Sign Language Production with Latent Motion Transformer

Pan Xie  Taiying Peng  Yao Du  Qipeng Zhang
Beihang University, Beijing, China
{panxie, taiyi, duyaoo, zhangqipeng}@buaa.edu.cn

Abstract

Sign Language Production (SLP) is the tough task of turning sign language into sign videos. The main goal of SLP is to create these videos using a sign gloss. In this research, we’ve developed a new method to make high-quality sign videos without using human poses as a middle step. Our model works in two main parts: first, it learns from a generator and the video’s hidden features, and next, it uses another model to understand the order of these hidden features. To make this method even better for sign videos, we make several significant improvements. (i) In the first stage, we take an improved 3D VQ-GAN to learn downscaled latent representations. (ii) In the second stage, we introduce sequence-to-sequence attention to better leverage conditional information. (iii) The separated two-stage training discards the realistic visual semantic of the latent codes in the second stage. To endow the latent sequences semantic information, we extend the token-level autoregressive latent codes learning with perceptual loss and reconstruction loss for the prior model with visual perception. Compared with previous state-of-the-art approaches, our model performs consistently better on two word-level sign language datasets, i.e., WLASL and NMFs-CSL.

1. Introduction

Sign language is the primary communication manner of the deaf community. As a visual language, it contains various hand gestures, movements, facial expressions, and complex grammatical structures. Therefore, it is difficult to communicate between sign and non-sign language speakers. To reduce the barrier, many studies [7, 8, 12, 14, 22, 25, 30, 34, 52, 58] have been dedicated to translating sign language into text and spoken language (SLR and SLT). More recent research [17, 38–41, 56] looks at doing the opposite: creating sign language videos from text (SLP). While most older SLP models focused on generating sign poses, Duarte et al. [17] and Stollet et al. [41] created sign videos in two steps: translating words to sign poses and then making videos from those signs. However, tools like OpenPose [9], used in these processes, sometimes struggle with fast hand movements.

In this work, we explore a novel SLP method to bypass the human pose estimator and directly generate sign videos. In the area of video generation, one strand of work towards the variations of GANs [13, 27, 31, 32, 48]. Other strands of work propose variational autoencoders (VAEs) [2, 15, 54], vector quantized VAE (VQ-VAE) [37, 46], autoregressive models [51], and flows [26]. Moreover, a recent strand of work is latent autoregressive method, such as LVT [36], VideoGPT [49], and PV-VQVAE [55]. They proposed to generate the video within two stages: first learn an image or a video generator using VQ-VAE, and then learn the prior of the latent codes with a causal Transformer (e.g., GPT [35]) or convolution networks (e.g., PixelCNN [45]). All these generative methods have their trade-offs in various aspects, e.g., sampling speed, sample quality, optimized stability, calculation requirements, and ease of evaluation.

Our SLP model is also built upon the latent autoregressive method. We make this choice for several reasons: (i) Compared with the adversarial models, the two stages of the latent autoregressive model are optimized by the reconstruction and likelihood, which are more stable and easier to evaluate. This advantage allows us to focus on complex video modeling. (ii) Compared with the autoregressive models in the pixel space, performing autoregressive modeling in the downscaled latent space is much more efficient in training and sampling. To this end, we propose a novel Latent Motion Transformer (LMT) for the word-level sign language production (see Figure 1). To better understand and produce high-quality sign videos, we make several modifications to existing methods.

In the first stage training of reconstruction: (i) We leverage the VQ-GAN [18], a brilliant merger of VQ-VAE and GANs. Modifying its application, we transition it into a 3D-dimension for video reconstructions. This change involves the employment of 3D convolution networks to facilitate the transformation of pixel-space videos into a compact latent space. Such a downsampling technique is not arbitrary; it serves the essential purpose of optimizing computational load, especially for the subsequent latent codes prior learn-
Figure 1. The overview of our Latent Motion Transformer (LMT) with two-stage training strategy. In the first stage, the video VQ-GAN model encodes the sign videos into discrete codes, and learn the latent space with reconstruction loss, perceptual loss, and a discriminator. In the second stage training on the flattened codes, the conditional autoregressive model predicts the future codes with the cross-entropy loss, perceptual loss and reconstruction loss. Note that the decoder parameters is fixed in the second stage.

In the second stage of prior learning: (iii) We substitute the casual self-attention, embracing instead the sentence-to-sentence attention, which in our assessment, holds promise in effectively harnessing conditional frames and labels.

