Using Sign Language Production as Data Augmentation to enhance Sign Language Translation

Harry Walsh
harry.walsh@surrey.ac.uk
University of Surrey
Guildford, United Kingdom

Maksym Ivashechkin
m.ivashechkin@surrey.ac.uk
University of Surrey
Guildford, United Kingdom

Richard Bowden
r.bowden@surrey.ac.uk
University of Surrey
Guildford, United Kingdom

Abstract

Machine learning models fundamentally rely on large quantities of high-quality data. Collecting the necessary data for these models can be challenging due to cost, scarcity, and privacy restrictions. Signed languages are visual languages used by the deaf community and are considered low-resource languages. Sign language datasets are often orders of magnitude smaller than their spoken language counterparts. Sign Language Production (SLP) is the task of generating sign language videos from spoken language sentences, while Sign Language Translation (SLT) is the reverse translation task. Here, we propose leveraging recent advancements in SLP to augment existing sign language datasets and enhance the performance of SLT models. For this, we utilize three techniques: a skeleton-based approach to production, sign stitching, and two photo-realistic generative models, SignGAN and SignSplat. We evaluate the effectiveness of these techniques in enhancing the performance of SLT models by generating variation in the signer's appearance and the motion of the skeletal data. Our results demonstrate that the proposed methods can effectively augment existing datasets and enhance the performance of SLT models by up to 19%, paving the way for more robust and accurate SLT systems, even in resource-constrained environments.

Keywords

Sign Language Translation, Data Augmentation, Sign Language Production, Generative Models

ACM Reference Format:

Harry Walsh, Maksym Ivashechkin, and Richard Bowden. 2025. Using Sign Language Production as Data Augmentation to enhance Sign Language Translation. In *Proceedings of* . ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Sign languages are visual languages that use multiple articulators, such as the hands, body, and facial expressions to convey meaning. They are natural languages used by the Deaf community and possess their own grammar and syntax [69]. Sign Languages can be classified as low-resource languages. Given their visual nature, recording and annotating them poses a significant challenge, especially in the quantities comparable to the spoken language domain. Acquiring real-world data presents numerous challenges, such as high collection and labelling costs, scarcity, and privacy restrictions [5]. Given that machine learning models rely on large quantities of high-quality data for training, this has been a limiting factor in computational Sign language research.

Sign language datasets are orders of magnitude smaller than their spoken language counterparts [5]. For instance, a common dataset used as a baseline in the field, the RWTH-PHOENIX-Weather-2014**T** (PHOENIX14**T**) dataset, only contains approximately 9,000 sentences [7]. While larger datasets exist, the linguistic annotation, which is required for state-of-the-art approaches, is often missing, incomplete, or non-existent [2]. Primarily due to the costs and expertise required to create such annotations.

A prevalent solution to address data scarcity is to augment existing datasets. Data augmentation encompasses a variety of techniques designed to artificially increase the size and diversity of training samples without collecting new data [17]. In contrast, an alternative approach is to generate synthetic data. Synthetic data is defined as artificially generated information that mimics the statistical properties and patterns of real-world data, produced via computational methods such as algorithms, simulations, or increasingly sophisticated generative AI models. Data augmentation and generation have become widely adopted practices in various domains, such as Natural Language Processing (NLP) [67].

As illustrated in Figure 1, SLT is the task of predicting a spoken language translation from a sign language video, Sign-to-Text (S2T). Whereas, SLP is the reverse task, aiming to generate sign language videos given spoken language sentences, Text-to-Sign (T2S). In this paper, we leverage three SLP techniques to augment and generate synthetic data [36, 61, 76]. We then leverage this data to enhance the translation ability of SLT models [7, 83]. We note the limitations of these approaches, as they often lack non-manual features, and significantly more work is required for the approaches to fully capture the subtleties of the language. However, we suggest that they capture enough of the manual features to be a useful tool for pre-training or to supplement data during training.

Some SLT approaches perform translation solely from video [77, 83], Video-to-Text (V2T), and others leverage a skeleton pose representation [9, 19, 72], Pose-to-Text (P2T). Our first technique focuses on enhancing P2T architectures and therefore synthesises synthetic skeleton data. For this, we leverage sign stitching [76], an approach that uses a dictionary of pre-recorded isolated signs and joins them together to create continuous sign language sequences. Previous research has suggested many problems with alternative

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

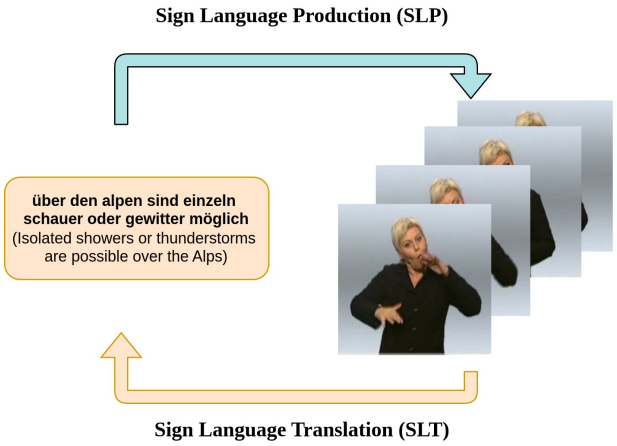

Figure 1: A visual overview of the SLT and SLP tasks. The SLT task is the process of translating sign language video into spoken language text. The SLP task is the reverse, generating sign language video from spoken language text.

methods that attempt to directly regress a sequence of poses from the spoken language text [63]. Mainly regression to the mean, which causes under-articulated and often incomprehensible signing. For this reason, we choose a stitching approach as it is guaranteed to produce expressive natural sequences.

The next two techniques focus on the V2T architectures for which both a Generative Adversarial Network (GAN) [64] and Gaussian Splatting [36] approaches are tested. Both methods use skeleton pose to generate a photo-realistic signer. GANs often struggle to generalise to out-of-domain poses and suffer from noisy conditioning. This can lead to several artefacts in production. Whereas, a Gaussian Splatting-based approach that uses SMPL-X mesh in addition to hammer features provides additional constraints on the generation process, resulting in fewer artefacts. Previous works emphasised that the quality of the pre-training data is paramount in achieving strong performance in downstream tasks [18, 47] and low-quality data can even harm model performance [66, 68].
We define the contributions of this paper as follows;

(1) To the best of our knowledge, we are the first to propose generating synthetic skeleton sequences for SLT.
(2) We propose augmenting the appearance of sign language data, testing two photo-realistic avatar models, SignGAN [64] and SignSplat [36].
(3) We evaluate the effectiveness of these techniques in enhancing the performance of SLT models, demonstrating that they can significantly improve translation accuracy.

