Bridging Text-to-Sign Translation via Codebook-Oriented Pretraining

Ninlawat Phuangchoke, Chantri Polprasert
Department of Information & Communications Technologies, Asian Institute of Technology, Thailand
st124847@ait.asia, chantri@ait.asia

Abstract

Sign Language Production (SLP), the automatic translation from spoken to sign languages, faces several challenges due to the intricate mapping between linguistic semantics and the spatial–temporal motion domain. Existing SLP methods employing a transformer model with a Vector Quantization (VQ) method exhibit poor translation performance due to weak semantic alignment between the codebook and the text representation. In this work, we propose a novel text-to-sign translation based on model pretraining, which enhances semantic alignment by inheriting codebook-oriented prior knowledge from masked self-supervised models. Our approach involves two stages: (i) transforming sign language into discrete values by employing VQ with masked self-attention learning to create pre-tasks that bridge the semantic gap between text and codebook representations, (ii) constructing an end-to-end architecture with an encoder-decoder-like structure that inherits the parameters of the model from the first stage. The integration of these designs forms a robust sign language representation and significantly improves the translation model, which surpass prior baselines.

Keywords: Vector Quantization, Sign Language Production, Pre-trained models

1. Introduction

Sign language is one of the primary forms of communication among Deaf individuals Sutton-Spence and Woll (1999). To facilitate communication between Deaf individuals and hearing people, Sign Language Production (SLP) techniques have been continuously developed. SLP involves translating fluent spoken language into a sign language sequence. This has led to continuous development of sign language animation systems using virtual avatars, which are more interesting and closer to human gestures. SLP is more challenging than Neural Machine Translation (NMT) due to its complex mapping process.

In the early stage, most studies Saunders et al. (2020); Zelinka and Kanis (2020) adopted an autoregressive approach for SLP, which suffers from regression-to-the-mean and error propagation problems. Conversely, subsequent works Hwang et al. (2021, 2022) explored non-autoregressive methods to alleviate these issues; however, the model operates with a fixed sequence length, resulting in some infomation being lost. Beyond the non-autoregressive paradigm, diffusion-based models Baltatzis et al. (2024); Tang et al. (2025) have been proposed to further enhance motion quality through iterative denoising, at the expense of substantial computational resources.

Recently, an increasing number of studies have focused on converting continuous data into discrete representations. For instance, Hwang et al. (2024) proposed the Concrete Distribution, a technique for approximating the sampling process of discrete variables from a continuous distribution. Similarly, Yin et al. (2024) introduced Vector Quantization-Variational Autoencoders (VQ-VAE), which employs a standard autoencoder framework that compares the encoder output with codebook vectors, selecting the nearest vector as the quantized representation passed to the decoder. These discretization techniques have been applied to SLP tasks Walsh et al. (2024); Hwang et al. (2024); Yin et al. (2024) using various quantization methods, such as Noise Substitution Vector Quantization (NSVQ), Gumbel-Softmax relaxation, and vanilla quantization, respectively. However, the problem of aligning the trained codebook with text representations remains challenging due to sparsely activated codebooks and weak semantic correspondence.

To address regression-to-the-mean and cumulative error propagation problems inherent in autoregressive models, we propose a non-autoregressive framework that leverages online codebook learning. Our method builds upon Clustering VQ-VAE Zheng and Vedaldi (2023) (CVQ-VAE), which introduces an online reinitialization mechanism that dynamically updates under-utilized codevectors based on the running averages of encoded features. This mechanism enables full codebook utilization and significantly improves both reconstruction and generation quality. Subsequently, we learn the mapping from text to discrete representations through a discrete diffusion model.

To address the mismatch learning codebook, we propose a novel pre-training approach that integrates masked self-supervised learning. Specifically, we incorporate masked self-supervised learning into the pre-training process to enhance the Codebook Decoder's ability to capture the syntactic

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 9504–9513
11–16 May 2026.

and semantic characteristics of textual sentences. In addition, we design the Text Encoder to align with the quantized representations, facilitating better correspondence between text and codebook embeddings. The second stage adopts an encoder–decoder structure and inherits parameters from the pre-trained Text Encoder and Codebook Decoder of the first stage. Furthermore, we integrate a DIFF Transformer Ye et al. (2024) into the vector quantization model to further strengthen the attention mechanism.

In summary, the main contribution of our work are listed as follows:

- We achieve improvement in the BLUE-4 score for SLP. In particular, compared with the SLP Progressive Transformer Saunders et al. (2020) method, our method has at least 7% improvements on the PHOENIX14T. These results demonstrate a substantial advancement in the task of SLP.
- To the best of our knowledge, this is the first attempt to extend DIFF Transformer to enhance capture sign pose sequence in vector quantization model.
- We propose a two-stage SLP system consisting of two components: (i) CVQ-VAE to transform continuous data into discrete data with a novel pre-training paradigm that applies masked self-supervised learning to facilitate mapping text and discrete data. (ii) A novel model to predict codes with inheriting parameters from stage (i). This approach represents a significant improvement over previous approaches and holds great promise for enhancing the accuracy of SLP systems.

