transition between the end of one sign and the start of the next. To accomplish this, we track the dominant hand of the signer and calculate the distance, Δ, between the end of the first sign and the start of the second sign. Then we can determine the required number of frames, U_{stitch}, needed to create a smooth transition. Such that:

$$U_{stitch} = \arg\min_{u}(V_{min} < \frac{\text{fps} * \Delta}{u} < V_{max}) \tag{4}$$

Where V_{max} and V_{min} are the start and end velocities of the two signs. This calculation ensures that the signer's velocity is bounded by the end velocity of sign one and the initial velocity of sign two. In cases where multiple solutions exist, we select U_{stitch} that minimizes the standard deviation between the start and end velocity. Following this, we employ linear interpolation to generate the missing frames.

Step 6) We concatenate the signs and the stitched frames to form a single sequence. We then sum all the predicted durations and resample the sequence to match the ground truth.

Step 7) Finally, we apply a low-pass Butterworth filter to each keypoint over time [3]. The predicted cutoff value determines which frequencies are removed, and corresponds to the -3dB attenuation point. This step aims to enhance the stylistic cohesiveness of the sequence by smoothing out any sharp, quick movements not present in the original sequence. The transfer function can be formulated as;

$$H(z) = \left((1 + \left(\frac{z}{e^{j \cdot \omega_c}}\right)^{2N}\right)^{-1} \tag{5}$$

Here we apply a 4th order filter thus, N is 4, and the angular cutoff frequency is given by $\omega_c = 2\pi C$. z corresponds to the z-transform of the pose sequence. Applying the bilinear transform to Eq. (5) gives the discrete formula that we apply. This process generates natural pose sequences, that remain expressive and are stylistically cohesive. Next, we map these poses to a photo-realistic signer.

3.3 SignGAN

Skeleton outputs have been shown to reduce Deaf comprehension compared to a photo-realistic signer [37]. Therefore, to gain valuable feedback from the Deaf community we train a SignGan model [25]. Given a pose sequence, $P = (p_1, p_2, \ldots, p_U)$, generated by our stitching approach the model aims to generate the corresponding video of sign language, $V = (v_1, v_2, \ldots, v_U)$ with U frames.

3.4 Facial Expression Generation

To be truly understandable and accepted by the Deaf community non-manual features must be present in the final output. Here we are using a discrete sequence-to-sequence model to generate the translation. Therefore, we must learn a discrete vocabulary of facial expressions, DF, that can be added to the isolated signs. We design a transformer base NSVQ architecture to learn a spatial-temporal dictionary of facial expressions. Using the ground truth gloss timing information, we extract the corresponding face mesh sequence, denoted as F. We then

resample each sequence to a constant length, U_f and scale it to be a constant size. Signers in the dataset are often looking off center, therefore we normalize the average direction of the face so that it is looking directly forward. Similar to Fig. 2 (Encoder) we add positional encoding and then embed the sequence using a single linear layer. After the embedding is passed through the transformer encoder to the codebook. The NSVQ codebook learns a set of N_f embeddings [34]. We denote the embedded face sequence and therefore each codebook entry as $F_i^z \in \mathbb{R}^{U_f \times H}$, where H is the embedding dimension. Each input is mapped to one codebook entry, the difference between the selected codebook entry and the input is then simulated using a normally distributed noise source. A product of the simulated noise and the encoder output is then passed to the decoder. We use the counter decoding technique from the PT [26], to drive the decoder. The decoder learns to reconstruct the original face sequence and the input counter values. Thus, the model is trained end-to-end with the following loss function;

$$L_{Face} = \frac{1}{U_f} \sum_{u=1}^{U_f} ((f_u - \hat{f}_u)^2 + \lambda_{CN}(c_u - \hat{c_u})^2) \tag{6}$$

Where λ_{CN} is a scaling factor and c is the counter value. Once fully trained we pass each codebook embedding, F_i^z, through the decoder to give the learnt dictionary of facial expressions in Euclidean space, $DF = [F_1, F_2, ..., F_{N_f}]$.

3.5 Cutoff Generation

To generate the ground truth cutoffs used in training, we once again apply our stitching approach. For each sequence in the ground truth data, P, we produce the equivalent stitched sequence, P_{stitch}. We then apply a low-pass filter to P_{stitch} within the range of 1 to 25 Hz and measure the intersection and set difference of the frequencies, denoted as $(FT(P) \cap FT(P_{stitch}))$ and $(FT(P) \setminus FT(P_{stitch}))$, where the Fourier transform is represented as $FT()$. Subsequently, we fit a parametric spline curve to the intersection and set difference. To determine the cutoff we find the frequency that maximises the intersection while minimising the set difference. This provides the cutoff frequency for that sequence. We opt to use this method over just analyzing the frequency in the original sequence as we do not have an ideal filter. Thus, the Butterworth filter has unintended effects on frequencies below the cutoff.

