# SignSynth: Data-Driven Sign Language Video Generation

Stephanie Stoll[0000−0002−3582−3969], Simon Hadfield[0000−0001−8637−5054], and Richard Bowden[0000−0003−3285−8020]

Centre for Vision, Speech and Signal Processing, University of Surrey
s.m.stoll@surrey.ac.uk

**Abstract.** We present SignSynth, a fully automatic and holistic approach to generating sign language video. Traditionally, Sign Language Production (SLP) relies on animating 3D avatars using expensively annotated data, but so far this approach has not been able to simultaneously provide a realistic, and scalable solution. We introduce a gloss2pose network architecture that is capable of generating human pose sequences conditioned on glosses.[1] Combined with a generative adversarial pose2video network, we are able to produce natural-looking, high definition sign language video. For sign pose sequence generation, we outperform the SotA by a factor of 18, with a Mean Square Error of 1.0673 in pixels. For video generation we report superior results on three broadcast quality assessment metrics. To evaluate our full gloss-to-video pipeline we introduce two novel error metrics, to assess the perceptual quality and sign representativeness of generated videos. We present promising results, significantly outperforming the SotA in both metrics. Finally we evaluate our approach qualitatively by analysing example sequences.

**Keywords:** Sign Language; Pose Generation; Human Motion

## 1 Introduction

Computational research into sign languages is an important, yet under-researched problem. Whilst there are some applications to translate sign languages into spoken languages [10, 32], their success is limited. The inverse process of translating spoken languages to sign languages is widely neglected. However, to provide the Deaf and Hard of Hearing with equal access and opportunities as hearing people, sign languages must become present in all parts of today's society. While sign language transcription is possible using human interpreters, it is simply infeasible to employ interpreters 24/7 at public places such as train stations and post offices, or to record video transcriptions for all web based content. An automatic, scalable solution is needed that can generate naturalistic sign language video from spoken or written language.

Traditionally, research into Sign Language Production (SLP) has focused on animating 3D avatars using sequences of parametrised glosses. However, given

[1] For sign languages a gloss is a written representation that describes a specific sign.

the complexity of sign language, the task of manually annotating data requires tremendous effort and expert knowledge. Sign languages are different from country to country and have the same local variations as dialects do in spoken languages. They also rely on much more than hand shape and motion to convey meaning, such as mouth/face gestures, eye gaze, and body pose. These non-manual features need to be annotated correctly and aligned with the gloss they belong to. Most sign avatars largely ignore non-manuals, making them hard to understand, and unnatural looking. Avatars using motion capture data provide a better sign quality, but are limited in their vocabulary. This is due to the cost associated with recording and storing high fidelity motion capture data.

Stoll et al. [34] were the first to present a neural network approach to SLP. They first translate written German into German sign gloss sequences using Neural Machine Translation (NMT), and use a look-up table (LUT) of mean sequences to generate 2D motion from automatically extracted pose information. This data provides the input to a Generative Adversarial Network (GAN) to produce sign language video. Their results for text to gloss translation are impressive and the approach is naturally scalable. However, the quality of the produced videos is lacking, given a low resolution of 128x128 pixels. Furthermore, the use of a LUT severely limits this approach and introduces artefacts and discontinuities in co-articulation between signs.

To further the field of SLP and address the shortcomings of previous approaches we present the following contributions:

1. In order to dramatically increase the quality of synthetic sign video generation, we propose a gloss-to-pose (gloss2pose) network capable of producing sign motion data of high fidelity, conditioned on sign glosses, and trained on weakly labelled data. To our knowledge, we are the first to address the generation of manuals and non-manuals in a holistic, data-driven way.
2. We combine the gloss2pose network with a pose-to-video (pose2video) network and are able to produce high definition sign language video.
3. We introduce two error metrics to assess the quality of automatically generated sign language videos.

We implemented our approach in Pytorch and will release the code upon acceptance of this manuscript.

## 2 Related Work

We provide an overview of recent developments in SLP, before describing the concept of motion graphs and recent approaches relevant to our work. Finally, the field of conditional image generation is presented.

**Approaches for Sign Language Production** Automatic SLP is traditionally achieved by animating avatars, using a sequence of parametrised glosses. Examples of these include VisiCast [2], eSign [40], Tessa [5], dicta-sign [9], and JASigning [17]. All these approaches rely on manually annotated data using a purpose-specific transcription language such as HamNoSys [29] or SigML [18]. Annotating the data requires expert knowledge and is generally carried out by trained linguists.

The resulting animations suffer from unnatural, under-articulated motion, that makes the avatars look robotic, hard to understand, and at times uncomfortable to view, due to the uncanny valley effect.[2] Given the tremendous annotation effort required, non-manuals are often neglected. Some work on integrating non-manuals has been carried out in recent years [7, 8, 24, 20], but it remains an unsolved problem. A possible method to circumvent these issues is to animate avatars directly from motion capture data [12]. This results in highly realistic, and expressive animations, including non-manuals. However, these systems are limited to pre-recorded phrases, or need complex re-assembly taking into account the effects of co-articulation. Additionally, the recording and cleaning of high-fidelity motion capture data is costly and time consuming, making this approach not scalable.

