**SynthSL: Expressive Humans for Sign Language Image Synthesis**

Jilliam M. Díaz Barros$^{1,2,*}$ Chen-Yu Wang$^{2,*}$ Jameel Malik$^{1,3}$ Abdalla Arafa$^{2}$ Didier Stricker$^{1,2}$  

$^1$ German Research Center for Artificial Intelligence (DFKI)  $^2$ RPTU Kaiserslautern  $^3$ NUST School of Electrical Engineering and Computer Science

---

**Abstract**—Around 5% of the world’s population live with disabling hearing loss. Despite recent advancements to improve accessibility to the Deaf community, research on sign language is still limited. In this work, we introduce a large-scale synthetic dataset on sign language, SynthSL, targeted to sign language production, recognition and translation. Using state-of-the-art methods for human body modeling, SynthSL aims to augment current datasets by providing additional ground-truth data such as depth and normal maps, segmentation masks and 2D/3D body joints. We additionally explore a generative architecture for the synthesis of sign images and propose a new generator based on Swin Transformers, conditioned on given body poses and appearance. We believe that an increase on the publicly available data on sign language would boost research and close the performance gap with related topics on human body synthesis. Our code, models and dataset are available at https://github.com/jilliam/SynthSL.

---

I. INTRODUCTION

Sign language (SL) is one of the main communication channels for people with severe or profound deafness. It is considered a natural language, where signs result from the combination of body postures, hand gestures and facial expressions, giving a representation in the spatio-temporal domain. It differs from the spoken language in both grammar and word order, without a one-to-one mapping from sign to word. Instead, the spoken sentences can be translated to glosses, which represent the written form of the signs [9].

Since SL perception is visual, it is actively investigated in computer graphics and vision, particularly for three main tasks: production, recognition and translation. SL production aims at generating sign videos or human pose sequences from written or spoken sentences, while recognition and translation refer to the interpretation of sign videos. Sign recognition targets the identification of glosses, whereas in translation, signs are mapped to written or spoken language words, in some cases using glosses in an intermediate step.

Communication with the Deaf community is a bidirectional process, and both recognition and production are essential to increase accessibility. Production, in particular, is a crucial step for them to understand live speeches, news, cultural activities, among others, specially when no interpreters are available. Signs are preferred over closed captions or subtitles, as SL users are accustomed to their own natural language, while written text is not always easily comprehensible for them [17], [40].

Although SL is used worldwide, it is not universal, since signs can vary widely in different regions, and several types and dialects might co-exist in the same country [17], [45], [46]. Each variation has a defined structure and its own grammar. This poses a challenge in sign recognition and translation, since datasets cannot be used cross-linguistically.

In the era of deep learning, large-scale datasets are of vital importance for training and testing. However, collecting such datasets is usually expensive and time consuming. Additionally, SL datasets require expert signers and annotators to label the data, in order to avoid interpretation errors due to similarity in the signs [46]. To address this limitation, some datasets have been collected from multiple sources, such as learning channels for SL [46] or from news or weather broadcasts [26], usually providing only RGB videos. Acquisition of additional ground-truth data such as 2D/3D skeletons, keypoints and body pose usually requires specialized systems such as 3D motion capture (MoCap) data or custom setups with hundreds of cameras and sensors [21]. Unfortunately, these systems are not easily accessible.

In this paper, we introduce a large-scale synthetic dataset on SL with expressive humans, SynthSL, based on the SMPL-X [59] model. Inspired by realistic synthetic datasets in human analysis such as SURREAL [77], we propose a pipeline for augmenting real RGB-based sign language...
datasets to include other modalities and add rich ground truth data such as depth information, rendered models, pose, body-part segmentation, normal maps and 2D/3D body joints. With the proposed data generation framework, we aim to extend existing datasets and advance research on SL production and interpretation. Furthermore, our rendering pipeline can contribute to close the gap between animated avatars used for SL production and synthetic sign videos. SynthSL and the synthetic sign generation code will be made publicly available for research purposes.

The use of synthetic data offers multiple advantages on SL research: (i) overcome the limitations of monocular RGB-based videos with extended ground truth data, especially for learning models; (ii) enable inter-signer variation. Body shapes and textures (with identity and clothing) are randomized, to increase the diversity in our dataset; (iii) ensure enough number of instances per word/sentence. Each sentence can be generated with different signers, adding random noise to the signs to increase variability. (iv) eliminate motion blurry, intrinsic to fast motions in video sequences; (v) control over the camera position and the acquisition of multiple views; and (vi) control of external factors such as illumination and background.

Finally, we propose a new pipeline for synthesizing sign language images, targeted to sign language production (SLP). This pipeline is based on Generative Adversarial Networks (GANs), where a base signer (or style) image, the current pose and a target pose are used as input. The model then generates an output image of the base signer in the target pose. We leverage the improved performance of transformers on vision tasks and introduce SwinGenerator, a new generator based on Swin Transformer V2 [50].

