Jointly Harnessing Prior Structures and Temporal Consistency for Sign Language Video Generation

YUCHENG SUO, Zhejiang University, Zhejiang, China
ZHEDONG ZHENG, University of Macau, Macau, China
XIAOHAN WANG, Zhejiang University, Zhejiang, China
BANG ZHANG, Alibaba Group, Zhejiang,China
YI YANG, Zhejiang University, Zhejiang, China*

Sign language provides a way for differently-abled individuals to express their feelings and emotions. However, learning sign language can be challenging and time-consuming. An alternative approach is to animate user photos using sign language videos of specific words, which can be achieved using existing image animation methods. However, the finger motions in the generated videos are often not ideal. To address this issue, we propose the Structure-aware Temporal Consistency Network (STCNet), which jointly optimizes the prior structure of humans with temporal consistency to produce sign language videos. We use a fine-grained skeleton detector to acquire knowledge of body structure and introduce short-term cycle loss and long-term cycle loss to ensure the continuity of the generated video. The two losses and keypoint detector network are optimized in an end-to-end manner. Quantitative and qualitative evaluations on three widely-used datasets, namely LSA64, Phoenix-2014T, and WLASL-2000, demonstrate the effectiveness of the proposed method. We hope this work can contribute to future studies on sign language production.

CCS Concepts: • **Computing methodologies → Animation**.

Additional Key Words and Phrases: Sign Language, Motion Transfer, Video Generation, Jointly Training.

ACM Reference Format:
Yucheng Suo, Zhedong Zheng, Xiaohan Wang, Bang Zhang, and Yi Yang. 2023. Jointly Harnessing Prior Structures and Temporal Consistency for Sign Language Video Generation. *ACM Trans. Multimedia Comput. Commun. Appl.* 9, 4 (June 2023), 17 pages. https://doi.org/10.1145/3648368

1 INTRODUCTION

Sign language is a type of visual language that conveys meanings through hand gestures and facial expressions [73]. According to the World Federation of the Deaf (WFD), approximately 72

Yucheng Suo, Xiaohan Wang and Yi Yang are with the College of Computer Science and Technology, Zhejiang University, China 310027. E-mail: suoych@gmail.com, xiaohan.wang@zju.edu.cn, yangyics@zju.edu.cn. Zhedong Zheng is with the Faculty of Science and Technology, and Institute of Collaborative Innovation, University of Macau, China. E-mail: zhedongzheng@um.edu.mo. Bang Zhang is with DAMO Academy, Alibaba Group, China 311121. E-mail: zhangbang.zb@alibaba-inc.com.
* Yi Yang is the corresponding author.
This work is partially supported by Major program of the National Natural Science Foundation of China (Grant Number: T2293723). This work is also supported in part by the Natural Science Foundation of Zhejiang Province (DT23F020008).
Authors' addresses: Yucheng Suo, Zhejiang University, 310013, Zhejiang, China; Zhedong Zheng, University of Macau, Macau, China; Xiaohan Wang, Zhejiang University, 310013, Zhejiang, China; Bang Zhang, Alibaba Group, 311121, Zhejiang,China; Yi Yang, Zhejiang University, 310013, Zhejiang, China*.

1551-6857/2023/6-ART $15.00
https://doi.org/10.1145/3648368

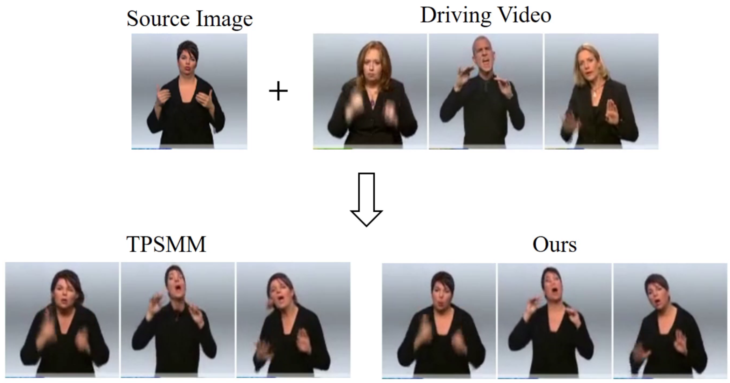

Fig. 1. The example picture of sign language motion transfer results. Given a source image and a driving video, the model generates a new video clip where the person in the source image performs the sign language motion in the driving video. Compared with the state-of-the-art method TPSMM [93], our method can generate smooth videos while preserving identity attributes such as hair and face. Check out the video example at https://youtu.be/2XL8o34hrHc.

million people worldwide use sign language [47]. However, learning sign language can be time-consuming and challenging, making it impractical for many people. Additionally, sign language varies depending on the local language and culture [32]. To address this, we aim to animate user photos based on sign language videos, allowing everyone to communicate using sign language without having to learn it.

Over the past few years, significant progress has been made in image animation [64–66, 93]. Given a video and an image containing the same type of object, the goal of image animation is to generate a new video whose object comes from the image and the motion comes from the video. However, when applying existing motion transfer methods to sign language generation, two main limitations arise. Firstly, the importance of body structure is often underestimated, as many works [64–66, 93] extract body keypoints in an unsupervised manner. These keypoints are not always aligned with the semantic body parts, making it difficult to capture detailed motions, especially for small-scale patterns like fingers. As shown in Figure 1, finger motions are often missing or blurred. Secondly, there is a lack of long-term temporal consistency in recent works [64–66, 93] that focus on short-term continuity between two frames. When given a pair of images, these methods only prioritize the quality of the reconstructed single frame during training, as shown in the top part of Figure 2, while ignoring the continuity and consistency of more frames in the future.

To overcome these limitations, we propose the Structure-aware Temporal Consistency Network (STCNet), a human body structure-aware network that generates sign language videos with high quality and continuity. The proposed framework has three main features. First, we employ a fine-grained keypoint detector network that provides strong human body structure knowledge, enhancing hand motion estimation. Second, we propose short-term cycle loss and long-term cycle loss to promote the continuity of the generated videos. Finally, to address the instability of the keypoint detector network's output, we adopt a jointly training strategy to fine-tune the pre-trained network without additional annotations.

We conduct extensive experiments on three sign language datasets: LSA64 [56], Phoenix-2014T [6], and WLASL-2000 [40]. The results demonstrate that our method outperforms state-of-the-art methods, including Monkey-Net [64], First Order Motion Model (FOMM) [65], Articulated Animation (AA) [66], and Thin-Plate Spline Motion Model (TPSMM) [93], in terms of the quality of the generated videos. As shown in Figure 1, our method generates smooth videos with correct

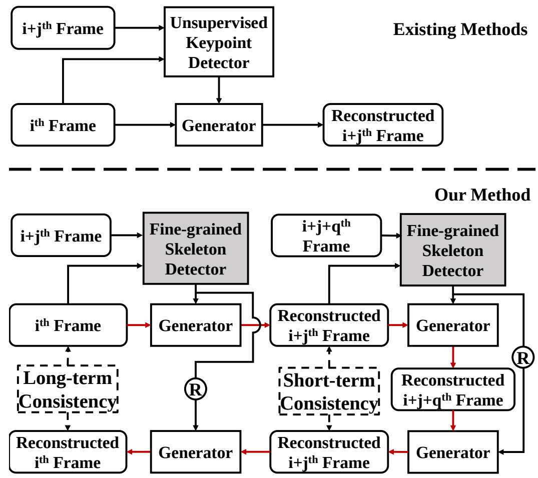

Fig. 2. **Comparison between existing methods and our method** Existing methods [64–66, 93] typically use an unsupervised keypoint detector and perform a single-frame generation procedure during training. In contrast, our method utilizes a fine-grained skeleton detector and enforces two types of temporal consistency. The Ⓡ in the figure indicates that we exchange the source image and the driving image to estimate the motion reversely. The red arrows show the generation order of our method during training.

motion details, as compared to the state-of-the-art method TPSMM. Briefly, our contributions can be concluded as follows:

- We propose a new Structure-aware Temporal Consistency network (STCNet). In particular, we explicitly introduce the prior human keypoints to guide the generation and involve the temporal consistency objective to further regularize the training process.
- Extensive experiments on LSA64 [56], Phoenix-2014T [6], and WLASL-2000 [40] datasets show that our approach surpasses several competitive methods, verifying the effectiveness of the proposed method.