2. Background

2.1. Sign Language Production

Prior studies on Sign Language Production (SLP) can be classified into five categories Rastgoo et al. (2021) : NMT, motion graph–based approaches, avatar-based methods, conditional image/video generation, and others. Initially, animated avatar methods were applied to SLP systems based on dictionary lookup and predefined gesture movements Glauert et al. (2006); Karpouzis et al. (2007); McDonald et al. (2016). Stoll et al. (2018) proposed the first SLP model integrating a Generative Adversarial Network (GAN) with Neural Machine Translation (NMT), dividing the SLP pipeline into two main steps: (1) translation from Text to Gloss (T2G) and (2) mapping from Gloss to Sign (G2S). Saunders et al. (2020) introduced the Progressive Transformer, which separates the generation process into two procedures: generating continuous pose sequences directly from spoken language sentences (T2S) and through an intermediate gloss representation (T2G2S), demonstrating superior translation performance. In contrast, this approach suffers from the problem of regression-to-the-mean, resulting in imprecise signing. Several attempts have been made to mitigate this issue, including the adoption of adversarial training and Mixture Density Networks (MDN) Saunders et al. (2021). Moreover, non-autoregressive Text-to-Sign (T2S) architectures had also been incorporated into SLP models Hwang et al. (2021), leveraging Gaussian latent to translate the source sentence to the target sign pose distribution. Furthermore, due to Large Language Models (LLMs) having become increasingly popular and continue to be actively developed, Fang et al. (2024) proposed SingLLM, which provides PROMT2SIGN enabling sign language generation based on query texts input and question-style prompt input respectively.

2.2. Alternative Representation to Gloss

Several alternative annotation systems to gloss have been explored, including the Hamburg Notation System (HamNoSys) Hanke (2004) and SignWriting Sutton (2022) have been proposed to describe sign language more precisely. HamNoSys is an alphabet notation framework for sign languages, developed to transcribe signs at the phonetic level, which decomposes sign language into three components: location, hand configuration (handshape), and movement. Uchida et al. (2024) employed HamNoSys components in a MoCap framework to contextually modify citation-form signs according to translated gloss sequences. Moreover, SignWriting is applied to help translating Jiang et al. (2022) and animating Bouzid and Jemni (2013) tasks, respectively.

2.3. Alternative Sign Representations

Most deep learning models require large amounts of data to learn meaningful representations Goodfellow et al. (2016). However, sign language datasets remain limited. For example, the mDGS dataset contains only about 50k parallel sentences annotated with both gloss and HamNoSys Konrad et al. (2020). These annotations are labor-intensive and costly to produce. Therefore, this paper proposes learning discrete representations from 2D sign pose data as an alternative to gloss and HamNoSys.

2.4. Discrete Representation

The first VAE was introduced by Kingma and Welling (2013) and showed promising results. However, they had difficulty capturing fine-grained struc-

tures and optimization issues. Van Den Oord et al. (2017) developed on this introducing the VQ-VAE architecture, which combines VQ with the latent space of a VAE, forcing the embedding space of the VAE to be discrete. This led to enabling state-of-the-art image and audio generation. Since then, VQ has been extended to a variety of applications, such as hierarchical music generation Zhu et al. (2022), self-supervised speech synthesis Chiu et al. (2022), and most recently, diffusion-based image generation Tang et al. (2022). In detail, the vanilla VQ-VAE architecture used an argmin employed to select the closest matching codebook entry. As a result, the model uses the Straight-Through Estimator (STE) to directly propagate gradients from the decoder to the encoder with three separate losses to train: a reconstruction loss, codebook loss and a commitment loss. However, during training, some embedding vectors are rarely utilized, leading to a phenomenon known as codebook collapse. Kaiser et al. (2018) adopted an Exponential Moving Average (EMA) scheme for updating the codebook parameters, enabling stabilized training and reducing the necessary loss functions to two. Jang et al. (2016) introduced an effective approach to learn a discrete latent distribution with Gumbel Softmax Relaxation. Recently, Zheng and Vedaldi (2023) proposed an online codebook learning strategy, which uses anchors from the encoder to update inactive codevectors, while underutilized codevectors are softly reinitialized. This simultaneously helps address the issues observed in previous vector quantization models. Therefore, we adopt this architecture in our approach.