The rest of this paper is organized as follows. Section 2 reviews related works in the field of data augmentation, generation, SLP, and SLT. Section 3 elaborates on the methodology for augmenting the data and the translation models used to verify its effectiveness. Section 4 details the evaluation protocol before moving on to the experimental results. Finally, Section 5 concludes the paper and suggests directions for future work.

2 Related Work

Data augmentation and generation have enhanced machine learning models. We review approaches from other fields that leverage a range of techniques to improve system performance. Before moving on to the field of SLP, which we leverage to produce synthetic skeleton data and augment the appearance of the signers. Finally, we review the SLT models that we use to evaluate the effectiveness of the proposed techniques.

2.1 Data Augmentation and Generation

Synthetic data is commonplace where the rules and constraints are well understood [12, 16, 24]. This can also help deal with rare scenarios where capturing additional data is expensive or unobtainable due to privacy concerns. In the related field of action recognition, simulated data was shown to be very effective and helped boost the recognition performance when fine-tuned on real data [39]. Statistical methods have been used to generate data by learning the underlying distribution of the real data. These methods often involve techniques such as Monte Carlo Simulation, Hidden Markov Models (HMMs), and other probabilistic models [26]. However, more recently, they have been replaced with deep learning based approaches that have shown superior performance.

Deep learning based approaches have utilised a variety of models in order to augment and generate data, such as GAN [20], Variational Autoencoders (VAE) [40], Diffusion [28], and transformer-based models [74]. All of which have been shown to generate high-quality synthetic data [26]. GANs and diffusion models have been used to generate high-quality images and videos for several applications like medical imaging [13], object detection [54], computer vision [29], etc. Large Language Models (LLMs) are transformer-base models, that are commonly used to generate synthetic data for various tasks such as knowledge distillation [22, 27], mathematical reasoning [50, 81], coding [32], or even refining its own predictions. For instance, LLAMA 3.1 is fine-tuned on synthetic data to tailor the output to a desired style [23].

Given that sign language is classified as a low-resource language, authors have previously attempted to augment monolingual text data to generate synthetic gloss[1] sequences. Gloss is a commonly used text intermediary for SLP and SLT, but is costly and time-consuming to create, and has been noted as a limiting factor in expanding state-of-the-art approaches to larger domains of discourse. Moryossef et al. [53] focused on Gloss-to-Text (G2T) translation, while Yao et al. [80] focused on the reverse, Text-to-Gloss (T2G). Both approaches used a deep translation model and a rule-based system to generate gloss sequences from spoken language sentences. Recently, Abdullah et al. leveraged GPT-4o, a LLMs to generate gloss sequences from spoken language sentences. Using this data, they showed improved performance [1]. The approaches were able to show the benefits of utilising synthetic data. As the field advances toward end-to-end methods, which do not require linguistic annotations, the relevance of such approaches has been reduced. Therefore, we propose creating synthetic skeleton sequences and appearance augmentation for enhanced SLT.

When augmenting data, there are two main approaches to integrating it into a model. Some methods suggest first pre-training on

[1]Gloss is the written word associated with each sign performed in a sequence.

Figure 2: An example from the PHOENIX14T dataset, showing left to right: skeleton pose, original video, SignSplat Avatar, and SignGAN Avatar.

the augmented data, followed by a second stage of fine-tuning on real data at a lower learning rate. Others propose simultaneously training on both real and augmented data [17]. In this work, we experiment with both of these approaches.

2.2 Sign Language Production

SLP is the task of generating realistic sign language sequences from spoken language text. To accomplish this, several intermediate representations can be employed within the translation pipeline. A common approach first generates a gloss representation, followed by a skeleton pose production that is used to drive a photo-realistic signer [65, 70, 71, 76]. An example of the skeleton pose representation can be seen in Figure 2

Early approaches to SLP used a graphical avatar-driven systems[4, 11, 14, 15, 84]. Some of the approaches looked for legal phrases, and, thus are limited to pre-recorded phrases [11]. An alternative approach performed a translation to a linguistic notation, then opted to play each sign in sequence with unnatural transitions in between [4]. The avatar was often unrealistic; and as a result, these early approaches were unpopular with the deaf community [41].

Deep learning based approaches have improved the realism of the signing sequences. Initially being tackled with Recurrent Neural Networks (RNNs) [70, 82], later being improved upon using transformer based architectures [60, 62, 63]. However, models that attempt to directly regress skeleton pose often suffer from regression to the mean. This is caused by the model attempting to minimise their loss function and therefore, they result in under-articulated and incompressible signing. Given that these models can also hallucinate, we avoid them for the P2T task.

Alternatively, other methods used a combination of deep learning and a pre-recorded dictionary of isolated signs. These methods attempt to learn the co-articulation between isolated signs in order to create fully continuous natural sequences. Walsh et al. [76] used a 7-step pipeline that included non-manual features. By including features such as the timing as well as the frequency components of a sequence, the method is able to capture signed prosody[2], this is utilized in Section 4.2.1 to generate our synthetic skeleton data. Other methods instead opt to use a diffusion model [73] or a transformer [79] to predict the transitions between isolated signs. Saunders et al. [65] proposed a keyframe selection network to select specific frames from isolated signs, and by concatenating and interpolating between them, were able to produce continuous sequences. The output was used to drive the SignGAN model, a model capable of generating photo-realistic sign language video given skeleton pose and a style image. The model is trained through a min-max game performed by the generator and discriminator. Here, the SignGAN model is used to generate appearance variations for the V2T task in Section 4.2.2.

Taking inspiration from computer graphics and the field of 3D rendering, Ivashechkin et al. [36] proposed a Gaussian Splatting based approach to generate photorealistic sign language video by attaching 3D Gaussian splatting primitives to the SMPL-X [56] human mesh model. While alternative Gaussian splatting approaches [30, 52] also tackle the problem of expressive avatar rendering, the *SignSplat* approach adds regularization and constraints to the underlying mesh geometry and appearance to minimize rendering artifacts and enforce the physical limits of a human. Furthermore, proposing a sign-stitching mechanism for gloss interpolation. Once again, we leverage this method in Section 4.2.2. Both a SignGAN and a SignSplat avatar are shown on the right side of Figure 2, respectively.