2 RELATED WORKS

2.1 Skeleton keypoint Detection

Skeleton keypoint detection, also known as pose estimation, is to locate the essential parts of people in an image or a video [14]. The pioneering work in deep learning-based pose estimation is DeepPose [74], which outperforms traditional methods based on regression or retrieval [15, 18]. State-of-the-art methods are typically derived from Convolutional Neural Networks [8, 50]. OpenPose [7, 8, 67, 82] is one of the most popular methods in the research community, capable of estimating whole-body pose. In essence, all pose estimation methods can be divided into top-down methods and bottom-up methods [90]. The bottom-up method involves detecting joints first and gathering several joints to estimate the pose of a human. Representative works include DeepCut [53], Associative Embedding [48], PifPaf [36], OpenPifPaf [37], Keypoint Communities [90], *etc.*

On the contrary, the top-down method involves detecting a human first and then estimating the joints within the bounding box. CFN [30] uses a "Coarse-Fine" network structure to exploit multi-level supervision. CPN [11] introduces a cascaded pyramid network that aims to deal with occluded keypoints. CrowdPose [42] designs a person-joint connection graph to deal with wrong joint assembling and redundant pose prediction. RMPE, also known as Alphapose [19, 43] designs Symmetric Spatial Transformer Network, Parametric Pose NonMaximum-Suppression, and Pose-Guided Proposals Generator to handle inaccurately detected bounding boxes. Inspired by recent sign language translation works [21, 28, 69, 96], we adopt a keypoint detector to facilitate the sign language understanding in this work. Since every sign language video only contains one signer in the center, we skip the human detection process in practice and fine-tune the off-the-shelf AlphaPose method to extract key points.

Video pose estimation is different from image-based pose estimation since video requires temporal continuity and identity tracking [14, 94]. Cherian *et al.* [13] propose extending the spatial graph model with temporal links to capture motions of specific human body parts. The extra links ensure temporal consistency with additional parameters. Nie *et al.* [85] propose a unified spatial-temporal model to jointly accomplish video pose estimation and action recognition, thus the estimated pose is aligned with action semantics. Deepflow [83] and Thin-Slicing [71] are two works using optical flow to improve continuity by introducing temporal information. Pfister *et al.* [52] utilize similar techniques, demonstrating the effectiveness of optical flow. UniPose [4] leverages the LSTM network to provide the memory of adjacent frames. Recent work DiffsionPose [55] uses a diffusion model to estimate human pose and achieves remarkable results on various datasets. Researchers also curate benchmarks like YoutubePose [10] and PoseTrack [3] for videos in various domains and complex poses. In this paper, we finetune the keypoint detector network in an end-to-end manner, ensuring the temporal consistency of sign language videos.

2.2 Image Animation

Image animation refers to synthesizing action given an image and a driving video. Efros *et al.* [17] propose a retrieval-based action synthesis method. In recent years, most motion transfer generation networks deploy deep networks [86].

For instance, MoCoGAN [75] decomposes motion and content into separate representations to generate video. Yang *et al.* propose a two-stage approach, *i.e.*, PSGAN and SCGAN [87], to transfer motions collaboratively. Everybody Dance Now (EDN) [9] is another two-stage motion transfer network. The first stage is to generate the image frame as a whole and the second stage is to realism to the face region. Zhou *et al.* propose a dance motion transfer network using a spatial transformer network [97]. G^3AN [80] is a three-stream video generation network to disentangle motion features. Siarohin *et al.* propose a series of works on motion transfer including Monkey-Net [64], First Order Motion Model (FOMM) [65], and Articulated Animation (AA) [66]. Monkey-Net is an end-to-end motion transfer network, which learns to detect keypoint in an unsupervised way. FOMM calculates the first-order Taylor expansion in a neighborhood of the keypoint locations. AA transfers the motion from the essential regions of the object. Liu *et al.* adopt neural-ODEs for motion deformation [44]. Yoon *et al.* design a network to animate images using UV maps produced by a 3D human model [89]. Thin-Plate Spline Motion Model (TPSMM) by Zhao *et al.* [93] applies Thin-Plate Spline (TPS) transformation based on FOMM. However, TPSMM needs five sets of keypoints which bring redundancy. In contrast, our work differs from existing works by mining semantic-aware keypoints. Moreover, inspired by the spirit of the cycle consistency [79], we consider the temporal consistency as an essential influence factor of the generated video quality, which is crucial for sign language understanding [12, 22, 41, 54].

Diffusion models [25, 70] recently achieved impressive results on generation tasks like image generation [49] and super-resolution [20]. Researchers also leverage diffusion to generate data for discriminative tasks [88]. Diffusion models sample data from the distribution and gradually add noise by the diffusion process. Then diffusion models learn the reverse process which is to denoise and reconstruct the sample. However, diffusion models require high computing resources. To address this challenge, DDIM [70] boosts the inference speed by skipping step sampling. For conditional generation, GLIDE [49] injects CLIP text features to guide the generation process. Current works substitute the text feature with human pose features to achieve pose-guided video generation. DreamPose [34] leverages pose sequences to generate fashion videos. DISCO [78] disentangles foreground and background features to improve background quality. Our method differs from these works in two points. First, we acquire human poses from raw videos. Second, our method captures fine-grained finger motion details.

2.3 Sign Language Production

Sign Language Production (SLP) is a research field related to generating sign language videos, as highlighted in several recent studies [29, 38, 58–60, 62, 63, 91]. Saunders *et al.* proposes an Adversarial Multi-Channel approach [58] to generate sign language pose sequences. Saunders *et al.* use progressive transformer [60] to generate consecutive pose sequences, improving the BLEU performance. Everybody Sign Now [59] takes the spoken language as input and samples skeleton pose to generate photorealistic sign language videos. Zelinka *et al.* [91] propose a CNN-based method to deal with text-to-video sign language pose synthesis. AnonySign proposed by Saunders *et al.* [61] is the most related work, they implicitly encode the style features to generate sign language videos. However, the video generation procedure in their method does not take time-consistency into consideration. Ventura *et al.* [77] and Duarte *et al.* [16] deploy Everybody Dance Now (EDN), to generate sign language videos. EDN is a two-stage network that directly takes keypoints as input, yet our method conforms to end-to-end training paradigm and extracts keypoints from sign language videos. SIGNGAN proposed by Saunders *et al.* [63] aims to produce sign language videos given spoken languages. The sign pose is selected in a given pose dictionary.

3 METHOD

We propose STCNet, a body structure-aware framework that focuses on sign language motion transfer while maintaining the cycle consistency of time. As Figure 3 shows, STCNet consists of four parts, a keypoint detector network, a motion estimation network, an encoder, and a decoder.