VQ has been widely employed in prior studies addressing the SLP task. Hwang et al. (2024) leverage spatio-temporal representations of sign poses with SignVQNet. This work attempts a latent-level alignment language and pose token representations; however, such alignment may fail under linguistically complex or structurally diverse conditions. Similarly, Walsh et al. (2024) apply NSVQ to transform sign sequences into codebook tokens, then apply a transformer to map codebooks to sign poses, which corporate a sign stitching method. Nevertheless, this method does not clearly describe the loss function training text-to-codebook translation. Yin et al. (2024) present a novel dynamic vector quantization (DVA-VAE) model, allowing dynamically adjusting the encoding length followed the information density in sign language. Compared to fixed-length quantization models, DVA-VAE provides greater flexibility but requires multiple loss objectives for training.

In summary, previous SLP algorithms, which applied vector quantization methods, train discrete data separately that affect the performance of the translation model. To address this problem, we present a proof of concept that uses a masked within the quantization process to predict discrete codes before quantization, and its parameters are inherited in the subsequent translation stage, enhancing the mapping from text representations to discrete data, facilitating its downstream application that trains codebooks separately and improving translation accuracy.

3. Methodology

In this section, we detail the CVQ-VAE, which learns discrete representations of sign sequences and supports masked self-supervised training, followed by a translation stage. We first formalize the SLP problem, then describe the first stage including learning codebook of tokens that can be directly mapped back to a sequence of poses. We then translate spoken-language text into a sequence of latent codes in the second stage. The pipeline's individual architectures are shown in Figure 1., and each stage is described in the following sections.

3.1. Problem Definition

To translate from spoken to signed languages, the following holds. Given a spoken language sequence $X = [x_1, x_2, \ldots, x_W]$ with W words, the goal is to generate a continuous sequence of sign poses $P = [p_1, p_2, \ldots, p_U]$ with U frames, where each pose is represented by J joints in a D-dimensional space. The SLP model is challenging since the target sequence is much longer than the source $(U \gg W)$, a difficulty that remains even with state-of-the-art sequence-to-sequence models Vaswani et al. (2017). The components of the pipeline are shown in Figure 1, and we detail each stage in the following sections.

3.2. Stage 1: Clustering VQ-VAE (CVQ-VAE)

The codebook aims to capture a set of motion patterns from continuous signing data. Our method utilizes a transformer-based encoder-decoder architecture integrated with CVQ-VAE. To address this, we first learn a codebook for tokens, each representing a short signing segment that can be directly reconstructed into pose sequences.

Encoder. The short sign language sequence $P = [p_1, p_2, \ldots, p_{U_{cb}}]$ contains U_{cb} frames with positional encoding added to each pose. Then, we embed the signed sequence using a linear layer. The sequence is then passed to the transformer encoder, which consists of spatial-temporal attention, enabling the network to learn and capture the long-range dependencies with the sequence. The embedded features are represented as $z \in \mathbb{R}^{U_{cb} \times H}$,

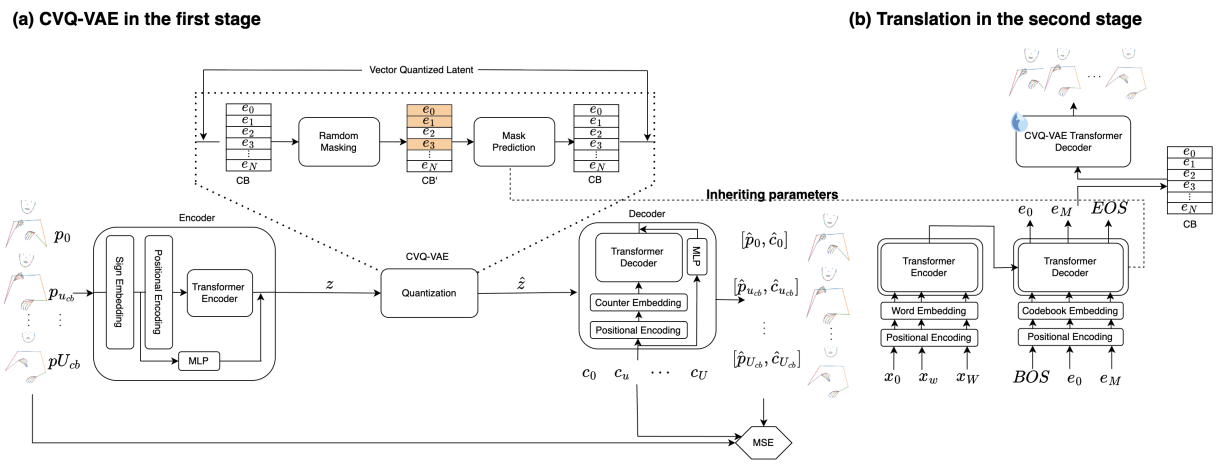

Figure 1: The overall framework of the proposed two-stage architecture.

where H denotes the embedding dimension. Each codebook vector is optimized to represent a localized sub-segment of the overall continuous sequence, thus $U_{cb} \ll U$.