2.3 Sign Language Translation

Initially, the field focused on isolated Sign Language Recognition (SLR), which aims to produce the corresponding gloss for a short video containing a sign [34, 37, 44]. Later, the field progressed to the more challenging task of Continuous SLR, which requires multiple signs to be recognised within a sequence [25, 31, 42, 51]. SLT requires an additional translation to spoken language.

Similar to the SLP field, the task was initially tackled using an RNN [7], before moving on to a transformer-based architecture [8] which employed a Connectionist Temporal Classification (CTC) loss on the encoder to simultaneously learn SLR and SLT in a single model. Modifying the network to work with skeleton keypoints, has become the standard for evaluating skeleton based SLP [60, 62, 63, 75, 76]. Therefore, we use this architecture in the P2T experiments in Section 4.2.1. Other approaches have made use of both skeleton keypoints and video features in a single model [9, 45]. While some rely solely on keypoints as features [19, 72]. Skeleton keypoints are appearance agnostic, while generally being computationally

[2]the natural rhythm, stress and intonation used to convey additional meaning in sign language

inexpensive when compared to video-based models. However, they can be susceptible to noise.

Several other video-based approaches have been proposed. Wong et al. were able to leverage the power of LLMs and adapt them for the use of sign language translation [77]. While Zhou et al. [83] initialised their models using pre-trained visual and language encoders. Using Contrastive Language-Image Pre-Training (CLIP) [58] the approach was able to learn an effective representation that helps in the downstream translation task. We adopt this architecture when experimenting with our V2T approach in Section 4.2.2.

It is commonplace for translation models to apply visual augmentations to the training data in order to make them more robust to colour variations and signer placement in the video. Such strategies include colour jittering, random Gaussian noise plus random, cropping, rotation, and scaling [3, 77, 78, 83]. However, to the best of our knowledge, we are the first to augment the appearance entirely to a new signer and use this data for training a SLT model.

3 Methodology

First, we explain the SLP techniques that are used to augment sign language data. We then describe the SLT models that are used to conduct the translation experiments.

3.1 Data Augmentation Techniques

The three proposed augmentation strategies can be categorised into two types: First, skeleton pose augmentation, where we use MediaPipe skeletons as a representation for sign language. By generating new synthetic sequences from isolated signs, we can create variations in the lexical form of the signs, in addition to altering the speed and order. Second, photo-realistic augmentations, where we use generative models to produce realistic sign language videos. This approach allows us to augment the appearance of the signer and create more variations in camera angle and background.

3.1.1 Sign Stitching. For a given dataset, we construct a dictionary of the performed signs. This dictionary comprises, for each sign, a video of an isolated sign and its corresponding spoken language gloss tag. From each video, we extract a 3D skeleton pose using the method described in [35]. This method employs inverse kinematics and a neural network to uplift a 2D MediaPipe pose to 3D. Solving for joint angles allows this method to enforce physiological constraints of the human body. The dictionary is stored as sequences of joint angles, a representation that facilitates the application of a canonical skeleton during pose generation. This ensures consistent bone lengths across all signers, irrespective of the original performer's physique.

Let the dictionary be denoted as $\mathcal{D}$, where each entry maps a gloss y to its corresponding sequence of angles A. This can be expressed as:

$$\mathcal{D} = \{(y_i, A_i) \mid y_i \in \mathcal{Y}, A_i = (a_{i,1}, a_{i,2}, \ldots, a_{i,U_i})\},$$

where $\mathcal{Y}$ is the set of all unique glosses in the dataset, and A_i is the sequence with length U_i for gloss y_i. If a specific sign is not available in $\mathcal{D}$, we employ a word embedding model to vectorise its gloss tag. Then we select the most similar sign from the dictionary based on the highest cosine similarity in the embedding space.

Sequence Generation - Given an input sequence of glosses $Y = (y_1, y_2, \ldots, y_G)$, their corresponding durations $D = (d_1, d_2, \ldots, d_G)$, each of length G, and a low-pass cutoff frequency C specified per sequence, we generate a continuous sequence of 3D skeleton poses $P = (p_1, p_2, \ldots, p_U)$ with U frames. The per-gloss durations d_i in D can be approximated using the method in [76]. We summarise this process into three steps:

1. Sign Retrieval: The angular sequences A_i for each gloss y_i in Y are retrieved from $\mathcal{D}$. These are then converted from their angular representation to 3D skeleton poses by applying the canonical skeleton. Concurrently, each sign's pose sequence is resampled to match its specified duration d_i from D.

2. Stitching: The resampled isolated sign pose sequences are concatenated. To ensure smooth coarticulation between the end of one sign and the start of the subsequent sign, transitions are generated using linear interpolation. The number of transition frames is determined by the distance between boundary poses, with the constraint that the transitional movement velocity remains consistent with, or bounded by, the velocities at the sign boundaries.

3. Motion Filtering: Finally, the entire stitched sequence of poses P is processed using a low-pass Butterworth filter [6] with the predefined cutoff frequency C. It is observed that natural signing exhibits distinct motion frequency ranges corresponding to sharpness or smoothness. This filtering step aims to remove sharp, unnatural movements not present in the original data, thereby emulating motion characteristics of the original signer.

Additional Augmentations - To introduce further variation into the skeleton data, two additional augmentation methods are implemented; First, gloss order permutation, where we apply random permutations to the input gloss order Y. Second, speed variation, we vary the total number of frames U in the generated sequence P, which effectively alters its performance speed.

3.1.2 SignGAN. Given a pose sequence, $P = (p_1, p_2, \ldots, p_U)$, the model aims to generate the corresponding video of a photorealistic signer, $V = (v_1, v_2, \ldots, v_U)$ with U frames. The model is trained using a GAN architecture [21] that comprises a generator and discriminator network. The generator takes the pose sequence as input and generates the corresponding video frames. We train the model on a single signer's appearance. Using a Convolutional Neural Network (CNN), we extract features from a style image and fuse them into the generator at multiple layers. While the discriminator evaluates the realism of the generated frames against real video frames. The generator is an encoder-decoder architecture that consists of a series of convolutional layers. The model uses residual connections to preserve spatial information and improve the quality of the generated frames.