The main contributions of this work are:

- A new rendering pipeline based on the SMPL-X model, which integrates the shape and pose of the body, hands and face, in addition to facial expressions.
- SynthSL, a large-scale synthetic dataset for research on sign language. Our dataset expands the PHOENIX 2014T dataset [9] with ground truth on rendered models, segmentation masks, depth and normal maps, 2D/3D body joints, body pose and shape, which are provided on a frame basis.
- A new deep-learning based model for sign image synthesis, with a novel SwinGenerator. Our model takes as input a source and target pose, plus a style image, and generates an image of the signer in the given style and target pose.

II. RELATED WORK

In this section, we review the state of the art in SLP and analyze current SL large-scale datasets. We additionally describe related work on human analysis with synthetic datasets. For a comprehensive study on methods for SL recognition, we refer the reader to [1].

Sign Language Production. Traditionally, SLP has been mainly focused on the animation of 3D avatars, using MoCap data [17], [29], [30] or pre-designed synthetic sequences [31], [40], [56]. Recent works have moved towards the generation of skeleton poses or realistic video sequences, using deep neural networks (DNN). Some approaches rely in an intermediate step to translate spoken sentences to skeletal poses, using glosses. [71] produces sign videos from spoken language, by combining Neural Machine Translation (NMT) and a generative model. The sentences are translated to glosses using an encoder-decoder, which in turn are mapped to skeletal pose sequences using a dictionary. [72] overcomes the look-up table and generates skeletal pose sequences via Motion Graph, which is conditioned by a sequence of gloss probabilities extracted from the written or spoken sentences. [73] maps glosses to signs using a tailored DNN.

Other works explore the direct transformation from text to skeletal poses, without an intermediate gloss representation [25], [83] used an architecture based on neural networks to generate fixed-length sign pose sequences. [68] introduced progressive transformers for SLP, from discrete spoken sentences to continuous sign pose sequences. [67] extended the framework to integrate skeletal pose sequences via Motion Graph, which is conditioned by a sequence of gloss probabilities extracted from the written or spoken sentences. [73] maps glosses to signs using a tailored DNN.

More relevant to our work is the pose-conditioned human synthesis for SLP. PSGN [71] proposed a hybrid architecture composed of a Variational Autoencoder (VAE) [42] generator and DCGAN [60], to synthesize sign sequences conditioned on the skeletal information and appearance. [78] produces realistic sign videos using the GAN architecture for human motion transfer in Everybody Dance Now [13], while the synthesis pipelines in [72], [73] are based on pix2pixHD [79], with an encoder-decoder-based generator. Similarly, SignGAN [67], [69] generates realistic sign videos using a conditional GAN based on [13] and [79], given a style image and a sign pose sequence. It additionally introduces a keypoint-based loss to improve the synthesis of hands. SignDIFF in [25] is based on dual-condition diffusion models for SLP on American Sign Language (ASL).

Sign Language Datasets. Datasets can be classified according to the language, if they were captured in a controlled or unconstrained (in the wild) scenario and if they are targeted to finger-spelling [20], word (isolated) or sentence level (continuous) sign language. Table I lists a set of large-scale datasets for isolated and continuous sign language, from different regions around the world.

Some datasets were collected from weather forecasts [10], [26], [27], TV broadcast [3], [10], videos on YouTube [75], or different online channels of the Deaf community [46]. Others datasets comprise videos recorded in controlled environments, with multiview RGB [43], or multimodal systems [22], [24], [57], and provide a large variety of ground truth data, e.g. depth maps, 2D/3D keypoints, facial landmarks, body and hand pose, usually extracted from Kinect v1/v2 and OpenPose [11]. How2Sign [21] includes data captured in a custom setup with hundreds of cameras and sensors placed in a geodesic dome. Other datasets in-
Synthetic Datasets for Human Analysis. Synthetic datasets have been widely used in recent years to compensate the scarcity of large-scale data for training deep learning models [74], [14]. Many human-based datasets synthesize images either from virtual characters generated with computer graphic software [6], [28], [49], or by using parametric body models such as SCAPE [4] or SMPL [52]. The latter has gained special attention for synthetic human datasets [77], due to its realistic appearance and motion. The bodies are animated using datasets collected with MoCap. Their experiments show that simple models trained on this synthetic data can improve more elaborated SOTA approaches in human pose and shape (HPS) estimation.