Vector Quantization. A set of tokens is learned by the encoder through the CVQ-VAE codebook, denoted the codebook as $CB = [e_1, e_2, \ldots, e_N]$, where N is the number of tokens in the codebook and each $e_i \in \mathbb{R}^{U_{cb} \times H}$. Thus, the length of each pose sequence, U_{cb} specifies the number of frames represented by each codebook token. To train CVQ-VAE, each embedded feature $\hat{z}_i$ from the encoder is quantized to its nearest codebook entry e_k , where $k = \arg\min_{e_k \in \mathcal{Z}} \|\hat{z}_i - e_k\|_2$ and:

$$z_q^i = q(\hat{z}_i) = e_k. \tag{1}$$

This operation discretizes the continuous latent space into a finite set of representative embeddings, enabling compact and interpretable discrete representations. The quantized features are then forwarded to the decoder for reconstruction. The training objective combines embedding L_{embed} and commitment terms L_{commit}:

$$\mathcal{L}_{vq} = \underbrace{\|z - sg[z_q]\|_2^2}_{\mathcal{L}_{\text{embed}}} + \underbrace{\beta \|sg[z] - z_q\|_2^2}_{\mathcal{L}_{\text{commit}}} \tag{2}$$

where sg denotes the stop-gradient operator, and β is a hyper-parameter for the commitment loss. The first aligns codevectors with the encoder outputs, and the second constrains the encoder to remain close to its assigned discrete code, ensuring stable training.

CVQ-VAE introduces an *online reinitialization mechanism* that dynamically updates rarely used codevectors based on moving averages of encoded features. $N_k^{(t)}$ denotes the running average usage frequency of the k-th codevector at iteration t:

$$N_k^{(t)} = N_k^{(t-1)} \cdot \gamma + \frac{n_k^{(t)}}{Bhw} \cdot (1 - \gamma) \tag{3}$$

A decay coefficient $\alpha_k^{(t)}$ is computed according to the average usage:

$$\alpha_k^{(t)} = \exp\left(-\frac{N_k^{(t)} K}{10(1-\gamma)} - \varepsilon\right) \tag{4}$$

where ε is a small constant ensuring stability. This weight determines the degree to which a codevector is updated low-usage entries (small $N_k^{(t)}$) receive stronger reinitialization. Each codevector is then updated using a moving-average combination between its previous state and a newly sampled *anchor feature* $\hat{z}_k^{(t)}$:

$$e_k^{(t)} = e_k^{(t-1)} \left(1 - \alpha_k^{(t)}\right) + \hat{z}_k^{(t)} \alpha_k^{(t)} \tag{5}$$

The anchor feature $\hat{z}_k^{(t)}$ can be selected via several strategies such as *random*, *unique*, *closest*, or *probabilistic sampling* from the current encoded features. In this work, we select *closest* for the anchor feature.

Decoder: The decoder reconstructs the original pose sequence from the quantized embeddings using the counter decoding method of Saunders et al. (2020). This non-autoregressive approach processes the entire sequence in one step, reducing computation and improving inference speed, while outperforming a simple MLP in reconstruction accuracy. The counter value is defined as

$$c_u = \frac{u+1}{U_{cb}} \tag{6}$$

where u denotes the current position and U_{cb} the total sequence length. The counter values are embedded through positional encoding and a linear layer, followed by a spatial-temporal transformer decoder with self- and cross-attention to generate

output embeddings. These are then projected back to pose and counter values via two linear layers. The reconstruction loss L_{re} is trained end-to-end with the following loss function:

$$\mathcal{L}_{re} = \frac{1}{U_{cb}} \sum_{u=1}^{U_{cb}} \left(P_u - \hat{P}_u\right)^2 + \alpha \left(c_u - \hat{c}_u\right)^2 \quad (7)$$

where α is a scaling factor that we determine empirically. $\hat{P}$ and $\hat{c}$ are the predicted pose and counter values, respectively. The final loss for the entire architecture is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{vq}} + \mathcal{L}_{\text{re}} \quad (8)$$

Attention Fusion Module. In our work, the attention module, inspired by MCST-Transformer Ma et al. (2024) is applied to both the encoder and the decoder architectures based on the entire embedding. We design a dual-attention fusion module that integrates Spatial Attention Vaswani et al. (2017) and Differential Attention Ye et al. (2024). The former enhances local spatial correlation among landmark channels, while the latter captures global temporal dynamics through differential attention. The two outputs are linearly fused with a learnable coefficient, enabling the model to effectively balance spatial precision and temporal consistency.

Multilayer Perceptrons (MLP). As shown in Figure 1., we employ residual connection with MLP in the encoder and decoder model, which refines per-frame keypoint features and injects them into both the encoder and decoder through residual fusion. By doing this, we use ReLu activation function to extract features.

