SIGNDICT: SIGN LANGUAGE PRODUCTION WITH A COMPRESSED POSE REPRESENTATION DICTIONARY AND ADAPTIVE FRAME WEIGHTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Sign Language Production (SLP) is a critical technology for breaking down communication barriers for people with hearing impairments and enabling interaction. Existing sign language production models are primarily divided into dictionary-based models and neural network-based models. Dictionary-based models offer high motion fidelity and good controllability, but typically suffer from cumbersome generation processes, stiff gesture transitions, and high storage costs. Neural network-based models can generate more fluid sign language movements and possess strong generalization capabilities, but they are prone to pose distortion and semantic misalignment. This paper proposes sign language production with a compressed pose representation dictionary and dynamic frame weighting (SignDict). By combining the pose priors of traditional dictionaries with the continuous generation capabilities of neural networks, SignDict achieves high-fidelity and fluent sign language pose generation. Unlike traditional dictionaries, which directly store complete discrete motion segments, SignDict constructs a pose feature dictionary that associates each gloss with a compact latent feature representation, representing sign language motions as latent feature vectors. During the generation process, the feature dictionary serves as a pose prior, guiding the model to generate pose sequences consistent with the semantic meaning. To achieve feature compression, this paper further designs an adaptive feature compression module that performs adaptive feature aggregation on pose segments of varying lengths corresponding to the same gloss, compressing them into a single latent feature vector. This significantly reduces the dictionary size while preserving motion information, achieving up to 1366-fold compression in storage footprint compared with traditional motion dictionaries. Furthermore, to address the issue of varying importance of different pose frames in semantic representation, this paper proposes an adaptive frame weight loss function. While maintaining constraints on action-transition frames, this function increases the weight of semantically critical frames, enabling the model to focus more on key actions and thereby further improving the accuracy and expressiveness of the generated poses. Experimental results on the two public datasets, PHOENIX14T and CSL-Daily, demonstrate that SignDict achieves competitive performance across multiple evaluation metrics and reaches state-of-the-art (SOTA) levels, validating the effectiveness of feature-dictionary-based sign language generation methods.

1 INTRODUCTION

According to a World Health Organization report, there are currently more than 430 million people worldwide with varying degrees of hearing loss, including a large number with severe hearing impairment Feng et al. (2025). Hearing loss can hinder effective communication with hearing individuals. Conventional speech-based communication can therefore be challenging for this population. They face long-term challenges such as social barriers, delays in accessing information, and difficulties in social integration, creating significant communication barriers and a social communication divide (Camgoz et al., 2018a; 2020; Gong et al., 2024). As the natural mother tongue and primary means of communication for people with hearing loss, sign language serves as a cru-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

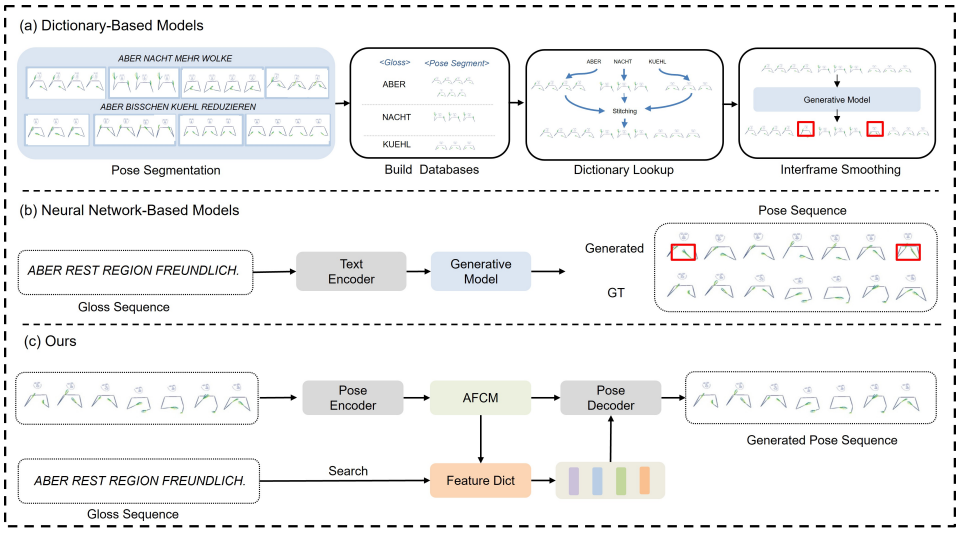

Figure 1: (a) For dictionary-based SLP models, the poses are accurate but the process is cumbersome, the movements are stiff, and the dictionary overhead is high. (b) Neural network-based models produce smooth movements but are prone to pose distortion. (c) The model proposed in this paper utilizes a lightweight feature dictionary to combine the advantages of both approaches, ensuring both pose fidelity and temporal smoothness, and provides a new implementation approach for SLP.

cial medium for enabling barrier-free communication and bridging the information gap between the hearing-impaired community and the general population. Therefore, high-precision and natural sign language generation is of significant practical importance. It can promote barrier-free communication and support equal access to communication for people with special needs (Liu et al., 2026).

Sign Language Production (SLP) aims to automatically convert sign language gloss sequences into continuous and semantically appropriate sign language gestures. It has emerged as an active research area at the intersection of computer vision and natural language processing. Figure 1 illustrates the two existing types of sign language production models: dictionary-based models and neural network-based models. **(a) Dictionary-based sign language production (SLP) models.** The implementation process for this type of model primarily consists of four stages (Zuo et al., 2024): First, the pose sequence is segmented according to glosses, a process that typically requires significant manual annotation effort. Second, a sign language dictionary database is constructed, storing corresponding pose segments for each gloss. A single gloss can be associated with one or more pose segments (generally, the number of pose segments generated corresponds to the number of times the gloss appears in the training set), so the dictionary database is typically at least as large as the training set. Taking the PHOENIX14T dataset as an example, the resulting dictionary is approximately 734MB in size. Third, based on the input gloss sequence, the system uses predefined retrieval and filtering algorithms to match corresponding pose segments in the dictionary and perform segment splicing. Fourth, a neural network or interpolation algorithm smooths the stitched pose segments across frames. However, since motion magnitude differs across pose segments, fixed-frame interpolation cannot guarantee globally smooth motion and often results in uneven movement speed. If adaptive frame-count interpolation is used, it is difficult to determine the number of interpolation frames between fragments. The above issues ultimately result in generated pose sequences with unnatural motion and poor continuity. **(b) Neural network-based SLP models.** These models use neural networks as their core: they feed gloss statements into a text encoder to extract language-conditional features, and then a generative model directly outputs continuous, complete sign language pose sequences. The model autonomously learns the mapping from text to human poses, eliminating the need to pre-construct a database of pose fragments and avoiding cumbersome processes such as manual boundary annotation, dictionary construction, and fragment retrieval and splicing. However, this paradigm still has significant limitations. The model must learn the global temporal distribution of poses, making it difficult to capture the complex temporal dynamics of sign language. Generated poses also often suffer from local distortions and pose drift, as

highlighted by the red boxes in the figure. Comparing the two types of models discussed above, we find that dictionary-based models provide strong motion priors and high motion fidelity, but rely on manual segmentation and large-scale pose sequence storage, often resulting in unnatural transitions between action segments. Neural network-based models, on the other hand, offer greater modeling flexibility; however, generating complete pose sequences directly from gloss requires learning complex high-dimensional spatiotemporal distributions, which can easily lead to pose distortion and semantic alignment errors. Thus, improving SLP generation requires preserving reliable action priors while avoiding explicit storage and retrieval of discrete pose fragments. Meanwhile, the complexity of directly generating complete pose sequences should be reduced.