[77] introduced SURREAL (Synthetic hUmans foR REAL tasks), the fist synthetic dataset with realistic rendered humans performing different actions. Pose and motion were captured using MoCap data and additionally to the 2D/3D body pose, the dataset provides depth map, segmentation masks, surface normals and optical flow. They demonstrated that CNNs trained on synthetic data can be used for human body segmentation and depth estimation on real RGB images. [76] presented SURREACT (Synthetic hUmans foR REal ACTions), a dataset intended to investigate the scope of temporal CNNs trained on synthetic data for human action recognition. They exploited the synthetic data to include augmentation, such as multiple viewpoints and variations in appearance and motion. [34] explored synthetic data for hand-object manipulation. They introduced a synthetic dataset, ObMan, which rendered the hands and body using the SMPL+H model [64], to produce synthetic RGB images, depth maps and segmentation masks. SMPL+H integrates a parametric hand model, MANO [64] to the SMPL model.

The images were rendered using Blender and its python API. [62] and [61] explore synthetic datasets with optical flow fields for single and multi-human motion analysis (MHOF). The output was rendered using a similar approach to [34]. For MHOF, additional ground truth such as depth maps, segmentation masks, normal maps, and SMPL+H shape and pose parameters are provided. This ground truth was created using the SMPL+H model and MoCap data.

[36] investigated multi-person pose estimation networks trained on a purely synthetic dataset and in an augmented dataset based on real images. Based on [23], they additionally presented a style-transfer method to map synthetic to real humans. BEDLAM [7] introduced a dataset of synthetic humans based on the SMPL-X model, with high level of realism. The bodies are animated using datasets collected with MoCap. Their experiments show that simple models trained on this synthetic data can improve more elaborated SOTA approaches in human pose and shape (HPS) estimation.

Previous work demonstrates that human-based synthetic datasets help improve DNN performance in vision-related tasks. With our dataset and rendering pipeline, we aim to help advance the research on SL, by providing additional types of ground-truth information to existing datasets.

### III. SYNTHSL AND RENDERING PIPELINE

We introduce a pipeline to render synthetic sign images, used to generate our large-scale dataset, SynthSL. Our rendering framework, depicted in Fig. 2, leverages the parametric model SMPL-X [59], to extend RGB-based sign videos with additional ground-truth annotations. To that end, we exploit a state-of-the-art model fitting pipeline that regresses the SMPL-X parameters from RGB images. These parameters are then used to animate the human model and generate synthetic data. Our pipeline can be applied to videos from different sources, such as existing SL datasets, broadcasts and YouTube videos. Nonetheless, its use is not limited to SL-related tasks and can be employed in the

### TABLE I

<table>
<thead>
<tr>
<th>Name</th>
<th>Language</th>
<th>Videos</th>
<th>Sentences</th>
<th>Signers</th>
<th>Vocab</th>
<th>Data type</th>
<th>Level</th>
<th>In the wild</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEVISION [12]</td>
<td>Chinese</td>
<td>24000</td>
<td>-</td>
<td>8</td>
<td>2000</td>
<td>RGB-D</td>
<td>Word</td>
<td>-</td>
</tr>
<tr>
<td>SMILE* [24]</td>
<td>Swiss German</td>
<td>12600**</td>
<td>-</td>
<td>42</td>
<td>100</td>
<td>RGB-D</td>
<td>Word</td>
<td>-</td>
</tr>
<tr>
<td>MS-ASL [75]</td>
<td>American English</td>
<td>25513</td>
<td>-</td>
<td>222</td>
<td>1000</td>
<td>RGB</td>
<td>Word</td>
<td>✓</td>
</tr>
<tr>
<td>BosphorusSign22k [57]</td>
<td>Turkish</td>
<td>22542</td>
<td>-</td>
<td>6</td>
<td>744</td>
<td>RGB-D</td>
<td>Word</td>
<td>-</td>
</tr>
<tr>
<td>PHOENIX 2012 [26]</td>
<td>German</td>
<td>190</td>
<td>1980</td>
<td>7</td>
<td>911</td>
<td>RGB</td>
<td>Sentence</td>
<td>-</td>
</tr>
<tr>
<td>PHOENIX 2014 [27]</td>
<td>German</td>
<td>645</td>
<td>6861</td>
<td>9</td>
<td>1558</td>
<td>RGB</td>
<td>Sentence</td>
<td>-</td>
</tr>
<tr>
<td>PHOENIX 2014T [9]</td>
<td>German</td>
<td>8257</td>
<td>8257</td>
<td>9</td>
<td>1066</td>
<td>RGB</td>
<td>Sentence</td>
<td>-</td>
</tr>
<tr>
<td>CSL [37]</td>
<td>Chinese</td>
<td>25000</td>
<td>-</td>
<td>100</td>
<td>50</td>
<td>RGB-D</td>
<td>Sentence</td>
<td>-</td>
</tr>
<tr>
<td>KETI* [43]</td>
<td>Korean</td>
<td>14672</td>
<td>-</td>
<td>105</td>
<td>14</td>
<td>RGB</td>
<td>Sentence</td>
<td>-</td>
</tr>
<tr>
<td>How2Sign [22]</td>
<td>American English</td>
<td>2500</td>
<td>38611</td>
<td>10</td>
<td>4000</td>
<td>RGB-D</td>
<td>Sentence</td>
<td>-</td>
</tr>
<tr>
<td>GSL [1]</td>
<td>Greek</td>
<td>10290</td>
<td>10290</td>
<td>7</td>
<td>310</td>
<td>RGB-D</td>
<td>Sentence</td>
<td>-</td>
</tr>
<tr>
<td>C4A - SWISSTXT-WEATHER [10]</td>
<td>Swiss German</td>
<td>183</td>
<td>811</td>
<td>-</td>
<td>1248</td>
<td>RGB</td>
<td>Sentence</td>
<td>-</td>
</tr>
<tr>
<td>C4A - SWISSTXT-NEWS [10]</td>
<td>Swiss German</td>
<td>181</td>
<td>6031</td>
<td>-</td>
<td>10561</td>
<td>RGB</td>
<td>Sentence</td>
<td>-</td>
</tr>
<tr>
<td>C4A - VRF-NEWS [10]</td>
<td>Flemish</td>
<td>120</td>
<td>7174</td>
<td>-</td>
<td>6875</td>
<td>RGB</td>
<td>Sentence</td>
<td>-</td>
</tr>
<tr>
<td>BOBSL [3]</td>
<td>British English</td>
<td>1962</td>
<td>1.2M</td>
<td>39</td>
<td>2281</td>
<td>RGB</td>
<td>Sentence</td>
<td>-</td>
</tr>
</tbody>
</table>