Masked Self-Supervised Module. To improve the mapping from text to discrete representations, we introduce a masked self-supervised model that enhances semantic alignment between text and codebook tokens. The model uses a Transformer decoder trained to predict randomly 15% masked codebook sequences; its parameters are then transferred to the translation stage as illustrated in Figure 1. We first train the CVQ-VAE to convergence and, only afterward, train the masked model, thereby avoiding interference with codebook learning. The masked model is optimized with the cross-entropy loss.

3.3. Stage 2: Translation (Text-to-Sign)

The predicted token sequence T can be directly converted into a corresponding sequence of poses P. We use the autoregressive transformer encoder-decoder architecture with inherited parameters from the first stage to translation from text to signed tokens where the decoder transformer applies an argmax operation to select the predicted sign token denoted as $e_i = \arg\max(h_i)$, where h_i is the i^{th} hidden representation from the CVQ-VAE encoder. Finally, given predicted tokens, we use pre-trained decoder to convert them into the sign sequence $P = D(T)$.

The model is optimized by minimizing the Negative Log-Likelihood loss:

$$\mathcal{L}_{\text{nll}} = -\sum_{i=1}^{M} \log\left(p(e_i \mid e_{<i}, x)\right), \quad (9)$$

where $p(e_i \mid e_{<i}, x)$ denotes the probability of the i^{th} token e_i given the previous tokens $e_{<i}$ and input x.

4. Experimental Setup

4.1. The Dataset

We use the PHOENIX-2014T dataset Camgoz et al. (2018). PHOENIX-2014T is an extension of the RWTH-PHOENIX-Weather-2014 corpus, originally created for Continuous Sign Language Recognition (CSLR), and was recorded by 9 professional sign language interpreters. The dataset consists of 7,096 training, 519 validation, and 642 test samples. The dataset contains vocabularies of 2,887 spoken words and 1,066 signs. Each video is annotated with both glosses which are written labels used to represent individual signs or meaningful units in sign language and German translation.

The training data consist of sequences of 2D skeleton poses formed by concatenating upper-body joint coordinates, including the face and both hands. Because PHOENIX14T provides only video frames, we extract 2D keypoints using MMPose (top-down) with RTMW-X (Jiang et al., 2023, 2024), yielding 78 joints in the COCO-WholeBody layout. Frames are cropped and resized to (256x192) prior to inference, and keypoint coordinates are normalized to [0,1] by image width/height. We then remove obviously noisy/jittery frames and idle frames with negligible hand motion, recentre poses at the nose, scale using the neck–shoulder geometry, and apply min–max normalization to [0,1] before downstream modeling.

4.2. Model Configuration

Our CVQ-VAE uses a Transformer encoder–decoder architecture, with 2 layers and 8 heads per layer, a 512-dimension embedding, a 2048-dimension feed-forward network Saunders et al. (2020), and dropout of 0.1, with about 37M parameters in total. We initialize the learning rate at 10^{-4} and train to convergence with a floor of 10^{-6}. The window size for each codebook entry is 4. The masked self-supervised model employs a 2-layer, 8-head Transformer decoder with the

same embedding, feed-forward sizes, learning rate and dropout of 0.2 with about 39M parameters in total.

After extensive hyperparameter tuning configuration: a 2-layer encoder–decoder Transformer with 8 attention heads, 512-dimension embeddings, and a 1024-dimension feed-forward network. Decoding uses beam search (beam size = 5) with a length penalty of 2.0. For encoding, we use the *distilbert-base-german-cased* model Sanh et al. (2019) for tokenization and to obtain text embeddings, which are fine-tuned during training with a learning rate of 10^{-5} and 73M parameters in total.

All models use ReLU activations between layers and apply pre-layer normalization for regularization and training stability. Training employs a Reduce-on-Plateau scheduler (patience = 5, reduction factor = 0.9). We initialize the Transformer encoder and decoder with Xavier initialization Glorot and Bengio (2010) zero biases and optimize with Adam optimizer Adam et al. (2014). The learning rate starts at 10^{-4}, and training proceeds to convergence.

4.3. Evaluation Metric

To assess our model, we adopt the back-translation system proposed by Saunders et al. (2020), translating produced sign-pose sequences back into the source modality (text or gloss). For comparability with prior work, we use the standard sign language translation system of Camgoz et al. (2020); configuration to ensure fair comparison: 3 layers, 8 heads, 512-dim embeddings (ReLU), hidden size 512, feed-forward size 2048, and dropout 0.1. On PHOENIX14T, we report BLEU-1–4 Papineni et al. (2002) and ROUGE Chin-Yew (2004) on the back-translated text, where higher scores indicate better performance.

4.4. Task

We consider two SLP tasks: gloss-to-sign (G2S) and text-to-sign (T2S). G2S generates signed sequences from gloss sequences, whereas T2S generates signed sequences from spoken-language text sequence.