3.1 Keypoint Detector Network

To obtain explainable and accurate keypoint locations, we adopt Alphapose model pre-trained on the Halpe dataset [43]. Sign language videos only contain the upper part human body, and the semantic information is revealed by the hand gestures and the facial expressions of the signer. Therefore, we select 21 vital keypoints with 12 on the hands, 5 on the upper body, and 4 on the face to remove redundancy without losing the details. Given the i^{th} frame $I^i \in \mathbb{R}^{H\times W\times 3}$ in a video clip, we pass the image to the Alphapose model F_P and get the result coordinates $K^i \in \mathbb{R}^{21\times 2}$. The keypoint detection procedure can be formulated as follow:

$$K^i = F_P(I^i). \tag{1}$$

Empirically, we find that the output of two continuous frames within a video clip generated by the pre-trained AlphaPose model differs a lot, especially when the frames are vague. One possible reason behind the large discrepancy between the detection results could be a lack of continuity of time. To address this problem, we fine-tune the keypoint detector along with the training procedure of the other modules. The keypoint detector shares the optimization goals with the other modules, thus no extra annotation is required. Therefore, it is conducted on single frames but preserves the temporal consistency in sign language videos. A violent fine-tuning procedure leads to missing the body structure information provided by the pre-trained model. Hence, we set the learning rate at a smaller value to fine-tune the detection module slowly.

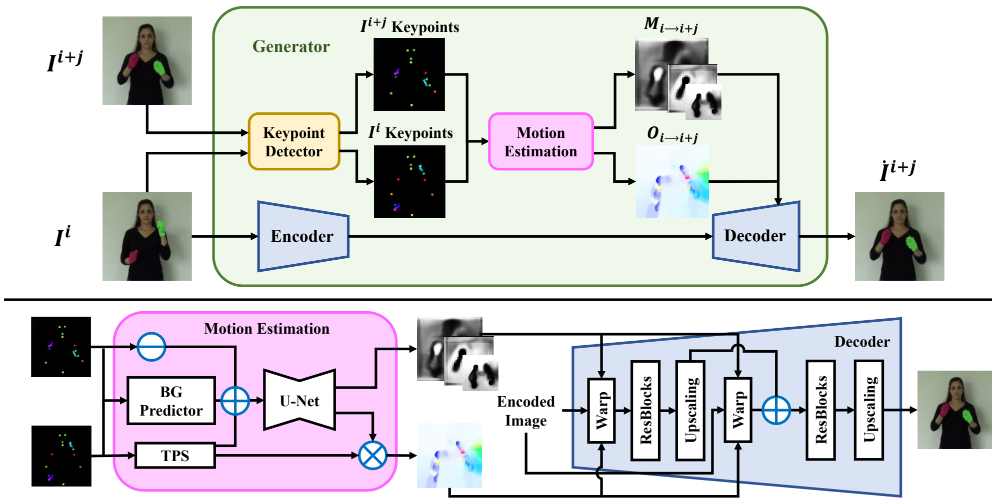

Fig. 3. **The Brief Framework.** The generator consists of an encoder, a decoder, a keypoint detector, and a motion estimation network. Specifically, the keypoint detector takes the source image I^i and the driving image I^{i+j} as input and passes the location of the keypoints to the motion estimation network to predict the optical flow O and the occlusion masks M. The encoder downsamples the source image to extract features whereas the decoder warps the encoded image according to the predicted optical flow O and occlusion masks M and generates the final output $\hat{I}^{i+j}$. The detailed structure of the motion estimation network and the decoder is depicted at the bottom. Note that $\ominus$ here indicates element-wise subtract, $\oplus$ indicates the concatenation operation, and $\otimes$ represents element-wise product.

3.2 Motion Estimation Network

The motion estimation network aims to predict a dense optical flow $O_{i\to i+j} \in \mathbb{R}^{H/4\times W/4\times 2}$ indicating the motion of the upper part of the body. $i \longrightarrow i+j$ means the model takes the i^{th} frame I^i as the source image and the $i+j^{th}$ frame I^{i+j} as the driving image.

Specifically, given 21 pairs of keypoint detected from the source image and driving image, we first use Thin Plate Spline (TPS) transformation [5] to estimate 21 corresponding optical flows. A learnable background predictor is adopted to predict an extra optical flow to approximate the motion of the background [66, 93]. We warp the downsampled source image according to each coarse optical flow mentioned above for later use. Every keypoint is modeled by a Gaussian in a heatmap, which means two sets of heatmaps can be obtained from the source image and the driving image. To emphasize the keypoint location difference between the source image and the driving image, the previously warped images are concatenated with the heatmap difference and then used as the input of the U-Net structure [57] to learn the residual motions. The output of the U-Net network is passed to a softmax layer and then multiplied with the coarse optical flows elementwisely. The final predicted optical flow $O_{i\to i+j}$ is obtained by summing the multiplied result along the channel axis. Meanwhile, the motion estimation network additionally predicts a set of occlusion masks M in different resolutions via applying a convolutional layer after the U-Net structure. The occlusion masks are applied in the decoder network to mask the unnecessary deformation within the feature map. Overall, the motion estimation process can be summarized as:

$$\begin{aligned} M_{i\to i+j} &= F_M(F_U(K^i, K^{i+j})), \\ O_{i\to i+j} &= F_O(F_U(K^i, K^{i+j})), \end{aligned} \tag{2}$$

where F_U denotes the U-Net structure. F_M is a convolutional layer used to predict the occlusion masks $M_{i\to i+j}$ in different resolutions, and F_O represents a softmax layer followed by multiplying the optical flows estimated by TPS, and a sum operation along the channel axis. Both F_M and F_O take the output of the U-Net model as the input. Note that in the motion estimation network, the driving images during inference only provide motion information and do not involve appearance textures. This enables animation between different signers, which means that even though during training the source image and driving image contain the same signer, we can still use videos of different signers to animate the source image.

3.3 Encoder and Decoder

As Figure 3 shows, the source image $I^i \in \mathbb{R}^{H\times W\times 3}$ is first passed to the encoder to extract features. We adopt a simple but effective "high to low" architecture for the encoder. The intuition is to combine the general information in the low-resolution feature maps and the detailed information in the high-resolution feature maps. The input image I^i is first passed to a convolutional layer to expand the feature channel. Followed by three downsampling blocks, the encoder aims to capture high-level features step by step. Every downsampling block consists of a 3×3 convolutional layer with a stride of 1, an instance normalization layer, a ReLU activation layer, and a 2×2 average pooling layer with a stride of 2. Let F_E denote the encoder model and F_D denote the decoder model. To get the generation result $\dot{I}^{i+j} \in \mathbb{R}^{H\times W\times 3}$, the deformation and decoding processes are carried on concurrently and progressively and can be formulated as:

$$\dot{I}^{i+j} = F_D(F_E(I^i), M_{i\to i+j}, O_{i\to i+j}). \tag{3}$$

As shown in the bottom of the Figure 3, the encoded feature map in the lowest resolution (*i.e.*, the output of the third downsampling block in the encoder) is first warped according to the optical flow $O_{i\to i+j}$ and then multiplied by the occlusion mask $M_{i\to i+j}$ in the corresponding resolution. Followed by a series of Resblocks, the model learns the residual information. In particular, a Resblock contains two 3×3 convolutional layers and a shortcut connection [24]. The input of each convolutional layer in the Resblock is normalized by an instance normalization layer and activated by a ReLU layer. We then warp the middle-resolution feature map (*i.e.*, the output of the second downsampling block in the encoder) according to the optical flow $O_{i\to i+j}$ and multiply the result by the occlusion mask. The warped middle-resolution feature map is concatenated with the upsampled output of the previous Resblocks and then passed to another series of Resblocks. Likewise, the feature map in the highest resolution (*i.e.*, the output of the first downsampling block in the encoder) is also warped by the optical flow, multiplied by the occlusion mask, concatenated with the output of the previous Resblocks, and passed to some other Resblocks. The decoding process ends up with a final convolutional layer which decreases the channel number to three. We use a sigmoid layer to restrict the output value and get the final result $\dot{I}^{i+j}$. Similarly, we can get the result of the other generation procedures depicted in the same training iteration, which can be formulated as:

$$\begin{aligned}
\dot{I}^{i+j+q} &= F_D(F_E(\dot{I}^{i+j}), M_{i+j\to i+j+q}, O_{i+j\to i+j+q}),\\
\ddot{I}^{i+j} &= F_D(F_E(\dot{I}^{i+j+q}), M_{i+j+q\to i+j}, O_{i+j+q\to i+j}),\\
\dot{I}^{i} &= F_D(F_E(\ddot{I}^{i}), M_{i+j\to i}, O_{i+j\to i}).
\end{aligned} \tag{4}$$

3.4 Optimization

During training, we apply a compound reconstruction loss as the optimization goal. Pyramid perceptual loss, middle feature loss, short-term cycle loss, and long-term cycle loss are the main components of reconstruction loss.

Perceptual Loss. Perceptual loss is proposed by Johnson *et al.* and is widely used in image transformation and reconstruction [33]. We minimize the L1 distance between two feature maps in five different middle layers extracted by a pre-trained VGG-19 network. The loss can be depicted as:

$$\mathcal{L}_p = \sum_n \left| V^n(I^{i+j}) - V^n(\dot{I}^{i+j}) \right|, \tag{5}$$

where I^{i+j} denotes the ground truth driving image and $\dot{I}^{i+j}$ means the generated image. V^n represents the n^{th} layer output of the pre-trained VGG-19 network [68]. In practice, we downsample the image pair and conduct a pyramid perceptual loss to facilitate the reconstruction supervision in different resolutions [64].

Warp Consistency Loss. We also constrain the warped encoded image to simulate the encoded driving image in the generation network [93]. To this end, the warp consistency loss is defined as:

$$\mathcal{L}_w = \sum_r \left| \mathcal{W}(F_E^r(I^i), O_{i\to i+j}) - F_E^r(I^{i+j}) \right|. \tag{6}$$

Note that I^i denotes the source image, and I^{i+j} represents the driving image. F_E^r means the r^{th} downsampling block in the encoder architecture. $\mathcal{W}$ is the warping operation according to the predicted optical flow $O_{i\to i+j}$.

Cycle-Consistency Losses. Temporal continuity is an essential influence factor for video generation since the real world is smooth and coherent. We propose two types of cycle-consistency losses: short-term cycle loss and long-term cycle loss to ensure temporal continuity. The short-term cycle loss is defined as:

$$\mathcal{L}_s = \left| \dot{I}^{i+j} - \dddot{I}^{i+j} \right|. \tag{7}$$

Let $\dot{I}^{i+j}$ denote the image generated by the first generation procedure, which means the model takes the i^{th} frame I^i as the source image and the $i+j^{th}$ frame I^{i+j} as the driving image. $\dddot{I}^{i+j}$ is the image generated by the third generation procedure. Although sharing the same driving image with the first generation procedure, the third procedure considers the output of the second generation procedure $\ddot{I}^{i+j+q}$ as the source image. Based on the temporal consistency hypothesis, the short-term cycle loss minimizes the $\mathcal{L}_1$ distance between $\dot{I}^{i+j}$ and $\dddot{I}^{i+j}$. In other words, the short-term cycle loss allows the model to generate the same results given different source images and the same driving image. We also adopt the long-term cycle loss, an augmented version of the original cycle loss. The long-term cycle loss is defined as:

$$\mathcal{L}_l = \left| \ddddot{I}^i - I^i \right|. \tag{8}$$

The long-term cycle loss performs a larger cycle compared to the previous short-term cycle loss as shown in Figure 2. We aim to consolidate the cycle consistency by minimizing the $\mathcal{L}_1$ distance between the output of the fourth generation procedure $\ddddot{I}^i$ and the i^{th} frame I^i. Overall, the short-term cycle loss and the long-term cycle loss provide temporal self-supervision, promoting the continuity and consistency of the generated videos.

The overall loss function is a combination of the above losses:

$$\mathcal{L}_{total} = \lambda_p \mathcal{L}_p + \lambda_w \mathcal{L}_w + \lambda_c(\mathcal{L}_s + \mathcal{L}_l). \tag{9}$$

The pyramid perceptual loss is considered the most essential reconstruction loss, thus we follow existing works [93] and set λ_p to 10. λ_w represents the weight of the warp consistency loss and is set to 1. The short-term cycle loss and long-term cycle loss share the same weight which is 2.

Training Strategy. As Figure 2 shows, existing motion transfer networks usually take an image pair as input and perform the generation procedure once to calculate the reconstruction loss. This training mode neglects the temporal information in a video clip, thus the generation results vary when given the same driving image but different source images picked from the same video. To address this problem, our framework conducts a cyclic end-to-end jointly training strategy. For every iteration, the model randomly chooses three frames from a video clip as input and performs

Fig. 4. **Qualitative comparison.** We compare our method with existing methods under the transfer setting on three datasets, *i.e.*, LSA64 [56], Phoenix-2014T [6], and WLASL-2000 [40]. Given the same source image driven by different driving images, we show the generated result. Note that the identity of the driving image is different from the identity of the source image. Our method keeps identity attributes of the source image while transferring fine-grained motion details from the driving image (highlighted in white dashed boxes).

the generation procedure four times. The short-term cycle loss is calculated between $\dot{I}^{i+j}$ and $\ddot{I}^{i+j}$, while the long-term cycle loss is calculated between $\dot{I}^{i}$ and I^{i}. We consider the temporal cycle-consistency as the prior hypothesis which helps the model learn the temporal information. The benefit of our training strategy is that the models can promise strong temporal robustness and generate videos with high continuity. Every rose has its thorn, the proposed strategy could be time-consuming in one iteration but converges quickly overall. Since the pre-trained image-based keypoint detector network does not maintain video continuity when facing blurred frames in a video clip, we jointly optimize the keypoint detector network and the generator network using the same optimization objective without extra human body structure annotations.

Inference Strategy. Similar to the training stage, the model takes two images as input for every generation procedure during testing. The model generates a new image, which resembles the target motion by deforming the source image. To generate the whole video clip, we further use the first image of a video clip as the source image and other frames as the driving images in sequence.

4 EXPERIMENTS

4.1 Dataset and Evaluation

LSA64 [56] is a small-scale dataset containing 64 words in Argentinian Sign Language (LSA). LSA64 consists of 3200 sign language videos performed by 10 different signers. We use videos in 8 word categories among 64 classes as the test set and the remaining vidoes as the training set.

Phoenix-2014T [6] is a German sign language dataset consists of 7738 videos performed by 9 different signers wearing dark clothes in front of a grey background. We follow the setting in the original dataset where 7096 videos are used for training and the rest 642 videos for testing.

WLASL-2000 [40] is a large-scale word-level American Sign Language (ASL) dataset including around 2000 words performed by more than 100 signers. There are 21083 videos in total and we use the official train-test split.