*Not publicly available. **This information was inferred from [72], where all signers perform each sign 3 times.
The SMPL-X model, taking its name from *SMPL eXPressive*, improves the SMPL model [52] by integrating detailed face and hand models to the given body template. The model was trained on multiple datasets of 3D human scans, incorporating the additional parametric models to increase the resolution of hands and face. Given a pose $\theta$, shape $\beta$, and facial expression $\psi$, the SMPL-X model is formulated as $M(\theta, \beta, \psi) : \mathbb{R}^{3\theta} \times \mathbb{R}^{\beta} \times \mathbb{R}^{\psi} \rightarrow \mathbb{R}^{3N}$, with $N = 10475$ being the number of vertices.

The pose parameters $\theta$ embody the pose for $K$ joints in the whole body, including jaw and finger joints. In total, SMPL-X has $K = 55$ joints, with one joint corresponding to the global orientation and the 54 left to body joints. 30 of these joints belong to the fingers, which are modelled based on MANO [64]. Such model uses a PCA pose space to reduce the dimension of the axis-angle rotations from 90 (3 angles per joint) to 24 parameters. Similarly, the facial expression is represented using a lower dimension PCA space with 10 parameters based on FLAME [47].

SMPL-X has in total 119 parameters: 99 for the pose (75 for \{ body, eyes and jaw \} and 24 for the hands), 10 parameters for the shape and 10 for the facial expression. In addition, it has the option to use a female, male or neutral body model.

**Augmented dataset.** SynthSL was generated from the training set of the publicly available dataset *PHOENIX 2014T* [9]. This dataset is targeted to continuous recognition and translation on German SL, and comprises sign videos describing the weather forecast. Even though this dataset has a large vocabulary size (1000+), the number of signers is relatively low (9) and is only RGB-based.

**Shape, pose and expressions.** Our dataset is based on the SMPL-X model, which comprises body, face and hands. It differs from related human body rendering pipelines such as SURREAL [77] and Obman [34], which use the SMPL (body) and SMPL+H (body and hands), respectively. The model parameters are estimated using SMPLer-X [8], a state-of-the-art fitting approach based on vision transformers (ViT) [19]. This method has currently the best performance for body modelling in the AGORA benchmark [58], which evaluates SMPL-X model fitting on a synthetic dataset with high realism. In SynthSL, we estimate the pose $\theta$ and facial expression $\psi$ from SMPLer-X with ViT-Small. These parameters are then passed to the rendering step. The body shape, on the other hand, is randomly generated for every sequence to increase the diversity of the signers.

**Texture and background.** The body textures are randomly sampled from the BEDLAM dataset [7]. It includes 50 skin textures for male and 50 for female models, with large diversity in terms of identity and skin tones. In addition, the dataset provides 1738 clothing textures which are overlaid on the given skins.

Our rendering pipeline offers the option to randomly select backgrounds from ImageNet [18] or LSUN [82]. In our dataset, we fixed the background to only one specific color, to simulate the controlled recording scenario in sign language videos.