5. Quantitative Results

In this section, we report quantitative results on PHOENIX14T. We first compare our model with prior methods and obtain state-of-the-art performance on both G2S and T2S. We then present an ablation study on PHOENIX14T to assess the contribution of each proposed module.

5.1. Baseline Comparison

For closely related methods Hwang et al. (2024); Yin et al. (2024), we could not find public implementations at the time of submission. While some works do provide code Walsh et al. (2024), but the released repositories diverge from the corresponding papers (e.g., in architecture). For a fair comparison, we train all models on the same preprocessed data and with identical back-translation configurations. Therefore, instead of relying on the scores reported in the original papers, we re-evaluate the baselines under our protocol. The 'ground truth' refers to the evaluation of the actual ground-truth sign sequences from the dev and test sets using the back-translation model. This is in contrast to the other results, which evaluate the sign sequences predicted by the models.

As shown in Table 2, our model combining Residual MLP and Masked Self-Supervised learning, consistently outperforms Progressive Transformer Saunders et al. (2020). The results support our claim that residual MLP connections enhance the extraction of fine-grained sign features, while the masked model bridges text and codebook representations. More specifically, on the test set, we obtain at least a 20% relative improvement in BLEU-4 for the G2S task over (PT+GN), increasing the score from 5.85 to 7.19. Compared with NSLP-G, our model improves by roughly 5% BLEU-4 on the dev set for the G2S task, with the remaining results broadly comparable. When focusing on the T2S task, our model substantially improves BLUE-4 from 8.71 to 9.35 on the test set and from 9.58 to 9.91 on the dev set, compared to PT+GN shown in Table 1.

5.2. Ablation Study

To assess the contribution of each component, we perform ablations on the T2S task using PHOENIX14T. Table 4 reports results for different codebook sizes. Performance is sensitive to the number of codewords: a 500-entry codebook yields the best BLEU-4 scores equal to 9.91 on the dev set and 9.35 on the test set. This suggests that per-frame keypoints are well captured by a relatively compact codebook. Increasing the size (e.g., 2000, 2500, 3000) leads to lower BLEU scores. This trend is broadly consistent with prior work that fixes the codebook at 1024 entries (Hwang et al., 2024; Yin et al., 2024).

In Table 3, we present a component comparison of our model, considering the codebook baseline, the residual MLP, and the masked self-supervised stage. Relative to the Codebook baseline, adding the residual MLP increases dev BLEU-4 from 8.24 to 9.40 and ROUGE from 26.63 to 30.07, with corresponding test improvements from 7.46 to 7.78

Approach	Dev Set					Test Set				
	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE
GT	30.36	17.50	12.12	9.21	29.86	30.16	16.83	11.32	8.47	29.69
PT	24.91	13.22	9.02	6.91	25.68	24.72	13.00	8.63	6.54	25.23
PT+GN	30.23	17.60	12.30	9.58	30.43	29.50	16.66	11.42	8.71	28.90
Codebook + MLP + Masked self-supervised	**30.68**	**18.19**	**12.86**	**9.91**	**31.24**	**30.69**	**17.67**	**12.19**	**9.35**	**30.36**

Table 1: Comparison of back-translation scores of Progressive Transformer (PT), Progressive Transformer with Gaussian Noise and our model on the PHOENIX14T Dataset. The table presents results for T2S task.

Approach	Dev Set					Test Set				
	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE
PT	23.16	11.72	7.79	5.87	23.20	22.57	11.03	7.16	5.31	22.83
PT+GN	24.19	12.32	8.19	6.19	24.10	23.99	11.71	7.71	5.85	23.50
NSLP-G	27.81	16.09	11.46	9.09	28.96	28.60	16.49	11.42	8.87	28.92
Codebook + MLP + Masked self-supervised	**29.27**	**16.00**	**10.85**	**8.28**	**28.62**	**27.21**	**14.66**	**9.70**	**7.19**	**27.02**

Table 2: Comparison of back-translation scores of Progressive Transformer (PT), Progressive Transformer with Gaussian Noise, Non-Autoregressive SLP (NSLP-G), and our model on the PHOENIX14T Dataset. The table presents results for our data G2S task.

Figure 2: Example translation produced by our top-performing model on the PHOENIX14T dataset.

and from 26.40 to 28.33. Adding Masked Self-Supervised learning on top yields further benefits dev BLEU-4 from 9.40 to 9.91, ROUGE from 30.07 to 31.24, and notably test BLEU-4 from 7.78 to 9.35 and ROUGE from 28.33 to 30.36. The full model (Codebook + MLP + Masked) achieves 9.91 (BLEU-4) on dev and 9.35 (BLEU-4) on test, indicating that residual MLPs capture fine-grained pose details and the masked stage strengthens text–codebook alignment.

6. Qualitative Analysis

Figure 2 presents an example of text-to-sign translation from the PHOENIX14T dataset, illustrating that the model can generate pose sequences from a given sentence. Some fine hand details, however, are lost due to quantization errors introduced by the codebook and mismatch mapping between text and codebook representation.

