Sign-MExD: An Expert-Infused Diffusion Model for Sign Language Production

Jiayu Shen*, Kalin Stefanov†, Vee Yee Chong*, Lay-Ki Soon*, and KokSheik Wong*
* Monash University Malaysia, Malaysia
E-mail: {jiayu.shen, anthonyalexanderveeyee.chong, soon.layki, wong.koksheik}@monash.edu
† Monash University, Australia
E-mail: kalin.stefanov@monash.edu

2025 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) | 979-8-3315-7206-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/APSIPAASC65261.2025.11249331

***Abstract*—Sign language production (SLP) aims to generate semantically aligned videos from textual statements, where the conversion from textual glosses to sign poses is a crucial step. In the field of SLP, the state-of-the-art is based on diffusion models that contain hundreds of millions of parameters and require numerous denoising steps to generate sign poses. Consequently, a major limitation is their high computational cost, resulting in slow training and inference speeds as the models become large. To address this challenge, we explore a mixture-of-experts (MoE) architecture that can scale model capacity while reducing computational cost for training and inference. We propose Sign-MExD, an expert-infused model for SLP that combines the strengths of the MoE mechanism with a diffusion model. Specifically, Sign-MExD learns to adaptively optimize the computational resources allocated to understanding input gloss sequences and generating the respective sign pose sequences, enabling heterogeneous computation aligned with varying gloss-to-pose complexities. Experiments on the PHOENIX14T and How2Sign datasets demonstrate that Sign-MExD offers competitive performance compared to the state-of-the-art and achieves significantly higher ROUGE scores.**

***Index Terms*—Sign Language Production, Diffusion Model, Mixture-of-Experts**

I. Introduction

Sign languages are natural languages with their own grammatical structures and lexicons, used mainly by Deaf and Hard-of-Hearing communities [1]. Sign language production (SLP) plays a crucial role in bridging the communication gap between Deaf and hearing individuals, promoting inclusion and accessibility. The SLP task is related to areas such as visual understanding [2], [3] and cross-media reasoning [4], [5], [6]. In addition, SLP is the reverse task of sign language recognition (SLR) [7], [8], [9], where SLP converts textual information into a visual representation of sign language. This task requires the model to understand textual semantics and generate matching visual sign representations (i.e., sign poses or videos) based on those semantics.

Glosses are written representations of signs using spoken-language text that preserve the meaning and grammatical structure [10]. As the basic semantic units in sign languages, glosses play a crucial transitional role in SLP. Previous methods first translate spoken language into gloss sequences (T2G) using neural machine translation (NMT) models, then generate sign pose sequences based on gloss sequences (G2P) [11]. Since G2P, the focus of this work, is a cross-media task that involves both textual understanding and visual generation, it is more challenging and decisive for the success of SLP.

Early approaches to SLP focus on animated avatars [12] and statistical machine translation methods [13]. These approaches rely on rule-based lookups of limited predefined phrases, resulting in high development costs. Recently, deep learning models have been used for SLP [14], [15], [16]. Stoll et al. [17] adopted a three-step approach (T2G $\rightarrow$ G2P $\rightarrow$ pose-to-video) to generate sign videos from text, by integrating NMT techniques with a generative adversarial network. Saunders et al. [11] proposed a progressive transformer to learn mappings with an encoder-decoder architecture and generate sign poses in an autoregressive manner.

Despite these innovations, sign languages present unique challenges due to their rich vocabularies, complex grammatical structures, and diverse expressions [1]. To overcome these challenges, diffusion-based solutions such as G2P-DDM [18] and GCDM [19] further advance SLP by generating sign poses under semantic guidance. By gradually denoising input Gaussian noise, these models can precisely depict the spatial positions and movement trajectories of signs, which exhibits great potential for SLP. However, a major limitation is their high computational cost; for example, standard architectures contain hundreds of millions of parameters [20] and require many denoising steps, affecting training and inference speeds.

In parallel, recent research [21], [22] highlights that sparse mixture-of-experts (MoE) techniques excel at effective processing of heterogeneous inputs through specialized expert networks with high computational efficiency and model scalability. Inspired by this, we propose Sign-MExD, a novel approach that integrates a MoE mechanism with a diffusion-based conditional generative model. Sign-MExD employs an expert-choice routing strategy to optimize computational resource allocation that is aligned with varying gloss-to-pose complexities, thus naturally aligning with the diffusion process.