4 Experiments

4.1 Implementation Details

We apply the approach to three datasets, the Public Corpus of German Sign Language, 3rd release, the Meine DGS Annotated (mDGS) dataset [15], RWTH-PHOENIX-Weather-2014**T** (PHOENIX14**T**) [4] and the BSL Corpus **T** (BSLCP**T**) [29]. To evaluate our approach we employ a CSLR model (Sign Language Transformers [5]) to conduct back-translation, the same as [10, 24, 28, 40]. BLEU [19], Rouge [17], and chrF [20] scores are computed between the predicted text and the ground truth. Finally, to evaluate the pose we employ Dynamic Time Warping Mean Joint Error (DTW-MJE). In the following experiment, we test two different dictionaries: 1) collected from isolated examples, and 2) a dictionary created from continuous data. Noted as Isolated and continuous in the following tables. Further information about the datasets, dictionaries and model implementation can be found in the supplementary material.

4.2 Quantitative Evaluation

Text-to-Gloss Translation Results: We start by evaluating the T2G translation performance described in Section 3.1. Table 1 shows the performance on all three datasets. We suggest that the difficulty of a dataset is proportional to the vocabulary and the total number of sequences used in training. We find the best performance on PHOENIX14**T** data which has the highest number of sequences per token, achieving 18.11 BLEU-4. In comparison to previous works, we find by having the model predict duration, face and cutoff we can achieve higher BLEU-1 scores, but at the cost of a lower BLEU-4 in comparison to [26]. On the more challenging mDGS dataset we find a considerably lower BLUE-4 score due to the larger domain of discourse. The BSLCP**T** has a smaller domain of discourse in comparison to mDGS but has the fewest sequences per token. Thus, understandably we only achieve a BLEU-4 of 1.67 on the test set. Overall we find the model to be performing as expected.

	TEST SET						DEV SET					
Dataset:	BLEU-1	BLEU-2	BLEU-3	BLEU-4	chrF	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	chrF	ROUGE
BSLCP**T**	26.02	11.19	4.15	1.67	23.54	25.06	26.88	11.40	4.72	1.28	23.59	26.96
mDGS	30.13	13.04	5.45	2.36	29.24	31.43	29.72	12.46	4.89	1.87	28.90	30.90
PHOENIX14**T**	55.48	36.54	25.18	18.11	49.30	53.83	56.55	37.32	25.85	18.74	48.91	54.81
PHOENIX14**T** [26]	55.18	37.10	26.24	19.10	-	54.55	55.65	38.21	27.36	20.23	-	55.41
PHOENIX14**T** [34]	50.67	32.25	21.54	15.26	-	48.10	50.15	32.47	22.30	16.34	-	48.42

Table 1: The results of translating from Text-to-Gloss on the BSL Corpus **T**, RWTH-PHOENIX-Weather-2014**T** and Meine DGS Annotated dataset.

Text-to-Pose Translation Results: Note in the following experiments the back-translation model's performance (shown as GT top row of Table 2, 3 and 4) should be considered the upper limit of performance. In this section, we evaluate the Text-to-Pose (T2P) performance using back translation. To allow for a comparison we train two versions of the PT with the setting presented in [26]. PT is the standard architecture, while PT + GN is trained with Gaussian Noise added to the input. In line with the original paper, we find Gaussian Noise improves the performance, however, our approach still outperforms both models except on DTW-MJE. As discussed previously, other works suffer from regression to the mean caused by the models attempting to minimise their loss function and thus, are incentivised to predict a mean pose. This metric fails to evaluate the content of the sequence, but the higher score does indicate our model is expressive as it is producing sequences further from the mean. For back-translation, we outperform the PT on all metrics. Showing significant improvements in BLEU-1 score of 98% and 269% on the mDGS and BSLCP**T** dev set (comparing PT + GN and Stitcher (continuous), Table 2 and 3).