To make automatic SLP feasible, Stoll et al. [34] propose generating synthetic sign language video using a LUT and GAN. Whilst this potentially overcomes some of the limitations of avatar technology, the low resolution of the produced videos and the use of a LUT to provide the pose information restricts the approach, particularly in terms of co-articulation between signs. Furthermore, non-manuals such as facial expressions are not addressed. In contrast, our approach learns to generate detailed pose and video sequences. Sequences of varying speed and expressiveness are automatically generated, including non-manuals.

**Motion Graphs** A popular concept in computer graphics, they are used to animate characters using a directed graph constructed from motion data, i.e. novel animations are created by re-combining short sequences of recorded motion. They were first introduced independently by Kovar et al. [21], Arikan et al. [1], and Lee et al. [22]. In recent years, deep-learning based approaches have emerged, most relevant to our work being Holden et al. [15], and Zhang, Starke et al. [39]. Holden et al. developed a regression network for generating cyclic motion such as walking and running, by predicting character joint positions, velocities, angles, and the character's global trajectory at $t + 1$ given the joint positions, velocities, the character's global trajectory, and a semantic variable describing the type of gait at time $t$. A Catmull-Rom Spline to calculate the weights of the regression network helps enforce the cyclic nature of the data to be generated. The system is trained on motion capture data. Zhang, Starke et al. [39] build on this approach, and apply it to quadruped characters. They use heavy supervision, such as the character's 3D joint positions, velocities, rotations, as well as a user-defined global character trajectory & velocity, plus action variables describing the type of gait, and footfall pattern. In contrast our system does not require heavy supervision and instead learns to decompose signs into simpler sub-units.

**Conditional Image Generation** The field has seen a number of different techniques emerge over the last few years. Convolutional Neural Networks (CNNs) [4], [27], as well as Recurrent Neural Networks (RNNs) [14], [28] have been

[2] The uncanny valley is a concept aimed at explaining the sense of unease people often experience when confronted with simulations that closely resemble humans, but are not quite convincing enough [26].

explored for generating images. Variational Auto-Encoders (VAEs) [19], and later conditional VAEs [37] have proven a popular choice. Since their initial conception, GANs [13] have provided many approaches to the task of image generation, such as conditional GANs [25], [30], [31]. VAEs and GANs are often combined to harness the VAE's stability, and the GAN's discriminative nature. Most relevant to our work, VAE - GAN hybrids have been applied to image generation of people [23], [33], and more specifically to produce videos of people performing sign language [34]. In image-to-image translation, pix2pix [16], and recently pix2pixHD [35] were able to produce high-definition images from semantic label maps using a multi-stage generator, and a multi-scale discriminator.

In our work we build on the recent success of pix2pixHD, and develop a VAE-GAN-based network that is capable of producing high resolution sign language productions from semantic label maps.

# 3 Synthetic Sign Video Generation

Our approach to generating sign language video from glosses works in two stages, see Figure 1. First a gloss is translated into a human pose sequence by our gloss2pose network. The acquired poses are then used to condition a generative network called pose2video.

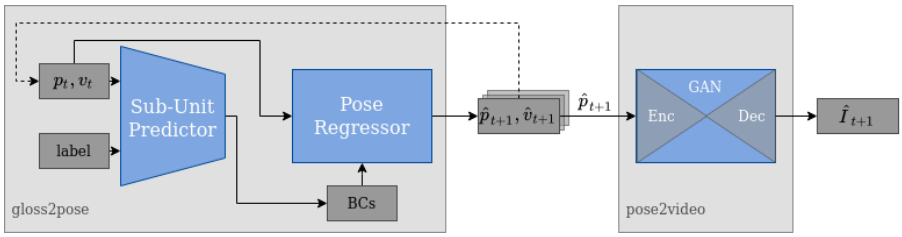


**Fig. 1.** Overview of our method at runtime. A sub-unit predictor estimates blending coefficients. These are used to generate the weights for the pose regressor. This network predicts poses and velocities for the next time step and is autoregressive. The generated poses are used as input to the video generator which produces sign video frames

## 3.1 Gloss to Pose

In contrast to previous work [39] our system works on 2D data, and does not require heavy supervision like user-defined global trajectories. We only use 2D skeletal pose and velocity data, as well as a label indicating the desired gloss(es). As the pose data can be automatically extracted using a detector such as [3] the data annotation effort is minimal. At run time the user only has to specify the desired gloss to generate a motion sequence of the target sign(s). The gloss2pose network predicts the state of pose keypoints for a future time step $Y$, given the current keypoints' state $X$. $X$ is defined as $X = \{p_t, v_t, l_t\}$ being a vector of the joint positions $p_t$, velocities $v_t$, and gloss label $l_t$ at time $t$. The velocity $v_t$ is defined as $v_t = p_t - p_{t-1}$, and $l_t$ is encoded as a one-hot vector, representing all

gloss classes. The output $Y$ is defined as $Y = \{\hat{p}_{t+1}, \hat{v}_{t+1}\}$ where $\hat{p}_{t+1}$ and $\hat{v}_{t+1}$ are the predicted positions and velocities of all keypoints at time $t+1$.