Our work makes the following contributions:

- We introduce Sign-MExD, a sparsely scaled diffusion model that incorporates expert-choice routing for SLP. This novel approach leverages global context at each generation step to achieve heterogeneous computational

*This research is supported, in part, by the Monash Future Centre Funding: Action Lab Malaysia (Grant No. I-M010-STG-000406).

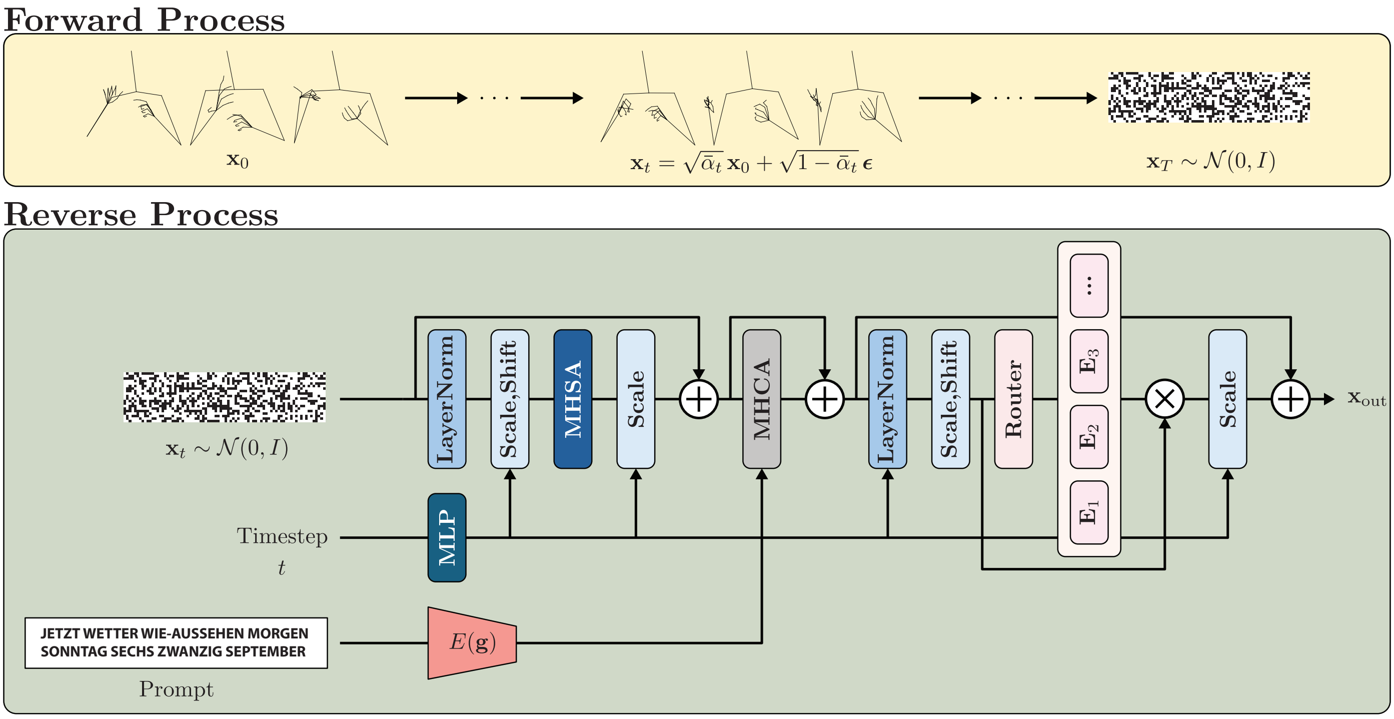
Forward Process
$\mathbf{x}_0$
$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}$
$\mathbf{x}_T \sim \mathcal{N}(0, I)$
Reverse Process
$\mathbf{x}_t \sim \mathcal{N}(0, I)$
LayerNorm
Scale,Shift
MHSA
Scale
MHCA
LayerNorm
Scale,Shift
Router
E1
E2
E3
...
Scale
$\mathbf{x}_{\text{out}}$
Timestep
t
MLP
JETZT WETTER WIE-AUSSEHEN MORGEN SONNTAG SECHS ZWANZIG SEPTEMBER
Prompt
$E(\mathbf{g})$

Fig. 1: **Overview of Sign-MExD.** *MHSA:* Multi-head self-attention; *MHCA:* Multi-head cross-attention; E_n*:* Expert n.

resources allocation, adaptive to different patterns within the generated sign poses;

- We examine Sign-MExD's scalability and effectiveness by configuring it with different model sizes and number of experts. Qualitative results further demonstrate that the model adaptively allocates computational resources based on textual significance, and;
- We conduct experiments on two widely used datasets, PHOENIX14T [11] and How2Sign [23], and illustrate that Sign-MExD significantly enhances pose accuracy, skeletal coherence, and linguistic fidelity while retaining the computational efficiency of MoE architectures.