3.1.3 SignSplat. We train a Gaussian splatting [38] model on multi-view capture data. We primarily utilized the framework from [36], which exploits the SMPL-X [56] human body parameterization with manually updated human constraints, especially for the hand joints (removing redundant degrees of freedom and limiting the angular range to be more realistic). To exploit the Gaussian splatting human appearance model for 3D reconstruction from a video input, we propose the following pipeline. First, we process an input image with MMPose [10] to obtain 2D OpenPose keypoints. Second, we estimate the intrinsic matrix with fixed focal length and principal

point based on the image resolution. We then run inverse kinematics optimization to fit the SMPL-X parameters to the 2D keypoints. This involves generating an SMPL-X mesh, applying linear blend skinning to obtain a 3D skeleton, projecting it using the estimated intrinsics, and minimizing the reprojection error.

Due to single-view ambiguities, the 3D reconstruction primarily suffers from poor hand reconstruction, because 2D hand keypoints do not carry depth information. To mitigate this issue, for the PHOENIX14**T** dataset, we exploited the HaMeR [57] hand angle parameters reconstruction for our initialization. The HaMeR hand angles (in MANO [59] format) have a good 3D prior and can be directly transferred to the SMPL-X mesh. Consequently, in the overall inverse-kinematics optimization, we only fine-tune the hands with a smaller learning rate. Such optimization leads to a better correspondence of the 3D mesh to 2D detections and improved hand-to-hand interaction, essential for finger-spelling. Once the SMPL-X parameters are optimized, we drive our Gaussian Splatting model to render the signer in real time.

3.2 Sign Language Translation (SLT) Models

Both V2T and P2T SLT models are transformer-based encoder-decoder architectures. We utilise Sign Language Transformers [8] for the P2T task, and GF-SLT [83] for the V2T task. We detail the difference in the models below.

3.2.1 Sign Language Transformers. This approach is commonly used in SLP work [33, 60, 62–64, 75, 76], hence we use it for the P2T task. The encoder processes the input skeleton pose sequence, P, and is supervised using a CTC loss so that it performs SLR. The decoder autoregressively predicts the spoken language translation, $X = (x_1, x_2, \ldots, x_W)$ with W words. Therefore, the model learns the conditional probability, $p(X|P)$.

3.2.2 GF-SLT. For the V2T we use the publicly available GF-SLT architecture. Given that this model comes with a pre-trained vision encoder and language encoder, we use this architecture for computational efficiency. Once again, the model follows the same transformer encoder-decoder architecture. The encoder processes a sequence of Video frames, V, and the decoder autoregressively predicts the spoken language translation, X. Therefore, the model learns the conditional probability, $p(X|V)$. The decoder's embedding layers are initialised from a pre-trained multilingual BART model for improved embeddings.

Furthermore, this model employs Visual Language Pre-training (VLP), which uses a contrastive learning framework, to learn the alignment between video and text features. This creates a shared embedding space between the two encoders, which is shown to improve the performance in the downstream translation task.

4 Experiments

4.1 Experimental Setup

Dictionary - We collect our isolated dictionary for German Sign Language - Deutsche Gebärdensprache (DGS), from a range of sources, such as [43]. In total, we collect a DGS sign vocabulary of 7,206 signs to experiment with. Given the PHOENIX14**T** dataset as a gloss vocabulary of 1,066. We are able to cover the vast majority of the vocabulary. However, we note the lack of specific place names.

Dataset - To test our approach, we utilize the PHOENIX14**T** dataset [7]. The dataset consists of 8,257 signed sentences in DGS, each labelled at the gloss level and translated into German. The dataset is divided into training, development, and test sets, at a ratio of 80:10:10, respectively.

Skeleton Keypoint - The skeleton keypoint representation comes from MediaPipe [49] and consists of 61 keypoints. 21 for each hand, 9 for the body, and 10 for the face. The 61 2D keypoints are uplifted to 3D using the method previously described [35]. From the 3D skeleton, we solve for joint angles, which correspond to 104 angles. Each joint node in the skeleton contains between one to three degrees of freedom. We extract this representation for both the dictionary and the dataset. The dataset extraction is used to drive the signGAN model and train the P2T model.

Evaluation - We evaluate the performance of our models using the Bilingual Evaluation Understudy (BLEU) [55] and Recall-Oriented Understudy for Gisting Evaluation (ROUGE) score [46], which is a widely used metric for evaluating machine translation systems. Both scores measure the similarity between the generated translations and the reference translations. BLEU score breaks down the translation into its word-level n-grams and calculates the precision of each, thus, we report BLEU-1 to 4 scores. We report the same metrics for both the P2T and V2T tasks.

SignGAN - The generator network is trained for 68,000 iterations on a single signer. The generator network has 170,363,715 parameters, while the discriminator network has 5,535,618 parameters. The model is trained to output video at a resolution of 1080p. However, in line with the original PHOENIX14**T** dataset, the videos are post-processed to a resolution of 256×256 at 25 FPS.

SignSplat - The Gaussian splatting appearance was trained on around 800 diverse frames with six camera views. The signer is rendered at 25 FPS at a resolution of 256×256. For SMPL-X reconstruction, the complete optimisation takes approximately 20–40 seconds per short video (100–300 frames) on a NVIDIA GeForce RTX 3090.

P2T Model - For this, we utilise the publicly available "Sign Language Transformers" [8], the same as [33, 60, 62–64, 75, 76]. We subsample the skeleton sequence to 12 frames per second for computational efficiency, and we normalise the skeleton such that the neck is set on the origin and the body is fixed on the xy-plane. The model is trained for 200 epochs with a batch size of 256 and a learning rate of 10^{-3}. We construct the encoder and decoder to be symmetrical, both containing 8 heads and 3 layers, with an embedding dimension and feedforward size of 512. During training, dropout was utilized with a probability of 0.1, and the optimum beam size and alpha were found to be 3 and -1, respectively. This model employs a reduced on-plateau scheduler with a patience of 5 and a decrease factor of 0.8.

V2T Model - For this we utilize the publicly available "GF-SLT" [83]. In line with the original paper, we perform 80 epochs of VLP. Here, the model employs a cosine scheduler with a warm-up of 10^{-6} and a learning rate of 5^{-3}. Following the VLP phase, we perform an additional 200 epochs on the translation task, translating from V2T. Employing the same learning rate scheduler, with an initial rate of 10^{-2}. The text encoder is initialised from a multilingual BART model [48], which consists of 12 layers and is pre-trained on 25

Table 1: The results of using sign stitching for data augmentation on the RWTH-PHOENIX-Weather-2014T dataset.

a) Baseline approach and different training strategies. The original data used in training is labelled as "GT" (ground truth).