	LSA64			Phoenix-2014T			WLASL-2000		
	$\mathcal{L}_1 \downarrow$	SSIM $\uparrow$	LPIPS $\downarrow$	$\mathcal{L}_1 \downarrow$	SSIM $\uparrow$	LPIPS $\downarrow$	$\mathcal{L}_1 \downarrow$	SSIM $\uparrow$	LPIPS $\downarrow$
Monkey-Net [64]	0.0121	0.9489	0.0217	0.0340	0.8314	0.0784	0.0242	0.8786	0.0623
FOMM [65]	0.0186	0.9151	0.0274	0.0253	0.8681	0.0461	0.0260	0.8726	0.0587
AA [66]	0.0110	0.9493	0.0187	0.0190	0.9077	0.0365	0.0178	0.9118	0.0415
TPSMM [93]	0.0109	0.9499	0.0203	0.0188	0.9149	0.0327	0.0158	0.9218	0.0374
Ours	**0.0104**	**0.9533**	**0.0170**	**0.0172**	**0.9211**	**0.0302**	**0.0153**	**0.9273**	**0.0313**

Table 1. Results for the four competitive methods and our method under the reconstruction setting on the three datasets. Three image quality evaluation metrics are adopted to test the reconstruction ability of models.

Evaluation Metrics. (1) Manhattan Distance ($\mathcal{L}_1$) [76] is the mean $\mathcal{L}_1$ distance between every pixel of the generated frame and the ground-truth frame. Lower $\mathcal{L}_1$ distance indicates higher reconstruction quality. **(2)** Structural SIMilarity (SSIM) [81] compares the resemblance between two images concerning luminance, contrast, and structure. Higher value means higher generation quality. **(3)** Learned Perceptual Image Patch Similarity (LPIPS) [92] proposed by Zhang *et al.* is a metric used to compare the perceptual similarity between two images. We adopt the default Alexnet [39] version here. Lower value indicates better reconstruction quality.

4.2 Implementation Details

We deploy a single Tesla V100 GPU to train models on every dataset. According to the dataset size, we train 100 epochs, 200 epochs, and 300 epochs for LSA64, Phoenix-2014T, and WLASL-2000 respectively. The resolution of frames is resized to 128×128. Following existing works [66, 93], the generator network and the motion estimation network are trained by the Adam Optimizer [35] with $\beta_1 = 0.5, \beta_2 = 0.99$. We set the initial learning rate to 0.0002 except for the keypoint detector network and apply a multistep scheduler to decay the learning rate. The learning rate decay factor γ is set to 0.1. To balance the memory cost and the training speed, we adopt a batch size of 16. In terms of the keypoint detector network, we set the initial learning rate to 0.00002 while the decay happens along with the generator network. The code is based on Pytorch[51]. We will make our code open-source for reproducing all experiments.

4.3 Quantitative Results

We compare our method with four previous representative works for motion transfer, including Monkey-Net [64], FOMM [65], AA [66], and TPSMM [93]. We re-trained all these works following the best setting reported in their papers for comparison. As shown in Table 1, the proposed method surpasses other methods and reaches the state-of-the-art results of all three metrics on three datasets. The results verify that our method can generate video in high fidelity by recovering more human body structure details. The reason is that our keypoint detector learns 21 explainable keypoints, making the motion estimation model focus on the motion of essential body parts accurately. Our method predicts 12 keypoints located on the hands, which explicitly makes the motion estimation network pay attention to fine-grained finger motions.

4.4 Qualitative Results

Different from the reconstruction setting in the quantitative comparison, we test the transferability of our method via qualitative analysis. As Figure 4 shows, we select a source image and multiple driving videos with different identities to compare the animation quality (details highlighted in the white dashed boxes). For all three datasets, our method maintains high identity consistency and correct motion, especially in facial expression and hand details. The pre-trained keypoint detector model explicitly raises the attention of the motion estimation network toward the hand motions

$\mathcal{L}_1$	$\mathcal{L}_s$	$\mathcal{L}_1 \downarrow$	SSIM $\uparrow$	LPIPS $\downarrow$
		0.0106	0.9516	0.0179
✓		0.0105	0.9519	0.0177
	✓	0.0105	0.9526	0.0175
✓	✓	**0.0104**	**0.9533**	**0.0170**

Table 2. Ablation study on the proposed short-term cycle loss and the long-term cycle loss. We train the models while removing the losses in turn on the LSA64 dataset and then test the reconstruction ability using three metrics.

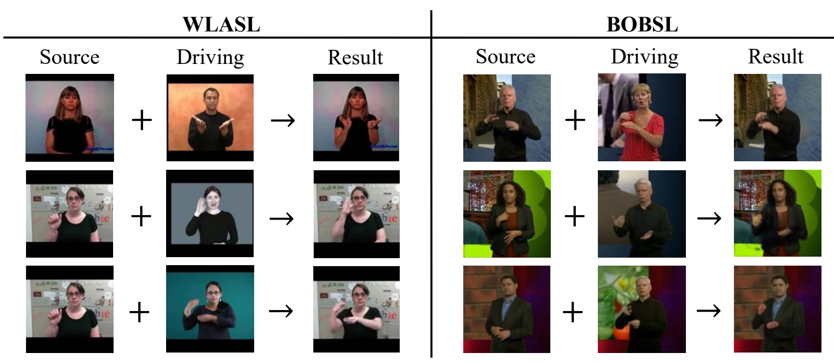

Fig. 5. Qualitative results of various complex backgrounds on WLASL and BOBSL datasets.

and the facial expressions, providing more details in the generation results. Monkey-Net [64] and FOMM [65] show poor motion transfer capability on every dataset. AA [66] and TPSMM [93] have a robust capability to capture the motion and preserve the correct body structure. However, the identity attributes of the generated image, such as hair and face, are relatively blurred with the identity of the driving image, especially on the Phoenix-2014T dataset.

Apart from the single-frame fidelity, we also provide visual results for the video continuity comparison in Figure 1. The motion in the reconstructed video is smoother than in the videos generated by other methods. The end-to-end training strategy empowers the generator network with strong temporal consistency.

4.5 Ablation Studies

Do the proposed cycle-consistency losses work? Table 2 shows the results of adopting the short-term cycle loss and long-term cycle loss on the LSA64 dataset. In particular, the $\mathcal{L}_1$, SSIM, and LPIPS of the vanilla model trained without the two losses arrive at 0.0106, 0.9516, and 0.0179 respectively. We could observe two points: (1) Compared with the vanilla model, the short-term cycle loss and the long-term cycle loss can individually boost the reconstruction quality of the trained model. The short-term cycle loss has a larger regularization impact on the reconstruction. (2) The short-term and long-term consistency losses are complementary. The model achieves the best performance (-0.0002 for L1, +0.0017 for SSIM, and -0.0009 for LPIPS) when both the two proposed losses are deployed. We think both losses ensure video continuity and robustness by refraining from the unrecoverable movements.

Is the model sensitive to the cycle-consistency losses? To test whether the model is sensitive to the weight of the cycle-consistency losses, we apply another experiment to explore the proper

λ_c	$\mathcal{L}_1 \downarrow$	SSIM $\uparrow$	LPIPS $\downarrow$
0.5	0.0106	0.9513	0.0179
1	0.0105	0.9528	0.0176
2	**0.0104**	**0.9533**	**0.0170**
5	0.0106	0.9518	0.0175
10	0.0105	0.9518	0.0177

Table 3. Ablation study on the weight of the proposed losses. We train the same model using different λ_c values and test the reconstruction ability on the LSA64 dataset.