Furthermore, delving into historical approaches, (iv) there’s a notable preference for the GPT-like model in prior works [18,36,49,55]. This method is inclined to unearth the latent discrete codes using a token-level cross-entropy loss function. However, such a bifurcated methodology appears to fall short in capturing the intricate visual semantics, risking subpar generation outcomes. Contrarily, our strategy integrates both the patch-level perceptual loss [57] and the reconstruction loss. This dual-pronged approach seeks to endow the prior model with richer visual semantics. Executing this involves culling discrete codes from the Transformer model’s predicted distribution. Despite the potential challenges (given its non-differentiable nature), we mitigate this through the Gumbel-softmax strategy [23]. Subsequent to this, the decoded predicted codes reveal the reconstructed videos, paving the way for a side-by-side comparison with their original versions, enabling computation of the perceptual and reconstruction losses.

Overall, our contributions are highlighted below:

1. We focus on generating realistic human-centered sign videos without using human pose sequences as the intermediate step. Furthermore, our innovative approach utilizes a 3D VQ-GAN with a motion transformer for latent sign video understanding, setting it apart from traditional axial attention and vanilla spatial-temporal atten-
tion methods.

2. Our method introduces sentence-to-sentence attention for prior learning on the flattened latent codes. This allows us to better harness conditional information and accurately predict coherent future frames. Additionally, our training method with patch-level perceptual loss and reconstruction loss enriches the prior model with visual perception, leading to a more potent model.

3. We conduct experiments on two public sign language datasets: WLASL [28] and NMFs-CSL [21]. Compared with previous state-of-the-art video generation methods, our proposed LMT model achieves significant improvements with −24.47 and −6.70 FVD scores on two datasets, respectively.

2. Related Work

**VQ-VAE and VQ-GAN.** VQ-VAE [46] is a specific autoencoder model which consists of three modules: an encoder, a codebook, and a decoder. Given an image $x \in \mathbb{R}^{H \times W \times C}$, the encoder $E(x)$ first compresses the high dimensional data into low-dimension features $z_e \in \mathbb{R}^{h \times w \times n_z}$. Different from reconstructing on the compressed features directly, it performs a nearest neighbors method to quantize the compressed feature $z_e$ to the discrete representations $\hat{z}_e \in \mathbb{R}^{h \times w \times n_z}$, where the discrete vectors are maintained by the codebook $Z = \{e_i \in \mathbb{R}^{n_z}\}_{i=1}^{K}$. This quantization process can be formulated as:

$$z_{q,i,j} = e_q(x_{i,j}), \quad q(x_{i,j}) = \arg \min_{k \in K} \| e_{q,i,j} - e_k \|,$$  \hspace{1cm} (1)

where $q(\cdot)$ is the quantization method for calculating the corresponding index of the codebook embedding $Z$.

After quantization, the decoder $D$ reconstructs the data from the discrete representations $z_{q}$. The whole model is optimized by minimizing the reconstruction loss and the gap between the $z_{e}$ and $z_{q}$ as the following object:

$$L_{vqvae}(x) = \left( \frac{||x - D(z_q)||^2_L}{\lambda_{\text{recon}}} + \frac{||sg[z_e] - z_q||^2_L}{\lambda_{\text{codebook}}} + \frac{||sg[z_q] - z_e||}{\lambda_{\text{commit}}}, \right)$$  \hspace{1cm} (2)

where $sg$ is the stop-gradient operator. $\lambda_{\text{recon}}$ is a reconstruction loss to encourage the model to learn meaningful representations for reconstructing the data. $\lambda_{\text{codebook}}$ is to bring the codebook vectors closer to the encoder output $z_e$. $\lambda_{\text{commit}}$ is a commit loss weighted by $\beta$ to prevent the encoder output from fluctuating between different code vectors.

VQ-GAN [18] is an improved model based on the VQ-VAE. To learn a perceptually rich codebook, it combines the reconstruction loss and a perceptual loss $L_{\text{lpips}}$ [57]. Moreover, it introduces a patch-based discriminator $D$ to discriminate the real and reconstructed images. The adversarial training procedure is optimized by the following loss:

$$L_{\text{gan}}(x) = [\log D(x) + \log(1 - D(G(z_q)))].$$  \hspace{1cm} (3)

Therefore, the complete objective for the VQ-GAN model can be formulated as:

$$\arg \min_{E,G,C} \max_D \mathbb{E}_{x \in p(x)}[(L_{\text{vqvae}} + L_{\text{lpips}}) + \lambda L_{\text{gan}}].$$  \hspace{1cm} (4)

where $\lambda = \frac{\nabla G_L[L_{\text{recon}}]}{\nabla G_L[L_{\text{gan}}] + \delta}$ is an adaptive weight. $\nabla G_L[\cdot]$ means the gradient of its input to the last layer of the decoder $G_L$. And $\delta = 10^{-6}$ is used for numerical stability.