Deep learning models exhibit a bias toward the data they were trained on and often show poor out-of-domain performance. Unsurprisingly, the performance improves when using the continuous dictionary. We find only a small increase in BLEU-1 of 0.13 on the BSLCP**T** dev set (Table 2), most likely due to the isolated dictionary containing the lexical variants found

BSLCP**T**	TEST SET							DEV SET						
Approach:	DTW-MJE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	chrF	ROUGE	DTW-MJE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	chrF	ROUGE
GT	0.000	17.3	3.96	1.37	0.54	13.00	21.76	0.000	17.32	3.71	1.08	0.39	13.04	21.89
PT [26]	0.288	4.40	0.65	0.18	0.00	5.80	8.22	0.292	4.00	0.61	0.10	0.00	5.69	8.02
PT + GN [26]	0.267	4.96	0.55	0.13	0.00	6.38	8.82	0.258	4.47	0.63	0.09	0.00	6.14	8.89
Stitcher (Isolated)	0.588	16.37	2.86	0.75	0.28	14.07	**20.84**	0.592	16.39	2.82	0.58	0.00	13.9	19.55
Stitcher (continuous)	0.575	**16.99**	**3.65**	**1.03**	**0.41**	**14.32**	20.65	0.573	**16.52**	**3.19**	**0.73**	0.00	**14.34**	**20.53**

Table 2: The results of translating from Text-to-Pose on the BSL Corpus **T** dataset.

mDGS	TEST SET							DEV SET						
Approach:	DTW-MJE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	chrF	ROUGE	DTW-MJE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	chrF	ROUGE
GT	0.000	20.87	5.60	1.89	0.80	17.56	23.78	0.000	20.75	5.43	1.81	0.76	17.63	23.41
PT [code]	0.229	6.11	0.94	0.21	0.05	8.07	8.36	0.228	6.22	0.98	0.17	0.00	8.23	8.44
PT + GN [code]	0.2245	7.18	1.48	0.40	0.01	8.46	8.38	0.2241	9.22	1.63	0.38	0.01	8.94	8.57
Stitcher (Isolated)	0.581	16.63	3.75	0.94	0.22	13.39	21.69	0.592	16.9	3.67	0.95	0.32	13.9	**21.34**
Stitcher (Continuous)	0.637	**18.64**	**4.17**	**1.07**	**0.39**	**16.86**	**21.80**	0.637	**18.27**	**4.07**	**1.19**	**0.43**	**16.75**	21.25

Table 3: The results of translating from Text-to-Pose on the Meine DGS Annotated (mDGS) dataset.

PHOENIX14**T**							
Approach:	DTW-MJE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	chrF	ROUGE
GT	0.000	32.41	20.19	14.41	11.32	33.84	32.96
PT [code]	0.197	6.27	3.33	2.14	1.59	14.52	9.50
PT + GN [code]	0.191	11.45	7.08	5.08	4.04	19.09	14.52
NAT-AT [code]	0.177	14.26	9.93	7.11	5.53	21.87	18.72
NAT-FA [code]	0.146	15.12	10.45	7.99	6.66	22.98	19.43
PoseVQ-MP [code]	0.146	15.43	10.69	8.26	6.98	-	-
PoseVQ-DDM [code]	0.116	16.11	11.37	**9.22**	**7.50**	-	-
Stitching G2P (Isolated)	0.593	21.47	8.79	4.25	2.49	23.74	20.32
Stitching G2P (Continuous)	0.587	23.58	12.31	8.05	5.95	28.85	24.84
Stitching T2P (Isolated)	0.594	22.78	9.68	5.17	3.12	24.27	21.30
Stitching T2P (Continuous)	0.572	**25.14**	**13.54**	8.98	6.67	**29.5**	**26.49**

Table 4: The results of translating from Gloss-to-Pose (G2P) and Text-to-Pose (T2P) on the RWTH-PHOENIX-Weather-2014**T** test set.

in the original data. Whereas we see a larger increase on the mDGS dataset (Table 3).

Previous work has primarily focused on G2P translation, therefore to facilitate a meaningful comparison we present two versions of the model. First, a G2P version, where we use the ground truth data and just apply the stitching module, and, secondly our T2P approach (translation then stitching). Results for comparison are provided by [40]. Note that the previous approaches do not use a dictionary of signs and instead attempt to regress the pose directly from the spoken language. We find our approach outperforms previous work on the BLEU-1 to 2 scores increasing the score by 56% and 19%, respectively (comparing Table 4, row 7 and 11). We also find significant improvement in ROUGE and chrF metrics.

Using a continuous dictionary we can outperform all models except for the Vector Quantisation (VQ) based approaches on BLEU-3 to 4. As the VQ model is learning sub-units of a gloss sequence we suggest this gives it an advantage on higher n-grams, as each token that is predicted can represent multiple signs.

4.3 Qualitative Evaluation

Visual Outputs: To demonstrate the approach's effectiveness, we present skeleton and video outputs for two sign languages (BSL and DGS)[1]. Furthermore, in the supplementary material, we share visualizations of the produced sequence.