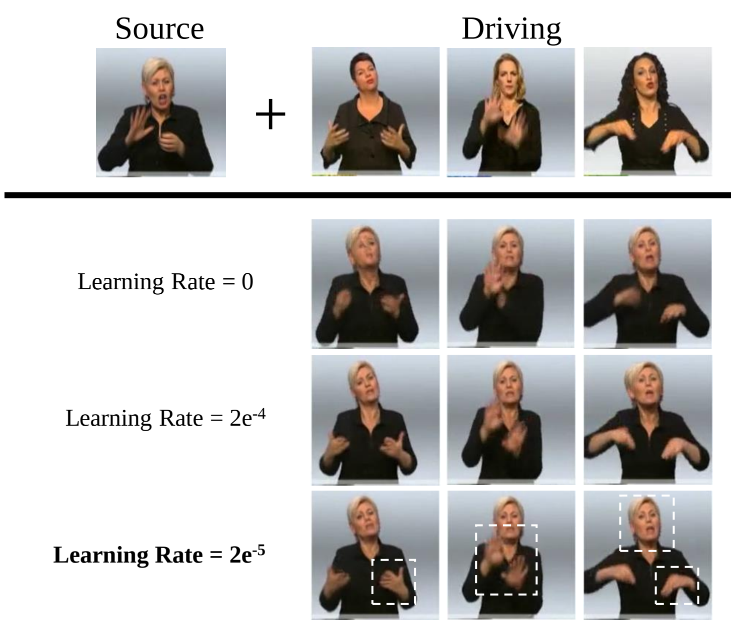

Fig. 6. Ablation study on different learning rates of the keypoint detector network. We provide the visual result on the Phoenix-2014T dataset using different learning rates of the keypoint detector network.

shared weight λ_c. As shown in Table 3, we attempt five different values of the weight including 0.5, 1, 2, 5, and 10. Under the same reconstruction setting carried out in the quantitative section, 2 is the optimal value for λ_c. The model is not sensitive to the value of λ_c, yet a too-large or too-small value of λ_c could harm the performance. For instance, compared with setting λ_c to 2, the performance decreases when setting λ_c to 0.5 (+0.0002 $\mathcal{L}_1$, -0.0020 SSIM and +0.0009 LPIPS) or 10 (+0.0001 $\mathcal{L}_1$, -0.0015 SSIM and +0.0007 LPIPS). We consider the reason is that lower weight for the cycle losses does not offer enough penalty on the cycle consistency. Meanwhile, a way larger weight could force the model to overfit the cycle losses and weaken the supervision provided by the reconstruction loss. Hence, we select 2 as the value of λ_c to balance the influence of the cycle losses and the reconstruction loss.

Shall we fine-tune the keypoint detection network? To explore whether the fine-tuning procedure of the keypoint detector network is essential, we conduct an ablation experiment on the Phoenix-2014T dataset. In particular, we set the learning rate of the keypoint detector network to 0, 0.0002, and 0.00002 and test the motion transfer ability of the models. The visual results are

Backbone	$\mathcal{L}_1 \downarrow$	SSIM $\uparrow$	LPIPS $\downarrow$
Ours	0.0104	**0.9533**	**0.0170**
HRNet	**0.0098**	0.9503	0.0176

Table 4. Ablation study on different encoder architecture.

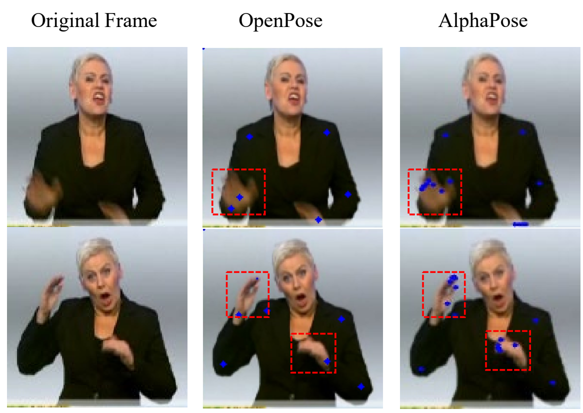

Fig. 7. Comparison between OpenPose and AlphaPose.

shown in Figure 6 and 0.00002 turns out to be the proper learning rate for the keypoint detector network. When we fix the parameters of the keypoint detector network, the transfer results are not ideal. The pre-trained keypoint detector model does not leverage the temporal information in video clips, leading to instability when facing blurred frames. The results demonstrate the importance of the fine-tuning process. Additionally, a large learning rate brings distortions in the face since it forces the pre-trained model to forget the prior human body structure information, making the generator model capture the wrong identity and background texture information. Therefore, to keep fine-grained finger details while avoiding losing body structure information, we set the learning rate of the keypoint detector network to 0.00002.

Is STCNet robust to complex backgrounds? To explore whether STCNet is robust to different backgrounds, we provide additional qualitative results on the WLASL dataset [40] and BOBSL dataset [2] as shown in Figure 5. BOBSL is a large-scale dataset of British Sign Language (BSL) containing about a total of 1400 hours videos of BBC broadcast footage in different backgrounds. Results show that the proposed STCNet is capable of videos with complex backgrounds like classrooms and studios. Although STCNet does not explicitly restrict keeping the background, we still observe that the learned decoder model is able to preserve the video background and correct human motion. It is worth mentioning that we directly use the model trained on the WLASL dataset to generate results on the BOBSL dataset. The impressive result indicates that STCNet has the ability to generalize to different datasets.

Can we use a different network architecture? We utilize the convolutional encoder and decoder architecture, following baseline methods for a fair comparison. We also try a different visual encoder backbone *i.e.*, HRNet [72], on the LSA64 dataset. The result is shown in Table 4. We find that our light-weight encoder model is competitive with the HRNet.

Why choosing Alphapose as the keypoint detection network? We test OpenPose[7, 8, 67, 82] and AlphaPose [19, 43]. As shown in Figure 7, AlphaPose detects accurate finger keypoints, yet OpenPose somehow fails to detect finger keypoints. Therefore, we select AlphaPose as default instead of OpenPose.

5 CONCLUSION

In this paper, we propose a sign language motion transfer framework called Structure-aware Temporal Consistency Network (STCNet). Different from existing works, STCNet leverages prior human body structure knowledge and temporal consistency for sign language video generation. We also introduce a pair of cycle-consistency losses to fully exploit temporal information within sign language videos and further improve the temporal continuity of the generated videos. Extensive experiments verify that our method could generate competitive videos with accurate motion and high-fidelity video continuity compared with existing works. In the future, we will continue exploring the potential of applying this method to other relevant research fields [26, 45, 46, 84], such as data augmentation for sign language recognition [1, 6, 28], clothing / makeup try-on according to keypoints [23, 27, 31] and 3D person re-identification [95].

Broad Impact. This research has the potential to improve social communication and inclusion for people who rely on sign language as a means of communication.

REFERENCES

[1] Samuel Albanie, Gül Varol, Liliane Momeni, Triantafyllos Afouras, Joon Son Chung, Neil Fox, and Andrew Zisserman. 2020. BSL-1K: Scaling up co-articulated sign language recognition using mouthing cues. In *ECCV*. 35–53.

[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] Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, and Bernt Schiele. 2018. Posetrack: A benchmark for human pose estimation and tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5167–5176.

[4] Bruno Artacho and Andreas Savakis. 2020. Unipose: Unified human pose estimation in single images and videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 7035–7044.

[5] Fred L. Bookstein. 1989. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence* 11, 6 (1989), 567–585.