**Vision Transformer.** The Transformer architecture [47] has led the way in many language tasks. In vision, some have used Transformers for image classification [16, 43]. But in video, it’s newer territory. A few works [1, 6] show promise using spatial-temporal attention for video. The main hurdle? Transformers are computationally heavy due to their quadratic complexity. Some solutions like Longformer [4], Linformer [50], and Nyströmformer [53] are trying to cut this down. We’re using trajectory attention [33] for dynamic scenes, which is faster due to its linear complexity.

**Autoregressive Vision Prediction Models.** PixelCNNs [45] and Image-GPT [11] are key models for generating images. Weissenborn et al. [51] used this idea for videos, breaking down the data distribution $P(x)$ into conditional probabilities for all pixels:

$$P(x) = \prod_{i=0}^{n} p_{\theta}(x_i|x_{<i}),$$  \hspace{1cm} (5)

where $n$ is the full dimensionality of the data. $x_{<i}$ means the pixels before $x_i$ in the raster-scan ordering. In this paper, we jointly train the autoregressive model with additional perceptual loss and reconstruction loss which endow the model with visual semantic awareness.

3. Latent Motion Transformer

Figure 1 illustrates the overall architecture of our proposed LMT. In the following subsections, we will give detailed descriptions about the two-stage training, i.e., the video VQ-GAN for latent codes learning and the latent Transformer model for prior learning.

3.1. 3D VQ-GAN

In the first stage, we utilize the 3D VQ-GAN to learn a set of latent codes on the sign videos. The 3D VQ-GAN
is composed of three module: a encoder, a codebook and a decoder.

**Encoder.** The encoder consists of a series of 3D convolution networks and a motion transformer network. Given a video $x \in \mathbb{R}^{T \times H \times W \times C}$, the 3D convolution networks reduce the size of the video to $x' \in \mathbb{R}^{t \times h \times w \times n_z}$, where $t=T/2^m$, $h=H/2^m$, $w=W/2^m$, and $m$ is the number of downsampling blocks.

**Motion Transformer.** We use the Transformer network to understand the long-term interactions in videos, as depicted in Figure 2. Given its quadratic complexity, direct self-attention on codes from convolution networks is challenging. To solve this, we utilize trajectory attention [33], which focuses on hand movement trajectories. This method identifies a reference point in space and time and finds its trajectory by comparing with other points over time. An approximation technique further accelerates this attention mechanism, as detailed in Algorithm 1.

![Figure 2. The architecture of the attention residual block with trajectory attention.](image)

**Algorithm 1 Prototype-based attention.**

1. $P \leftarrow \text{MostOrthogonalSubset} (Q, K, R)$;
2. $\Omega_1 = S(Q^TP/\sqrt{D})$;
3. $\Omega_2 = S(P^TK/\sqrt{D})$;
4. $Y = \Omega_1(\Omega_2Y)$;

The attention matrix is approximated using intermediate prototypes, selected as the most orthogonal subset of the queries and keys, given a desired number of prototypes $R$. This algorithm reduces the computation dependency of the attention to a linear complexity $O(N)$. To focus on our main contributions, we omit the detailed architecture and refer readers to [33] for reference.

**Codebook and Decoder.** The encoded low-dimension features are mapped into the discrete latent codes $z_q$ by calculating a nearest neighbor from the codebook embedding as shown in Eq. (1). Then we feed the corresponding embedding vector of $z_q$ to the decoder module. The decoder is composed of a motion Transformer and a series of 3D transposed convolution networks.

**Training.** The training objective of the video VQ-GAN is similar to the Eq. (4). The only difference is that our patch-level discriminator is composed of 3D convolutions instead of 2D convolutions used in image VQ-GAN [18].