	TEST SET					DEV SET				
Training Data:	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE
GT	32.41	20.19	14.41	11.32	32.96	32.36	20.02	14.25	11.13	33.46
Stitched	16.74	6.88	4.53	3.49	17.72	16.78	7.67	5.29	4.21	18.34
GT + Stitched	31.84	19.45	13.71	10.73	32.04	31.19	19.63	14.26	11.37	32.57
GT + Stitched Pre-Training	**37.86**	**24.36**	**17.58**	**13.68**	**37.61**	**37.26**	**23.71**	**17.13**	**13.52**	**37.95**

b) Stitched data pre-training with N random permutations in sign order.

	TEST SET					DEV SET				
Permutations:	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE
0	**37.86**	**24.36**	17.58	13.68	37.61	**37.26**	23.71	17.13	13.52	37.95
1	36.22	22.84	16.22	12.65	36.09	36.26	22.98	16.45	12.76	36.49
3	37.04	24.22	**17.69**	**13.89**	**37.83**	37.18	**24.07**	**17.57**	**13.81**	**38.00**
10	37.67	24.18	17.23	13.24	37.36	36.47	23.26	16.83	13.16	36.91

c) Stitched data pre-training with variations in the duration of each sequence.

	TEST SET					DEV SET				
Duration scale:	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE
0.5	35.75	22.49	15.84	12.27	35.94	34.28	21.05	14.89	11.53	34.94
0.7	37.23	24.65	18.09	14.29	38.39	37.67	**25.43**	**19.17**	**15.38**	39.05
1.1	36.24	23.31	16.94	13.27	36.55	36.49	23.33	16.88	13.13	36.84
1.5	**38.61**	**25.66**	**18.75**	**14.71**	**38.91**	**38.28**	24.97	18.13	14.08	**39.07**
0.7 + 1.0 + 1.5	37.60	23.64	16.56	12.65	36.89	36.67	23.1	16.66	13.02	36.92

languages. Whereas the visual encoder consists of multiple layers of 2D and 1D convolutions, with ReLU and max-pooling.

4.2 Quantitative Results

4.2.1 Pose-to-Text. Table 1.a shows the results of the P2T task. We observe that the SLP techniques significantly improve the performance of the SLT model. Utilising the skeleton-based augmentation for pre-training outperforms the baseline model by a large margin, achieving an BLEU-1 score of 37.86 compared to 32.41 for the baseline. Higher n-grams also show similar results, with a BLEU-4 score increase of 2.36 on the test set. This suggests that pre-training the model using different lexical variants not only allows the model to better recognise individual signs, but also improves the model's ability to understand the grammatical order in which they occur, as indicated by the increased BLEU-4 score.

Row 2 of Table 1.a shows the results of training the model solely on the stitch data and then testing it on the original PHOENIX14**T** test set. Given the model has never seen real continuous sign language data, the results are impressive and indicate the stitch sequences contain features that are in line with the original. We find that jointly training on both the real and the stitched data is detrimental to the model's performance on most metrics. Providing only marginal improvement for BLEU-3 and 4 scores on the dev set. Overall, the best performance is achieved with a pre-training phase followed by fine-tuning on the real data. It provides at least 15% improvement for all metrics, compared to the model trained only on real data. Therefore, the following two tables, Table 1.b and c, employ this pre-training strategy.

In Table 1.b we experiment with randomly permuting the order of N sequential signs in the pre-training data. We find that this has a marginal detrimental effect on the model's BLEU-1 score, decreasing it by up to 4%. However, permuting up to 3 glosses in a sequence gives marginal improvements in the higher N-grams BLEU scores, indicating that creating small variations in the grammatical ordering makes the model more robust. However, permuting more than 3 glosses in a sequence results in a drop in performance. This suggests that the model is sensitive to the order of signs and that the grammatical structure of the sentences is important for achieving good translation results.

Our final skeleton pose-based experiment investigates if creating variations in the signing speed in the pre-training data affects the performance on downstream tasks. As can be seen in Table 1.c we determine that the best performance on the test set comes from increasing the speed by 1.5 times. This is achieved by resampling the sequence to the desired number of frames. Possibly, reducing the total number of frames increased the difficulty in the pre-training data, allowing for the best performance on the Test set. However, on the Dev set, we find that reducing the speed results in better

Table 2: GFSLT Translation Metrics on the RWTH-PHOENIX-Weather-2014T DEV and TEST Sets

	TEST SET					DEV SET				
Training Data:	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE
PHIX	40.59	30.52	23.83	19.53	40.27	40.51	30.37	24.26	20.15	41.06
PHIX (VLP PreTrain)	43.06	32.68	25.88	21.35	42.67	43.39	32.94	26.24	21.65	43.62
GAN	2.41	0.95	0.38	0.19	4.69	2.71	1.00	0.42	0.19	5.19
PHIX (Gan PreTrain)	41.46	31.20	24.65	20.28	41.63	41.93	31.68	25.17	20.85	42.37
PHIX + GAN	43.12	32.15	25.26	20.77	42.01	**44.07**	**33.24**	26.42	21.84	**43.89**
Splat	6.51	2.20	0.83	0.40	5.30	7.13	2.55	1.15	0.62	5.85
PHIX (Splat PreTrain)	41.15	30.79	23.90	19.47	40.14	42.50	31.92	25.38	20.97	42.18
PHIX + Splat	**43.31**	**33.02**	**26.30**	**21.79**	**43.02**	43.29	33.02	**26.48**	**22.00**	43.58

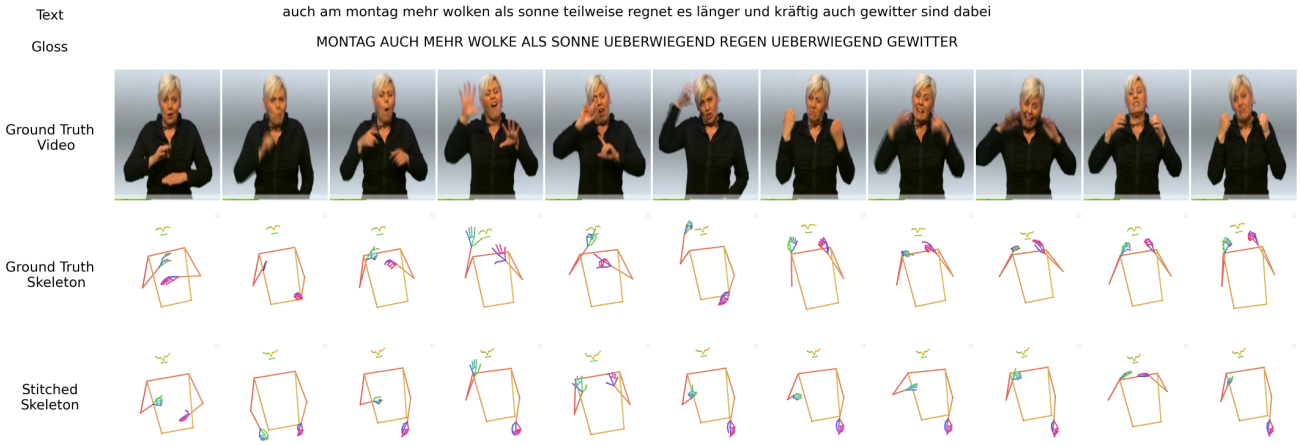

Figure 3: An example from the PHOENIX14T dataset, showing top to bottom: spoken language, gloss, original video, extracted skeleton, and Stitched sequence.

