LLM Is a Good Conditioner: End-to-End Sign Language Video Generation with VQ-Diffusion

Xiao Liu
State Key Laboratory of Novel Software Technology
Nanjing University
Suzhou, China
liuxiaox@smail.nju.edu.cn

Shiwei Gan*
State Key Laboratory of Novel Software Technology
Nanjing University
Nanjing, China
sw@nju.edu.cn



Figure 1: Generated sign language videos from our SLVG framework. Given a spoken-language sentence and a single reference frame, the model generates semantically aligned and temporally smooth sign videos without the need for intermediate cues (e.g., pose or hands).

Abstract

Sign Language Video Generation (SLVG) aims to generate realistic and motion-accurate sign language videos from spoken text. Most existing studies heavily rely on intermediate representations (*e.g.*, poses or 3D meshes), and adopt a two-stage generation paradigm, where the text is first converted into poses or other visual modalities, which are then used as conditions to synthesize sign language video frames via diffusion models (*e.g.*, pose-to-image models). However, the end-to-end SLVG task remains largely unexplored. In this paper, we demonstrate that end-to-end SLVG is feasible without relying on intermediate representations (*e.g.*, poses), and we argue that a key challenge lies in how to generate effective and temporally consistent conditions. With well-designed condition modules, it becomes feasible to build an end-to-end SLVG model without relying on any intermediate modalities. To this end, we propose LLMCond, a first end-to-end text-to-video SLVG framework that requires no auxiliary visual cues during either training or inference. LLMCond consists of four key components: (1) a video VQ-VAE that compresses sign language videos into a sequence of discrete latent tokens; (2) **an LLM-based conditioner** that generates temporally and semantically aligned conditions corresponding to the sign sequences; (3) **a Gaussian Condition Refiner** designed to enforce temporal smoothness across conditions, enabling the generation of coherent and natural sign language motions; and (4) a discrete diffusion model that synthesizes motion-accurate sign language videos conditioned on the refined condition sequences. Extensive experiments on public SL datasets demonstrate that LLMCond achieves highly competitive performance, producing temporally coherent and accurate sign language motions without relying on auxiliary visual modalities such as pose, depth, or optical flow.

*Corresponding author.

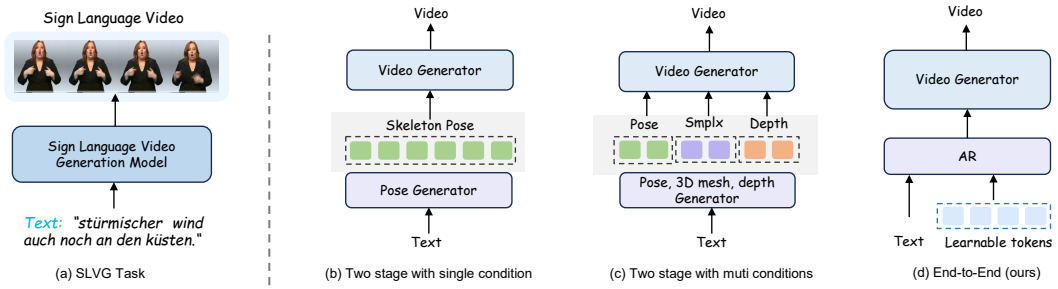


Figure 2: Difference between two-stage SLVG models and our end-to-end approach. Our model directly learns conditional tokens from text and generates videos without relying on intermediate modalities (pose, 3D mesh, depth).

1 Introduction

Research on sign language (SL) aims to enable bidirectional conversion between SL and spoken language. Accordingly, it is typically divided into two main tasks: Sign Language Understanding and Sign Language Generation. The former focuses on translating sign language sequences into glosses or textual sentences, which encompasses various sub-tasks such as Isolated SL Recognition (ISLR) [Hu et al., 2021, Huang et al., 2018], Continuous SL Recognition (CSLR) [Gan et al., 2024, Wei and Chen, 2023], Sign Language Translation (SLT) [Gan et al., 2026a, Chen et al., 2022] and SL Question Answering (SLQA) [Gan et al., 2026b]. The latter, in contrast, aims to generate corresponding sign pose or video sequences from text sequences, leading to tasks such as Sign Language Pose Production (SLP) [Saunders et al., 2021a, Baltatzis et al., 2024, Liu et al., 2026a], Sign Language Video Generation (SLVG) [Qi et al., 2024, Saunders et al., 2020b], and Pose to SL Video Generation (P2V) [Shi et al., 2024].

Among them, SLVG has recently gained increasing attention, aiming to generate realistic and expressive SL videos from spoken language inputs. Most existing studies adopt a two-stage generation paradigm, *i.e.*, first generating SL pose sequences (SLP) or other intermediate representations (*e.g.*, local regions, depth, or optical flow) [Saunders et al., 2022, Fang et al., 2023], and then using these representations as conditions to synthesize corresponding SL frames (Pose2Video). Since pose sequences contain much less visual and spatial information, they are considerably easier to model than raw video frames, and thus are widely employed as intermediate representations. However, heavy dependence on intermediate modalities increases the complexity of model training, limits the scalability, and also introduces cumulative errors at each stage. In contrast, compared with pose sequences, video frames capture much richer spatial and appearance details, making the development of end-to-end[2] text-to-video SLVG models considerably more challenging.