II. Methodology

Given a gloss sequence, the goal is to generate the corresponding sign pose sequence. The generative process learns motion patterns through a diffusion-based model conditioned on the input gloss sequence. At each step, a transformer encoder is employed to learn the spatial and temporal relationships from the sign pose sequence. To reduce computational complexity while simultaneously enabling high-quality gloss-driven motion generation, we augment the gloss-conditioned diffusion model with expert-choice routing. Finally, a series of poses is generated, optimized by applying joint and bone constraints. Fig. 1 illustrates the method, while the following subsections provide detailed information.

A. Preliminaries

1) Diffusion Model: The pipeline of a diffusion model [24] consists of three processes, namely, a forward process that gradually diffuses noise into samples, a reverse process that optimizes a network to eliminate the above perturbations from noisy samples, and an inference process that utilizes the trained network to iteratively denoise noisy samples.

Specifically, let $\mathbf{x}_0 \in \mathbb{R}^{N\times M}$ represent a sequence of sign poses for N frames, where M is the dimensionality of the sign pose representations. For a timestep $t \sim U[0, T]$ the noisy motion $\mathbf{x}_t$ after t steps of diffusion is obtained by

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}, \tag{1}$$

where $\boldsymbol{\epsilon}$ is a Gaussian noise. Here, $\sqrt{\bar{\alpha}_t}$ and $\sqrt{1-\bar{\alpha}_t}$ are the strengths of the signal and noise, respectively, and α_t decreases as t increases. Specifically, when $\sqrt{\bar{\alpha}_t}$ is sufficiently small, we can approximate $\mathbf{x}_t \sim \mathcal{N}(0, I)$.

Given a motion denoising model $G_\theta(\mathbf{x}_t, t, \mathbf{g})$ to predict the original motion, parameterized by θ, where $\mathbf{g}$ denotes the gloss prompt that guides the denoising process, the optimization can be formulated as follows,

$$\min_\theta \; \mathbb{E}_{t\sim U[0,T],\, \mathbf{x}_0\sim p_{\text{data}}} \left\| G_\theta(\mathbf{x}_t, t, \mathbf{g}) - \mathbf{x}_0 \right\|_2^2 . \tag{2}$$

In the inference process, a trained motion denoising model can progressively generate samples from noise with various samplers. For example, DDPM [24] iteratively denoises the noisy data from t to a previous timestep t'.

2) Mixture-of-Experts: The mixture-of-experts (MoE) [25] consists of a set $\{\mathcal{E}_i(\bar{x})\}_{i=1}^{E}$ of E experts (each of which is often a feedforward network) and a learnable router with weights W_r. For a given token representation $\bar{x}$, the router selects the top-k experts based on the gating value of $\bar{x} \cdot W_r$. The output of the MoE becomes the weighted combination of the selected experts' computations, where the experts weights are the normalized gating values via the softmax distribution. Since the token chooses its best set of experts, this top-k routing is also termed *token-choice routing*.

B. Gloss Conditioning

In gloss-to-pose generation, the model generates a pose sequence $\mathbf{x}_{\text{out}}$ conditioned on the gloss prompt $\mathbf{g}$. We utilize a CLIP-based [26] gloss encoder $E(\mathbf{g})$ to extract embeddings $\bar{g}$. Following the approach in [27], we incorporate a multi-head cross-attention layer (MHCA) [28] between the self-attention and feedforward layers in each transformer block. The cross-attention C_h for head h is defined as

$$C_h = \text{softmax}\left(\frac{Q_{\bar{x}} \cdot K_{\bar{g}}^\top}{\sqrt{d_{\bar{x}}}}\right) \cdot V_{\bar{g}}, \tag{3}$$

with

$$\begin{aligned} Q_{\bar{x}} &= \bar{x} \cdot W_{\bar{x}}^{\text{query}} \\ K_{\bar{g}} &= \bar{g} \cdot W_{\bar{g}}^{\text{key}} \\ V_{\bar{g}} &= \bar{g} \cdot W_{\bar{g}}^{\text{value}}, \end{aligned} \tag{4}$$

where $Q_{\bar{x}}$ is the query matrix for $\bar{x}$, and $K_{\bar{g}}$, $V_{\bar{g}}$ are the key and value matrices for $\bar{g}$, respectively. $W_{\bar{x}}^{\text{query}}$ and $W_{\bar{g}}^{\text{key}}$, $W_{\bar{g}}^{\text{value}}$ are the projection matrices for $\bar{x}$ and $\bar{g}$. Subsequently, the outputs from all H heads are concatenated and transformed by W, i.e., $\text{MHCA}(\bar{x}) = \text{concat}([C_1, \ldots, C_H]) \cdot W$. This process injects textual information into the motion generation.