### 3.2. Latent Transformer

In the second stage, we learn an autoregressive prior model over the flattened latent codes from the first stage. More precisely, the quantized encoding of a video is given by the encoder $E(x)$ and quantization $q(\cdot)$ as $z_q = q(E(x)) \in \mathbb{R}^{t \times h \times w \times n_z}$. We flatten the spatial-temporal indices in the raster scan order, and the indices can be seen as a sequence $S = \{s_1, s_2, \ldots, s_i, \ldots, s_{(t \times h \times w - 1)}\}$, where $0 \leq s_i \leq |Z| - 1$, $|Z|$ is the size of the codebook. Following the previous works, we generate the latent indices of videos in an autoregressive manner. Typically, the training objective of the autoregressive model is expressed as a chain of conditional probabilities in a left-to-right manner:

$$
\mathcal{L}_{ce} = - \sum_{t=1}^{t \times h \times w - 1} \log p(s_t | s_{\leq t}, s_c, l),
$$

where $\mathcal{L}_{ce}$ is the cross-entropy loss function.

**Conditional Generation.** In our task, we generate the conditional sign videos given by a specifical label and a still human-center image. Specifically, we utilize a learnable embedding $L \in \mathbb{R}^{n_z \times n_{cls}}$ to embed the one-hot label to a dense vector $l \in \mathbb{R}^{n_z}$ where $n_{cls}$ is the number of sign language category. And we also encode the conditional still image into discrete latent codes $s_c$. As shown in Figure 1, we introduce a sequence-to-sequence attention mechanism to better leverage the conditional information. In this attention mechanism, the latent codes of conditional frame and the label vector can see each other, but the future latent codes can only see previous codes before themselves with a causal mask. Then the conditional Transformer of our model can be formulated as:

$$
\mathcal{L}_{ce} = - \sum_{t=1}^{t \times h \times w - 1} \log p(s_t | s_{\leq t}, s_c, l).
$$

### 3.3. Learning a Perceptually Prior Model

Many past studies mainly use the token-level cross-entropy loss when learning the prior model. However, this doesn’t account for the visual meaning of the predicted codes, leading to a decline in generation quality. To address this, our approach first translates these predicted codes back into videos, then calculates the perceptual loss [57] and the reconstruction loss. Due to the non-differentiability of predicting codes from the Transformer output, the Gumbel-softmax technique [23] is employed, using the straight-through estimator for gradient approximation [5]. As illustrated by the red line in Figure 1, after deducing the indices $\hat{s}$ and converting them to codebook entries $z_q(\hat{s})$, we decode them into a video $\hat{x} = G(z_q(\hat{s}))$. The final step involves computing the perceptual and reconstruction losses by comparing the predicted video $\hat{x}$ to the actual video $x$. The complete objective of the latent Transformer is defined by:

$$
\mathcal{L}_{\text{Transformer}} = \mathcal{L}_{ce} + \mathcal{L}_{\text{lpips}}(\hat{x}, x) + \mathcal{L}_{\text{recon}}(\hat{x}, x).
$$

3027
Table 1. The evaluation scores of all models on the two benchmark datasets. ↓ means the lower the better. ↑ means the higher the better. “∗” indicates the results obtained by our implementation.

<table>
<thead>
<tr>
<th>Models</th>
<th>WLASL</th>
<th>NMFs-CSL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Reconstruction</td>
<td>Predicting</td>
</tr>
<tr>
<td></td>
<td>R-FVD ↓</td>
<td>LPIPS ↓</td>
</tr>
<tr>
<td>pose-sequence based method</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Text2Sign [41]∗</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>GAN based method</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TFGAN [3]</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MoCoGAN-HD [42]</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Latent Transformer method</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LVT [36]</td>
<td>160.10</td>
<td>0.131</td>
</tr>
<tr>
<td>VideoGPT [55]</td>
<td>148.27</td>
<td>0.094</td>
</tr>
<tr>
<td>LMT (Ours)</td>
<td><strong>120.77</strong></td>
<td><strong>0.048</strong></td>
</tr>
</tbody>
</table>

To make a summary, equipping with our several improvements, we finally arrive at our proposed latent motion Transformer for sign language production.

4. Experiments

4.1. Experimental Settings

**Datasets.** We evaluate our proposed model on two public word-level sign language datasets: WLASL [29] and NMFs-CSL [20]. (i) WLASL is an American sign language dataset (ASL) that are collected from the web videos. It contains 2,000 words and 21,083 samples in total for training, validation, and test, respectively. Among them, the Top-100 and Top-300 most frequent words are released as WLASL100 and WLASL300, respectively. (ii) NMFs-CSL is a Chinese sign language dataset (CSL) which are collected in the laboratory environment. It totally contains 1,067 words, 25,608 and 6,402 samples for training and testing, respectively.