Meanwhile, current mainstream text-to-video (T2V) frameworks in the general domain (*e.g.*, CogVideoX [Yang et al., 2024], VideoCrafter [Chen et al., 2024], AnimateDiff [Guo et al., 2023], Wan [Wan et al., 2025]) generally follow a de facto paradigm: a VAE is employed to compress videos into compact latent representations, enabling efficient modeling of high-dimensional video data. A DiT-like Transformer is then used to generate these latent video representations conditioned on text inputs, after which the pretrained VAE decoder reconstructs the final video frames. Other models like CogVideo [Hong et al., 2022], VideoGPT [Yan et al., 2021], Phenaki [Villegas et al., 2022] adopt a discrete token modeling paradigm, where videos are quantized into sequences of discrete token IDs using a VQ-VAE tokenizer. The generation process is then formulated as a sequence modeling task, in which autoregressive, discrete diffusion, or masked language modeling techniques are applied to predict the discrete video tokens. In this paper, we show that directly fine-tuning current state-of-the-art (SOTA) T2V models on the SLVG task may yield unsatisfactory results, since current T2V models directly adopt text sequences as conditions, which are not aligned with SL sequences and often fail to effectively guide SL video generation.

Different from previous works, we focus on the end-to-end Text-to-Video SLVG task and build a novel yet effective LLM-based Vector-Quantized Diffusion framework. We tackle two problems

[2]Here, end-to-end means that no intermediate modalities (pose, depth etc.) are produced or required during training or inference.

that are not addressed by existing frameworks: *(1) how to align the textual conditions with the SL sequence*, since SL exhibits different grammar and non-monotonic, variable-length alignments with spoken text, making it hard for textual conditions to directly guide SL video generation; and *(2) how to ensure temporal coherence across consecutive frames/clips*, as natural sign language expressions require smooth and contextually consistent transitions between signs in SL.

In our approach, LLMs act as powerful condition generators based on the input text. We decompose this challenging task into four essential components: (1) a Video VQ-VAE: SignVQ, which compresses and quantizes SL videos into discrete video tokens, enabling compact yet expressive latent representations; (2) to address the misalignment between textual conditions and SL sequences, we introduce an **LLM-based conditioner** that generates semantically consistent conditions for each video clip, which is constrained by gloss sequences via a CTC-based alignment loss during training; (3) to ensure temporal coherence across frames, we propose a **Gaussian Condition Refiner** (GCR) that smooths the generated conditions over time to promote temporally coherent SL video generation; and (4) a VQ diffusion model is employed to generate latent video tokens conditioned on the refined condition sequence, which are then decoded by the SignVQ decoder to produce realistic sign language videos. During inference, the input text is first transformed into a sequence of semantically aligned conditions by the LLM conditioner. These conditions are refined by the GCR module and then passed into a VQ diffusion model to generate discrete video tokens, which are finally decoded into SL videos. Our contributions are:

- We propose an LLM-based conditioner for end-to-end SLVG to generate fine-grained semantic conditions, incorporating a CTC alignment constraint to promote temporal semantic correspondence with the sign sequences.
- To enhance temporal coherence across consecutive frames, we introduce a Gaussian Condition Refiner that generates temporally smooth but semantically discriminative conditions, ensuring natural and consistent sign transitions in the generated videos.
- To the best of our knowledge, we propose LLMCond, the first simple yet effective end-to-end T2V SLVG model that does not rely on auxiliary visual modalities (*e.g.*, pose, body parts, depth, or optical flow). Extensive experiments on public SL datasets show that our model achieves highly competitive performance even compared with two-stage models.

2 Related Work

Text to Video Generation. Current end-to-end T2V models generally fall into two paradigms: (1) modeling continuous latent representations extracted by VAE-based tokenizers, reformulating the video generation task as a problem of latent representation modeling; [Yang et al., 2024, Chen et al., 2023, 2024, Guo et al., 2023], or (2) compressing video into discrete token IDs via VQVAE-based tokenizers, converting the video generation task into a discrete token sequence modeling problem, which is then addressed using autoregressive, discrete diffusion or masked language modeling methods [Hong et al., 2022, Yan et al., 2021, Villegas et al., 2022]. The continuous-latent approach operates on low-dimensional real-valued embeddings, reducing pixel-level complexity and enabling latent diffusion pipelines to synthesize high-resolution frames. In contrast, the discrete-token paradigm compresses videos into sequences of quantized IDs and formulates video synthesis as a sequence modeling problem. Early influential works, such as VideoGPT [Yan et al., 2021], demonstrate that VQ-VAE tokenizers combined with autoregressive Transformers can effectively predict discrete video latents. Later models, including Phenaki [Villegas et al., 2022], extend this approach to variable-length T2V generation using discrete tokenizers and causal attention. *Despite these advances, in this paper, we show that directly fine-tuning current SOTA T2V models on SLVG task yields unsatisfactory results*, often producing inaccurate and temporally inconsistent hand movements, as they still struggle to (1) align textual conditions with sign sequences, since the ordering of signs often differs from that of spoken text, making it difficult for textual cues to effectively guide video generation, and (2) ensure temporal coherence across consecutive frames, which is essential for generating smooth and contextually consistent sign transitions.

Sign Language Video Generation. Sign language research is typically divided into understanding tasks (e.g., CSLR [Wei and Chen, 2023], SLT [Gan et al., 2025, Shi et al., 2022]) and generation tasks (SLP [Saunders et al., 2020a, 2021b, Tang et al., 2025, Xie et al., 2024, Stoll et al., 2018, 2022, Yin et al., 2023, Rastgoo et al., 2021, Liu et al., 2026b, Zuo et al., 2025, Tang et al., 2022],