C. Expert-Choice Routing

Since the majority of computation takes place in the dense feedforward layers [27], replacing those layers with a MoE layer is an efficient method of scaling as it decouples the effective computation from the model capacity. That is, we can effectively scale up the model capacity without a significant decrease in inference speed, since only a selected subset of experts will be activated conditioned on the input representation.

Therefore, we propose to upscale the diffusion model by judiciously exploiting the structure of the diffusion process. Let x_c be the output of the cross-attention module. Each expert is a two-layer feedforward network, where the i-th expert is represented as $\mathcal{E}_i(x_c) = \text{GeLU}(x_c \cdot W_1^i) \cdot W_2^i$. Here, W_1^i and W_2^i are the weight matrices for the i-th expert. For each MoE layer, the router is parameterized by the expert embedding W_r. Given the input x_c, the router first produces a token-expert affinity score tensor A via a softmax along the expert dimension:

$$A_{s,i} = \frac{\exp\left((x_c \cdot W_r)_{s,i}\right)}{\sum\limits_{i=1}^{E} \exp\left((x_c \cdot W_r)_{s,i}\right)}. \tag{5}$$

This affinity score tensor assesses the relevance between each pair of expert and input token. Unlike the token-choice routing where each token in x_c selects the top-k experts from $A_{s,i}$, Sign-MExD works from the expert-choice [29] view where each expert selects the top-C tokens in descending order from $A_{s,i}$. Here, $C = S \times f_c / E$ represents the average capacity of each expert, where f_c denotes the *capacity factor* and reflects the average number of experts assigned to process each token, and S is the sequence length.

Compared to the token-choice routing, which assigns each token independently, Sign-MExD selects the most relevant C tokens from the entire sequence. To achieve this, we compute the gating tensor G as follows

$$G_{s,i} = \begin{cases} A_{s,i} & \text{if } A_{s,i} \in \text{top-}k\left(\{A_{s,i} \mid 1 \leq s \leq S\},\ k = C\right); \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where $G_{s,i}$ is the weight score representing expert i's preference over the s-th token. This process directs the computational focus of the experts towards tokens with significant textual information and pose patterns, while also being adaptive to the denoising steps, resulting in adaptive and efficient computation, concentrating resources where they are most needed.

Finaly, we define a set of indexing vectors $\{\mathcal{I}_i \mid 1 \leq i \leq E\}$ to filter the input tokens allocated to each expert

$$\mathcal{I}_i = \{s \mid G_{s,i} > 0,\ 1 \leq s \leq S\}. \tag{7}$$

The output $\mathbf{x}_{\text{out}}$ is then obtained by combining the results of each expert using the gating tensor G as follows:

$$\mathbf{x}_{\text{out}} = \sum_{i=1}^{E} \left(G_{\mathcal{I}_i, i}\right)^\top \mathcal{E}_i(x_{c_{\mathcal{I}_i,:}}). \tag{8}$$

D. Loss Function

We employ two loss functions to train Sign-MExD. The $\mathcal{L}_{\text{joint}}$ is utilized to enforce the precision of the joints, while the $\mathcal{L}_{\text{bone}}$ serves to constrain the orientation of the bones.

1) Joint Loss: Following [30], [31], we adopt a joint loss for the joint positions, ensuring precise match with the ground truth. The joint loss $\mathcal{L}_{\text{joint}}$ is defined as

$$\mathcal{L}_{\text{joint}} = \frac{1}{J} \sum_{j=1}^{J} \left| x_j - x'_j \right|, \tag{9}$$

where J represents the total number of joints, and x'_j and x_j are the predicted and true joint positions, respectively.

2) Bone Loss: To better depict complex motion details, we introduce $\mathcal{L}_{\text{bone}}$ to improve the accuracy of bone orientations in the generated poses, defined as:

$$\mathcal{L}_{\text{bone}} = \frac{1}{B} \sum_{b=1}^{B} \left(q_b - q'_b\right)^2, \tag{10}$$

where $q_b = (\vec{x}_b, \vec{y}_b, \vec{z}_b)$ and $q'_b = (\vec{x'}_b, \vec{y'}_b, \vec{z'}_b)$ represent the ground truth and predicted bone orientations, respectively, and B represents the number of bones.