To address these issues, we propose SignDict, a sign language generation framework (Figure 1(C)). SignDict compresses pose fragments from traditional dictionaries into compact pose features, providing a lightweight representation while preserving sign language motion priors. It then uses neural networks to perform temporal generation and feature fusion at the feature level, producing smooth and natural pose sequences. Unlike traditional dictionary-based models, SignDict no longer stores complete, discrete pose fragments for each gloss. Instead, it constructs a compact pose feature dictionary, thereby avoiding the direct storage of large-scale pose fragments and the complex processes of fragment retrieval and concatenation. Taking the PHOENIX14T dataset as an example, a traditional pose segment dictionary occupies approximately 734 MB, whereas SignDict requires only about 550 KB of feature storage space-a reduction in storage size of approximately 1366x (734MB/550KB). At the same time, compared to neural network models that generate complete pose sequences directly from gloss, SignDict leverages the motion priors provided by the pose feature dictionary to transform the inherently complex high-dimensional pose generation problem into a feature-level representation and temporal modeling problem. This reduces the difficulty of the direct learning of the complete pose space by the model and provides clearer motion priors for the generated results.

The main contributions of this paper are as follows:

1. We propose a new paradigm for sign language generation that offers fresh insights into sign language generation tasks.
2. We propose an adaptive feature compression module to compress variable-length pose segments corresponding to the same gloss. It adaptively aggregates segment features based on their temporal characteristics.
3. We propose an adaptive frame weight loss that dynamically weights frames according to their semantic importance. It guides the model to focus on key actions, improving the semantic accuracy and motion quality of generated poses.
4. Experimental results on the PHOENIX14T and CSL-Daily datasets show that SignDict outperforms existing mainstream methods across multiple evaluation metrics, validating its effectiveness in sign language pose generation tasks. Further ablation experiments demonstrate that the proposed adaptive feature compression module and adaptive frame weight loss improve the generative performance of the model. These results confirm the contribution of each module.

2 Related Work

2.1 Sign Language Production

Early sign language generation relied primarily on symbolic representations, linguistic rules, and virtual avatar animations. Marshall & Sáfár (2004) used HamNoSys to describe sign language handshapes, direction, position, and movement, and combined this with the HPSG linguistic framework to implement text-to-sign generation. Subsequently, Elliott et al. (2004) proposed SiGML, which converted HamNoSys into an XML format suitable for computer processing and used avatars to generate sign language animations in real time. Building on this, Elliott et al. (2008) further integrated natural language processing, sign language grammar, and avatar animation to achieve automatic generation from English text to British Sign Language. Lu & Huenerfauth (2010) established the American Sign Language motion capture corpus, which provides a foundation for sign language generation and data-driven methods based on real-world motion data. Subsequently, with the ad-

vancement of deep learning, research gradually shifted toward end-to-end generation based on neural networks. Stoll et al. (2020) introduced Text2Sign, which integrates neural machine translation, motion graphs, and generative models to achieve end-to-end generation of continuous sign language videos from text, laying the foundation for this research direction. To address the prediction drift issue in long-sequence generation, Saunders et al. (2020) proposed the Progressive Transformer, which uses reverse decoding to output variable-length 3D sign language poses and employs data augmentation to mitigate temporal drift. To model the inherent multimodal distribution of sign language movements, Saunders et al. (2021) combines adversarial training with a mixed-density network (MDN) to enhance the naturalness of generated gestures. Hwang et al. (2021) proposes the non-autoregressive model NSLP-G, which models pose distributions in a Gaussian space to enhance movement diversity. To capture the coordinated movement of human joints, Cui et al. introduces a spatio-temporal graph Transformer, which models the sign language human mesh through a joint graph structure. Building on this, Chen et al. (2024) proposes SDD-GPLA, which aligns semantic dependencies in the gloss–pose latent space and uses a semantic-driven diffusion model to generate latent pose features. To address the issues of missing hand details and temporal discontinuity, Ma et al. (2024) designed a cascaded dual-decoder Transformer that models hand and full-body poses separately and introduces a spatiotemporal loss to simultaneously improve hand details and motion continuity.

2.2 VAE Model

The Variational Autoencoder (VAE) was introduced by Kingma & Welling (2013) and is a probabilistic generative model that combines variational inference with the autoencoder architecture. Given an observed sample x, VAE employs an encoder $q_\phi(z|x)$ to approximate the posterior distribution of the latent variable z, and a decoder $p_\theta(x|z)$ to reconstruct the input data from the latent variable. Since the true posterior distribution $p(z|x)$ is generally intractable, VAE optimizes the model by maximizing the Evidence Lower Bound (ELBO):

$$\mathcal{L}(\theta, \phi; x) = \mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(x|z)\right] - D_{\mathrm{KL}} \left(q_\phi(z|x) \,\|\, p(z)\right). \tag{1}$$

The first term represents the reconstruction loss, which encourages the model to accurately reconstruct the input data. The second term is the Kullback–Leibler (KL) divergence regularization term, which constrains the posterior distribution to be close to the predefined prior distribution, thereby encouraging a latent representation with desirable continuity and sampling properties. To enable the stochastic sampling process to participate in gradient backpropagation, VAE adopts the reparameterization trick, expressing the latent variable as:

$$z = \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{2}$$

By transferring the randomness to the parameter-independent noise variable ϵ, VAE enables end-to-end optimization.

3 Model

3.1 Model Architecture

Overall. Figure 2 illustrates the overall architecture of SignDict. The model employs a two-stage training strategy. In the first stage, the length prediction module, PreLong, is pre-trained to predict the length of the corresponding body pose sequence based on sign language glosses, and its parameters are frozen during subsequent VAE training. In the second stage, the VAE is trained: first, the VAE encoder maps the input pose sequence to the latent space; subsequently, the adaptive Feature Compression Module (AFCM) uses the length prior predicted by PreLong to adaptively compress the latent features and obtains sampled latent variables through the VAE's reparameterization technique. Finally, the latent variables are sequentially extended based on the target length predicted by PreLong, and the extended features are fed into the pose decoder to reconstruct the complete body pose sequence.