**Evaluation Metrics.** Our evaluation consists of two stages, each with specific metrics:

- For the initial reconstruction phase, we use: (i) LPIPS (Learned Perceptual Image Patch Similarity) [57], a frame-centric metric that gauges the perceptual likeness of two frames. (ii) R-FVD (reconstruction Fréchet Video Distance) [44], a dynamic-oriented metric derived from comparing statistics of the I3D network trained on the Kinetics-400 dataset [10].

- In the second stage’s assessment, we employ: (i) FVD to compare the resemblance between the sampled videos and the original ones. (ii) Due to the unsuitability of LPIPS for the sampling stage, we utilize an accuracy metric, SLR-Acc, for the conditional generation. This metric assesses if our videos align with the designated categories. We opt for the publicly available leading I3D model as our SLR. It showcases a Top-5 accuracy of 88.0% on NMFs-CSL and 78.38% on the WLASL 300 dataset.

**Training Details.** We standardize all image data to a range of $[-0.5, 0.5]$. Each video’s resolution is set at 128×128, with 16 frames sampled per sign word. Depending on the dataset, we adjust our sampling approach. For the web-sourced WLASL dataset, we apply uniform sampling. As NMFs-ASL is lab-gathered, with key frames predominantly in the middle, we sample 16 frames from the central 48 frames. During inference, we generate 16 frames for each sign word. In the second training stage, the initial static image serves as the conditional frame. For in-depth model specifics and hyperparameters, please refer to the Appendix.

In the first stage, we use the AdamW optimizer to optimize the generator and discriminator with learning rate as $3e^{-4}$. In the second stage, we use the AdamW optimizer to optimize the latent Transformer model. The learning rate is also set as $3e^{-4}$ with a cosine annealing schedule. We utilize up to 8 Nvidia GTX 3090 GPUs (24G) to achieve all the experiments.

4.2. Experimental Results

In this section, we provide the quantitative comparison with existing methods for conditional word-level sign language production. We assess the performance of our model in terms of the mentioned evaluation metrics: R-FVD, LPIPS, FVD, and SLR-Acc.

4.2.1 Baselines

**Pose-sequence based method.** Text2Sign [41] converts spoken language into sign pose sequences using an NMT network and a motion graph, and then creates realistic sign videos from these poses. We reimplemented their method on our two word-level sign language datasets, adjusting the input to the NMT network to be a word rather than a sentence.

**GAN based method.** TFGAN [3] is a conditional GAN...
with an effective multi-scale text-conditioning scheme based on discriminative convolutional filter generation. MoCoGAN-HD [42] is built on top of a pre-trained image generator [24]. It first learns the distribution of video frames as independent images, and then introduces a motion generator to discover the desired trajectory between the continuous latent codes, in which content and motion are disentangled.

**Latent Transformer method** VideoGPT [55] and LVT [36] are both two-stage latent transformer methods which are similar to our proposed approach. They use VQ-VAE to learn discrete latent representations of a raw video. And then a simple Transformer architecture is used to autoregressively model the discrete latents.

### 4.2.2 Main Results.

Table 1 reveals our model’s notable advancements over latent autoregressive methods like VideoGPT and LVT, evident in the R-FVD and FVD score improvements. Against GAN-based models, our results are superior by marked FVD score gains. Compared to pose-sequence methods, we show clear progress. Figures 3 and 4 showcase our model’s capability in producing detailed future frames and consistent sign videos with accurate semantics. While our SLR-Acc outperforms others, all models still underperform in this metric, suggesting both the need for a more refined SLR model and challenges in generating precise sign videos, leaving scope for enhancement.

Furthermore, models perform differently on the WLASL and NMFs-CSL datasets. While they reconstruct better on WLASL, their predictive abilities are stronger on NMFs-CSL. This could be due to the shorter gesture duration and faster hand movements in NMFs-CSL. However, the lab-curated nature of NMFs-CSL, with central sign movements, aids in better understanding and forecasting.

### 4.3. Ablation Study

In this section, we present the ablation study that provides some intuitions as to why our approach works better...
Do 3D VQ-GAN with Motion Transformer Generates High-Fidelity Sign Videos? In the first stage model, we make two improvements: replacing the VA-VAE model with VQ-GAN and incorporating the trajectory attention into the VQ-GAN architecture. To demonstrate the effectiveness of the modifications, we conduct comparable experiments on the WLASL dataset. As shown in Table 2, trajectory attention performs much better than axial attention which is used in VideoGPT [55]. Moreover, by jointly training with perceptual loss and a patch-level discriminator, the VQ-GAN model can generate high-quality future frames than the VQ-VAE model. Figure 5 and Figure 6 show some videos reconstructed by the video VQ-GAN with a motion Transformer on the two WLASL and NMFs-CSL datasets, respectively.