Survey Results: The survey presented a comparison to PT, followed by an ablation of different components of the stitching approach. 17% of people surveyed were native Deaf signers, while 34% were L2 signers or language learners. 87.5% preferred our approach compared to the PT, while the rest selected no preference. 100% of people agreed that applying the filter improved the realism compared, while resampling was found to be less important, with 37.5% selecting no preference between the resampled and normal sequence.

[1] https://github.com/walsharry/Sign_Stitching_Demos

5 Conclusion

In this paper, we presented a novel approach to SLP. Previous works have suffered from the problem of regression to the mean and have mainly focused on manual features. Here we have overcome the problem by using a dictionary of expressive examples. The stitching effectively joins the signs together creating a natural continuous sequence and by clustering facial expressions into a vocabulary we can create a sequence that contains both manual and non-manual features. We eliminated unnatural transitions and enhanced the stylistic cohesiveness through the approach. As a result, we present state-of-the-art performance. Finally, the user evaluation agrees with the quantitative results, indicating our approach can produce realistic expressive Sign language.

6 Acknowledgement

We thank Adam Munder, Mariam Rahmani, and Abolfazl Ravanshad from OmniBridge, an Intel Venture. We also thank Thomas Hanke and the University of Hamburg for the use of the Meine DGS Annotated (mDGS) data. This work was supported by Intel, the SNSF project 'SMILE II' (CRSII5 193686), the European Union's Horizon2020 programme ('EASIER' grant agreement 101016982) and the Innosuisse IICT Flagship (PFFS-21-47). This work reflects only the author's view and the Commission is not responsible for any use that may be made of the information it contains.

References

[1] J Andrew Bangham, SJ Cox, Ralph Elliott, John RW Glauert, Ian Marshall, Sanja Rankov, and Mark Wells. Virtual signing: Capture, animation, storage and transmission-an overview of the visicast project. In *IEE Seminar on speech and language processing for disabled and elderly people (Ref. No. 2000/025)*, pages 6–1. IET, 2000.

[2] Diane Brentari. *A prosodic model of sign language phonology*. Mit Press, 1998.

[3] Stephen Butterworth et al. On the theory of filter amplifiers. *Wireless Engineer*, 7(6): 536–541, 1930.

[4] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. Neural sign language translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7784–7793, 2018.

[5] Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard 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*, pages 10023–10033, 2020.

[6] Stephen Cox, Michael Lincoln, Judy Tryggvason, Melanie Nakisa, Mark Wells, Marcus Tutt, and Sanja Abbott. Tessa, a system to aid communication with deaf people. In *Proceedings of the fifth international ACM conference on Assistive technologies*, pages 205–212, 2002.

[7] Eleni Efthimiou, Stavroula-Evita Fotinea, Thomas Hanke, John Glauert, Richard Bowden, Annelies Braffort, Christophe Collet, Petros Maragos, and François Lefebvre-Albaret. The dicta-sign wiki: Enabling web communication for the deaf. In *Computers Helping People with Special Needs: 13th International Conference, ICCHP 2012, Linz, Austria, July 11-13, 2012, Proceedings, Part II 13*, pages 205–212. Springer, 2012.

[8] Oussama ElGhoul and Mohamed Jemni. Websign: A system to make and interpret signs using 3d avatars. In *Proceedings of the Second International Workshop on Sign Language Translation and Avatar Technology (SLTAT), Dundee, UK*, volume 23, 2011.

[9] Alexis Heloir and Sylvie Gibet. A qualitative and quantitative characterisation of style in sign language gestures. In *International Gesture Workshop*, pages 122–133. Springer, 2007.

[10] Wencan Huang, Wenwen Pan, Zhou Zhao, and Qi Tian. Towards fast and high-quality sign language production. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3172–3181, 2021.

[11] Eui Jun Hwang, Huije Lee, and Jong C Park. Autoregressive sign language production: A gloss-free approach with discrete representations. *arXiv preprint arXiv:2309.12179*, 2023.

[12] Maksym Ivashechkin, Oscar Mendez, and Richard Bowden. Improving 3d pose estimation for sign language. In *2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, pages 1–5, 2023. doi: 10.1109/ICASSPW59220.2023.10193629.

[13] Richard Kennaway. Avatar-independent scripting for real-time gesture animation. *arXiv preprint arXiv:1502.02961*, 2015.

[14] Oscar Koller, Jens Forster, and Hermann Ney. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding*, 141:108–125, 2015. ISSN 1077-3142. doi: https://doi.org/10.1016/j.cviu.2015.09.013. URL https://www.sciencedirect.com/science/article/pii/S1077314215002088. Pose & Gesture.