Length Prediction Module (PreLong). PreLong is a lightweight model designed to predict the lengths of pose segments corresponding to glosses. It consists of an embedding layer, a GRU tem-

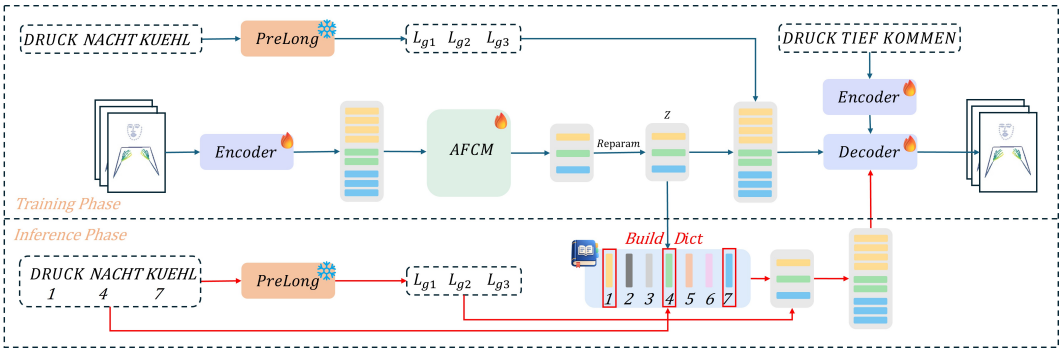

Figure 2: SignDict model overview. The upper section shows the training process. The lower section shows the inference process. Note that the feature dictionary (Dict) needs to be constructed only once. PreLong is a length prediction model used to predict the length of the pose segment corresponding to gloss. AFCM is the adaptive feature compression module.

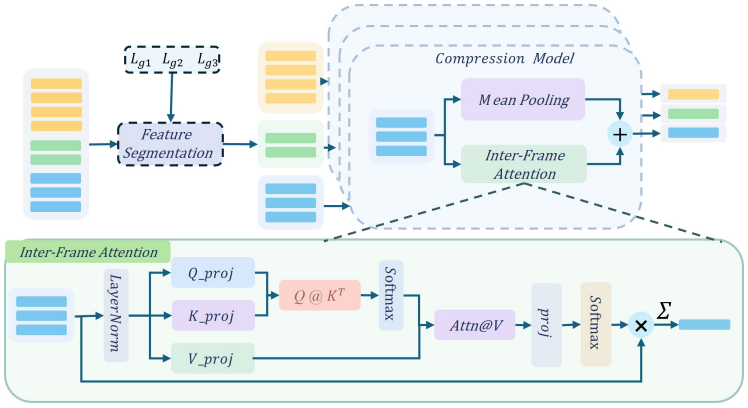

Figure 3: Adaptive Feature Compression Module (AFCM).

poral encoder, and a hierarchical decoding prediction module. The detailed architecture of the model can be found in the supplementary materials (Appendix A). During the training phase, we first use a pre-trained length prediction model (PreLong) to predict the length of the pose segment corresponding to each gloss, yielding $L_g = \{L_{gi}\}$.

Pose Encoder. The pose encoder encodes the pose sequence into continuous feature vectors, where each pose frame outputs a corresponding feature vector. It maps sequential pose data into a latent feature space.

Adaptive Feature Compression Module (AFCM). The architecture of AFCM is illustrated in Figure 3. This module compresses the feature vectors belonging to the same gloss token into a single feature vector. The detailed pipeline is described as follows. First, we partition pose features according to the predictions from the PreLong to obtain pose feature segments corresponding to each gloss token. Then, the feature segment of each gloss is fed into the compression module for feature aggregation. The compression module consists of an average pooling branch and an inter-frame attention branch. We adopt this dual-branch design for the following reasons. Average pooling serves as a deterministic baseline that preserves the global information of pose segments, stabilizes training, and prevents weight collapse in the attention branch. Average pooling computes the mean over the pose feature segment. In contrast, inter-frame attention assigns learnable weights to distinguish key frames and transition frames in sign movements, adaptively emphasizing core pose information. The inter-frame attention module learns weights for each pose vector, multiplies these weights with the corresponding pose vectors, and accumulates the weighted features. Finally, the outputs of the average-pooling and inter-frame-attention branches are summed to obtain the aggregated feature vector for each gloss.

Reparameterization. To enable probabilistic modeling in the latent space, we adopt the reparameterization trick. The features output by AFCM are encoded to produce the mean and variance of the latent distribution. Afterwards, we sample the latent variable z from this Gaussian distribution via the reparameterization trick.

Pose Decoder. The latent variable z is temporally expanded by repeating features according to the segment lengths predicted by PreLong, yielding a feature sequence consistent with the total number of target frames. The expanded feature sequence is then fed into the decoder to reconstruct the final pose sequence.

3.2 Loss Function

Total Loss Function. During training, the model employs three loss terms: the adaptive frame-weighted loss $\mathcal{L}_{\text{weight}}$, the KL divergence loss $\mathcal{L}_{\text{KL}}$, and the motion loss $\mathcal{L}_{\text{move}}$.

Adaptive Frame-Weighted Loss. An independent weight is assigned to each frame in the pose sequence. For each pose segment corresponding to a gloss, the weights are determined according to the relative temporal position of each frame within the segment. Specifically, frames located at the center of a segment, which generally contain the core semantic motion, are assigned higher weights, while transition frames near the boundaries are assigned lower weights. The weight is constrained to the range $[1, 2]$. This weighting strategy strengthens the constraint on the core semantic motions within each sign segment and reduces the influence of reconstruction errors in transitional frames, thereby encouraging the model to focus on semantically important poses.

$$\mathcal{L}_{\text{weight}} = \frac{1}{F} \sum_{t=1}^{F} w_t \left\| P_t - G_t \right\|_1 , \tag{3}$$

where P_t and G_t represent the predicted pose coordinates and the ground truth pose coordinates for frame t, respectively. The frame weight w_t is defined as

$$w_t = 1 + \frac{1}{2} \left(1 + \cos \left(\pi \left| \frac{2t - (L-1)}{L-1} \right| \right) \right) , \tag{4}$$

where L denotes the length of the corresponding pose segment and t represents the temporal position of the current frame within the segment.

KL Divergence Loss. To regularize the latent representation, we minimize the KL divergence between the posterior distribution $q_\phi(z|x)$ and the standard Gaussian prior $\mathcal{N}(0, I)$. This constraint prevents the latent distribution from deviating excessively from the prior and encourages a continuous and structured latent space. Such regularization facilitates the learning of compact and consistent pose representations, which is important for constructing the feature dictionary and generating pose sequences from latent features. The KL divergence loss is defined as

$$\mathcal{L}_{\text{KL}} = -\frac{1}{2D} \sum_{d=1}^{D} \left(1 + \log \sigma_d^2 - \mu_d^2 - \sigma_d^2 \right) , \tag{5}$$

where μ and $\log \sigma^2$ denote the mean and logarithmic variance of the latent distribution, respectively, and D denotes the latent dimension.