[6] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. 2018. Neural sign language translation. In *CVPR*.

[7] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. 2019. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

[8] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*.

[9] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. 2019. Everybody dance now. In *ICCV*.

[10] James Charles, Tomas Pfister, Derek Magee, David Hogg, and Andrew Zisserman. 2016. Personalizing human video pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3063–3072.

[11] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. 2018. Cascaded pyramid network for multi-person pose estimation. In *CVPR*. 7103–7112.

[12] Yutong Chen, Fangyun Wei, Xiao Sun, Zhirong Wu, and Stephen Lin. 2022. A simple multi-modality transfer learning baseline for sign language translation. In *CVPR*.

[13] Anoop Cherian, Julien Mairal, Karteek Alahari, and Cordelia Schmid. 2014. Mixing body-part sequences for human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2353–2360.

[14] Qi Dang, Jianqin Yin, Bin Wang, and Wenqing Zheng. 2019. Deep learning based 2d human pose estimation: A survey. *Tsinghua Science and Technology* 24, 6 (2019), 663–676.

[15] Matthias Dantone, Juergen Gall, Christian Leistner, and Luc Van Gool. 2013. Human pose estimation using body parts dependent joint regressors. In *CVPR*.

[16] Amanda Duarte, Shruti Palaskar, Lucas Ventura, Deepti Ghadiyaram, Kenneth DeHaan, Florian Metze, Jordi Torres, and Xavier Giro-i Nieto. 2021. How2sign: a large-scale multimodal dataset for continuous american sign language. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2735–2744.

[17] Alexei A Efros, Alexander C Berg, Greg Mori, and Jitendra Malik. 2003. Recognizing action at a distance. In *ICCV*.

[18] Marcin Eichner, Manuel Marin-Jimenez, Andrew Zisserman, and Vittorio Ferrari. 2012. 2d articulated human pose estimation and retrieval in (almost) unconstrained still images. *International journal of computer vision* 99, 2 (2012), 190–214.

[19] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. 2017. Rmpe: Regional multi-person pose estimation. In *ICCV*. 2334–2343.
[20] Sicheng Gao, Xuhui Liu, Bohan Zeng, Sheng Xu, Yanjing Li, Xiaoyan Luo, Jianzhuang Liu, Xiantong Zhen, and Baochang Zhang. 2023. Implicit diffusion models for continuous super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10021–10030.
[21] Ivan Gruber, Zdenek Krnoul, Marek Hrúz, Jakub Kanis, and Matyas Bohacek. 2021. Mutual support of data modalities in the task of sign language recognition. In *CVPR Workshop*.
[22] Dan Guo, Shuo Wang, Qi Tian, and Meng Wang. 2019. Dense Temporal Convolution Network for Sign Language Translation.. In *IJCAI*. 744–750.
[23] Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, and Larry S. Davis. 2018. VITON: An Image-Based Virtual Try-On Network. In *CVPR*.
[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.
[25] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.
[26] Trang-Thi Ho, John Jethro Virtusio, Yung-Yao Chen, Chih-Ming Hsu, and Kai-Lung Hua. 2020. Sketch-guided deep portrait generation. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 16, 3 (2020), 1–18.
[27] Bingwen Hu, Ping Liu, Zhedong Zheng, and Mingwu Ren. 2022. SPG-VTON: Semantic Prediction Guidance for Multi-pose Virtual Try-on. *IEEE Transactions on Multimedia* (2022).
[28] Hezhen Hu, Weichao Zhao, Wengang Zhou, Yuechen Wang, and Houqiang Li. 2021. Signbert: pre-training of hand-model-aware representation for sign language recognition. In *ICCV*.
[29] Hezhen Hu, Wengang Zhou, Junfu Pu, and Houqiang Li. 2021. Global-local enhancement network for NMF-aware sign language recognition. *ACM transactions on multimedia computing, communications, and applications (TOMM)* 17, 3 (2021), 1–19.
[30] Shaoli Huang, Mingming Gong, and Dacheng Tao. 2017. A coarse-fine network for keypoint localization. In *ICCV*. 3028–3037.
[31] Zhikun Huang, Zhedong Zheng, Chenggang Yan, Hongtao Xie, Yaoqi Sun, Jianzhong Wang, and Jiyong Zhang. 2021. Real-world automatic makeup via identity preservation makeup net. In *IJCAI*.
[32] Songyao Jiang, Bin Sun, Lichen Wang, Yue Bai, Kunpeng Li, and Yun Fu. 2021. Sign Language Recognition via Skeleton-Aware Multi-Model Ensemble. *arXiv:2110.06161* (2021).
[33] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*. Springer, 694–711.
[34] Johanna Karras, Aleksander Holynski, Ting-Chun Wang, and Ira Kemelmacher-Shlizerman. 2023. Dreampose: Fashion image-to-video synthesis via stable diffusion. *arXiv preprint arXiv:2304.06025* (2023).
[35] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6980
[36] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. 2019. Pifpaf: Composite fields for human pose estimation. In *CVPR*. 11977–11986.
[37] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. 2021. Openpifpaf: Composite fields for semantic keypoint detection and spatio-temporal association. *IEEE Transactions on Intelligent Transportation Systems* (2021).
[38] Shyam Krishna et al. 2021. SignPose: Sign Language Animation Through 3D Pose Lifting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2640–2649.
[39] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *NeurIPS* 25 (2012).
[40] 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 *WACV*.
[41] Dongxu Li, Chenchen Xu, Xin Yu, Kaihao Zhang, Benjamin Swift, Hanna Suominen, and Hongdong Li. 2020. Tspnet: Hierarchical feature learning via temporal semantic pyramid for sign language translation. *NeurIPS* (2020).
[42] Jiefeng Li, Can Wang, Hao Zhu, Yihuan Mao, Hao-Shu Fang, and Cewu Lu. 2019. Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. In *CVPR*. 10863–10872.
[43] Yong-Lu Li, Liang Xu, Xinpeng Liu, Xijie Huang, Yue Xu, Shiyi Wang, Hao-Shu Fang, Ze Ma, Mingyang Chen, and Cewu Lu. 2020. Pastanet: Toward human activity knowledge engine. In *CVPR*. 382–391.
[44] Peirong Liu, Rui Wang, Xuefei Cao, Yipin Zhou, Ashish Shah, Maxime Oquab, Camille Couprie, and Ser-Nam Lim. 2021. Self-appearance-aided Differential Evolution for Motion Transfer. *arXiv:2110.04658* (2021).
[45] Shiguang Liu and Huixin Wang. 2023. Talking Face Generation via Facial Anatomy. *ACM Transactions on Multimedia Computing, Communications and Applications* 19, 3 (2023), 1–19.