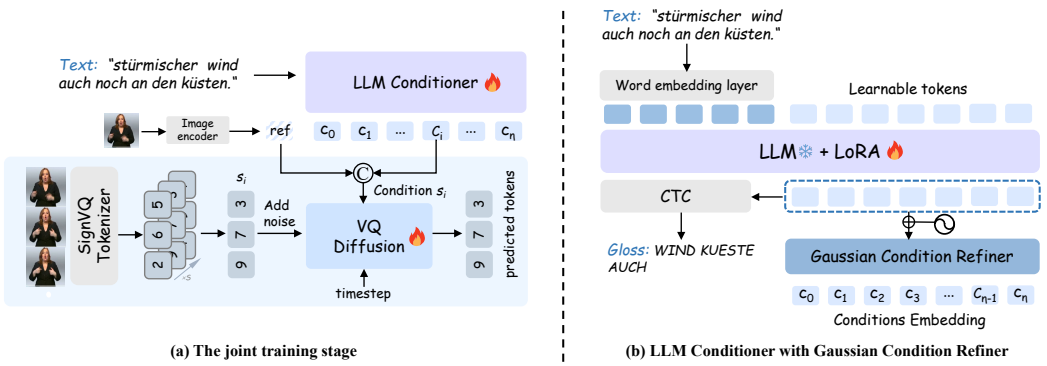


Figure 3: Overview of LLMCond. (a) Joint training, where SignVQ encodes videos into discrete tokens and VQ-Diffusion generates video tokens conditioned on a reference image and conditions. (b) LLM conditioner, which fuses text embeddings with learnable tokens, enforces gloss alignment via CTC, and refines conditions using a Gaussian refiner for temporal consistency.

SLVG [Saunders et al., 2022, Fang et al., 2023, Qi et al., 2024, Fang et al., 2025]). The former converts SL sequences into gloss or spoken text, while the latter maps gloss or text back into sign language. Recently, SLVG has attracted increasing attention for its potential to synthesize realistic SL videos from text, improving information accessibility for the hearing impaired community. Most existing SLVGs follow a two-stage design [Saunders et al., 2022, Fang et al., 2023], where pose generation serves as an intermediate representation. Within this framework, autoregressive or diffusion models predict poses from text, while GAN or diffusion based models synthesize video frames from the generated pose sequences. Such dependence on intermediate modalities increases the complexity of SLVG model training, limits scalability, and can introduce cumulative errors at each stage. In contrast, image sequences contain richer visual details, making end-to-end SLVG modeling substantially more challenging. Early attempts, such as SignGen [Qi et al., 2024], employ auxiliary conditions (*e.g.*, optical flow, pose, depth) during training; however, these signals are unavailable during inference, which highly constrains the generation quality. In this work, we propose a simple yet effective end-to-end SLVG model that eliminates reliance on auxiliary visual conditions in training and inference.

3 Method

Overall Framework. We formulate the SLVG task as a condition-based SL clip generation process which consists of five components: video quantization, condition generation, condition refinement, video latent generation, and video latent dequantization. Specifically, given a text sequence $x = \{x_i\}_{i=0}^{\varsigma}$ with ς words, the goal of SLVG is to generate a SL video with F frames $f = \{f_i\}_{i=0}^{F}$ based on input x. As shown in Figure 3, our model tackles the SLVG as follows: (1) a video tokenizer that compresses and quantizes video sequences into discrete tokens $v = \mathcal{VQ}_E(f)$, where $v = \{v_i\}_{i=1}^{\eta}$ and η denotes the temporal length after the tokenizer's downsampling; (2) an LLM-based conditioner that generates temporally aligned condition embeddings $c = \mathcal{LLM}(x)$, where $\mathbf{c} = \{\mathbf{c}_i\}_{i=0}^{\eta}$; (3) a Gaussian Condition Refiner (GCR) that smooths the generated conditions over time to alleviate jitters in generated videos; (4) a VQ diffusion that produces discrete video tokens $\hat{v} = \mathcal{LDM}(\mathbf{c}, c_{\text{ref}})$, where c_{ref} is the reference image embedding used to guide the generation; and (5) the predicted tokens are subsequently decoded into full videos by the video tokenizer decoder $\hat{f} = \mathcal{VQ}_D(\{\hat{v}_i\}_{i=0}^{\eta})$.

3.1 Sign Language Video Tokenizer

Current video generation models typically use VAE or VQ-VAE to compress a video f into a latent representation, reducing video generation to latent modeling. Inspired by VQ-VAE-2 [Razavi et al., 2019], we propose SignVQ, a sign language tokenizer that encodes raw SL videos into discrete tokens with a two-level codebook. Given a video $f \in \mathbb{R}^{F\times H\times W\times 3}$, where F, H, and W denote the number of frames, height, and width, the encoder $\mathcal{VQ}_E$ first performs spatial and temporal downsampling by factors of d_b and d_f via 3D convolutional blocks, producing a coarse spatiotemporal feature map e_b.

This representation is further spatially downsampled to obtain a high-level feature e_t.

$$e_t, e_b = \mathcal{VQ}_E(f). \tag{1}$$

Next, two vector quantizers, Q_t and Q_b, quantize the top and bottom features e_t and e_b, respectively:

$$v_t = Q_t(e_t), \quad v_b = Q_b([U_t(\hat{e}_t), e_b]), \tag{2}$$

where U_t denotes the upsampling operation applied to the quantized top-level features $\hat{e}_t$. The resulting discrete indices v_t and v_b form a sequence of quantized video tokens $v = \{v_i\}_{i=1}^{\eta}$, where $v_i = \{v_{t,i}, v_{b,i}\}$. The total video id length *i.e.*, (the number of ids) is computed as:

$$L_v = \eta \times (L_{v_t} + L_{v_b}) = \frac{F}{d_f} \times \left(\frac{H \times W}{d_t^2} + \frac{H \times W}{d_b^2} \right). \tag{3}$$

Thus, the SLVG objective shifts from modeling pixel-space likelihood $p(f|x)$ to discrete latent likelihood $p(v|c)$. During reconstruction, the SignVQ decoder $\mathcal{VQ}_D$ retrieves the nearest embeddings from the corresponding codebooks using discrete indices $\{v_i\}_{i=1}^{\eta}$ and reconstructs the SL video $\hat{f}$ as:

$$\hat{f} = \mathcal{VQ}_D(v_t, v_b), \tag{4}$$

3.2 LLM Conditioner

Traditionally, T2V models directly embed text inputs as condition embeddings, using pretrained language models such as CLIP or BERT. While effective in general domains, this approach often fails in SLVG, due to the unique grammatical structure of SL, *i.e.*, spoken texts are not monotonically aligned with sign sequences. Consequently, these embeddings fail to capture semantic alignment with SL sequences, leading to temporally inconsistent and semantically distorted SL video generation.