**Rendering.** The RGB images are rendered using the Python API from Blender [16], similarly to [34]. However, our pipeline generates multiple types of ground truth data, such as depth, segmentation and normal maps, following SUR-REAL and MHOF. In contrast to these pipelines, the facial expression and jaw pose are also integrated, to render expressive models. The RGB images, segmentation masks, depth and normal maps are rendered with a default resolution of $512 \times 512 \times 3$. SMPL-X model parameters such as body pose, shape, facial expression and 2D/3D body joints are provided in SynthSL as well. SMPL-X is available in Autodesk’s Filmbox (FBX) file format, which enables the animation in software such as Blender and Maya [5].

Our rendering pipeline provides multiple benefits to SL research: (1) to augment existing RGB-based datasets; (2) to assist in the design of DNN architectures, where various...
types of data could be integrated into SL-based models; and (3) to guide in the collection of new datasets, where different technologies and algorithms could be evaluated beforehand.

IV. SIGN LANGUAGE IMAGE SYNTHESIS

In addition to the rendering pipeline and our synthetic dataset SynthSL, we investigate the synthesis of sign images for SLP. This task can be divided into three main stages: (i) transcription of spoken to written words, (ii) translation from written words to glosses and (iii) mapping from glosses to signs for production. In this work, we focus on the third stage, the generation of sign images and videos from glosses. More specifically, we investigate the generation of sign images from skeleton poses representing the glosses.

Inspired by [71], we designed a GAN architecture to generate sign images from an input base signer and a target pose. This problem is often termed as image to image translation, where the generator $G$ maps the input data to a given visual domain, conditioned on a given input data. The discriminator $D$ decides whether the generated image is fake or not, and trains in tandem along with the generator. Our pipeline introduces the pose of the base signer as an additional input to $G$, as shown in Fig. 3. We additionally propose a new generator architecture for image synthesis, based on Swin Transformers [50], [51].

A. Generator and Discriminator Training

The generator $G$ and discriminator $D$ are trained in intervals with an adversarial training, where $G$ tries to maximize the number of images that deceives $D$, and $D$ penalizes $G$ for generating unrealistic images [32].

$D$ is trained using the adversarial loss [32], given by:

$$L_{adv} = \mathbb{E}_y[\log(D(y))] + \mathbb{E}_x[\log(1 - D(G(x)))],$$  \hspace{1cm} (1)

where $\mathbb{E}[]$ is the expected value and $D(\cdot)$ is the probability estimated by the discriminator that the given input corresponds to real or generated data. In the first term in (1), $D$ takes as input the ground-truth sign image, $y$. This term enhances the ability of $D$ to correctly identify real images.

In the second term, $D$ assesses the image reconstructed by the generator, $G(x)$, and is used to improve the capability of $D$ to distinguish $G(x)$ as fake or real.

In our architecture, the input of the generator $G$ corresponds to the concatenation of the base signer, base pose and target pose, $x$, while the output $G(x)$ is an RGB image depicting the base signer in the given target pose. We propose to train $G$ with the loss function:

$$L_G = \lambda_{rec} L_{rec} + \lambda_{adv} L_{adv} + \lambda_{perc} L_{perc},$$  \hspace{1cm} (2)

with $L_{rec}$ being the reconstruction loss given by the L1 pixel-wise distance between the ground truth and reconstructed image at $G(x)$; $L_{adv}$ is the adversarial image loss in (1); $L_{perc}$, the perceptual loss [39]; and $\lambda(\cdot)$ is the weight of each loss.

The final objective is then formulated as:

$$L = \lambda_{rec} L_{rec} + \lambda_{adv} \arg \min_G \max_D \lambda_{adv} L_{adv} + \lambda_{perc} L_{perc}. \hspace{1cm} (3)$$

Generator Network. We introduce a new generator model, SwinGenerator, as depicted in Fig. 4. The backbone of our generator $G$ is based on Swin Transformer V.2 [50], which improves previous models such as ViT, by integrating shifting windows [51]. Since Swin Transformer applies the attention mechanism to patches of the window instead of the full image, the model complexity is reduced, with improved computational performance. This enhancement allows the use of this model on higher-resolution sign images.

In our SwinGenerator, the Swin Transformer V.2 serves as the encoder, with each block containing paired W-MSA (Windows Multi-Head Attention) and SW-MSA (Shifted Windows Multi-Head Attention) for feature extraction. Patch merging is used for down-sampling, reducing the size of the output feature map by half and doubling the number of channels.

To enable the generation of images conditioned on a given pose and appearance, we propose to add up-sampling and un-patching layers, which act as a decoder and map the embedded code to the synthesized sign image. The upsampling layers are based on transposed convolutions,
followed by two convolution layers, and the un-patching layer recovers the original image structure from the patches.

We additionally integrate skip connections between the corresponding down-sampling and up-sampling layers, to improve the convergence of the model. Furthermore, the target pose is resized and concatenated to the output of every Swin block and to the corresponding skip connection required for the decoder.