The final loss, where the weight λ is 0.1, is defined as

$$\mathcal{L} = \mathcal{L}_{\text{joint}} + \lambda \mathcal{L}_{\text{bone}}. \tag{11}$$

TABLE I: **Model configurations.** For *Total Params*, nE denotes Sign-MExD with n experts in a single MoE layer. *Activate Params* indicates the average number of active parameters per token.

Config	Total Params				Active Params
	DENSE	4E	8E	16E	Sign-MExD
XL	16.94M	29.53M	46.34M	79.95M	16.78M
XXL	23.24M	35.84M	52.65M	86.25M	23.07M
3XL	35.85M	48.45M	65.26M	98.86M	35.65M

III. Experiments

A. Datasets

Following existing literature [11], [31], [32], we evaluated Sign-MExD on two widely used sign language datasets: PHOENIX14T [11] and How2Sign [23]. PHOENIX14T is a German Sign Language (DGS) dataset collected from weather forecasts. It contains 8,247 sentences with a vocabulary of 1,085 signs, divided into 7,096 training instances, 519 development instances, and 642 testing instances. On the other hand, How2Sign is a large-scale multimodal American Sign Language (ASL) dataset with 35,000 high-resolution clips of co-articulated signing and a vocabulary of 16,000 signs. It contains 31,165 training samples, 1,741 development samples, and 2,357 testing samples.

B. Evaluation metrics

Following the standard evaluation protocols in SLP [33], a sign language translation model named NSLT [34] is employed to back-translate sign poses into textual glosses and compare them with the ground truth for calculating the BLEU1-4 and ROUGE scores. Additionally, we adopted Fréchet Inception Distance (FID) to evaluate the overall quality of the generated motions by comparing feature distributions.

C. Implementation Details

For the PHOENIX14T dataset, where keypoints are not provided, we used the model in [11] to extract 2D joints from the original videos and followed the approach in [7] to apply a skeleton model improvement estimation algorithm to convert these 2D joint coordinates into 3D sign poses. In this work, we consider the transformed 3D pose sequences as ground truth.

We evaluated three configurations Sign-MExD: XL, XXL, and 3XL, as detailed in Table I. Each configuration varies the model architecture with 2, 4, and 8 transformer layers, respectively. We maintained a capacity factor $f_c = 2.0$ throughout both training and inference stages for all sparse models. We compared dense models with sparse Sign-MExD implementations at three scaling levels: 4, 8, and 16 experts per MoE layer. Furthermore, we applied Gaussian noise to the pose coordinates, with rate set to 5%. During training, we employed the Adam [35] optimizer with a learning rate of 1×10^{-3}. All experiments were conducted using PyTorch [36].

TABLE II: **Comparison with the state-of-the-art.** Back-translation results on PHOENIX14T and How2Sign datasets.

Method	BLEU-1↑	BLEU-2↑	BLEU-3↑	BLEU-4↑	ROUGE↑
PHOENIX14T [11]					
Ground Truth	18.12	12.77	7.57	5.65	16.08
PTR [11]	9.45	5.93	3.73	2.57	11.00
SIGNGAN [37]	12.64	6.70	4.74	4.63	13.95
G2P-DDM [18]	15.87	**10.07**	6.45	4.94	10.08
Ours	**16.12**	9.26	**6.48**	**4.96**	**14.12**
How2Sign [23]					
Ground Truth	20.63	14.07	9.93	7.62	21.26
PTR [11]	10.30	6.96	4.47	3.39	13.14
SIGNGAN [37]	13.36	6.77	5.11	5.23	14.69
G2P-DDM [18]	**18.06**	10.83	6.63	5.88	13.37
Ours	17.75	**11.32**	**8.37**	**6.65**	**15.81**

IV. Results

A. Back-Translation

We compare the performance of Sign-MExD against three baselines: PTR [11], SIGNGAN [37], and G2P-DDM [18], as detailed in Table II. On PHOENIX14T, Sign-MExD achieves substantial improvements, outperforming PTR by 6.67 points in BLEU-1 and 3.12 points in ROUGE. On How2Sign, it also significantly improves performance, with BLEU-1 increasing from 10.30 to 17.75 and ROUGE from 13.14 to 15.81. These results suggest that Sign-MExD captures a greater extent of the reference sequences than the other methods. This performance gain can be attributed to the integration of a MoE architecture, which dynamically allocates more computational resources to semantically important gloss tokens. Unlike the baseline methods, which treat all tokens uniformly, Sign-MExD emphasizes the linguistically critical components of the input, leading to more coherent and informative outputs.