As shown in Figure 1, the gloss2pose network consists of two sub-networks, which are trained end-to-end. The main network, called pose regressor, predicts $Y$ given a subset of $X$, $\hat{X} = \{p_t, v_t\}$:

$$Y = \Phi(\hat{X}|\omega_\Phi), \tag{1}$$

where $\Phi$ is the pose regressor network and $\omega_\Phi$ are its weights. The pose regressor is a three-layer network consisting of two 1D-convolutional residual layers, and one fully connected layer. We found a convolutional architecture to be superior to a fully connected one, for both spatial accuracy in predicting joint poses per frame, as well as learning trajectories given the frame velocities. We achieved this by reshaping the input $\hat{X}$ to the pose regressor from a 1D to a 2D vector, with $p_t$ occupying the first, and $v_t$ the second row. Using a filter of height two we teach the network to learn the relationship between keypoint positions and velocities. To express in theory any number of signs and smoothly blend between them, we want the gloss2pose network to learn their composition in terms of sub-units. We achieve this by learning a set of blending coefficients given $X$, using a secondary neural network, called the sub-unit predictor. This is a fully connected network, consisting of four fully connected layers. We estimate the set of blending coefficients for $X$:

$$BC = \iota(\varsigma(X|\omega_\varsigma), l_t|\omega_\iota), \tag{2}$$

where $\varsigma$ is the sub-unit predictor, $\omega_\varsigma$ the weights of $\varsigma$, and $\iota$ is the reduction layer, with weights $\omega_\iota$. The blending coefficients $BC$ are a vector of length $b$, with $BC \in \mathbb{R}^b$. BC is used to generate the pose regressor network weights $\omega_\Phi$:

$$\omega_\Phi = \sum_{i=1}^{b} BC_i \Omega_i, \tag{3}$$

where $\Omega$ is a bank of pose regressor weights $\omega_\Phi$ of size $b$. This allows us to dynamically blend between weights depending on the target sign's sub-unit composition over time.

The predicted output $Y$ is compared to the ground truth using the mean-square error. Therefore the loss of the gloss2pose network is defined as:

$$L_P = \frac{1}{N} \sum_{k=1}^{N} (Y_{gt}^k - Y^k)^2, \tag{4}$$

where $k$ is an iterator over all keypoints up to the maximum number of keypoints $N$, and $Y_{gt}$ is the ground truth position and velocity.

**Error-Correcting Data Augmentation** We developed a two-time-step training scheme that allows us to double the amount of training data, and teach the network to correct its own mistakes in pose and velocity predictions. At time $t$

we let the network predict the output $Y$ for $t+1$. After calculating a loss, we use this result as input $X$ for the next time step, to predict $Y$ at $t+2$. After this the next ground truth from the original training data is used and the process repeated. The generated training samples are discarded after their use, to keep the ratio of ground truth and generated training data constant.
In addition to serving as a data augmentation scheme, we argue that this training scheme helps the network to correct itself from mis-predictions. At $t+1$ the predicted result $Y = \hat{p}_{t+1}, \hat{v}_{t+1}$ has a prediction error of $\epsilon_{t+1}$, making the total error for this time step $E = \epsilon_{t+1}$. When using the predicted result as the input for the next time step the total error at $t+2$ now becomes $E = \epsilon_{t+1} + \epsilon_{t+2}$. This is penalised with a higher loss, than just $E = \epsilon_{t+2}$. This means the network learns to correct for drift from error accumulation, and to handle data samples from previously unexplored space.

### 3.2 Pose to Video

Like pix2pixHD [35], our pose2video network is the combination of a convolutional image encoder and a Generative Adversarial Network (GAN), conditioned on semantic label maps. Inside the GAN a generator $G$ is engaged in a minimax game against a set of multi-scale discriminators $D$. $G$ is generating new data instances, which are evaluated by $D$ to be either "fake" (as in not belonging to the same data distribution), or "real" (part of the same data distribution). $G$'s aim is to maximise the likelihood of $D$ choosing incorrectly, whereas $D$ tries to maximise its chance of choosing correctly. Trained in conjunction, the networks improve each other, with $G$ creating more and more realistic data samples.

The input to the generator $G$ is the positional information generated by the gloss2pose network $\hat{p}$. The discriminator $D$ evaluates either the generated image $G(\hat{p}_t)$, or the real image $I_t$. We can therefore define the adversarial loss as

$$L_{GAN}(G, D_k) = \mathop{\mathbb{E}}_{\hat{p}_t}[log(D_k(\hat{p}_t, I_t))] + \mathop{\mathbb{E}}_{\hat{p}_t}[log(1 - D_k(\hat{p}_t, G(\hat{p}_t)))], \tag{5}$$

where $k$ indicates the discriminator scale. To combine the adversarial losses of all $D_k$, we sum:

$$L_{GAN}(G, D) = \sum_{k=1,2,3} L_{GAN}(G, D_k). \tag{6}$$

Additionally we apply a feature matching loss as presented in [35]:

$$L_f(G, D_k) = \mathop{\mathbb{E}}_{\hat{p}_t} \sum_{i=1}^{T} \frac{1}{N_i} \Big[ \sum |D_k^{(i)}(\hat{p}_t, I_t) - D_k^{(i)}(\hat{p}_t, G(\hat{p}_t)| \Big], \tag{7}$$

where $T$ is the total number of layers in $D_k$, $i$ is the current layer of $D_k$, and $D_k^{(i)}$ is the $i^{\text{th}}$ layer feature extractor of $D_k$. Again we sum the $L_f$ losses of all $D_k$ to obtain the overall $L_f$ loss:

$$L_f(G, D) = \sum_{k=1,2,3} L_f(G, D_k). \tag{8}$$

The total loss $L_{p2v}$ is a combination of adversarial and L1 loss:

$$L_{p2v} = L_{GAN} + \delta L_f, \tag{9}$$

where $\delta$ weighs the influence of $L_f$.