To bridge the gap between spoken text and SL sequence, we design an LLM-based conditioner that provides temporally and semantically aligned conditional embeddings, to guide SL frame generation. As shown in Figure 3(b), given a text sequence x, and its corresponding embedding $e_x = \{e_x^i\}_{i=0}^{\ell}$, we introduce a sequence of learnable condition tokens $e_l = \{e_l^i\}_{i=0}^{\eta}$ that serve as alignment anchors between text and sign language. The text embeddings and learnable condition tokens are then concatenated and fed into the LLM for contextual modeling:

$$h = \mathcal{LLM}([e_x, e_l]) \tag{5}$$

We take the hidden states h at learnable tokens as the target conditional embeddings, denoted as $c = h[\ell\ :,]$. To ensure that LLM-generated conditions preserve text semantics and temporal consistency with SL videos, we impose a CTC alignment loss $\mathcal{L}_{CTC}$ between the generated conditions and corresponding gloss sequences g during training.

$$\Theta_{\text{LLM}}^*,\ e_l^* \leftarrow \underset{\Theta_{\text{LLM}},\ e_l}{\arg\min}\, \mathbb{E}_{(x,g)\sim\mathcal{D}}\big[\mathcal{L}_{\text{CTC}}(c * W_g, g)\big], \tag{6}$$

where Θ_{LLM} denotes the trainable parameters of LLM and W_g denotes the classification layer weights. Through these constraints, the condition embeddings c are temporally aligned with the gloss sequence, allowing each embedding to independently provide semantically aligned and temporally consistent guidance for SL video generation. *The length of the learnable token ℓ is determined by a length prediction module; see the Appendix for details.*

3.3 Gaussian Condition Refiner

Although the LLM conditioner module ensures that the generated conditions c are temporally aligned with the SL sequences, aligning conditions c with the gloss sequence discretizes its distribution, which may lead to discontinuities between adjacent sign clips. To alleviate this issue, we further smooth c using a Gaussian Condition Refiner (GCR) to promote seamless transitions across clips.

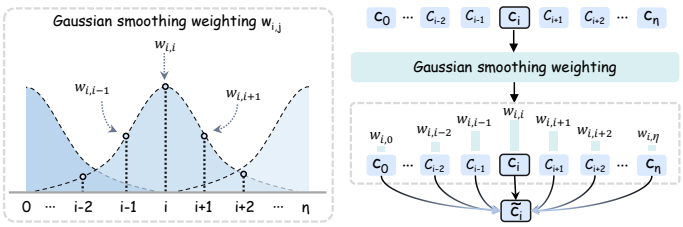


Figure 4: The Gaussian Condition Refiner (GCR) module.

Specifically, as shown in Figure 4, given a condition sequence $c = \{c_i\}_{i=0}^{\eta}$, we compute Gaussian weights $\{w_{i,j}\}_{j=0}^{\eta}$ centered at each position i and apply them to neighboring conditions via weighted summation, to obtain a refined condition $\tilde{c}_i$. This encourages smooth transitions and temporally coherent conditioning signals, where σ controls the smoothing strength.

$$\tilde{c}_i = \frac{\sum_{j=0}^{\eta} \left\{ \exp\left(-\frac{(j-i)^2}{2\sigma^2}\right) \cdot c_j \right\}}{\sum_{j=0}^{\eta} \exp\left(-\frac{(j-i)^2}{2\sigma^2}\right)}, \tag{7}$$

3.4 VQ-Diffusion Model for SLVG

After obtaining the refined condition sequence $c = \{c_i\}_{i=0}^{\eta}$, we adopt a VQ-Diffusion model [Gu et al., 2022] to generate v_i conditioned on c_i. During training, VQ-Diffusion gradually corrupts v_i via a fixed Markov chain $q(v_{i,t}|v_{i,t-1})$, *e.g.*, randomly replacing tokens of $v_{i,t-1}$, where $v_{i,0} = v_i$. By the Markov property, we derive the probability of $v_{i,t}$ as:

$$q(v_{i,t}|v_{i,0}) = e^{\top}(v_t)\, \overline{Q}_t\, e(v_0), \quad \overline{Q}_t = Q_t \cdots Q_1, \tag{8}$$

where $e(v_t) \in \mathbb{R}^{K+1}$ is the one-hot representation of a token id (with an additional id for the `[MASK]` token) and K is codebook size. The transition matrix $Q_t \in \mathbb{R}^{(K+1)\times(K+1)}$ is defined as:

$$Q_t = \begin{bmatrix} \alpha_t + \beta_t & \beta_t & \cdots & 0 & \cdots & 0 \\ \beta_t & \alpha_t + \beta_t & \cdots & \vdots & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ \beta_t & \cdots & \cdots & \alpha_t + \beta_t & \cdots & 0 \\ \gamma_t & \gamma_t & \cdots & \gamma_t & \cdots & 1 \end{bmatrix}. \tag{9}$$