B. Inference Efficiency

We evaluated the inference efficiency of Sign-MExD on PHOENIX14T, as shown in Fig. 2. Across all model configurations, Sign-MExD consistently improves motion generation quality. As shown in Table I, Sign-MExD reduces the number of active parameters by 0.56% to 0.94% compared to dense counterparts. Nevertheless, it incurs an inference-time overhead of 20% to 28%. This discrepancy likely stems from the computational costs of sign-based operations and specialized matrix computations, which increase runtime despite the parameter savings. Additionally, hardware-level inefficiencies in sparse computations and differences in inference-time parallelism may contribute further to this. However, Sign-MExD reliably improves gloss-to-pose alignment performance while maintaining superior parameter efficiency and keeping inference overhead below 30%.

C. Heterogeneous Compute Allocation

Fig. 3 presents a heatmap illustrating the active expert frequency for each gloss token, capturing the heterogeneous compute allocation learned by Sign-MExD. The model routes

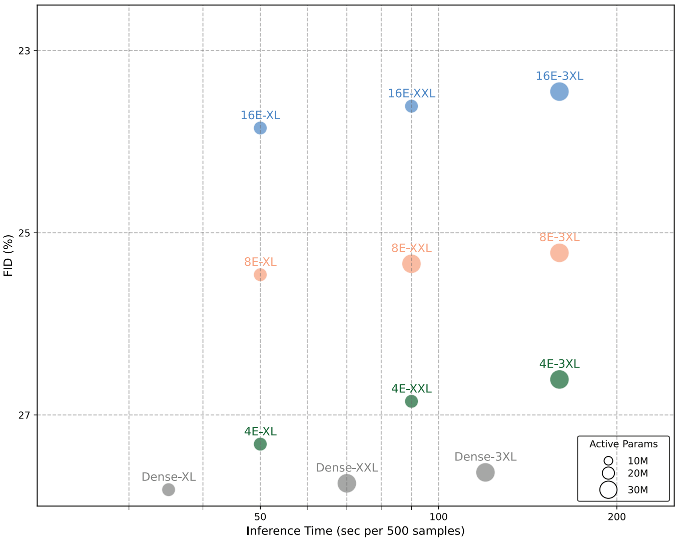

Fig. 2: **Inference time efficiency.** The circle size is proportional to the total active parameters. Inference time represents the time taken to generate 500 samples on 8 A100 GPUs.

different tokens through distinct combinations of experts, revealing specialized computational pathways tailored to linguistic characteristics. For instance, the token SONNE predominantly activates experts 0, 2, 3, and 7, while WETTER shows concentrated activation in experts 2 and 6. These differentiated patterns reflect the model's capacity for adaptive expert selection, whereby each token engages a unique subset of the MoE network. This behavior highlights the model's ability for token-specific, specialized processing —- moving beyond uniform resource allocation to dynamically optimized computation.

D. *Qualitative Evaluations*

To more effectively assess the quality of the generation, we conducted a qualitative comparison of Sign-MExD with other methods on samples from the How2Sign dataset. As shown in Fig. 4, the sign poses generated by Sign-MExD are noticeably superior to PTR, particularly in the movements of the limbs and demonstrate more accurate hand details compared to G2P-DDM. Furthermore, even in cases where the ground truth provides inaccurate poses due to motion blur, Sign-MExD consistently generates clear and precise poses.

V. Conclusions

In this work, we introduced Sign-MExD, a novel sign language production method that combines a conditional generative approach with a mixture-of-experts design. Sign-MExD employs an adaptive expert-choice routing strategy that leverages global pose-level information to optimize computational resource allocation based on pose complexity, thereby enhancing the efficiency of the underlying diffusion process. Experimental results demonstrate that Sign-MExD achieves superior performance compared to existing methods on the PHOENIX14T and How2Sign datasets.