# 4 Experiments and Results

We evaluate the gloss2pose and pose2video parts of our approach separately, before quantitatively and qualitatively analysing their combined performance.

We use the SMILE Sign Language Assessment Dataset [6] for training and testing the gloss2pose as well as pose2video networks. It consists of 42 signers performing 105 signs in isolated form, with three repetitions each, in Swiss-German Sign Language (DSGS). The SMILE dataset is multi-view, however we only utilise the Kinect colour stream, which is of 1920x1080 resolution at 30fps. We extract 2D human pose estimations from the video data, using OpenPose [3] for the upper body (14 keypoints), face (70 keypoints) and hands (21 keypoints per hand). For the pose2video network, the extracted keypoints of one chosen signer are used to generate semantic label maps encoding the position and type of joint of each keypoint. At run time the positional information to generate these maps is provided from the output of the gloss2pose network, $Y$. We split our remaining dataset into train, test and validation sets. We use the training set to train the gloss2pose network, and evaluate it using the test set. The validation set is used to assess the performance of our whole system.

## 4.1 Gloss to Pose

We compare our gloss2pose network against the approach of Stoll et al. [34], who use a LUT to transform glosses to a dynamically time-warped mean sequence built from all example sequences for the gloss. They populate their LUT with the RWTH-PHOENIX-Weather 2014**T**[11] dataset, which is of continuous German Sign Language (DGS). However as it is of low resolution (227x227), the quality of pose information is poor. Furthermore, as the data is continuous, the glosses were extracted using a forced-alignment approach, meaning boundaries between glosses are not exact. These factors led us to decide it would be unfair to directly compare their results to our results obtained using high-resolution isolated data. We therefore contacted the authors of [34] for their code and populated their LUT using the SMILE data instead. The quantitative comparison in this section will be against this LUT of dynamically time warped mean sequences created out of all examples per gloss.

**Quantitative Evaluation** For our first experiment we train with 10, 30, 50, and 80 blending coefficients to find the optimal configuration to encode the 105 gloss classes into sub-units. The data per epoch consists of 859,522 real, and 859,522 synthetic data frames, given the regime described in Section 3.1. Batch size is set to 32, the learning rate to 0.0001, and ADAM optimisation is used.

We evaluate by sampling 20 different sequences per gloss and taking the Mean Squared Error (MSE) between sampled positions & velocities, and their ground

truth. The MSE per frame is accumulated and divided by the total number of frames across all sequences for all glosses. A sequence is sampled by giving the network a starting input $X_t$ from the test dataset. The network generates a prediction $Y_{t+1}$ which we use as the input for the next time step, and let the network feed back on itself for 150 frames in total. This is a very challenging test environment as there is a huge scope for drift caused by errors propagating as the network feeds back on itself over such a long time. To put our findings into context we compare it to the performance of the network when feeding the next ground truth frame at $t+1$ instead of a generated frame. We found that after 20 epochs the best and most stable performance was achieved with 50 blending coefficients, see Figure 2. We did not find that, in general, increasing the number of blending coefficients improves performance, meaning our subunit-predictor is able to dissect sign motions into sub-motions, rather than learning one set of coefficients for each of the 105 signs in the dataset.

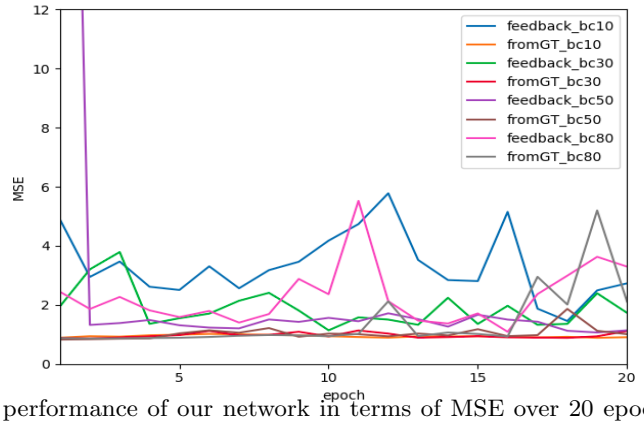


**Fig. 2.** The performance of our network in terms of MSE over 20 epochs. Different numbers of blending coefficients are explored $(10, 30, 50, 80)$. As a means of comparison we also provide results when the network is fed the next ground truth frame, rather than the sample generated at the previous time step ($fromGT$)

We next compare the MSE of our best performing network against the LUT approach of [34]. The result is presented in Table 1. To test the performance of [34] we compare the dynamically time warped mean sequence for each gloss against all ground truth sequences. As before, the MSE per frame is accumulated and divided by the total number of frames across all sequences. Our network outperforms [34] by a factor of 18. We suspect this has two causes. Firstly, averaging across all signers in the training set to acquire one representative mean sequence per gloss robs the LUT of the capability to express different people's skeletal builds, and a signer's natural variance in expressiveness. However, our network is capable of intelligently managing this spatial variance, as it learns from the data. Secondly, the dynamic time warping needed by [34] removes any variability in speed. However, a sign's duration can vary immensely, depending on repetition, or to convey e.g. excitement. In contrast, our network can express this natural variance in speed.