V. EXPERIMENTS AND RESULTS

A. Implementation

The proposed synthesis pipeline was developed in PyTorch and trained on a Nvidia RTX A6000 GPU. Inference time was computed on an Nvidia RTX2070.

B. Datasets

We trained and evaluated our synthesis pipeline using SynthSL, PHOENIX 2014T and BosphorusSign22k. Images in PHOENIX 2014T have resolution of $210 \times 260 \times 3$, while BosphorusSign22k has full HD images of $1920 \times 1080 \times 3$ pixels. Since BosphorusSign22k is very large, we used 4% of the training data, which we found was sufficient for generating realistic sign language images. Nonetheless, we used the complete test set for evaluation. In SynthSL, we selected 1000 sequences, with 600 for training, 100 for validation and 300 for testing. The training set alone has 79362 frames in total, with an average of $\sim 132$ frames per sequence. The test set has 35848 frames, with $\sim 119$ frames per sequence.

C. Data pre-processing and model architecture

During pre-processing, input images were resized to $256 \times 256 \times 3$. For BosphorusSign22k, we trimmed 1/4 from each side, to have square images. The face and body keypoints were resized accordingly.

SwinGenerator uses a Swin-T V2, with number of channels $C = 96$ and block $\{2, 2, 6, 2\}$. Its input corresponds to the base signer RGB image, plus the base and target poses. The output is an RGB image of the signer in the new pose, with the same size of the base signer image, $256 \times 256 \times 3$. The base and target poses have each $k = 96$ 2D keypoints extracted with MediaPipe [53], where 42 belong to the hands, 46 to the face, and 8 to the body. Although there are many recent works on pose estimation from images [81], [54], we employ MediaPipe due to its highly robust estimations from RGB data and real-time performance. We map the extracted keypoints to Gaussian heatmaps, with one channel per keypoint.

The discriminator is composed of two convolutional layers and three residual blocks. It takes as input the ground truth image and the generated image, as depicted in Fig. 3. The discriminator is updated one time per every generator update. We used the Adam optimizer [41] with learning rate $\alpha_G = 2e^{-3}$ for $G$ and $\alpha_D = 2e^{-4}$ for $D$. In (2), we set $\lambda_{rec} = 5$, $\lambda_{adv} = 2$ and $\lambda_{perc} = 0.5$. All the models are trained for 30 epochs, with batch size of 32. For the perceptual loss, we used layers $\{1, 3, 5, 9, 13\}$ from VGG19 [70].

D. Baseline architectures

Since existing sign language synthesis models are not publicly available, we implemented a baseline architecture inspired by [71]. We introduced five main changes to the architecture in [71]: (i) the generator is based on U-Net [65] instead of VAE, similarly to pix2pix [38]; (ii) the keypoints are mapped to Gaussian heatmaps, instead of binary heatmaps; (iii) we used the same keypoints $k = 96$ as in our architecture, and not 10; (iv) we added convolutional blocks in the input of the encoder and output of the decoder, to generate higher quality images of size $256 \times 256 \times 3$, instead of $128 \times 128 \times 3$; and (v) we introduced the perceptual loss to train the generator. We refer to this model as “U-Net”.

Additionally, we implemented a second baseline generator model, named as ”ResNet”. This models is composed of deep residual blocks [35] in the encoder, following related works on image synthesis [39], [79], [84] and SLP [69], [72], while the decoder is given by transposed convolutions.

In both models, the input is given by the concatenation of the base signer and target pose, following [69], [71], [72]. We use the same loss function in (3) and discriminator as in our model for training and evaluation, for a fair comparison.

E. Metrics

The performance of the proposed sign image synthesis pipeline in Fig. 3 is evaluated using the structural similarity index metric (SSIM), the peak-signal-to-noise ratio (PSNR), the position errors (PE) and the Fréchet Inception Distance (FID). In the SSIM, PSNR and FID the generated RGB image is compared to the respective ground truth RGB image in the given pose. For the PE, we extracted the keypoints from the ground truth and generated images using MediaPipe, and summed the Euclidean distance between the respective keypoints.

F. Evaluation

The quantitative results of the proposed synthesis approach on PHOENIX 2014T, SynthSL and BosphorusSign22k are reported in Tables II, III and IV, respectively, under SwinGenerator. Qualitative results for the three datasets are shown in Fig. 5, and in the Supplementary material. In contrast to PHOENIX 2014T and BosphorusSign22k, SynthSL has large diversity in terms of signer and clothes. Furthermore, the images do not suffer from blurriness, thereby enabling the synthesis of sharp hands without additional datasets.

Generator architecture. We evaluated the influence of the proposed SwinGenerator and compared it to the generator architectures based on U-Net and ResNet, described in Section V-D. The results for PHOENIX 2014T, SynthSL and BosphorusSign22k are shown in Tables II, III and IV, respectively, and in Fig. 5.