After a fixed number of T timesteps, the $v_{i,T}$ becomes pure noise tokens. Starting from the noise $v_{i,T}$, the reverse process gradually denoises the latent variables, by sampling from the reverse distribution $q(v_{i,t-1}|v_{i,t}, \tilde{c}_i, c_{ref})$, where c_{ref} is reference image embedding, and then restores the $v_{i,0}$. For reverse process, we train a Transformer denoiser to model the probability $p_\theta(v_{i,t-1} \mid v_{i,t}, \tilde{c}_i, c_{ref})$ as:

$$p_\theta(v_{i,t-1}|v_{i,t}, \tilde{c}_i, c_{ref}) = \sum_{\tilde{v}_{i,0}} q(v_{i,t-1}|v_{i,t}, \tilde{v}_{i,0})\, p_\theta(\tilde{v}_{i,0}|v_{i,t}, \tilde{c}_i, c_{ref}), \tag{10}$$

Then, the VQ-Diffusion module is trained under:

$$\mathcal{L}_{\mathcal{DI}} = \mathbb{E}_{t\sim\mathcal{U}(1,T),\,(v_{i,0},v_{i,t})\sim q}\Big[D_{\mathrm{KL}}\big(q(v_{i,t-1} \mid v_{i,t}, v_{i,0}) \parallel p_\theta(v_{i,t-1} \mid v_{i,t}, \tilde{c}_i, c_{ref})\big)\Big]. \tag{11}$$

3.5 Training and Inference

Training. The final training process of LLMCond first trains SignVQ model with reconstruction loss, embedding loss and commitment loss as follows:

$$\arg\min_{\Theta_{\mathcal{VQ}}} \mathbb{E}_{(f)\sim\mathcal{D}}\Big[\|f - \hat{f}\|_2^2 + \|\mathrm{sg}[\hat{e}] - e\|_2^2 + \beta\|\hat{e} - \mathrm{sg}[e]\|_2^2 \Big], \tag{12}$$

where $||\cdot||_2$ denotes the L2 norm and β is a weighting coefficient. After the SignVQ is trained, we train the LLM Conditioner together with the VQ-Diffusion model using the following joint loss:

$$\arg\min_{\Theta_{\mathcal{LLM}},\Theta_{\mathcal{DI}}} \mathbb{E}_{(f,x,g)\sim\mathcal{D}}\Big[\alpha\, \mathcal{L}_{\mathrm{CTC}}(c * W_g, g) + \lambda \sum_{i=0}^{\eta} \mathcal{L}_{DI}(v_i, \tilde{c}_i, c_{ref}) \Big], \tag{13}$$

where α and λ are hyperparameters. The condition sequence c is optimized under CTC supervision and diffusion constraints, encouraging temporally aligned, visually coherent sign video generation.

Inference. During inference, the text sequence is fed into the LLM conditioner to generate the condition sequence. The VQ-diffusion model then produces a sequence of latent discrete video tokens $\{\hat{v}_i\}_{i=0}^{\eta}$ based on $\tilde{c}$, which are decoded by the SignVQ decoder into the final SL video $\hat{f}$. No gloss annotations or intermediate modalities are required or generated in this process.

Table 1: Comparison of SLVG performance on Phoenix14T. (P: pose, H: 3D hands, D: depth, O: optical flow, * denotes our finetuning results. The same notation applies to the tables below.)

Two-Stage	Extra Data				DEV						TEST					
	P	H	D	O	ROUGE	BLEU1	BLEU2	BLEU3	BLEU4	gFVD↓	ROUGE	BLEU1	BLEU2	BLEU3	BLEU4	gFVD↓
ProTran [Saunders et al., 2020c] + Ani [Hu, 2024]	✓				9.74	9.09	2.95	1.51	0.75	2125.19	10.33	9.79	3.60	1.93	0.82	2017.90
SignGAN [Saunders et al., 2022]	✓				13.90	13.27	4.91	2.55	1.36	1986.33	15.97	14.02	6.74	3.86	2.67	1918.42
End-to-End																
SignGen [Qi et al., 2024]	✓		✓	✓	10.41	9.36	2.15	0.83	0.39	2565.03	10.08	10.61	2.47	1.03	0.45	2553.14
VideoGPT* [Yan et al., 2021]					10.83	9.85	3.60	1.71	0.86	2021.74	11.10	10.71	3.66	2.03	0.87	2011.89
CogVideoX* [Yang et al., 2024]					11.24	11.18	3.75	2.06	1.19	2008.25	11.67	11.32	4.18	2.02	1.14	1998.81
Wan2.1* [Wan et al., 2025]					12.14	11.75	4.36	2.48	1.26	2074.31	12.56	12.79	5.64	2.41	1.40	1894.75
LLMCond					**26.57**	**27.06**	**14.95**	**10.68**	**7.93**	**1124.10**	**27.29**	**26.55**	**15.36**	**10.43**	**7.91**	**1107.46**

Table 2: Comparison of SLVG performance on the CSL-Daily dataset.

Two-Stage	Extra Data			DEV						TEST					
	P	D	O	ROUGE	BLEU1	BLEU2	BLEU3	BLEU4	gFVD↓	ROUGE	BLEU1	BLEU2	BLEU3	BLEU4	gFVD↓
ProTran [Saunders et al., 2020c] + Ani [Hu, 2024]	✓			8.93	8.81	2.74	1.55	0.63	2158.04	9.41	9.48	3.50	1.82	0.69	2077.45
SignGAN [Saunders et al., 2022]	✓			14.02	14.74	5.17	2.68	1.73	1923.53	13.42	13.85	4.11	1.86	1.09	1946.27
End-to-End															
SignGen [Qi et al., 2024]	✓	✓	✓	10.44	10.98	3.04	2.27	1.30	2406.46	11.45	11.39	4.24	2.76	1.38	2551.87
VideoGPT* [Yan et al., 2021]				9.17	9.55	2.52	1.73	0.69	2674.20	9.60	9.91	3.16	1.49	0.66	2694.93
CogVideoX* [Yang et al., 2024]				11.56	10.22	3.02	2.18	1.66	1944.53	11.42	10.90	4.21	2.94	1.28	1962.08
Wan2.1* [Wan et al., 2025]				11.87	11.20	3.39	2.36	1.74	1932.50	12.33	12.61	4.59	3.25	1.58	1985.47
LLMCond				**24.59**	**24.31**	**14.93**	**9.65**	**7.78**	**1500.27**	**24.80**	**25.26**	**15.04**	**9.89**	**7.94**	**1412.48**