7. Conclusions

We propose a new perspective for the SLP task by using discrete representation, which helps to alleviate the problem of regression-to-the-mean which previous approaches suffer from and mapping alignment between text and discrete representations. We introduce a novel pretraining paradigm that combines masked self-supervision learning to bridge employing codebooks in different stages. We test our proposed SLP method on the PHOENIX14T dataset and achieve a BLUE score equal to 9.91 on the dev set and 9.35 on the test set, yielding 3.44% and 7.35% improvement over baselines (PT+GN), respectively. Our experiments reveal that codebooks size affects on the performance of this method. Future work will focus on implementing our proposed method to other sign language datasets such as How2Sign.

8. References

Kingma DP Ba J Adam et al. 2014. A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 1412(6).

Vasileios Baltatzis, Rolandos Alexandros Potamias, Evangelos Ververas, Guanxiong Sun, Jiankang Deng, and Stefanos Zafeiriou. 2024. Neural sign actors: A diffusion model for 3d sign language production from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1985–1995.

Yosra Bouzid and Mohamed Jemni. 2013. An avatar based approach for automatically interpreting a sign language notation. In *2013 IEEE 13th International Conference on Advanced Learning Technologies*, pages 92–94. IEEE.

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

Approach	Dev Set					Test Set				
	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE
Codebook	25.19	14.22	10.26	8.24	26.63	25.77	13.79	9.54	7.46	26.40
Codebook + MLP	29.11	17.04	12.05	9.40	30.07	27.70	15.19	10.28	7.78	28.33
Codebook + MLP + Masked self-supervised	**30.68**	**18.19**	**12.86**	**9.91**	**31.24**	**30.69**	**17.67**	**12.19**	**9.35**	**30.36**

Table 3: The results of translating from spoken language text to sign (T2S) with different components on the PHOENIX14T dataset.

Vocabulary Size	Dev Set					Test Set				
	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE
500	**30.68**	**18.19**	**12.86**	**9.91**	**31.24**	**30.69**	**17.67**	**12.19**	**9.35**	**30.36**
1,000	29.52	17.36	12.12	9.27	29.78	29.01	15.94	10.81	8.07	28.15
1,500	30.50	17.49	12.04	9.26	30.51	29.65	16.59	11.05	8.15	29.08
2,000	26.88	15.22	10.58	8.19	27.91	26.75	14.65	10.05	7.69	27.11
2,500	29.49	16.59	11.53	8.91	29.14	28.67	15.84	10.77	8.09	28.21
3,000	28.04	16.14	11.19	8.56	28.54	27.35	14.35	9.70	7.24	26.72

Table 4: The results of translating from spoken language text to sign (T2S) with different codebook vocabulary sizes on the PHOENIX14T dataset.

of the IEEE conference on computer vision and pattern recognition, pages 7784–7793.

Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. 2020. Sign language transformers: Joint end-to-end sign language recognition and translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10023–10033.

Lin Chin-Yew. 2004. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out, 2004*.

Chung-Cheng Chiu, James Qin, Yu Zhang, Jiahui Yu, and Yonghui Wu. 2022. Self-supervised learning with random-projection quantizer for speech recognition. In *International Conference on Machine Learning*, pages 3915–3924. PMLR.

Sen Fang, Chen Chen, Lei Wang, Ce Zheng, Chunyu Sui, and Yapeng Tian. 2024. Signllm: Sign language production large language models. *arXiv preprint arXiv:2405.10718*.

John RW Glauert, Ralph Elliott, Stephen J Cox, Judy Tryggvason, and Mary Sheard. 2006. Vanessa–a system for communication between deaf and hearing people. *Technology and disability*, 18(4):207–216.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*, volume 1. MIT press Cambridge.

Thomas Hanke. 2004. Hamnosys–representing sign language data in language resources and language processing contexts. In *sign-lang@ LREC 2004*, pages 1–6. European Language Resources Association (ELRA).

Eui Jun Hwang, Jung Ho Kim, Suk Min Cho, and Jong C Park. 2022. Non-autoregressive sign language production via knowledge distillation. *arXiv preprint arXiv:2208.06183*.

Eui Jun Hwang, Jung-Ho Kim, and Jong C Park. 2021. Non-autoregressive sign language production with gaussian space. In *BMVC*, page 197.

Eui Jun Hwang, Huije Lee, and Jong C Park. 2024. A gloss-free sign language production with discrete representation. In *2024 IEEE 18th International Conference on Automatic Face and Gesture Recognition (FG)*, pages 1–6. IEEE.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Tao Jiang, Peng Lu, Li Zhang, Ningsheng Ma, Rui Han, Chengqi Lyu, Yining Li, and Kai Chen. 2023. Rtmpose: Real-time multi-person pose estimation based on mmpose. *arXiv preprint arXiv:2303.07399*.