Motion Loss. The motion loss is introduced to explicitly supervise the temporal variations of sign language movements. Specifically, the frame-wise motion differences are first calculated for both the predicted pose sequence and the ground-truth pose sequence. The motion loss is then defined as

$$\mathcal{L}_{\text{move}} = \frac{1}{F} \sum_{t=1}^{F-1} \left\| (P_{t+1} - P_t) - (G_{t+1} - G_t) \right\|_1 , \tag{6}$$

where P_t and G_t denote the predicted and ground-truth pose at the t-th frame, respectively. By constraining the difference between the predicted and ground-truth frame-wise motions, the motion loss

explicitly supervises the temporal dynamics of sign language movements, encouraging the generated pose sequence to follow the motion trends of the ground-truth sequence and reducing undesirable jitter, abrupt changes, and discontinuities.

Overall Loss. The overall training objective is formulated as a weighted combination of the adaptive frame-weighted loss, KL divergence loss, and motion loss:

$$\mathcal{L} = \mathcal{L}_{\text{weight}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}} + \mathcal{L}_{\text{move}}. \tag{7}$$

where λ_{KL} is a balancing coefficient that controls the contribution of the KL divergence loss.

Inference Pipeline. After training, we construct the gloss feature dictionary from the training data. Specifically, pose sequences are encoded into feature segments, and features associated with the same gloss are aggregated to obtain a single representative feature. The dictionary is constructed once and remains fixed during inference. Given an input gloss sequence, the corresponding feature vectors are first retrieved from the dictionary. PreLong then predicts the duration of each gloss, and the retrieved features are temporally expanded according to the predicted lengths. Finally, the expanded feature sequence is fed into the pose decoder to generate the final pose sequence.

4 Experiment

4.1 Experimental Settings

Dataset. We evaluate our model on two public sign language datasets, including the German sign language dataset PHOENIX14T (Camgoz et al., 2018b) and the Chinese sign language dataset CSL-Daily (Zhou et al., 2021). The PHOENIX14T dataset comprises 7096 training samples, 519 validation samples, and 642 test samples collected from 9 signers, covering a total of 1066 distinct glosses. The CSL-Daily dataset contains 18401 training samples, 1077 validation videos, and 1176 test videos from 10 signers, with 2000 unique glosses and paired sentence-level annotations and translation labels. For pose estimation, we adopt HRNet (Yu et al., 2021) to extract 2D keypoints for each frame. The extracted keypoints consist of 42 hand keypoints (21 keypoints for each hand), 68 facial keypoints, and 6 upper-body keypoints.

Evaluation. We employ a pre-trained Sign Language Translation (SLT) model (Gan et al., 2026) as a back-translation evaluator to assess the semantic fidelity of the generated sign language pose sequences. Specifically, the generated pose sequences are first fed into the pre-trained SLT model to obtain the corresponding translated text sequences, which are then compared with the reference text sequences. We adopt BLEU, ROUGE, and Word Error Rate (WER) as text-based evaluation metrics. BLEU measures the n-gram precision between the generated and reference sequences, ROUGE primarily measures n-gram recall, and WER measures the word-level edit distance between the two sequences. These metrics evaluate how well the generated sign language preserves the linguistic content of the input. MPJPE (Mean Per-Joint Position Error) is used to quantify the spatial deviation between the predicted sign language pose and the actual pose, thereby evaluating the geometric accuracy of sign language motion generation at the joint-level.

Implementation Details. We train the SignDict model using the Adam optimizer with an initial learning rate of 5×10^{-4}, which is decayed by a factor of 0.8 every 10 epochs, and a batch size of 16. The encoder and decoder are both built upon Transformer blocks with 8-head multi-head self-attention, a hidden dimension of 512, and a feed-forward dimension of 1024. The latent dimension is set to 64. Specifically, the encoder contains 2 Transformer blocks, and the decoder contains 6 blocks. The KL loss weight λ_{KL} is set to 0.001. All experiments are implemented based on PyTorch and conducted on an NVIDIA GeForce RTX 5070 Ti GPU.

4.2 Comparison with State-of-the-Arts

Table 1 presents a quantitative comparison between SignDict and existing state-of-the-art (SOTA) methods on the PHOENIX14T dataset. SignDict achieved the best results on all seven metrics across both the Dev and Test sets. These results demonstrate that SignDict can simultaneously improve the correspondence between generated sign language and target semantics as well as the overall quality of motion sequences. The performance improvements are primarily attributed to three factors:

Table 1: Performance Comparison of PHOENIX14T. † indicates that we reproduced the results using the code provided by the authors and conducted tests using the same dataset and metrics.

Model	Dev							Test						
	BLEU1↑	BLEU2↑	BLEU3↑	BLEU4↑	ROUGE↑	WER↓	MPJPE↓	BLEU1↑	BLEU2↑	BLEU3↑	BLEU4↑	ROUGE↑	WER↓	MPJPE↓
NAT-EA (Huang et al., 2021)	–	–	–	–	–	–	–	15.12	10.45	7.99	6.66	19.43	82.01	–
GEN-OBT (Tang et al., 2022)	24.92	15.72	11.20	8.68	25.21	82.36	41.47	23.08	14.91	10.48	8.01	23.49	81.78	52.9
DET Viegas et al. (2023)	17.25	10.17	7.04	5.32	12.85	–	–	12.18	10.39	7.39	5.76	17.64	–	–
T2S-GPT (Yin et al., 2024)	–	–	–	–	–	–	–	33.16	21.09	15.26	11.87	34.65	–	–
GCDM Tang et al. (2025c)	22.88	14.28	10.01	7.64	23.35	82.81	–	22.03	14.21	10.16	7.91	23.20	81.94	–
Sign-D2C (Tang et al., 2025a)	22.46	–	–	8.06	22.07	80.73	–	22.36	–	–	8.73	22.36	79.06	–
PT † (Saunders et al., 2020)	15.61	4.31	2.20	1.43	14.77	98.98	34.55	16.57	4.92	2.66	1.80	15.62	98.78	34.53
PT-GN † (Saunders et al., 2020)	15.86	4.37	2.24	1.45	14.92	98.98	34.53	16.62	5.12	2.79	1.88	15.65	98.78	34.53
FGDM † (Yu et al., 2026)	22.08	10.53	6.96	5.30	22.89	91.54	32.27	22.30	10.66	7.10	5.41	22.71	92.49	32.03
Sign-IDD † (Tang et al., 2025b)	26.91	15.00	10.09	7.60	27.70	84.07	21.34	26.84	15.09	10.04	7.46	26.95	84.87	21.57
SignDict	**33.83**	**21.78**	**15.73**	**12.31**	**35.59**	**67.66**	**19.08**	**36.03**	**23.65**	**17.02**	**13.16**	**36.34**	**66.46**	**19.11**

Table 2: Performance Comparison of CSL-Daily. † indicates that we reproduced the results using the code provided by the authors and conducted tests using the same dataset and metrics.