4 Experiments

Datasets. We evaluate our model on two widely used SL datasets with both gloss and text annotations: (1) *Phoenix14T* [Camgoz et al., 2018], a German SL corpus from weather broadcasts, consists of 8257 videos from nine signers with 1066 glosses and 2877 German words, split into 7096, 519, and 642 videos for training, validation, and testing, respectively. (2) *CSL-Daily* [Zhou et al., 2021], a Chinese SL dataset with a vocabulary of 2000 glosses and 2343 Chinese words, comprising 18401, 1077, 1176 videos for training, validation and testing respectively.

Evaluation Metrics. To evaluate **semantic accuracy**, we employ a pretrained SLT model [Gan et al., 2025] to translate generated videos back into text. The resulting translations are compared with reference sentences using BLEU [Papineni et al., 2002] and ROUGE-L [Lin and Och, 2004]. These scores reflect how accurately generated videos convey sign meaning. To evaluate **video quality**, we adopt common T2V metrics, including FID [Heusel et al., 2017], PSNR, SSIM, and LPIPS for reconstruction, and FVD [Unterthiner et al., 2018] for generation. These metrics measure pixel-level similarity, perceptual fidelity, and temporal consistency between generated and GT videos.

Implementation Details. (1) *SignVQ*: In the baseline setup, the spatial and temporal downsampling factors d_f, d_b, and d_t are set to 4, 16, and 32, respectively. The latent channel dimension is 128. (2) *LLM Conditioner*: We employ the LLaMA-3.2 1B model fine-tuned using LoRA [Hu et al., 2022]. The LoRA modules are inserted into the `q_proj`, `v_proj`, and `o_proj` layers, with the rank set to 64. (3) *GCR*: The Gaussian smoothing parameter σ is set to 0.5. (4) *VQ Diffusion*: We adopt a VQ-Diffusion backbone consisting of 15 Transformer layers. The generated condition tokens $\tilde{c}$ are fused through cross-attention layers in every Transformer block, while reference image conditions are fused at the 3rd, 9th, and 12th layers. *More implementation details are provided in the Appendix.*

5 Comparisons

We use MixSignGraph [Gan et al., 2025], an SLT model, to evaluate videos generated by our SLVG model and the baseline methods. For a fair comparison, we reproduce several open-source baselines and report their results under a unified evaluation protocol. We compare our method with end-to-end SLVG models as well as two-stage approaches that rely on explicit auxiliary modalities, such as pose or depth, during training or inference. Results are reported in Tables 1 and 2.

Evaluation of Generation on Phoenix14T. We compare two groups of baselines on Phoenix14T, reporting back-translation results and generated-video FVD (gFVD). *(1) Comparison with End-to-End T2V Models.* We fine-tune three representative open-source T2V models on our dataset, including continuous latent-based CogVideoX-2B and Wan2.1-1.3B, and discrete latent-based VideoGPT. As shown in Table 1, directly adapting general-purpose T2V models to SLVG leads to inaccurate sign generation and unsatisfactory performance. These models are mainly optimized for globally coherent visual content and smooth motion, rather than fine-grained human actions, making it difficult to synthesize subtle and temporally precise sign gestures. This limitation is also noted in the official

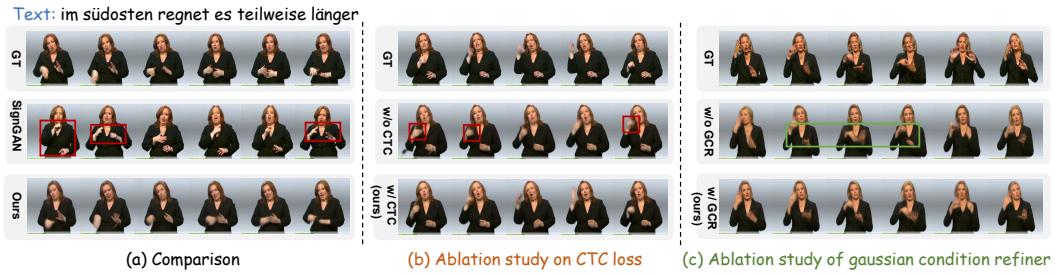


Figure 5: (a) Qualitative comparison on Phoenix14T with SignGAN. (b) Ablation study on CTC loss. (c) Ablation study on Gaussian Condition Refiner (GCR). Red boxes denote incorrect motions, while green boxes indicate temporal discontinuities.

documentation [3], which reports strong global consistency and appearance modeling but limited capability for detailed human actions. For SignGen, although it uses pose, depth, and optical flow during training, its reliance on these modalities makes it difficult to generate semantically correct sign videos when they are unavailable during inference. *(2) Comparison with Two-Stage SLVG Models.* We further compare with current two-stage SLVG models. Without requiring intermediate modalities during training or inference, our model achieves SOTA back-translation performance, demonstrating the effectiveness of the LLM-based conditioner in aligning text conditions with sign sequences. Moreover, it obtains lower gFVD compared to SignGAN, which uses pose as an intermediate representation.

Evaluation of Generation on CSL-Daily. We further evaluate our model on the CSL-Daily, comparing it with both T2V models (Wan2.1 etc.) and SLVG models (SignGAN etc.). As shown in Table 2, our method achieves clear improvements in both back-translation accuracy and video quality. Compared with the T2V model Wan2.1, ROUGE increases from 12.33 to 24.80, while gFVD decreases from 1985.47 to 1412.48. Compared with the SLVG model SignGAN, our LLMCond achieves gains of +11.38 ROUGE, along with a 533.79 reduction in gFVD.