[46] Zhiming Liu, Kai Niu, and Zhiqiang He. 2023. ML-CookGAN: Multi-Label Generative Adversarial Network for Food Image Generation. *ACM Transactions on Multimedia Computing, Communications and Applications* 19, 2s (2023), 1–21.
[47] Joseph J Murray, Maartje De Meulder, and Delphine Le Maire. 2018. An Education in Sign Language as a Human Right: The Sensory Exception in the Legislative History and Ongoing Interpretation of Article 24 of the UN Convention on the Rights of Persons with Disabilities. *Hum. Rts. Q.* 40 (2018), 37.
[48] Alejandro Newell, Zhiao Huang, and Jia Deng. 2017. Associative embedding: End-to-end learning for joint detection and grouping. *NeurIPS* 30 (2017).
[49] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. 2021. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741* (2021).
[50] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. 2018. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *ECCV*.
[51] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS* 32 (2019).
[52] Tomas Pfister, James Charles, and Andrew Zisserman. 2015. Flowing convnets for human pose estimation in videos. In *Proceedings of the IEEE international conference on computer vision.* 1913–1921.
[53] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. 2016. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *CVPR*.
[54] Junfu Pu, Wengang Zhou, and Houqiang Li. 2019. Iterative alignment network for continuous sign language recognition. In *CVPR*.
[55] Zhongwei Qiu, Qiansheng Yang, Jian Wang, Xiyu Wang, Chang Xu, Dongmei Fu, Kun Yao, Junyu Han, Errui Ding, and Jingdong Wang. 2023. Learning Structure-Guided Diffusion Model for 2D Human Pose Estimation. *arXiv preprint arXiv:2306.17074* (2023).
[56] Franco Ronchetti, Facundo Quiroga, César Armando Estrebou, Laura Cristina Lanzarini, and Alejandro Rosete. 2016. LSA64: an Argentinian sign language dataset. In *XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016)*.
[57] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 234–241.
[58] Ben Saunders, Necati Cihan Camgöz, and Richard Bowden. [n. d.]. Adversarial Training for Multi-Channel Sign Language Production. In *The 31st British Machine Vision Virtual Conference*. British Machine Vision Association.
[59] Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. 2020. Everybody sign now: Translating spoken language to photo realistic sign language video. *arXiv:2011.09846* (2020).
[60] Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. 2020. Progressive transformers for end-to-end sign language production. In *ECCV*. Springer, 687–705.
[61] Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. 2021. Anonysign: Novel human appearance synthesis for sign language video anonymisation. In *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*. IEEE, 1–8.
[62] Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. 2021. Mixed signals: Sign language production via a mixture of motion primitives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1919–1929.
[63] 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.
[64] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. 2019. Animating arbitrary objects via deep motion transfer. In *CVPR*.
[65] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. 2019. First order motion model for image animation. *NeurIPS* (2019).
[66] Aliaksandr Siarohin, Oliver J Woodford, Jian Ren, Menglei Chai, and Sergey Tulyakov. 2021. Motion Representations for Articulated Animation. In *CVPR*.
[67] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. 2017. Hand Keypoint Detection in Single Images using Multiview Bootstrapping. In *CVPR*.
[68] K. Simonyan and A. Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*.
[69] Ozge Mercanoglu Sincan, Julio Junior, CS Jacques, Sergio Escalera, and Hacer Yalim Keles. 2021. Chalearn LAP large scale signer independent isolated sign language recognition challenge: Design, results and future research. In *CVPR Workshop*.

[70] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502* (2020).
[71] Jie Song, Limin Wang, Luc Van Gool, and Otmar Hilliges. 2017. Thin-slicing network: A deep structured model for pose estimation in videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4220–4229.
[72] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. 2019. Deep high-resolution representation learning for human pose estimation. In *CVPR*.
[73] Federico Tavella, Aphrodite Galata, and Angelo Cangelosi. 2022. Phonology Recognition in American Sign Language. In *IJCAI*.
[74] Alexander Toshev and Christian Szegedy. 2014. Deeppose: Human pose estimation via deep neural networks. In *CVPR*.
[75] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. 2018. Mocogan: Decomposing motion and content for video generation. In *CVPR*.
[76] AKMSSA Vadivel, AK Majumdar, and Shamik Sural. 2003. Performance comparison of distance metrics in content-based image retrieval applications. In *CIT*. 159–164.
[77] Lucas Ventura, Amanda Duarte, and Xavier Giró-i Nieto. 2020. Can everybody sign now? Exploring sign language video generation from 2D poses. *arXiv:2012.10941* (2020).
[78] Tan Wang, Linjie Li, Kevin Lin, Chung-Ching Lin, Zhengyuan Yang, Hanwang Zhang, Zicheng Liu, and Lijuan Wang. 2023. Disco: Disentangled control for referring human dance generation in real world. *arXiv preprint arXiv:2307.00040* (2023).
[79] Xiaolong Wang, Allan Jabri, and Alexei A Efros. 2019. Learning correspondence from the cycle-consistency of time. In *CVPR*.
[80] Yaohui Wang, Piotr Bilinski, Francois Bremond, and Antitza Dantcheva. 2020. G3AN: disentangling appearance and motion for video generation. In *CVPR*.
[81] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.
[82] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. 2016. Convolutional pose machines. In *CVPR*.
[83] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. 2013. DeepFlow: Large displacement optical flow with deep matching. In *Proceedings of the IEEE international conference on computer vision*. 1385–1392.
[84] Xintian Wu, Huanyu Wang, Yiming Wu, and Xi Li. 2023. D3T-GAN: Data-Dependent Domain Transfer GANs for Image Generation with Limited Data. *ACM Transactions on Multimedia Computing, Communications and Applications* 19, 4 (2023), 1–20.
[85] Bruce Xiaohan Nie, Caiming Xiong, and Song-Chun Zhu. 2015. Joint action recognition and pose estimation from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1293–1301.
[86] Chengming Xu, Yanwei Fu, Chao Wen, Ye Pan, Yu-Gang Jiang, and Xiangyang Xue. 2020. Pose-Guided Person Image Synthesis in the Non-Iconic Views. *IEEE Transactions on Image Processing* 29 (2020), 9060–9072. https://doi.org/10.1109/TIP.2020.3023853
[87] Ceyuan Yang, Zhe Wang, Xinge Zhu, Chen Huang, Jianping Shi, and Dahua Lin. 2018. Pose guided human video generation. In *ECCV*.
[88] Shuyu Yang, Yinan Zhou, Zhedong Zheng, Yaxiong Wang, Li Zhu, and Yujiao Wu. 2023. Towards unified text-based person retrieval: A large-scale multi-attribute and language search benchmark. In *Proceedings of the 31st ACM International Conference on Multimedia*. 4492–4501.
[89] Jae Shin Yoon, Lingjie Liu, Vladislav Golyanik, Kripasindhu Sarkar, Hyun Soo Park, and Christian Theobalt. 2021. Pose-Guided Human Animation from a Single Image in the Wild. In *CVPR*.
[90] Duncan Zauss, Sven Kreiss, and Alexandre Alahi. 2021. Keypoint Communities. In *ICCV*. 11057–11066.
[91] Jan Zelinka and Jakub Kanis. 2020. Neural sign language synthesis: Words are our glosses. In *WACV*.
[92] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*.
[93] Jian Zhao and Hui Zhang. 2022. Thin-Plate Spline Motion Model for Image Animation. In *CVPR*.
[94] Ce Zheng, Wenhan Wu, Chen Chen, Taojiannan Yang, Sijie Zhu, Ju Shen, Nasser Kehtarnavaz, and Mubarak Shah. 2023. Deep learning-based human pose estimation: A survey. *Comput. Surveys* 56, 1 (2023), 1–37.
[95] Zhedong Zheng, Xiaohan Wang, Nenggan Zheng, and Yi Yang. 2022. Parameter-Efficient Person Re-identification in the 3D Space. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)* (2022).
[96] Hao Zhou, Wengang Zhou, Yun Zhou, and Houqiang Li. 2020. Spatial-temporal multi-cue network for continuous sign language recognition. In *AAAI*.
[97] Yipin Zhou, Zhaowen Wang, Chen Fang, Trung Bui, and Tamara Berg. 2019. Dance dance generation: Motion transfer for internet videos. In *ICCV Workshops*.

Received June 2023; revised xx xxxx; accepted xx xxxx