Tao Jiang, Xinchen Xie, and Yining Li. 2024. Rtmw: Real-time multi-person 2d and 3d whole-body pose estimation. *arXiv preprint arXiv:2407.08634*.

Zifan Jiang, Amit Moryossef, Mathias Müller, and Sarah Ebling. 2022. Machine translation between spoken languages and signed languages represented in signwriting. *arXiv preprint arXiv:2210.05404*.

Lukasz Kaiser, Samy Bengio, Aurko Roy, Ashish Vaswani, Niki Parmar, Jakob Uszkoreit, and Noam Shazeer. 2018. Fast decoding in sequence models using discrete latent variables. In *International Conference on Machine Learning*, pages 2390–2399. PMLR.

Kostas Karpouzis, George Caridakis, S-E Fotinea, and Eleni Efthimiou. 2007. Educational resources and implementation of a greek sign language synthesis architecture. *Computers & Education*, 49(1):54–74.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Reiner Konrad, Thomas Hanke, Gabriele Langer, Dolly Blanck, Julian Bleicken, Ilona Hofmann, Olga Jeziorski, Lutz König, Susanne König, Rie Nishio, et al. 2020. Meine dgs–annotiert. öffentliches korpus der deutschen gebärdensprache, 3. release/my dgs–annotated. public corpus of german sign language, 3rd release. *Language Resource*, 3.

Xiaohan Ma, Rize Jin, and Tae-Sun Chung. 2024. Multi-channel spatio-temporal transformer for sign language production. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 11699–11712.

John McDonald, Rosalee Wolfe, Jerry Schnepp, Julie Hochgesang, Diana Gorman Jamrozik, Marie Stumbo, Larwan Berke, Melissa Bialek, and Farah Thomas. 2016. An automated technique for real-time production of lifelike animations of american sign language. *Universal Access in the Information Society*, 15(4):551–566.

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

Razieh Rastgoo, Kourosh Kiani, Sergio Escalera, and Mohammad Sabokrou. 2021. Sign language production: A review. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3451–3461.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. 2020. Progressive transformers for end-to-end sign language production. In *European Conference on Computer Vision*, pages 687–705. Springer.

Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. 2021. Continuous 3d multi-channel sign language production via progressive transformers and mixture density networks. *International journal of computer vision*, 129(7):2113–2135.

Stephanie Stoll, Necati Cihan Camgöz, Simon Hadfield, and Richard Bowden. 2018. Sign language production using neural machine translation and generative adversarial networks. In *Proceedings of the 29th British Machine Vision Conference (BMVC 2018)*. British Machine Vision Association.

Valerie Sutton. 2022. *Lessons in SignWriting*. SignWriting Press.

Rachel Sutton-Spence and Bencie Woll. 1999. *The linguistics of British Sign Language: an introduction*. Cambridge University Press.

Shengeng Tang, Feng Xue, Jingjing Wu, Shuo Wang, and Richang Hong. 2025. Gloss-driven conditional diffusion models for sign language production. *ACM Transactions on Multimedia Computing, Communications and Applications*, 21(4):1–17.

Zhicong Tang, Shuyang Gu, Jianmin Bao, Dong Chen, and Fang Wen. 2022. Improved vector quantized diffusion models. *arXiv preprint arXiv:2205.16007*.

Tsubasa Uchida, Taro Miyazaki, and Hiroyuki Kaneko. 2024. Hamnosys-based motion editing method for sign language. In *Proceedings of the LREC-COLING 2024 11th Workshop on the Representation and Processing of Sign Languages: Evaluation of Sign Language Resources*, pages 376–385.

Aaron Van Den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30.

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

Harry Walsh, Abolfazl Ravanshad, Mariam Rahmani, and Richard Bowden. 2024. A data-driven representation for sign language production. In *2024 IEEE 18th International Conference on*

Automatic Face and Gesture Recognition (FG), pages 1–10. IEEE.

Tianzhu Ye, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei. 2024. Differential transformer. *arXiv preprint arXiv:2410.05258*.

Aoxiong Yin, Haoyuan Li, Kai Shen, Siliang Tang, and Yueting Zhuang. 2024. T2s-gpt: Dynamic vector quantization for autoregressive sign language production from text. *arXiv preprint arXiv:2406.07119*.

Jan Zelinka and Jakub Kanis. 2020. Neural sign language synthesis: Words are our glosses. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3395–3403.

Chuanxia Zheng and Andrea Vedaldi. 2023. Online clustered codebook. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22798–22807.

Ye Zhu, Kyle Olszewski, Yu Wu, Panos Achlioptas, Menglei Chai, Yan Yan, and Sergey Tulyakov. 2022. Quantized gan for complex music generation from dance videos. In *European Conference on Computer Vision*, pages 182–199. Springer.