6 Qualitative Results

Visualization Comparison. To intuitively demonstrate the generation quality of our model, we visualize one sample from the Phoenix14T dataset and compare it with the previous SLVG model SignGAN. As shown in Figure 5 (a), compared with the ground-truth SL sequences (GT), our model produces an SL sequence that more closely matches the ground truth in both visual quality and temporal coherence. In contrast, SignGAN often exhibits misaligned gestures, leading to reduced semantic accuracy. These results indicate that LLMCond models spatiotemporal dependencies more effectively, achieving tighter alignment between text semantics and sign sequences.

Visualization Ablation. As shown in Figure 5 (b), we compare the ground truth (GT), a variant trained without the CTC constraint (w/o CTC), and our model (w/ CTC). In Figure 5 (c), we further compare GT, a variant without Gaussian condition refiner (w/o GCR), and our model (w/ GCR). It can be clearly observed that removing the CTC constraint and GCR respectively results in semantic inaccuracy and temporal inconsistency, highlighting the importance and effectiveness of our proposed design. *Please refer to the supplementary materials for more visualization.*

7 Ablation Study

In this subsection, we report ablations on the Phoenix14T dataset that verify the effectiveness of LLMCond. *Please refer to the Appendix for more ablation study.*

Reconstruction Results of Different Compression Rate. We first analyze how compression affects quantization and reconstruction in SignVQ. As reported in Table 3, higher compression degrades VQ-VAE reconstruction but shortens the latent sequence. Increasing compression from $2 \times 8 \times 8$ to $4 \times 16 \times 16$ (time$\times$height$\times$width) raises FVD, while making latent codes $8\times$ shorter. Overall,

[3] https://huggingface.co/docs/diffusers/training/cogvideox

Table 3: Effect of different variants of SignVQ.

Variants	Baseline	A	B	C
Compression	4×16×16	2×16×16	4×8×8	2×8×8
Latent Codes	$\frac{L}{4}$×245	$\frac{L}{2}$×245	$\frac{L}{4}$×980	$\frac{L}{2}$×980
Codebook Size	2048	1024	1024	512
rFID↓	0.23	0.16	0.13	0.03
rLPIPS↓	0.0611	0.0545	0.0492	0.0425
rFVD↓	363.75	319.60	259.27	194.46
rPSNR↑	28.88	29.86	30.68	32.01
rSSIM↑	0.89	0.90	0.91	0.93
Hand SSIM↑	0.89	0.90	0.91	0.92
Face SSIM↑	0.88	0.89	0.90	0.91

Table 4: Effect of LLM conditioner and GCR.

CTC	GCR	ROUGE-L	BLEU-1	BLEU-4	gFVD↓
		20.95	21.22	5.20	1533.16
✓		24.41	24.92	6.98	1257.95
	✓	21.36	21.60	5.53	1412.28
✓	✓	**27.29**	**26.55**	**7.91**	**1107.46**

Table 5: Ablation on Signer-Independent test.

Split	ROUGE-L	BLEU-1	BLEU-4	gFVD↓
Norm-Split	27.29	26.55	7.91	1107.46
Signer-Independent	25.67	25.41	7.69	1345.36

stronger compression trades reconstruction fidelity for compactness. We also evaluate key local regions (*e.g.*, hands and face) cropped using HRNet keypoints, where the model captures fine-grained SL motion details. To balance compression and reconstruction, we adopt $4 \times 16 \times 16$ compression.

Ablation study on Conditioner Modules. We investigate the impact of the LLM Conditioner and GCR in Table 4. As shown, introducing the LLM Conditioner with CTC supervision significantly improves semantic accuracy and video quality, demonstrating its effectiveness in aligning textual and visual representations. Removing CTC supervision causes a severe performance drop, indicating that temporal alignment is essential for guiding the conditioner. Similarly, excluding GCR reduces generation quality, as indicated by higher gFVD, suggesting that Gaussian smoothing helps refine temporal coherence. Overall, our model with both the LLM Conditioner and GCR achieves the best performance across all metrics, confirming the effectiveness of the proposed components.

Ablation Study on σ in Gaussian Smoothing. To investigate the impact of Gaussian standard deviation σ in the GCR module, we conduct experiments with different σ values. A larger σ strengthens the influence of neighboring conditions on each clip's generation. As shown in Figure 6(a), the generated results vary significantly with σ, and the model performs best when $\sigma = 0.5$. A smaller σ makes each clip more independent, causing discontinuous motion, whereas an excessively large σ leads to over-smoothing with overly similar neighboring clips.

Generation Results of Different Compression Rate. We analyze the effect of video compression on LLMCond. As shown in Figure 6(b), stronger compression improves ROUGE and FVD, despite modestly reducing VQ reconstruction quality (Table 3). The shorter latent sequence eases generation, outweighing reconstruction loss and improving overall performance.

Ablation Study on Codebook Size for Quantization. We evaluate tokenizer codebook size for 3D video reconstruction in Figure 6(c). Small codebooks (256) increase reconstruction error, lowering PSNR and raising FVD. Performance improves up to 1024, where PSNR peaks and FVD is lowest. Increasing to 2048 gives no gain and slightly degrades generation.

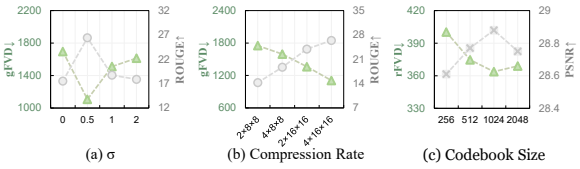


Figure 6: Effect of SignVQ and GCR hyperparameters. Triangles: gFVD; circles: ROUGE; crosses: PSNR.