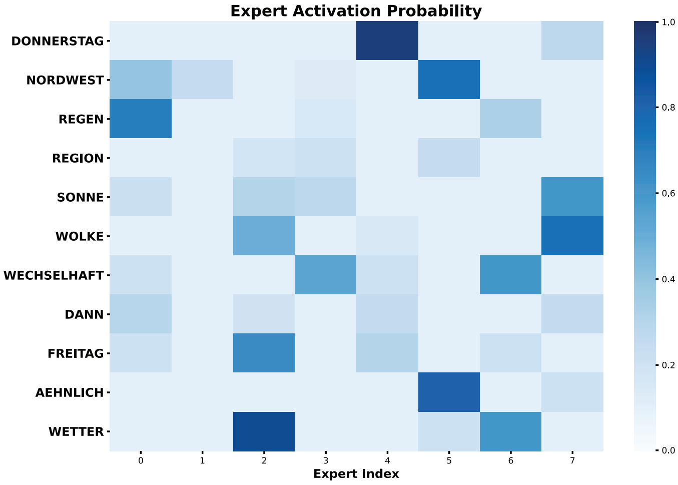

Fig. 3: **Heterogeneous compute allocation.** The model routes different tokens through distinct combinations of experts. *Prompt:* DONNERSTAG NORDWEST REGEN REGION SONNE WOLKE WECHSELHAFT DANN FREITAG AEHNLICH WETTER (am donnerstag regen in der nordhälfte in der südhälfte mal sonne mal wolken ähnliches wetter dann auch am freitag).

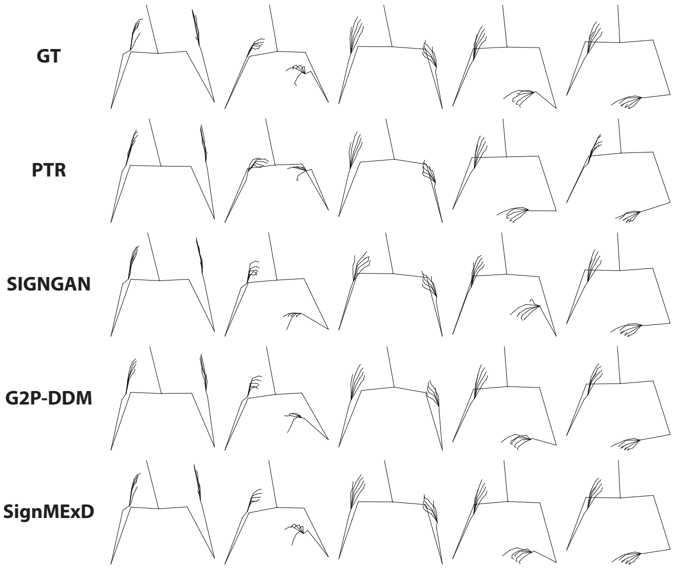

Fig. 4: **Comparison with the state-of-the-art.** Qualitative results on samples form the How2Sign dataset.

Despite these promising results, the current diffusion model lacks explicit temporal modeling, limiting the method's ability to learn the temporal evolution of pose sequences—an aspect that could further enhance generation quality. As a direction for future work, we plan to explore explainable architectures that explicitly capture temporal dependencies to further improve both the interpretability and performance.

References

[1] N. S. Glickman, *Deaf identity development: Construction and validation of a theoretical model*. University of Massachusetts Amherst, 1993.

[2] K. Li, D. Guo, and M. Wang, “Vigt: proposal-free video grounding with a learnable token in the transformer,” *Science China Information Sciences*, vol. 66, no. 10, p. 202102, 2023.

[3] F. Wang, D. Guo, K. Li, and M. Wang, “Eulermormer: Robust eulerian motion magnification via dynamic filtering within transformer,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 6, 2024, pp. 5345–5353.

[4] K. Li, D. Guo, and M. Wang, “Proposal-free video grounding with contextual pyramid network,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 3, 2021, pp. 1902–1910.

[5] P. Song, D. Guo, X. Yang, S. Tang, and M. Wang, “Emotional video captioning with vision-based emotion interpretation network,” *IEEE Transactions on Image Processing*, vol. 33, pp. 1122–1135, 2024.

[6] P. Song, D. Guo, X. Yang, S. Tang, E. Yang, and M. Wang, “Emotion-prior awareness network for emotional video captioning,” in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 589–600.

[7] L. Hu, L. Gao, Z. Liu, and W. Feng, “Continuous sign language recognition with correlation network,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2529–2539.

[8] W. Xue, J. Liu, S. Yan, Y. Zhou, T. Yuan, and Q. Guo, “Alleviating data insufficiency for chinese sign language recognition,” *Visual Intelligence*, vol. 1, no. 1, p. 26, 2023.

[9] R. Zuo, F. Wei, and B. Mak, “Natural language-assisted sign language recognition,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 14 890–14 900.

[10] S. K. Liddell, *Grammar, gesture, and meaning in American Sign Language*. Cambridge University Press, 2003.