**Table 1.** MSE for [34]'s LUT and our gloss2pose (g2p) network

| | LUT [34] | g2p (ours) |
|---|---|---|
| MSE | 19.2886 | **1.0673** |

**Qualitative Evaluation** We analyse two generated pose sequences, see Figure 3. Each sequence and canonical example is sampled at every $10^{\text{th}}$ frame. We would like to point out that the example video sequences cannot be considered a direct ground truth, but merely a reference for the reader. The sequences depict the signs JAHR (YEAR) and ERZHLEN (TELL), respectively. We chose two signs that are close in their overall motion to showcase our network's ability to still generate them distinctively. Both hands for the sign JAHR form fists, whereas for ERZHLEN they are flat and open. This is recreated in detail in the pose sequences generated, together with the correct motion for each sign. We also want to point out the variability in speed of the signs produced. The sequence generated for ERZHLEN is shorter than the video example provided, whereas for JAHR it is slightly longer. Our network can produce sequences of variable length and expressiveness, something a mean sequence based approach such as [34] is incapable of. Furthermore, we are able to automatically produce aligned non-manuals such as facial expressions, which is not addressed in the SotA [34].


**Fig. 3.** Two pose sequences generated by the gloss2pose network. Canonical example sequences are provided to give the reader a reference of the signs. The top example is conditioned on the gloss JAHR (YEAR), the bottom example on ERZHLEN (TELL). This figure is best viewed in colour and digital format

## 4.2 Pose to Video

We evaluate the performance of pose2video against the Pose-Conditioned Sign Generation Network (PSGN) by Stoll et al. [34]. We trained the pose2video network for 100 epochs, with a training set of 12,500 and a test set of 1,400 image-semantic label map pairs. We used ADAM optimisation with an initial learning rate of 0.0002 that we linearly decay to zero.

**Quantitative Evaluation** We evaluate the image generation of each network using Structural Similarity Index Measurement (SSIM) [36], Peak Signal-to-Noise Ratio (PSNR)[38] and MSE. SSIM measures a perceptual degradation of down-

sampled or corrupted images compared to their originals. We use this to measure the perceptual degradation of a generated image $\hat{I}_t = G(P_t)$ to its ground truth $I_t$. PSNR and MSE are also used to assess compressed or corrupted image quality compared to their original. For images the MSE is used to calculate the average squared per-pixel error between $\hat{I}_t$ and $I_t$. Using the MSE, PSNR measures the peak error in dB. Table 2 compares the SSIM, PSNR, and MSE scores for PSGN [34] and our pose2video network over 1200 frames respectively. PSGN scores marginally higher for SSIM, however as it is a metric focussing on overall image structure and appearance rather than fine details, this is unsurprising. As the original ground truth of 1920x1080 is encoded to 128x128 pixels by PSGN most fine detail is already lost. In contrast pose2video beats PSGN by a large margin for both PSNR and MSE. This is due to to high resolution and detail of the generated images.

**Table 2.** Mean SSIM, PSNR, and MSE values for PSGN [34] and our pose2video network. SSIM ranges between -1 and +1, with +1 indicating identical images. Lower MSE indicates more likeness, whereas higher PSNR indicates images are more alike

| | SSIM | PSNR | MSE |
|---|---|---|---|
| PSGN [34] | **0.9434** | 24.7248 | 226.2474 |
| p2v (ours) | 0.9428 | **29.0181** | **86.0553** |

### 4.3 Synthetic Sign Video Generation from Gloss

Finally, we evaluate the performance of the gloss2pose and pose2video networks in conjunction (SignSynth). We compare our approach to that of [34] before discussing qualitative results.

**Quantitative Evaluation** Evaluating the quality of generated sign language videos in a quantitative fashion is challenging for multiple reasons. For each sign there is a natural variability in terms of size, motion and speed, but also context-specific differences, such as repetitions, or negations. Furthermore, transitions between signs can influence the individual signs' trajectories and positions. A logical option that comes to mind is to use a Sign Language Recognition (SLR) system to assess the quality and accuracy of produced sign language videos. Since the advent of deep learning such systems have certainly improved. However, they are still far from accurate and highly depend on the data they were trained on. To our knowledge there is no publicly available SLR system that is trained on the SMILE dataset. Furthermore, we want to accurately measure the quality of synthetic sign video, rather than diluting the measurements with inevitable errors produced by an SLR network.

We therefore devise two metrics. The first one is a confidence score that assesses the level of detail and human characteristics of the generated videos. The second is a distance measure that assesses how closely generated sign videos resemble the ground truth videos for that gloss. Both metrics make use of the OpenPose [3] human pose detector. To be more specific, we take the generated sign language videos and let OpenPose detect pose, face and hand keypoints.