Effect of the Spatial-Temporal Downsampling Factor. Choosing the right spatial-temporal downsampling factor for sign videos is crucial. In experiments on the WLASL dataset, as per Table 3, larger temporal latent sizes lead to improved reconstruction quality, and the trend is similar for spatial dimensions. However, the maximum spatial latent size of 64×64 isn’t optimal, suggesting patch-level Transformers outdo pixel-level ones.

The optimal spatial-temporal downsampling factor is (2, 4, 4), equating to a latent size of 8×32×32. Yet, bigger latent sizes increase computational demands in the subse-
Table 4. The effect of the size of the codebook on the WLASL dataset.

<table>
<thead>
<tr>
<th>Size of Codebook</th>
<th>R-FVD↓</th>
<th>FVD↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>2048</td>
<td>121.32</td>
<td>300.25</td>
</tr>
<tr>
<td>1024</td>
<td>120.77</td>
<td>301.62</td>
</tr>
<tr>
<td>512</td>
<td>136.65</td>
<td>324.11</td>
</tr>
</tbody>
</table>

Table 5. The effect of different attention mechanism over the flattened codes in the second stage on the WLASL dataset.

<table>
<thead>
<tr>
<th>Attention of Transformer</th>
<th>FVD↓</th>
<th>SLR-Acc↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Casual Attention</td>
<td>319.36</td>
<td>0.416</td>
</tr>
<tr>
<td>Sent.-to-Sent. Attention (Ours)</td>
<td>301.62</td>
<td>0.430</td>
</tr>
</tbody>
</table>

Do Larger Size of Codebook Help? Unlike the vanilla VAE model, the VQ-VAE and VQ-GAN models use discrete latent codes that do not suffer from “posterior collapse”. Similar to this collapse problem, using discrete latent space also faces the “index collapse”, where only a few of the embedding vectors get trained due to a rich getting richer phenomenon. To explore this problem, we conduct experiments with different numbers of codebook vectors on the WLASL dataset. As shown in Table 4, using 512 codes obtains a worse generation quality. Then increase the number of codes from 512 to 1024, the reconstruction and prediction performances are improved. When the code number reaches 2048, the performance is not affected. This phenomenon indicates that more codes increase the expression of the latent codebook, and using 1024 codes access a base threshold of the generation quality.

Effect of Different Attentions over the Flattened Latent Codes. In the second stage of prior training, we replace the pure casual attention with the sentence-to-sentence attention to better leverage the conditional information. The attention mask mechanism is shown in Figure 1, the latent codes of the conditional frame can see each other, thus learning a better representation of a single frame. In Table 5, we compare LMT with different attention mechanism. We can see that our proposed attention brings benefits to our generation quality.

Do Perceptual Loss and Reconstruction Loss Help for Prior Model Learning? Traditional latent autoregressive transformer models are primarily optimized using token-level cross-entropy loss, focusing on the autoregressive nature of the codes to produce meaningful latent representations. This method, however, lacks a visual evaluation of the generated codes. As evidenced in Table 6, training the model with both perceptual and reconstruction losses results in a substantial FVD improvement of −58.58. Figure 7 further illustrates the training loss curves. By integrating these two additional losses, the Transformer not only converges quicker but also yields a lower cross-entropy loss. This highlights that incorporating perceptual and reconstruction losses grants the model better visual understanding, leading to a more proficient prior model.

4.4. Limitations

Using a two-stage strategy, our model produces conditional sign videos, a more intricate approach than one-stage methods. While it creates coherent, high-quality frames, it has potential for better accuracy in generating videos for specific sign words.

5. Conclusion

In this work, we propose a novel Latent Motion Transformer (LMT) model to realize the word-level sign language production without human pose sequences. We’ve developed a two-stage model inspired by recent latent autoregressive approaches to enhance sign video generation. First, we utilize the VQ-GAN for effective latent code learning and the motion Transformer to track hand movements. In the next stage, we integrate sentence-to-sentence attention for better conditional and categorical data use. Instead of using the cross-entropy loss function, we combine perceptual and reconstruction losses, resulting in enhanced generation quality. Tests on two sign language datasets confirm our model’s superior performance against existing methods.
References


[23] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In 5th Interna-