Model	Dev							Test						
	BLEU1↑	BLEU2↑	BLEU3↑	BLEU4↑	ROUGE↑	WER↓	MPJPE↓	BLEU1↑	BLEU2↑	BLEU3↑	BLEU4↑	ROUGE↑	WER↓	MPJPE↓
PT † (Saunders et al., 2020)	11.46	1.59	0.32	0.11	10.03	98.99	27.27	11.81	1.82	0.51	0.17	10.10	99.07	27.38
PT-GN † (Saunders et al., 2020)	11.57	1.60	0.32	0.10	9.99	98.99	27.20	11.75	1.76	0.50	0.17	10.11	99.07	27.29
FGDM † (Yu et al., 2026)	26.25	14.62	8.98	5.79	24.20	79.19	22.69	25.82	14.18	8.57	5.57	23.93	79.04	22.84
Sign-IDD † (Tang et al., 2025b)	28.62	16.06	9.80	6.28	24.65	74.02	19.25	29.39	16.65	10.47	6.88	25.23	73.72	19.17
SignDict	**34.41**	**21.61**	**14.62**	**10.39**	**31.11**	**61.50**	**16.43**	**34.92**	**22.17**	**15.08**	**10.83**	**31.38**	**59.99**	**16.61**

First, the feature dictionary provides a clear motion prior for the generation of poses from vocabulary, thereby reducing ambiguity in the semantic mapping. Second, AFCM performs adaptive feature compression on pose sequences, reducing redundancy while preserving the core motion patterns corresponding to the gloss. Finally, the adaptive frame weight loss strengthens reconstruction constraints for key action frames, minimizing the interference of low-information transition frames during training. The synergy of these three approaches improves the semantic consistency of the model and the quality of motion generation, enabling SignDict to achieve comprehensive and stable performance improvements on PHOENIX14T. Table 2 shows the comparison results on the CSL-Daily dataset, where SignDict also achieved the best results.

Figure 4 compares the 2D skeleton keypoint sequences generated by GT, the Sign-IDD baseline, and SignDict using the PHOENIX14T German Sign Language weather-forecast domain. The results generated by Sign-IDD exhibit distortions in hand poses and spatial offsets. SignDict better reproduces the fine morphology and temporal motion of hand joints, and the generated skeletons are closer to the ground truth in terms of relative limb positions and motion continuity.

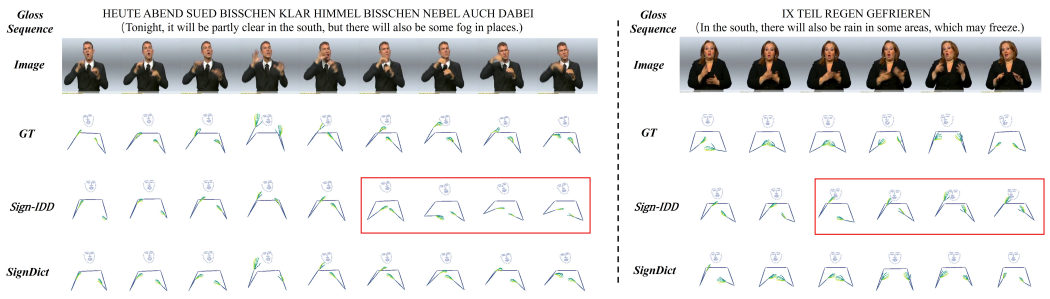

Figure 4: Qualitative comparison of generation quality on PHOENIX14T with previous models. We compare SignDict with Sign-IDD, the state-of-the-art model among publicly available models, and ground truth (GT) poses.

4.3 Ablation Experiments

To validate the effectiveness of the proposed method, we conduct ablation experiments on the VAE architecture, Adaptive Feature Compression Module (AFCM), and adaptive frame weight loss, with the results presented in Table 3. First, we evaluate the necessity of the VAE-based latent modeling framework. Compared with the standard autoencoder (AE), VAE introduces KL divergence to regularize the latent space toward a standard Gaussian distribution. This regularization produces

Table 3: Ablation experiments on the PHOENIX14T dataset.(AE and VAE incorporate feature compression using average pooling.

Model	Dev							Test						
	BLEU1↑	BLEU2↑	BLEU3↑	BLEU4↑	ROUGE↑	WER↓	MPJPE↓	BLEU1↑	BLEU2↑	BLEU3↑	BLEU4↑	ROUGE↑	WER↓	MPJPE↓
AE	23.28	12.22	8.22	6.25	24.31	87.03	23.98	23.14	11.96	7.95	6.05	23.41	87.75	24.55
VAE	30.36	18.73	13.48	10.66	32.58	74.14	20.13	31.89	20.09	14.26	11.15	33.27	72.98	20.05
VAE+AFCM	**33.99**	**21.81**	15.31	12.06	**35.60**	68.89	19.88	34.28	22.31	16.25	12.71	35.29	67.54	19.32
VAE+AFCM+L_{weight}	33.83	21.78	**15.73**	**12.31**	35.59	**67.66**	**19.08**	**36.03**	**23.65**	**17.02**	**13.16**	**36.34**	**66.46**	**19.11**

a smoother and more structured latent space, reducing the risk of overfitting and improving the generalization of pose generation. Second, we investigate the contribution of AFCM. Unlike static strategies such as fixed pooling or direct truncation, AFCM adaptively aggregates frame-level pose features according to the duration of each gloss segment. By assigning greater importance to informative motion frames while reducing the influence of redundant and transitional frames, AFCM produces compact and discriminative gloss-level representations, facilitating the alignment between semantic tokens and continuous pose sequences. Finally, we evaluate the adaptive frame weight loss, which further improves temporal modeling during pose reconstruction. By assigning different weights according to the motion contribution of each frame, the loss encourages the model to focus on informative gesture movements while reducing the influence of less informative transitions. This improves both the temporal consistency and naturalness of the generated poses. Overall, the three components play complementary roles: VAE provides a structured latent space, AFCM learns compact gloss-level motion representations, and the adaptive frame weight loss enhances temporal modeling. Their combination consistently improves the overall performance of SignDict, demonstrating the effectiveness of the proposed design.

Table 4: Ablation study on the weight of KL loss λ_{KL} for SignDict.