For [34] we do the same on the mean video sequences. For each keypoint $k$ the coordinates $x$ and $y$ as well as a detection confidence $c$ is inferred. For our first metric we utilise $c$ to assess the pose detector's ability to detect human keypoints from the generated videos, the intuition being that the more detailed and "human-like" the generated video, the higher the detector's confidence. We divide our keypoints into regions of interest. This lets us assess confidences on specific body parts, such as the hands and the face. For each region we define the regional confidence as

$$C = \frac{1}{T-1} \sum_{t=0}^{T-1} \left( \frac{1}{I} \sum_{i=1}^{I} \alpha_i c(k_i) \right), \tag{10}$$

where $I$ is the total number of keypoints in the region, $T$ is the total number of frames assessed, and $\alpha_i$ is the importance of a keypoint in the region. We can further obtain the overall confidence by summing over the regions:

$$C_{total} = \frac{1}{R} \sum_{r=1}^{R} \beta_r C_r \tag{11}$$

where $\beta_r$ is the importance of the region in the total score, and $R$ is the number of regions. In our experiment we define four regions: *pose* (14 keypoints defining the upper body skeleton), $face$ (70 facial keypoints), *handl* and *handr* (21 keypoints for the left and right hand respectively). We set $\alpha$ and $\beta$ to 1.0. Whilst we believe there is scientific value in identifying the importance of specific regions and keypoints for sign language, this is future work and beyond the scope of this manuscript. Table 3 compares confidence scores of [34] and SignSynth. Both methods perform well in the *pose* region. However, the detector fails to detect any facial keypoints in the output of [34] and has very low confidence in the keypoints of both hands. For our method, the detector has a high confidence for the $face$ region, and beats the hand confidences of [34] by an order of magnitude. This showcases our approach's superior ability to produce detailed signings with manuals and non-manuals clearly present.

**Table 3.** Confidence scores for Stoll et al. [34] and SignSynth. Confidences are given for four regions, as well as an overall score. The confidence is measured between 0 and 1

| | $C_{pose}$ | $C_{face}$ | $C_{handl}$ | $C_{handr}$ | $C_{total}$ |
|---|---|---|---|---|---|
| Stoll et al. [34] | 0.499 | 0.000 | 0.026 | 0.025 | 0.138 |
| SignSynth | **0.791** | **0.766** | **0.120** | **0.266** | **0.485** |

For our second experiment we analyse the behaviour of keypoints over time. Rather than just looking for overall smoothness, we want to also relate the trajectory of points to the signs they are meant to represent, whilst taking into account different levels of speed and expression. For this, we first perform hierarchical clustering with average linking of the SMILE validation set. The metric used is based on dynamic time warping (dtw). We define the similarity of two sequences $S1$ and $S2$ as the euclidean distance between $S1$ and $S2$ when the

alignment path $P$ is optimal:

$$dtw(S1, S2) = \sqrt{\sum_{(m,n)\in P} ||(S1_m - S2_n)||^2}. \tag{12}$$

We again divide our keypoints into regions and treat each keypoint's trajectory independently:

$$D = \frac{1}{I}\sum_{i=1}^{I} \alpha_i dtw(S1(k_i), S2(k_i)), \tag{13}$$

where $D$ is the regional distance between S1 and S2, $\alpha_i$ again is the importance of a keypoint in the region. To obtain the overall distance between S1 and S2 we sum over the regions as before:

$$D_{total} = \frac{1}{R}\sum_{r=1}^{R} \beta_r D_r, \tag{14}$$

where $R$ is the number of regions and $\beta_r$ the importance of each region.
After clustering the validation set we use the same metric to measure the distance of each generated sample to each cluster. We then report the mean distance between clusters and samples per sign class. Table 4 shows results for three sign classes. Again, we compare our approach against that of [34]. However, we also provide a reference to put the distances obtained into perspective for the reader. As the reference we take ground truth samples of a signer from the SMILE test set and measure the distance to each cluster. The reference's samples per sign class measure closest to the sign cluster they belong to. [34]'s approach measures more or less the same distances to all clusters per sign class, meaning their sequences are not descriptive of any sign. Overall their distances are significantly larger than that of the reference. Our results lie in the same range as the reference, and their variability per sign class showcases the generated samples' descriptiveness. Two out of three signs are correctly identified, whereas for the third all samples score similarly regardless of which sign class they belong to. When inspecting the samples for the third gloss we saw that our network performs a variable number of repetitions for the circular hand motion in front of the body. While repetition in sign language is common, there are no sequences with repetitions in the data used to form the clusters. We wish to take into account the occurrence of repetitions in future work.

**Table 4.** $D_{total}$ for test samples of three signs to each sign cluster. ERZHLEN is abbreviated to ERZ in the table

| | SignSynth | | | Stoll et al. [34] | | | Reference | | |
|---|---|---|---|---|---|---|---|---|---|
| Clusters | ABEND | ABER | ERZ | ABEND | ABER | ERZ | ABEND | ABER | ERZ |
| ABEND | **12.51** | 14.49 | 16.17 | 19.82 | **16.36** | 19.29 | **12.28** | 14.61 | 14.69 |
| ABER | 15.58 | **14.72** | 17.02 | 18.13 | **16.29** | 19.35 | 14.86 | **13.34** | 15.43 |
| ERZ | 15.51 | **15.32** | 15.93 | 20.68 | **17.04** | 19.78 | 14.70 | 15.03 | **13.54** |