In all datasets, the ResNet-based architecture showed an improved performance with respect to U-Net in terms of SSIM and PE. For the PSNR and FID, the results differ in each dataset for U-Net and ResNet. Our SwinGenerator outperforms both architectures in every dataset and metric, as shown in Tables II, III and IV. Furthermore, the image
Fig. 5. Sign images synthesized with the proposed pipeline and baseline architectures on SynthSL, PHOENIX 2014T and BosphorusSign22k.

### TABLE II
**Quantitative evaluation on Phoenix 2014T**

<table>
<thead>
<tr>
<th>Method</th>
<th>SSIM↑</th>
<th>PSNR↑</th>
<th>PE↓</th>
<th>FID↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-Net</td>
<td>0.684</td>
<td>30.999</td>
<td>0.884</td>
<td>42.624</td>
</tr>
<tr>
<td>ResNet</td>
<td>0.745</td>
<td>28.692</td>
<td>0.832</td>
<td>47.624</td>
</tr>
<tr>
<td>SwinGenerator</td>
<td>0.835</td>
<td>33.971</td>
<td>0.675</td>
<td>31.678</td>
</tr>
</tbody>
</table>

### TABLE III
**Quantitative evaluation on SynthSL**

<table>
<thead>
<tr>
<th>Generator Architecture</th>
<th>SSIM↑</th>
<th>PSNR↑</th>
<th>PE↓</th>
<th>FID↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-Net</td>
<td>0.885</td>
<td>33.050</td>
<td>0.469</td>
<td>14.160</td>
</tr>
<tr>
<td>SwinGenerator</td>
<td>0.911</td>
<td>36.382</td>
<td>0.391</td>
<td>7.544</td>
</tr>
</tbody>
</table>

quality is degraded for both U-Net and ResNet using the same discriminator and loss function as in our model. In many cases, SwinGenerator provides steadier hands reconstructions (see 1st column of PHOENIX 2014T in Fig. 5), specially if the hands suffer from blurriness. However, in some cases ResNet generates slightly better hands as in the last column of BosphorusSign22k in Fig. 5.

### G. Ablation Study

Transformer model. In this ablation study, we investigated the influence of different modules within SwinGenerator for synthesizing images using SynthSL. We first evaluated the SwinGenerator without concatenating the resized target pose in the residual connections. This architecture is shown in Table V as Swin-T_{noTP}. The results show that adding the target pose in the residual connections is advantageous for SwinGenerator, with improvements in every metric, specially the PE and FID.

We additionally implemented a pipeline with only the source image as input, and the residual connection incorporating the difference between target pose and base pose. This model is shown in Table V as Swin-T_{Res}. This architecture reduces the model size, but compromises some level of quality as compared to the proposed SwinGenerator.

Transformer backbone. Our architecture was also evaluated with different backbone sizes of Swin V2 on SynthSL. More specifically, we compared the tiny Swin-T with the small (Swin-S) and big (Swin-B) versions of Swin Transformers.

### TABLE IV
**Quantitative evaluation on BosphorusSign22k**

<table>
<thead>
<tr>
<th>Generator Architecture</th>
<th>SSIM↑</th>
<th>PSNR↑</th>
<th>PE↓</th>
<th>FID↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-Net</td>
<td>0.769</td>
<td>29.673</td>
<td>0.420</td>
<td>34.161</td>
</tr>
<tr>
<td>ResNet</td>
<td>0.865</td>
<td>32.150</td>
<td>0.302</td>
<td>21.356</td>
</tr>
<tr>
<td>SwinGenerator</td>
<td>0.930</td>
<td>37.589</td>
<td>0.211</td>
<td>15.824</td>
</tr>
</tbody>
</table>

### TABLE V
**Evaluation of different SwinGenerator architectures on SynthSL**

<table>
<thead>
<tr>
<th>Generator Architecture</th>
<th>SSIM↑</th>
<th>PSNR↑</th>
<th>PE↓</th>
<th>FID↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Swin-T_{noTP}</td>
<td>0.892</td>
<td>36.170</td>
<td>0.511</td>
<td>16.301</td>
</tr>
<tr>
<td>Swin-T_{Res}</td>
<td>0.895</td>
<td>36.362</td>
<td>0.468</td>
<td>12.633</td>
</tr>
<tr>
<td>SwinGenerator</td>
<td>0.911</td>
<td>36.382</td>
<td>0.391</td>
<td>7.544</td>
</tr>
</tbody>
</table>