λ_{KL}	Dev							Test						
	BLEU1↑	BLEU2↑	BLEU3↑	BLEU4↑	ROUGE↑	WER↓	MPJPE↓	BLEU1↑	BLEU2↑	BLEU3↑	BLEU4↑	ROUGE↑	WER↓	MPJPE↓
0.01	33.71	21.02	14.32	10.75	32.76	74.20	20.54	34.42	21.97	14.31	10.83	31.49	72.25	20. 48
0.001	**33.83**	**21.78**	**15.73**	**12.31**	**35.59**	**67.66**	**19.08**	**36.03**	**23.65**	**17.02**	**13.16**	**36.34**	**66.46**	**19.11**
0.0001	30.12	19.45	13.24	9.52	29.81	83.14	21.35	30.78	19.96	13.48	9.91	29.75	81.48	21.31

4.4 Hyperparameter Settings

To explore the effect of the KL divergence regularization weight on the performance of SignDict for sign language generation, we conduct ablation experiments with $\lambda_{\text{KL}} \in \{0.01, 0.001, 0.0001\}$. When $\lambda_{\text{KL}} = 0.01$, the KL loss accounts for a relatively large proportion of the total loss, as shown in Table 4. The model optimization is biased toward constraining the latent distribution, which squeezes the optimization space of the pose sequence reconstruction task. The excessively strong regularization on the latent space degrades the semantic alignment between sign poses and text, leading to lower BLEU and ROUGE scores as well as higher WER. When the weight is reduced to $\lambda_{\text{KL}} = 0.0001$, the regularization term is too weak to effectively mitigate latent space collapse, the latent variable distribution tends to degenerate, and the semantic consistency of the generated sign sequences deteriorates, resulting in decreased metrics. In contrast, $\boldsymbol{\lambda_{\text{KL}} = 0.001}$ achieves the best results, yielding the highest BLEU, ROUGE and the lowest WER on both the Dev and Test sets. This demonstrates that a moderate KL weight imposes effective regularization on the latent space without harming the learning of pose reconstruction, which verifies the rationality of our selected $\lambda_{\text{KL}} = 0.001$.

5 Conclusion

Combining the advantages of dictionary-based and neural network-based models, this paper proposes SignDit. This model utilizes a VAE to learn compact latent representations of pose and employs an adaptive feature compression module to compress variable-length pose features into a single feature vector, thereby constructing a lightweight pose feature dictionary. Additionally, an adaptive frame weight loss function is designed to enhance the learning of key semantic frames and motion changes. Experimental results on the PHOENIX14T and CSL-Daily datasets demonstrate that SignDict achieves superior generation performance compared to existing models while reducing dictionary storage overhead. Ablation experiments further validate the effectiveness of the proposed modules.

AI USE STATEMENT

In this work, we used generative AI tools for English language polishing, grammatical error correction, sentence structure optimization, and literature expression refinement. We have not used generative AI tools for research idea design, core algorithm design, experimental data generation, result analysis, and manuscript key conclusion writing, and automated generation of research figures are not applicable to this work. Additionally, we used generative AI tools for improving the readability and fluency of the full manuscript text and standardizing academic writing formats. We have thoroughly reviewed and revised all AI-assisted work in this paper. Specifically, all AI-polished text was manually checked word-by-word by the authors to eliminate logical confusion and inconsistent academic expressions; all AI-optimized sentences were verified against the core research content and experimental conclusions to ensure no distortion of research facts; we further conducted manual literature cross-checks to confirm that AI-assisted writing sections contain no plagiarized content and conform to academic writing norms. All authors of this paper have independently verified the rationality, accuracy and originality of the entire manuscript. We take full and ultimate responsibility for the final content of this work, including all text, research claims, experimental descriptions, and academic artifacts produced with the aid of generative AI. This disclosure complies with the ICLR 2027 AI Policy for Authors.

ETHICS STATEMENT

This study strictly adheres to the ICLR Code of Ethics and standard academic research integrity principles. No human subjects, private personal data, or animal experiments were involved in this work. All datasets adopted in this research are publicly available, legally accessible, and have been previously used in peer-reviewed academic studies, with legitimate open-source usage licenses and complete attribution to original sources. The proposed methodologies, experimental analyses, and research findings of this paper pose no potential safety risks, discriminatory biases, unfair implications, or harmful societal impacts. There are no conflicts of interest, commercial sponsorships, or financial benefits that could influence the research design, experimental results, or manuscript conclusions. All research processes comply with academic norms and legal compliance requirements. The authors ensure full transparency, authenticity, and reproducibility of the experimental settings, data processing, and result analysis. No academic misconduct, data falsification, or improper manipulation exists in this work.

REPRODUCIBILITY STATEMENT

We have made comprehensive efforts to ensure the full reproducibility of all experimental results and theoretical findings presented in this work. All model architectures, training configurations, hyperparameter settings, and evaluation metrics are elaborated in detail in the main paper. Complete data collection criteria, standardized preprocessing pipelines, and dataset partitioning strategies are documented in the supplementary materials to facilitate consistent data preparation and usage. For the proposed algorithm and experimental implementations, we will release the complete anonymous source code and running scripts upon paper acceptance, including all core modules, training and testing pipelines, and result evaluation programs. All theoretical assumptions, derivation details, and rigorous proofs corresponding to our technical claims are fully supplemented in the appendix of this paper. All experimental settings follow standard academic paradigms, and all variable conditions and baseline comparison schemes are clearly specified to eliminate ambiguous operations. With the provided manuscript content, appendix descriptions, and to-be-released supplementary resources, researchers can completely replicate our experimental procedures, reproduce all quantitative and qualitative results, and verify the validity and robustness of the proposed method.

REFERENCES

Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. Neural sign language translation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7784–7793. Ieee, 2018a.

Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. Neural sign language translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7784–7793, 2018b.

Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. 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*, pp. 10023–10033, 2020.

Sheng Chen, Qingshan Wang, and Qi Wang. Semantic-driven diffusion for sign language production with gloss-pose latent spaces alignment. *Computer Vision and Image Understanding*, 246: 104050, 2024.

Z Cui, Z Chen, Z Li, and Z Wang. Spatial-temporal graph transformer with sign mesh regression for skinned-based sign language production. ieee access 10, 127530–127539 (2022).

Ralph Elliott, John Glauert, Vince Jennings, and Richard Kennaway. An overview of the sigml notation and sigmlsigning software system. *sign-lang@ LREC 2004*, pp. 98–104, 2004.

Ralph Elliott, John RW Glauert, JR Kennaway, Ian Marshall, and Eva Safar. Linguistic modelling and language-processing technologies for avatar-based sign language presentation. *Universal access in the information society*, 6(4):375–391, 2008.

Liqian Feng, Lintao Wang, Kun Hu, Dehui Kong, and Zhiyong Wang. Text2sign diffusion: A generative approach for gloss-free sign language production. In *2025 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–8. IEEE, 2025.

Shiwei Gan, Yafeng Yin, Zhiwei Jiang, Lei Xie, Sanglu Lu, and Hongkai Wen. Mixsigngraph: A sign sequence is worth mixed graphs of nodes. *Advances in Neural Information Processing Systems*, 38:139371–139395, 2026.