[15] Reiner Konrad, Thomas Hanke, Gabriele Langer, Dolly Blanck, Julian Bleicken, Ilona Hofmann, Olga Jeziorski, Lutz König, Susanne König, Rie Nishio, Anja Regen, Uta Salden, Sven Wagner, Satu Worseck, Oliver Böse, Elena Jahn, and Marc Schulder. Meine dgs – annotiert. öffentliches korpus der deutschen gebärdensprache, 3. release / my dgs – annotated. public corpus of german sign language, 3rd release, 2020. URL https://doi.org/10.25592/dgs.corpus-3.0.

[16] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

[17] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

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

[19] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[20] Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In Ondřej Bojar, Rajan Chatterjee, Christian Federmann, Barry Haddow, Chris Hokamp, Matthias Huck, Varvara Logacheva, and Pavel Pecina, editors, *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/W15-3049. URL https://aclanthology.org/W15-3049.

[21] Siegmund Prillwitz and Hamburg Zentrum für Deutsche Gebärdensprache und Kommunikation Gehörloser. *HamNoSys: Version 2.0; Hamburg notation system for sign languages; an introductory guide*. Signum-Verlag, 1989.

[22] Razieh Rastgoo, Kourosh Kiani, Sergio Escalera, and Mohammad Sabokrou. Sign language production: A review. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3451–3461, 2021.

[23] Judy S Reilly, Marina L McIntire, and Howie Seago. Affective prosody in american sign language. *Sign Language Studies*, pages 113–128, 1992.

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

[25] Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. Everybody sign now: Translating spoken language to photo realistic sign language video. *arXiv preprint arXiv:2011.09846*, 2020.

[26] Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. Progressive transformers for end-to-end sign language production. In *European Conference on Computer Vision*, 2020.

[27] Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. Continuous 3d multi-channel sign language production via progressive transformers and mixture density networks. *International journal of computer vision*, 129(7):2113–2135, 2021.

[28] Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. Mixed signals: Sign language production via a mixture of motion primitives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1919–1929, 2021.

[29] Adam Schembri. British sign language corpus project: Open access archives and the observer's paradox. In *sign-lang@ LREC 2008*, pages 165–169. European Language Resources Association (ELRA), 2008.

[30] Stephanie Stoll, Necati Cihan Camgöz, Simon Hadfield, and Richard Bowden. Sign language production using neural machine translation and generative adversarial networks. In *Proceedings of the 29th British Machine Vision Conference*, 2018.

[31] Stephanie Stoll, Necati Cihan Camgoz, Simon Hadfield, and Richard Bowden. Text2sign: towards sign language production using neural machine translation and generative adversarial networks. *International Journal of Computer Vision*, 128(4): 891–908, 2020.

[32] Shinichi Tamura and Shingo Kawasaki. Recognition of sign language motion images. *Pattern Recognition*, 1988.

[33] Shengeng Tang, Richang Hong, Dan Guo, and Meng Wang. Gloss semantic-enhanced network with online back-translation for sign language production. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 5630–5638, 2022.

[34] Mohammad Hassan Vali and Tom Bäckström. Nsvq: Noise substitution in vector quantization for machine learning. *IEEE Access*, 10:13598–13610, 2022.

[35] Desmond Eustin Van Wyk. Virtual human modelling and animation for real-time sign language visualisation. 2008.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[37] Lucas Ventura, Amanda Duarte, and Xavier Giró-i Nieto. Can everybody sign now? exploring sign language video generation from 2d poses. *arXiv preprint arXiv:2012.10941*, 2020.

[38] Ronnie B Wilbur. Stress in a sl: Empirical evidence and linguistic i ssues. *Language and speech*, 42(2-3):229–250, 1999.

[39] Ronnie B Wilbur. Effects of varying rate of signing on asl manual signs and nonmanual markers. *Language and speech*, 52(2-3):245–285, 2009.

[40] Pan Xie, Qipeng Zhang, Zexian Li, Hao Tang, Yao Du, and Xiaohui Hu. Vector quantized diffusion model with codeunet for text-to-sign pose sequences generation. *arXiv preprint arXiv:2208.09141*, 2022.

[41] Pan Xie, Qipeng Zhang, Peng Taiying, Hao Tang, Yao Du, and Zexian Li. G2p-ddm: Generating sign pose sequence from gloss sequence with discrete diffusion model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 6234–6242, 2024.

[42] Inge Zwitserlood, Margriet Verlinden, Johan Ros, Sanny Van Der Schoot, and T Netherlands. Synthetic signing for the deaf: Esign. In *Proceedings of the conference and workshop on assistive technologies for vision and hearing impairment (CVHI)*, 2004.