### TABLE VI
**Evaluation of different sizes of Swin V2 on SynthSL**

<table>
<thead>
<tr>
<th>Swin Architecture</th>
<th>SSIM↑</th>
<th>PSNR↑</th>
<th>PE↓</th>
<th>FID↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Swin-T</td>
<td>0.911</td>
<td>36.382</td>
<td>0.391</td>
<td>7.544</td>
</tr>
<tr>
<td>Swin-S</td>
<td>0.912</td>
<td>36.810</td>
<td>0.387</td>
<td>7.346</td>
</tr>
<tr>
<td>Swin-B</td>
<td>0.918</td>
<td>37.114</td>
<td>0.374</td>
<td>6.304</td>
</tr>
</tbody>
</table>

### TABLE VII
**Evaluation of different adversarial losses on SynthSL**

<table>
<thead>
<tr>
<th>Loss function</th>
<th>SSIM↑</th>
<th>PSNR↑</th>
<th>PE↓</th>
<th>FID↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vanilla</td>
<td>0.911</td>
<td>36.382</td>
<td>0.391</td>
<td>7.544</td>
</tr>
<tr>
<td>LS-GAN [55]</td>
<td>0.910</td>
<td>36.716</td>
<td>0.391</td>
<td>7.845</td>
</tr>
<tr>
<td>Hinge [48]</td>
<td>0.909</td>
<td>36.628</td>
<td>0.386</td>
<td>8.856</td>
</tr>
<tr>
<td>WGAN-GP [33]</td>
<td>0.913</td>
<td>36.803</td>
<td>0.394</td>
<td>7.927</td>
</tr>
</tbody>
</table>

The quantitative results are shown in Table VI.

As expected, enhanced results were observed when using larger Swin Transformer models. However, this comes at a computational cost as the model size and complexity increase 2 and 4 times respectively, when using Swin-S and Swin-B. In Swin-S and Swin-B, the number of layers are \( \{2, 2, 18\} \) while the channel number of the hidden layers in the first stage \( C \) is 96 for Swin-S and 128 for Swin-B.

**Loss function.** We additionally compared different adversarial loss functions to train the generator and discriminator on SynthSL. The results in Table VII show that the choice of the adversarial loss does not have a significant impact on the performance of our pipeline.

**Pre-trained model.** Finally, we evaluated the effect on image synthesis when pre-training the networks in other datasets. Results are shown in Table VIII. Overall, pre-training on other datasets result on an improved performance in every metric in PHOENIX 2014T and BosphorusSign22k.

### H. Runtime Analysis

We estimated the average inference time for the different generator architectures after 300 iterations. For U-Net, ResNet and SwinGenerator, a single frame was generated on 43.59ms, 112.60ms and 46.34ms, respectively. Our SwinGenerator has a frame rate of \( \sim 22 \) FPS.

### I. Limitations and Failure Cases

In SynthSL, we observed some artifacts if the clothes of the signer had similar texture to the skin, as shown in Fig. 6. This problem did not occur in PHOENIX 2014T and BosphorusSign22k, as the signers always wear dark clothes. These artifacts were also present in images generated with U-Net and ResNet. For PHOENIX 2014T and BosphorusSign22k, some synthesized images had noisy regions in the face or clothes of the signers, as shown in Fig. 6. We believe this noise is due to low variability in these datasets regarding skin color and clothes, as it did not occur in SynthSL.

### J. Discussion

Our evaluation shows that the proposed SwinGenerator introduces a significant improvement with respect to the U-Net and ResNet-based generators, widely used for image synthesis, including SLP. These improvements were observed in all the evaluated datasets in Tables II, III and IV.

We additionally found that the ResNet-based generator tends to suffer from mode collapse with the current pipeline. In the literature, ResNet-based architectures for SLP [69], [72] are usually trained with the multi-scale discriminator from pix2pixHD [79]. In our experiments, we used a simple discriminator model to focus mainly on the performance of the evaluated generators. The multi-scale discriminator could be used as an alternative for an improved performance on the synthesis of high-resolution images.

Finally, we observed that on real datasets for SLP hand synthesis suffers from blurriness, due to the inherent fast motion on sign gestures. In that regard, [69] provides a framework and keypoint-based loss with outstanding hand reconstruction. However, a private dataset with high quality hands and manually selected frames without blurriness were used. Since this dataset is not publicly available, SynthSL could be used as an alternative. With our dataset SynthSL and the proposed rendering pipeline, we aim to provide additional resources for research on SLP.

### VI. Conclusions

In this paper, we introduced SynthSL, a large-scale synthetic dataset for research on sign language. We also proposed a rendering pipeline, to extend RGB-based sign videos to include rich ground truth data such as body, hands and head poses, segmentation masks, depth and normal maps. In addition, we presented a SL synthesis architecture conditioned on a given body pose and appearance. To that end, we leverage a GAN to generate realistic sign sequences with a newly introduced SwinGenerator for image synthesis. For future work, we would like to investigate the effect of additional input data such as depth map and segmentation masks for image synthesis. Furthermore, we aim to integrate additional ground truth data in our rendering pipeline such as optical flow, which can be exploited in SL-related tasks as in [63].