Jia Gong, Lin Geng Foo, Yixuan He, Hossein Rahmani, and Jun Liu. Llms are good sign language translators. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18362–18372. IEEE, 2024.

Wencan Huang, Wenwen Pan, Zhou Zhao, and Qi Tian. Towards fast and high-quality sign language production. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 3172–3181, 2021.

Eui Jun Hwang, Jung-Ho Kim, and Jong C Park. Non-autoregressive sign language production with gaussian space. In *BMVC*, volume 1, pp. 3, 2021.

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

Xiao Liu, Shiwei Gan, Yafeng Yin, Jiaxin Yin, Bowen Guo, Yaqi Sun, Zhiwei Jiang, and Lei Xie. Bridging the gap between semantics and reconstruction: Unifying sign language translation and production. *arXiv preprint arXiv:2608.09045*, 2026.

Pengfei Lu and Matt Huenerfauth. Collecting a motion-capture corpus of american sign language for data-driven generation research. In *Proceedings of the NAACL HLT 2010 workshop on speech and language processing for assistive technologies*, pp. 89–97, 2010.

Xiaohan Ma, Rize Jin, Jianming Wang, and Tae-Sun Chung. Attentional bias for hands: Cascade dual-decoder transformer for sign language production. *IET Computer Vision*, 18(5):696–708, 2024.

Ian Marshall and Éva Sáfár. Sign language generation in an ale hpsg. In *Proceedings of the International Conference on Head-Driven Phrase Structure Grammar*, pp. 189–201, 2004.

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

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

Stephanie Stoll, Necati Cihan Camgoz, Simon Hadfield, and Richard Bowden. Text2sign: towards sign language production using neural machine translation and generative adversarial networks. *International Journal of Computer Vision*, 128(4):891–908, 2020.

Shengeng Tang, Richang Hong, Dan Guo, and Meng Wang. Gloss semantic-enhanced network with online back-translation for sign language production. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 5630–5638, 2022.

Shengeng Tang, Jiayi He, Lechao Cheng, Jingjing Wu, Dan Guo, and Richang Hong. Discrete to continuous: Generating smooth transition poses from sign language observations. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3481–3491. IEEE, 2025a.

Shengeng Tang, Jiayi He, Dan Guo, Yanyan Wei, Feng Li, and Richang Hong. Sign-idd: Iconicity disentangled diffusion for sign language production. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7266–7274, 2025b.

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

Carla Viegas, Mert Inan, Lorna Quandt, and Malihe Alikhani. Including facial expressions in contextual embeddings for sign language generation. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (* SEM 2023)*, pp. 1–10, 2023.

Aoxiong Yin, Haoyuan Li, Kai Shen, Siliang Tang, and Yueting Zhuang. T2s-gpt: Dynamic vector quantization for autoregressive sign language production from text. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3345–3356, 2024.

Changqian Yu, Bin Xiao, Changxin Gao, Lu Yuan, Lei Zhang, Nong Sang, and Jingdong Wang. Lite-hrnet: A lightweight high-resolution network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10440–10450, 2021.

Yiheng Yu, Sheng Liu, Yuan Feng, Zhelun Jin, Yining Jiang, and Min Xu. Focal-general diffusion model with semantic consistent guidance for sign language production. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35915–35925, 2026.

Hao Zhou, Wengang Zhou, Weizhen Qi, Junfu Pu, and Houqiang Li. Improving sign language translation with monolingual data by sign back-translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1316–1325, 2021.

Ronglai Zuo, Fangyun Wei, Zenggui Chen, Brian Mak, Jiaolong Yang, and Xin Tong. A simple baseline for spoken language to sign language translation with 3d avatars. In *European Conference on Computer Vision*, pp. 36–54. Springer, 2024.

A LENGTH PREDICTION MODEL(PRELONG)

Figure 5 shows the detailed architecture of the PreGlossLengthModel. This length prediction model first maps a sequence of sign language words (glosses) to low-dimensional dense vectors via an embedding layer; it then uses a unidirectional GRU to encode the temporal features of the sequence, modeling the contextual dependencies of the gloss sequence. Finally, a multi-layer MLP predicts the corresponding number of frames for each gloss, and the ReLU activation function is used to ensure that the predicted values are non-negative.

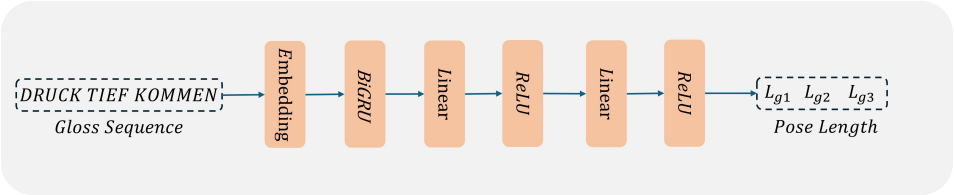

Figure 5: Architecture of a Length Prediction Model.

Loss Function To predict the duration (number of frames) of each gloss, we design a multi-constraint loss function. The total loss is composed of four L1 loss terms: mask loss $\mathcal{L}_{\text{mask}}$, global frame-matching loss $\mathcal{L}_{\text{real}}$, minimum average length constraint $\mathcal{L}_{\text{nonzero}}$, and variance constraint $\mathcal{L}_{\text{var}}$.

$$\mathcal{L}_{\text{total}} = \lambda_{\text{mask}}\mathcal{L}_{\text{mask}} + \lambda_{\text{real}}\mathcal{L}_{\text{real}} + \lambda_{\text{nonzero}}\mathcal{L}_{\text{nonzero}} + \lambda_{\text{var}}\mathcal{L}_{\text{var}}. \tag{8}$$

where $\lambda_{\text{mask}} = 1.0$, $\lambda_{\text{real}} = 1.0$, $\lambda_{\text{nonzero}} = 0.2$, and $\lambda_{\text{var}} = 0.1$.

The mask loss $\mathcal{L}_{\text{mask}}$ enforces the predicted lengths of padding tokens to approach zero. No mask multiplication is applied in the forward pass; this loss suppresses invalid frame predictions at padding positions:

$$\mathcal{L}_{\text{mask}} = \text{L1}\left(\sum_i (1 - \boldsymbol{m}_i)\hat{l}_i, \mathbf{0}\right). \tag{9}$$

Here, $\boldsymbol{m}_i$ denotes the gloss mask where valid glosses are marked as 1 and padding positions as 0; $\hat{l}_i$ is the predicted frame count of the i-th gloss.

The global frame-matching loss $\mathcal{L}_{\text{real}}$ is the primary objective. It constrains the sum of predicted frames over valid glosses to match the ground-truth total frames of the sign skeleton sequence:

$$\mathcal{L}_{\text{real}} = \text{L1}\left(\sum_i \boldsymbol{m}_i\hat{l}_i, L_{\text{gt}}\right), \tag{10}$$

where L_{gt} is the ground-truth total frame number of the sample.