[11] B. Saunders, N. C. Camgoz, and R. Bowden, “Progressive transformers for end-to-end sign language production,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*. Springer, 2020, pp. 687–705.

[12] K. Karpouzis, G. Caridakis, S.-E. Fotinea, and E. Efthimiou, “Educational resources and implementation of a greek sign language synthesis architecture,” *Computers & Education*, vol. 49, no. 1, pp. 54–74, 2007.

[13] D. Kouremenos, K. S. Ntalianis, G. Siolas, and A. Stafylopatis, “Statistical machine translation for greek to greek sign language using parallel corpora produced via rule-based machine translation.” in *CIMA@ ICTAI*, 2018, pp. 28–42.

[14] S. Krishna and J. Ukey, “Gan based indian sign language synthesis,” in *Proceedings of the Twelfth Indian Conference on Computer Vision, Graphics and Image Processing*, 2021, pp. 1–8.

[15] Q. Xiao, M. Qin, and Y. Yin, “Skeleton-based chinese sign language recognition and generation for bidirectional communication between deaf and hearing people,” *Neural networks*, vol. 125, pp. 41–55, 2020.

[16] J. Zelinka, J. Kanis, and P. Salajka, “Nn-based czech sign language synthesis,” in *Speech and Computer: 21st International Conference, SPECOM 2019, Istanbul, Turkey, August 20–25, 2019, Proceedings 21*. Springer, 2019, pp. 559–568.

[17] S. Stoll, N. C. Camgoz, S. Hadfield, and R. Bowden, “Text2sign: towards sign language production using neural machine translation and generative adversarial networks,” *International Journal of Computer Vision*, vol. 128, no. 4, pp. 891–908, 2020.

[18] P. Xie, Q. Zhang, P. Taiying, H. Tang, Y. Du, and Z. Li, “G2p-ddm: Generating sign pose sequence from gloss sequence with discrete diffusion model,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 6, 2024, pp. 6234–6242.

[19] S. Tang, F. Xue, J. Wu, S. Wang, and R. Hong, “Gloss-driven conditional diffusion models for sign language production,” *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 21, no. 4, pp. 1–17, 2025.

[20] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, “Diffusion policy: Visuomotor policy learning via action diffusion,” *The International Journal of Robotics Research*, p. 02783649241273668, 2023.

[21] B. Zoph, I. Bello, S. Kumar, N. Du, Y. Huang, J. Dean, N. Shazeer, and W. Fedus, “St-moe: Designing stable and transferable sparse expert models,” *arXiv preprint arXiv:2202.08906*, 2022.

[22] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen, “Gshard: Scaling giant models with conditional computation and automatic sharding,” *arXiv preprint arXiv:2006.16668*, 2020.

[23] A. Duarte, S. Palaskar, L. Ventura, D. Ghadiyaram, K. DeHaan, F. Metze, J. Torres, and X. Giro-i Nieto, “How2sign: a large-scale multimodal dataset for continuous american sign language,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 2735–2744.

[24] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[25] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, “Outrageously large neural networks: The sparsely-gated mixture-of-experts layer,” *arXiv preprint arXiv:1701.06538*, 2017.

[26] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, “Learning transferable visual models from natural language supervision,” in *International conference on machine learning*. PmLR, 2021, pp. 8748–8763.

[27] W. Peebles and S. Xie, “Scalable diffusion models with transformers,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4195–4205.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[29] Y. Zhou, T. Lei, H. Liu, N. Du, Y. Huang, V. Zhao, A. M. Dai, Q. V. Le, J. Laudon *et al.*, “Mixture-of-experts with expert choice routing,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 7103–7114, 2022.

[30] W. Huang, W. Pan, Z. Zhao, and Q. Tian, “Towards fast and high-quality sign language production,” in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 3172–3181.

[31] C. Viegas, M. Inan, L. Quandt, and M. Alikhani, “Including facial expressions in contextual embeddings for sign language generation,” *arXiv preprint arXiv:2202.05383*, 2022.

[32] S. Tang, R. Hong, D. Guo, and M. Wang, “Gloss semantic-enhanced network with online back-translation for sign language production,” in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 5630–5638.

[33] B. Saunders, N. C. Camgoz, and R. Bowden, “Adversarial training for multi-channel sign language production,” *arXiv preprint arXiv:2008.12405*, 2020.

[34] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, “Neural sign language translation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7784–7793.

[35] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[36] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in PyTorch,” 2017.

[37] B. Saunders, N. C. Camgoz, and R. Bowden, “Signing at scale: Learning to co-articulate signs for large-scale photo-realistic sign language production,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5141–5151.