BLEU-2 to 4 scores. Given only a 70% sign overlap in the dev and test split, we suspect speed augmentation might introduce bias for some signs.

4.2.2 Video-to-Text. Table 2 shows the results of the V2T task. We observe mixed results with the appearance-based augmentations. In line with the original paper, we find that applying VLP improves the baseline model's performance. However, the best performance comes from including additional data created using the SignSplat approach. Increasing the BLEU-4 score on both the test and dev splits by 0.49 and 0.68, respectively. Surprisingly, given that the PHOENIX14**T** dataset contains only 9 signers and most models overfit to the appearance of these individuals. We hypothesise that greater performance gains can be achieved by including more appearances rendered from multiple viewpoints.

We achieve inconsistent performance using the data generated with the SignGAN approach, even though the qualitative results show the model produces a realistic synthetic signer. Artifacts caused by noise in the generation process can degrade the quality of the data. Issues such as disconnected limbs, loss of detail between fingers and hands, and artefacts in the face are all possible causes. On the Dev set, we find the best improvements in BLEU-1 and 2 score using this data, while it is detrimental on most other metrics.

4.3 Qualitative Results

Figure 3 and Figure 4 show examples of the generated data from the PHOENIX14**T** dataset. Both figures show the original video, the gloss, and the spoken language translation. Figure 3 shows the extracted skeleton and the stitched sequence. We note that the two sequences share many features like handshape and locations. However, the temporal alignment between the two differs, mostly due to our approximations, as the timing information is not available in the metadata.

Figure 4 shows the visual augmentation from both SignSplat and SignGAN. Both approaches can faithfully capture the motion and maintain a realistic appearance, although with slight changes in camera angle compared to the ground truth.

The SignSplat model can produce a more realistic signer, with fewer artefacts and a more natural appearance. The SignGAN model, while able to produce realistic video sequences, suffers from artefacts and noise in the data, which can detract from the overall quality of the generated video. This is likely due to the limitations

Figure 4: An example from the PHOENIX14T dataset, showing top to bottom: spoken language, gloss of the original video, SignSplat Avatar, and SignGAN Avatar.

Figure 5: Examples from the PHOENIX14T dataset, showing top to bottom: the original video, SignSplat Avatar, and SignGAN Avatar. The bottom row shows issues with blurring, hand merging and artefacts in the face.

of the GAN architecture and the training data used. As shown in Figure 5, some of the artefacts include loss of detail between fingers and hands, and artefacts in the face.

5 Conclusion

In this paper, we propose leveraging three different SLP techniques to augment and generate synthetic sign language data. We discovered that stitching together isolated signs to create synthetic skeleton sequences allowed us to pretrain SLT models, leading to significant performance gains. Further improvements across all metrics can be achieved by introducing temporal variations in these sequences. For high N-gram scores (specifically BLEU-3 to BLEU-4), creating grammatical variations also proves beneficial.

We tested two different approaches for generating visual augmentations that were able to transfer the motion of the original dataset to an entirely new appearance. Qualitatively, we found that the GAN was able to produce realistic video sequences, but was prone to unrealistic artefacts caused by noise in the data. This method proved effective on a subset of metrics. The Gaussian Splatting approach was able to produce realistic sequences with fewer artefacts, given that the approach is constrained by the underlying human mesh. As a result, we demonstrated the benefits of introducing visual augmentations.

Given the low-resource nature of sign language translation datasets, we suggest this is a promising direction for future research. These simple approaches could be considered a baseline for many more experiments, which we believe these results indicate could generate much greater improvements in performance. For instance, combining the visual augmentations with the novel skeleton motion to generate entirely new video sequences rendered from multiple viewpoints. Furthermore, we believe the power of LLMs is yet to be leveraged here to generate novel sentences.

Acknowledgments

This work was supported by the SNSF project 'SMILE II' (CRSII5 193686), the Innosuisse IICT Flagship (PFFS-21-47), EPSRC grant APP24554 (SignGPT-EP/Z535370/1), Google DeepMind and through funding from Google.org via the AI for Global Goals scheme. This work reflects only the author's views and the funders are not responsible for any use that may be made of the information it contains.

References

[1] Sharif Md Abdullah, Abhijit Paul, Shebuti Rayana, Ahmedul Kabir, and Zarif Masud. 2025. State-of-the-Art Translation of Text-to-Gloss using mBART: A case study of Bangla. *arXiv preprint arXiv:2504.02293* (2025).

[2] Samuel Albanie, Gül Varol, Liliane Momeni, Hannah Bull, Triantafyllos Afouras, Himel Chowdhury, Neil Fox, Bencie Woll, Rob Cooper, Andrew McParland, et al. 2021. Bbc-oxford british sign language dataset. *arXiv preprint arXiv:2111.03635* (2021).

[3] Hasan Algafri, Hamzah Luqman, Sarah Alyami, and Issam Laradji. 2025. SSLR: A Semi-Supervised Learning Method for Isolated Sign Language Recognition. *arXiv preprint arXiv:2504.16640* (2025).

[4] Andrew Bangham, SJ Cox, Ralph Elliott, John RW Glauert, Ian Marshall, Sanja Rankov, and Mark Wells. 2000. 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)*. IET, 6–1.

[5] Danielle Bragg, Oscar Koller, Mary Bellard, Larwan Berke, Patrick Boudreault, Annelies Braffort, Naomi Caselli, Matt Huenerfauth, Hernisa Kacorri, Tessa Verhoef, et al. 2019. Sign language recognition, generation, and translation: An interdisciplinary perspective. In *Proceedings of the 21st international ACM SIGACCESS conference on computers and accessibility*. 16–31.