**Qualitative Evaluation** We present sequences generated by our SignSynth method, and compare it to results from [34], (see Figure 4). For each sign, a canonical sequence is provided. For those example sequences and SignSynth results, every $10^{\text{th}}$ frame is shown. For spatial reasons we only show every $20^{\text{th}}$ frame of LUT+PSGN, as the dynamic time warping needed by [34]'s approach results in sequences that are much slower than many real life examples.
We study two signs with similar hand motion, but different hand shape. The first sign ABER (BUT), is shown in the top section of Figure 4, the second VORGESTERN (DAY-BEFORE-YESTERDAY) in the section below. Our generations for both signs follow the correct trajectory, with slight variations in speed and expressiveness, showcasing our networks' ability to learn natural variations in sign language production. Furthermore, our approach generates significant detail such as an extended index finger in the dominant hand. The hand shape for the sign VORGESTERN (an extended thumb pointing backwards) is also generated. The sequences generated by [34] follow the global trajectories for both signs, but are executed at less than half the speed. Any detail of hands or facial expression is lost completely.
Finally, we show a sequence generated from multiple glosses. Even though our approach is trained on isolated data, it is capable of generating smooth pose and video sequences without artefacts between signs, see Figure 5. It depicts the generated pose and video data conditioned on the gloss sequence FUSSBALL SPIELEN (PLAY FOOTBALL). As before, every $10^{\text{th}}$ frame is shown. The SignSynth approach generates detailed sequences for pose and video that represent the gloss input sequence, with smooth transitions between signs. Detail in the hands is well preserved, especially for the sign SPIELEN. For more results we refer to the supplementary material.

# 5 Conclusion

We presented a novel approach to Sign Language Production (SLP) that only requires minimal user input. Our approach is capable of producing sign language video of high resolution, where sequences contain hand motion with a natural variance in speed, expressiveness, and distinctive hand shape. Non-manuals are also generated and naturally aligned with the rest of the sign, as our approach directly learns from sign language data. Additionally, we are able to smoothly and automatically transition between glosses, making our approach superior to approaches relying on manually enforcing co-articulation. When comparing our method to the current SotA [34], we were able to surpass its performance for pose and video generation, as well as generating videos from gloss information. We evaluated and compared our approach using MSE and popular metrics from broadcast quality assessment. We then developed two new metrics to assess the quality of sign language videos, which we used to compare our approach to [34], and reported highly promising results.
In the future, there are a number of avenues to pursue. The gloss2pose network could be extended to 3D data, and be used to drive an avatar without the drawbacks found with current approaches. We also want to incorporate techniques

from NMT to address complete spoken language to sign language translation. Furthermore we want to explore the intricacies of sign language, as mentioned in Section 4. Finally, we are excited to continue working with linguists on solving the problem of automatic SLP to further the integration of the Deaf community.


**Fig. 4.** SignSynth output compared to Stoll et al. [34]. The top half depicts the sign ABER (BUT), the bottom half VORGESTERN (DAY-BEFORE-YESTERDAY). This figure is best viewed in colour and digital format


**Fig. 5.** Generated pose and video sequences for the gloss sequence FUSSBALL SPIELEN (PLAY FOOTBALL). This figure is best viewed in colour and digital format

**Acknowledgements** This work received funding from the SNSF Sinergia project SMILE (CRSII2 160811), the European Unions Horizon2020 research and innovation programme under grant agreement no. 762021 Content4All and the EPSRC project ExTOL (EP/R03298X/1). This work reflects only the authors view and the Commission is not responsible for any use that may be made of the information it contains.

# References


1. Arikan, O., Forsyth, D.A.: Interactive motion generation from examples. In: Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques. pp. 483–490. SIGGRAPH '02, ACM, New York, NY, USA (2002)
2. Bangham, J.A., Cox, S.J., Elliott, R., Glauert, J.R.W., Marshall, I., Rankov, S., Wells, M.: 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). pp. 6/1–6/7 (2000)
3. Cao, Z., Simon, T., Wei, S., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). vol. 00, pp. 1302–1310 (July 2017)
4. Chen, Q., Koltun, V.: Photographic image synthesis with cascaded refinement networks. In: ICCV. pp. 1520–1529. IEEE Computer Society (2017)
5. Cox, S., Lincoln, M., Tryggvason, J., Nakisa, M., Wells, M., Tutt, M., Abbott, S.: Tessa, a system to aid communication with deaf people. In: Proceedings of the fifth international ACM conference on Assistive technologies. pp. 205–212. ACM (2002)
6. Ebling, S., Camgoz, N.C., Braem, P., Tissi, K., Sidler-Miserez, S., Stoll, S., Hadfield, S., Haug, T., Bowden, R., Tornay, S., Razavi, M., Magimai-Doss, M.: Smile swiss german sign language dataset. In: 11th Edition of the Language Resources and Evaluation Conference (LREC) (2018)
7. Ebling, S., Glauert, J.: Exploiting the full potential of jasigning to build an avatar signing train announcements (10 2013)
8. Ebling, S., Huenerfauth, M.: Bridging the gap between sign language machine translation and sign language animation using sequence classification. In: SLPAT@Interspeech (2015)
9. Efthimiou, E.: The Dicta-Sign Wiki: Enabling Web Communication for the Deaf (2012)
10. Elwazer, M.: Kintrans (2018), http://www.kintrans.com/
11. Forster, J., Schmidt, C., Koller, O., Bellgardt, M., Ney, H.: Extensions of the sign language recognition and translation corpus rwth-phoenix-weather. In: Language Resources and Evaluation. pp. 1911–1916. Reykjavik, Island (May 2014)
12. Gibet, S., Lefebvre-Albaret, F., Hamon, L., Brun, R., Turki, A.: Interactive editing in french sign language dedicated to virtual signers: Requirements and challenges. Univers. Access Inf. Soc. **15**(4), 525–539 (Nov 2016)
13. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada. pp. 2672–2680 (2014)
14. Gregor, K., Danihelka, I., Graves, A., Rezende, D., Wierstra, D.: Draw: A recurrent neural network for image generation. In: Bach, F., Blei, D. (eds.) Proceedings of the 32nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 37, pp. 1462–1471. PMLR, Lille, France (07–09 Jul 2015)
15. Holden, D., Komura, T., Saito, J.: Phase-functioned neural networks for character control. ACM Trans. Graph. **36**(4), 42:1–42:13 (Jul 2017). https://doi.org/10.1145/3072959.3073663, http://doi.acm.org/10.1145/3072959.3073663
16. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 5967–5976 (2017)