To prevent the model from collapsing to zero predictions for valid glosses, we introduce a one-sided constraint loss $\mathcal{L}_{\text{nonzero}}$. Penalty is activated **only** when the average length of valid glosses is smaller than the threshold $\tau_1 = 0.5$:

$$\bar{\hat{l}} = \frac{\sum_i \boldsymbol{m}_i\hat{l}_i}{\sum_i \boldsymbol{m}_i}, \quad \mathcal{L}_{\text{nonzero}} = \text{L1}\big(\max(\tau_1 - \bar{\hat{l}}, 0), 0\big). \tag{11}$$

We further adopt a variance constraint loss $\mathcal{L}_{\text{var}}$ to mitigate the uniform-averaging prediction issue. Penalty is imposed only if the variance of valid gloss lengths is below threshold $\tau_2 = 0.1$, encouraging differentiated frame allocation for different sign words:

$$\sigma^2(\hat{l}) = \text{Var}\big(\{\hat{l}_i \mid \boldsymbol{m}_i = 1\}\big), \quad \mathcal{L}_{\text{var}} = \text{L1}\big(\max(\tau_2 - \sigma^2(\hat{l}), 0), 0\big). \tag{12}$$

Gradient clipping is used during training to limit the gradient norm within 1.0 and stabilize optimization. All loss terms adopt L1 loss for robustness against outlier predictions. The variance is computed with `unbiased=False`, which is suitable for sentences with a small number of glosses. Hyperparameters $\tau_1 = 0.5$ and $\tau_2 = 0.1$ are tuned and verified via ablation studies.

B User Study

In addition to objective evaluation metrics, we conducted user research to assess the quality of sign language gestures generated by different methods from the perspective of subjective perception. We invited 10 researchers with backgrounds in sign language research to participate in the experiment and asked them to rate the generated sign language gestures. For each test sample, we provided both the ground truth and sign language gestures generated by different methods. We then presented all

gestures to the participants in a randomized order to avoid any potential influence on the evaluation results from the model's identity or the presentation sequence. Participants rated the semantic consistency between the generated sign language gestures and the input text, as well as the overall quality of the gestures, on a 1–10 scale. Higher scores indicate better semantic alignment with the input text and greater comprehensibility. As shown in Figure 6, participants independently scored 15 test samples; therefore, this user study not only calculated the overall average score but also retained the specific scores assigned by participants for each test sample. The Ground Truth received the highest average score of 8.6, indicating that real sign language gestures possess high semantic expression quality. The method proposed in this paper achieved a score of 7.8, the highest average score among all generative models, significantly higher than Sign-IDD (6.46), and PT (2.12). In terms of the distribution of individual sample scores, the method described in this paper received high and relatively stable evaluations on most test samples, indicating that the generated gestures can more accurately convey the semantic information contained in the input text.

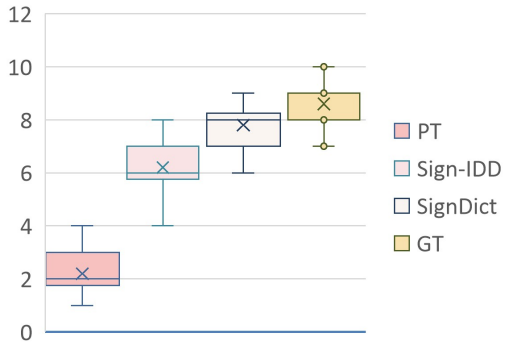

Figure 6: Comparison of scores for different sign language production models.

C Effect of Model Depth.

We investigate the depth of the gloss encoder and pose decoder. Since the gloss encoder and pose decoder constitute the full inference pipeline of our model, we first conduct ablation experiments on these two modules. Table 5 presents the performance comparison with different numbers of layers for the pose decoder, and Table 6 reports the results when varying the depth of the gloss encoder. As shown in Table 5, the pose decoder with 6 layers achieves the best BLEU and ROUGE scores as well as the lowest WER on both the development and test sets. Increasing the layers from 4 to 6 significantly improves representation capacity and generation quality. However, further stacking layers to 8 leads to a clear performance drop and higher WER. This suggests that excessive network depth may introduce unnecessary parameters and make optimization more difficult, which can negatively affect sequence generation. Similar trends can be observed for the gloss encoder in Table 6. The 2-layer gloss encoder yields optimal results. Raising the layer count from 1 to 2 strengthens the encoder's ability to model the gloss symbol sequence and improves generation metrics. When the depth increases to 3, performance degrades again. This suggests that deeper networks are not always beneficial for the sign generation task. Over-stacked layers bring redundant features and hinder effective temporal feature extraction.

Table 5: The impact of pose decoders with different numbers of layers on performance.

Model	Dev							Test						
	BLEU1↑	BLEU2↑	BLEU3↑	BLEU4↑	ROUGE↑	WER↓	MPJPE↓	BLEU1↑	BLEU2↑	BLEU3↑	BLEU4↑	ROUGE↑	WER↓	MPJPE↓
4	31.28	19.22	14.33	11.75	32.25	73.14	19.68	30.46	18.80	13.92	11.65	31.53	74.50	19.85
6	**33.83**	**21.78**	**15.73**	**12.31**	**35.59**	**67.66**	**19.08**	**36.03**	**23.65**	**17.02**	**13.16**	**36.34**	**66.46**	**19.11**
8	33.39	20.94	15.33	12.22	33.42	73.10	19.23	33.17	21.05	15.06	12.76	33.60	73.07	19.19

Table 6: The impact of different numbers of layers in the gloss encoder on performance.

Model	Dev							Test						
	BLEU1↑	BLEU2↑	BLEU3↑	BLEU4↑	ROUGE↑	WER↓	MPJPE↓	BLEU1↑	BLEU2↑	BLEU3↑	BLEU4↑	ROUGE↑	WER↓	MPJPE↓
1	31.36	19.93	13.88	10.27	31.07	76.70	19.65	30.51	20.87	14.43	10.63	31.65	75.60	19.79
2	**33.83**	**21.78**	**15.73**	**12.31**	**35.59**	**67.66**	**19.08**	**36.03**	**23.65**	**17.02**	**13.16**	**36.34**	**66.46**	**19.11**
3	32.60	20.80	15.30	12.33	34.63	70.66	19.20	33.61	21.77	15.82	12.57	34.61	70.08	19.18

D LIMITATIONS

Although SignDict effectively combines the strengths of these two types of generative models to improve the quality of pose generation, the method still has two major limitations. First, consistent with other dictionary-based models, when a sign language word not included in the input feature dictionary is inputted, the model can only rely on neighboring features to generate the output. Second, the length prediction module lacks ground-truth labels, so its predictions can only achieve approximate accuracy. This paper proposes a new research path for sign language generation, and we look forward to subsequent researchers further refining this approach.