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

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

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

[9] Yutong Chen, Ronglai Zuo, Fangyun Wei, Yu Wu, Shujie Liu, and Brian Mak. 2022. Two-stream network for sign language recognition and translation. *Advances in Neural Information Processing Systems* 35 (2022), 17043–17056.

[10] MMPose Contributors. 2020. OpenMMLab Pose Estimation Toolbox and Benchmark. https://github.com/open-mmlab/mmpose.

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

[12] Shengliang Deng, Mi Yan, Songlin Wei, Haixin Ma, Yuxin Yang, Jiayi Chen, Zhiqi Zhang, Taoyu Yang, Xuheng Zhang, Heming Cui, et al. 2025. GraspVLA: a Grasping Foundation Model Pre-trained on Billion-scale Synthetic Action Data. *arXiv preprint arXiv:2505.03233* (2025).

[13] Kenneth W Dunn, Chichen Fu, David Joon Ho, Soonam Lee, Shuo Han, Paul Salama, and Edward J Delp. 2019. DeepSynth: Three-dimensional nuclear segmentation of biological images using neural networks trained with synthetic data. *Scientific reports* 9, 1 (2019), 18295.

[14] Eleni Efthimiou, Stavroula-Evita Fotinea, Thomas Hanke, John Glauert, Richard Bowden, Annelies Braffort, Christophe Collet, Petros Maragos, and François Lefebvre-Albaret. 2012. The dicta-sign wiki: Enabling web communication for the deaf. In *International Conference on Computers for Handicapped Persons*. Springer, 205–212.

[15] Oussama ElGhoul and Mohamed Jemni. 2011. 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*, Vol. 23.

[16] Tom Erez, Yuval Tassa, and Emanuel Todorov. 2015. Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx. In *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 4397–4404.

[17] Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. *arXiv preprint arXiv:2105.03075* (2021).

[18] Aymane El Firdoussi, Mohamed El Amine Seddik, Soufiane Hayou, Reda Alami, Ahmed Alzubaidi, and Hakim Hacid. 2024. Maximizing the Potential of Synthetic Data: Insights from Random Matrix Theory. *arXiv preprint arXiv:2410.08942* (2024).

[19] Edward Fish and Richard Bowden. 2025. Geo-Sign: Hyperbolic Contrastive Regularisation for Geometrically Aware Sign Language Translation. arXiv:2506.00129 [cs.CV] https://arxiv.org/abs/2506.00129

[20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

[21] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).

[22] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision* 129, 6 (2021), 1789–1819.

[23] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).

[24] Xi Guo, Wei Wu, Dongliang Wang, Jing Su, Haisheng Su, Weihao Gan, Jian Huang, and Qin Yang. 2022. Learning video representations of human motion from synthetic data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20197–20207.

[25] Aiming Hao, Yuecong Min, and Xilin Chen. 2021. Self-mutual distillation learning for continuous sign language recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*. 11303–11312.

[26] Shuang Hao, Wenfeng Han, Tao Jiang, Yiping Li, Haonan Wu, Chunlin Zhong, Zhangjun Zhou, and He Tang. 2024. Synthetic data in AI: Challenges, applications, and ethical implications. *arXiv preprint arXiv:2401.01629* (2024).

[27] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).

[28] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.

[29] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. 2022. Video diffusion models. *Advances in Neural Information Processing Systems* 35 (2022), 8633–8646.

[30] Hezhen Hu, Zhiwen Fan, Tianhao Wu, Yihan Xi, Seoyoung Lee, Georgios Pavlakos, and Zhangyang Wang. 2024. Expressive Gaussian Human Avatars from Monocular RGB Video. In *NeurIPS*.

[31] Lianyu Hu, Liqing Gao, Zekang Liu, and Wei Feng. 2022. Temporal lift pooling for continuous sign language recognition. In *European conference on computer vision*. Springer, 511–527.

[32] Qisheng Hu, Kaixin Li, Xu Zhao, Yuxi Xie, Tiedong Liu, Hui Chen, Qizhe Xie, and Junxian He. 2023. Instructcoder: Empowering language models for code editing. *CoRR* (2023).

[33] Wencan Huang, Wenwen Pan, Zhou Zhao, and Qi Tian. 2021. Towards Fast and High-Quality Sign Language Production. In *Proceedings of the 29th ACM International Conference on Multimedia*. 3172–3181.

[34] Alfarabi Imashev, Medet Mukushev, Vadim Kimmelman, and Anara Sandygulova. 2020. K-RSL: a corpus for linguistic understanding, visual evaluation, and recognition of sign languages. In *Proceedings of the 24th Conference on Computational Natural Language Learning*. Association for Computational Linguistics.

[35] Maksym Ivashechkin, Oscar Mendez, and Richard Bowden. 2023. Improving 3D Pose Estimation For Sign Language. In *2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*. 1–5. doi:10.1109/ICASSPW59220.2023.10193629

[36] Maksym Ivashechkin, Oscar Mendez, and Richard Bowden. 2025. SignSplat: Rendering Sign Language via Gaussian Splatting. *arXiv preprint arXiv:2505.02108* (2025).

[37] Hamid Reza Vaezi Joze and Oscar Koller. 2018. Ms-asl: A large-scale data set and benchmark for understanding american sign language. *arXiv preprint arXiv:1812.01053* (2018).

[38] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4 (July 2023). https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

[39] Yo-whan Kim, Samarth Mishra, SouYoung Jin, Rameswar Panda, Hilde Kuehne, Leonid Karlinsky, Venkatesh Saligrama, Kate Saenko, Aude Oliva, and Rogerio Feris. 2022. How transferable are video representations based on synthetic data? *Advances in Neural Information Processing Systems* 35 (2022), 35710–35723.

[40] Diederik P Kingma, Max Welling, et al. 2013. Auto-encoding variational bayes.

[41] Michael Kipp, Quan Nguyen, Alexis Heloir, and Silke Matthes. 2011. Assessing the deaf user perspective on sign language avatars. In *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*. 107–114.

[42] Oscar Koller, Necati Cihan Camgoz, Hermann Ney, and Richard Bowden. 2019. Weakly supervised learning with multi-stream CNN-LSTM-HMMs to discover sequential parallelism in sign language videos. *IEEE transactions on pattern analysis and machine intelligence* 42, 9 (2019), 2306–2320.