17. JASigning: Virtual humans research for sign language animation (2017), http://vh.cmp.uea.ac.uk/index.php/Main_Page
18. Kennaway, R.: Avatar-independent scripting for real-time gesture animation. CoRR **abs/1502.02961** (2013)
19. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings (2014)
20. Kipp, M., Héloir, A., Nguyen, Q.: Sign language avatars: Animation and comprehensibility. In: IVA (2011)
21. Kovar, L., Gleicher, M., Pighin, F.: Motion graphs. In: Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques. pp. 473–482. SIGGRAPH '02, ACM, New York, NY, USA (2002)
22. Lee, J., Chai, J., Reitsma, P.S.A., Hodgins, J.K., Pollard, N.S.: Interactive control of avatars animated with human motion data. In: Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques. pp. 491–500. SIGGRAPH '02, ACM, New York, NY, USA (2002)
23. Ma, L., Jia, X., Sun, Q., Schiele, B., Tuytelaars, T., Van Gool, L.: Pose guided person image generation. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 406–416. Curran Associates, Inc. (2017)
24. McDonald, J., Wolfe, R., Schnepp, J., Hochgesang, J., Jamrozik, D.G., Stumbo, M., Berke, L., Bialek, M., Thomas, F.: An automated technique for real-time production of lifelike animations of american sign language. Universal Access in the Information Society **15**(4), 551–566 (2016)
25. Mirza, M., Osindero, S.: Conditional generative adversarial nets. CoRR **abs/1411.1784** (2014), http://arxiv.org/abs/1411.1784
26. Mori, M., MacDorman, K., Kageki, N.: The uncanny valley [from the field] **19**, 98–100 (06 2012)
27. van den Oord, A., Kalchbrenner, N., Espeholt, L., kavukcuoglu, k., Vinyals, O., Graves, A.: Conditional image generation with pixelcnn decoders. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29, pp. 4790–4798. Curran Associates, Inc. (2016)
28. Oord, A.V., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. In: Balcan, M.F., Weinberger, K.Q. (eds.) Proceedings of The 33rd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 48, pp. 1747–1756. PMLR, New York, New York, USA (20–22 Jun 2016)
29. Prillwitz, S.: HamNoSys Version 2.0. Hamburg Notation System for Sign Languages: An Introductory Guide. Intern. Arb. z. Gebärdensprache u. Kommunik, Signum Press (1989)
30. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. CoRR **abs/1511.06434** (2015), http://arxiv.org/abs/1511.06434
31. Reed, S.E., Akata, Z., Mohan, S., Tenka, S., Schiele, B., Lee, H.: Learning what and where to draw. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29, pp. 217–225. Curran Associates, Inc. (2016)
32. Robotka, Z.: Signall (2018), http://www.signall.us/
33. Siarohin, A., Sangineto, E., Lathuilière, S., Sebe, N.: Deformable gans for pose-based human image generation. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 3408–3416. Salt Lake City, United States (Jun 2018)

34. Stoll, S., Camgoz, N.C., Hadfield, S., Bowden, R.: Sign language production using neural machine translation and generative adversarial networks. In: British Machine Vision Conference (BMVC) (2018)
35. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
36. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4), 600–612 (April 2004)
37. Yan, X., Yang, J., Sohn, K., Lee, H.: Attribute2image: Conditional image generation from visual attributes. In: ECCV (4). Lecture Notes in Computer Science, vol. 9908, pp. 776–791. Springer (2016)
38. Yao, S., Lin, W., Ong, E., Lu, Z.: Contrast signal-to-noise ratio for image quality assessment. In: IEEE International Conference on Image Processing 2005. vol. 1, pp. I–397 (Sep 2005). https://doi.org/10.1109/ICIP.2005.1529771
39. Zhang, H., Starke, S., Komura, T., Saito, J.: Mode-adaptive neural networks for quadruped motion control. ACM Trans. Graph. **37**(4), 145:1–145:11 (Jul 2018). https://doi.org/10.1145/3197517.3201366, http://doi.acm.org/10.1145/3197517.3201366
40. Zwitserlood, I., Verlinden, M., Ros, J., Schoot, S.V.D.: Synthetic signing for the deaf: esign (2004)