[43] 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. 2020. MEINE DGS – annotiert. Öffentliches Korpus der Deutschen Gebärdensprache, 3. Release / MY DGS – annotated. Public Corpus of German Sign Language, 3rd release. doi:10.25592/dgs.corpus-3.0

[44] Dongxu Li, Cristian Rodriguez, Xin Yu, and Hongdong Li. 2020. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 1459–1469.

[45] Zecheng Li, Wengang Zhou, Weichao Zhao, Kepeng Wu, Hezhen Hu, and Houqiang Li. 2025. Uni-Sign: Toward Unified Sign Language Understanding at Scale. arXiv:2501.15187 [cs.CV] https://arxiv.org/abs/2501.15187

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

[47] Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, et al. 2024. Best practices and lessons learned on synthetic data. *arXiv preprint arXiv:2404.07503* (2024).

[48] Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual Denoising Pre-training for Neural Machine Translation. *Transactions of the Association for Computational Linguistics* 8 (2020), 726–742. doi:10.1162/tacl_a_00343

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

[50] Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583* (2023).

[51] Yuecong Min, Aiming Hao, Xiujuan Chai, and Xilin Chen. 2021. Visual alignment constraint for continuous sign language recognition. In *proceedings of the IEEE/CVF international conference on computer vision*. 11542–11551.

[52] Gyeongsik Moon, Takaaki Shiratori, and Shunsuke Saito. 2024. Expressive Whole-Body 3D Gaussian Avatar. In *ECCV*.

[53] Amit Moryossef, Kayo Yin, Graham Neubig, and Yoav Goldberg. 2021. Data augmentation for sign language gloss translation. *arXiv preprint arXiv:2105.07476* (2021).

[54] Michael Niemeyer and Andreas Geiger. 2021. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11453–11464.

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

[56] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. 2019. Expressive Body Capture: 3D Hands, Face, and Body from a Single Image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[57] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. 2024. Reconstructing Hands in 3D with Transformers. In *CVPR*.

[58] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. *CoRR* abs/2103.00020 (2021). arXiv:2103.00020 https://arxiv.org/abs/2103.00020

[59] Javier Romero, Dimitrios Tzionas, and Michael J. Black. 2017. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)* 36, 6 (Nov. 2017).

[60] Ben Saunders, Necati Cihan Camgöz, and Richard Bowden. 2020. Adversarial Training for Multi-Channel Sign Language Production. In *British Machine Vision Virtual Conference*.

[61] Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. 2020. Everybody Sign Now: Translating Spoken Language to Photo Realistic Sign Language Video. *arXiv preprint arXiv:2011.09846* (2020).

[62] Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. 2020. Progressive transformers for end-to-end sign language production. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*. Springer, 687–705.

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

[64] Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. 2022. Signing at Scale: Learning to Co-Articulate Signs for Large-Scale Photo-Realistic Sign Language Production. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[65] Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. 2022. Signing at scale: Learning to co-articulate signs for large-scale photo-realistic sign language production. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5141–5151.

[66] Mohamed El Amine Seddik, Suei-Wen Chen, Soufiane Hayou, Pierre Youssef, and Merouane Debbah. 2024. How bad is training on synthetic data? a statistical analysis of language model collapse. *arXiv preprint arXiv:2404.05090* (2024).

[67] Connor Shorten, Taghi M Khoshgoftaar, and Borko Furht. 2021. Text data augmentation for deep learning. *Journal of big Data* 8, 1 (2021), 101.

[68] Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. 2023. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arXiv:2305.17493* (2023).

[69] William C Stokoe. 1980. Sign Language Structure. *Annual Review of Anthropology* (1980).

[70] Stephanie Stoll, Necati Cihan Camgöz, Simon Hadfield, and R. Bowden. 2018. Sign Language Production using Neural Machine Translation and Generative Adversarial Networks. In *British Machine Vision Conference*. https://api.semanticscholar.org/CorpusID:52288950

[71] Stephanie Stoll, Necati Cihan Camgoz, Simon Hadfield, and Richard Bowden. 2020. Text2Sign: Towards Sign Language Production Using Neural Machine Translation and Generative Adversarial Networks. *Int. J. Comput. Vision* 128, 4 (apr 2020), 891–908. doi:10.1007/s11263-019-01281-2

[72] Shengeng Tang, Dan Guo, Richang Hong, and Meng Wang. 2021. Graph-Based Multimodal Sequential Embedding for Sign Language Translation. *IEEE Transactions on Multimedia* PP (10 2021), 1–1. doi:10.1109/TMM.2021.3117124

[73] Shengeng Tang, Jiayi He, Lechao Cheng, Jingjing Wu, Dan Guo, and Richang Hong. 2024. Discrete to Continuous: Generating Smooth Transition Poses from Sign Language Observation. *arXiv preprint arXiv:2411.16810* (2024).

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

[75] Harry Walsh, Abolfazl Ravanshad, Mariam Rahmani, and Richard Bowden. 2024. A Data-Driven Representation for Sign Language Production. In *Proceedings of the 18th International Conference on Automatic Face and Gesture Recognition (FG 2024)*. Institute of Electrical and Electronics Engineers (IEEE).

[76] Harry Walsh, Ben Saunders, and Richard Bowden. 2024. Sign Stitching: A Novel Approach to Sign Language Production. In *The 35th British Machine Vision Conference (BMVC)*.

[77] Ryan Wong, Necati Cihan Camgoz, and Richard Bowden. 2024. Sign2GPT: Leveraging large language models for gloss-free sign language translation. *arXiv preprint arXiv:2405.04164* (2024).

[78] Ryan Wong, Necati Cihan Camgoz, and Richard Bowden. 2025. SignRep: Enhancing Self-Supervised Sign Representations. *arXiv preprint arXiv:2503.08529* (2025).

[79] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. 2021. A simple baseline for zero-shot semantic segmentation with pre-trained vision-language model. *arXiv preprint arXiv:2112.14757* (2021).

[80] Huijie Yao, Wengang Zhou, Hao Zhou, and Houqiang Li. 2024. Semi-Supervised Spoken Language Glossification. *arXiv preprint arXiv:2406.08173* (2024).

[81] Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284* (2023).

[82] Jan Zelinka and Jakub Kanis. 2020. Neural Sign Language Synthesis: Words Are Our Glosses. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 3384–3392. doi:10.1109/WACV45572.2020.9093516

[83] Benjia Zhou, Zhigang Chen, Albert Clapés, Jun Wan, Yanyan Liang, Sergio Escalera, Zhen Lei, and Du Zhang. 2023. Gloss-free sign language translation: Improving from visual-language pretraining. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 20871–20881.

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