Ablation Study on signer-independent test. To evaluate generalization to unseen signers, we re-split Phoenix14T, using one signer as the test set and the remaining eight for training. As shown in Table 5, performance only slightly decreases compared to the Norm-Split (ROUGE 27.29 to 25.67), indicating that LLMCond achieves comparable performance to the standard split, with ROUGE-L decreasing only slightly from 27.29 to 25.67, suggesting good generalization to unseen signers.

Ablation Study on Generation Models. To examine the effectiveness of our VQ-Diffusion model, we compare it with several representative discrete sequence generation approaches, including an AR-based model (Llama3.2-1B), a masked diffusion model (MDM)-based variant (LLaDA), and VQ-Diffusion. As shown in Table 6, the AR-based Llama model yields the weakest performance, as it struggles to capture spatial dependencies in sign sequences. In contrast, our VQ-Diffusion model

Table 6: Comparison of different generation models.

Model	Type	ROUGE-L	BLEU-1	BLEU-4	gFVD↓
Llama-3.2 [Grattafiori et al., 2024]	AR	20.25	19.96	3.65	1579.86
LLaDA [Nie et al., 2025]	MDM	24.73	23.14	6.20	1306.30
VQ-Diffusion	**VQD**	**27.29**	**26.55**	**7.91**	**1107.46**

Table 7: Comparison of representation paradigms.

Model	ROUGE-L	BLEU-1	BLEU-4	gFVD↓
Continuous	26.10	25.74	7.65	1194.32
Discrete	**27.29**	**26.55**	**7.91**	**1107.46**

achieves the best results in both semantic accuracy and video quality, demonstrating its superior capability for coherent SL video generation.

Comparison of Different Representation Paradigms. In mainstream text-to-video frameworks, videos are often represented either as continuous or discrete latent representations. Here, we evaluate our framework under different representation paradigms. We replace the VQ-VAE with a VAE whose latent channel is set to 16, and replace the discrete diffusion model with a continuous diffusion model, while keeping other components unchanged. As shown in Table 7, models based on continuous representations also achieve good performance, demonstrating the generalizability of our method. However, in our experiments, discrete-based models achieve slightly better generation results on both back-translation metrics and video quality.

Ablation of Training Strategies. We also present generation results under different training strategies. In the first strategy, the LLM Conditioner is trained with a CTC constraint and then frozen before training the VQ-Diffusion model (denoted as 'Separate'). In the second strategy, the LLM Conditioner and VQ-Diffusion model are trained jointly under the constraint of Equation 13 (denoted as 'Joint'). As shown in Table 8, our joint training strategy achieves better performance.

Table 8: Comparison of training strategies for the LLM and diffusion components.

Training strategy	ROUGE-L	BLEU-1	BLEU-4	gFVD↓
Separate	24.09	23.86	6.36	1373.61
Joint	**27.29**	**26.55**	**7.91**	**1107.46**

8 Conclusion

In this paper, we present a first end-to-end SLVG framework: LLMCond, which generates SL videos directly from text without relying on intermediate modalities. Specifically, to align textual input with the sign sequence, we introduce an LLM-based conditioner trained with a CTC alignment, to produce semantically consistent clip-level conditions. To ensure temporal coherence across clips, we employ a GCR that smooths the generated conditions over time. To enable efficient yet expressive video synthesis, we compress videos into discrete tokens with the SignVQ, and generate latent video tokens using a VQ-diffusion model conditioned on the refined conditions. Extensive experiments on public SL datasets demonstrate that LLMCond delivers highly competitive performance, producing temporally coherent and visually expressive videos without relying on extra visual modalities such as pose, depth, or optical flow.

9 Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant No. 62606221, the Postdoctoral Fellowship Program of the China Postdoctoral Science Foundation (CPSF) under Grant No. GZB20260384, and the Jiangsu Funding Program for Excellent Postdoctoral Talent under Grant No. 2026ZB254.

References

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

Vasileios Baltatzis, Rolandos Alexandros Potamias, Evangelos Ververas, Guanxiong Sun, Jiankang Deng, and Stefanos Zafeiriou. 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, 2024.

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*, pages 7784–7793, 2018.

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*, pages 10023–10033, 2020.

Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, et al. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023.

Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7310–7320, 2024.

Yutong Chen, Ronglai Zuo, Fangyun Wei, Yu Wu, Shujie Liu, and Brian Mak. Two-stream network for sign language recognition and translation. *Advances in Neural Information Processing Systems*, 35:17043–17056, 2022.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

Sen Fang, Chunyu Sui, Xuedong Zhang, and Yapeng Tian. Signdiff: Learning diffusion models for american sign language production. *arXiv e-prints*, pages arXiv–2308, 2023.

Sen Fang, Chen Chen, Lei Wang, Ce Zheng, Chunyu Sui, and Yapeng Tian. Signllm: Sign language production large language models. In *2025 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 6681–6693. IEEE, 2025.

Shiwei Gan, Yafeng Yin, Zhiwei Jiang, Hongkai Wen, Lei Xie, and Sanglu Lu. Signgraph: A sign sequence is worth graphs of nodes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13470–13479, 2024.

Shiwei Gan, Yafeng Yin, Zhiwei Jiang, Lei Xie, Sanglu Lu, and Hongkai Wen. Mixsigngraph: A sign sequence is worth mixed graphs of nodes. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Shiwei Gan, Xiao Liu, Yafeng Yin, Zhiwei Jiang, Bowen Guo, Lei Xie, Sanglu Lu, and Hongkai Wen. Signllama: Enhancing gloss-free sign language translation by prioritizing visual features for llms. *arXiv preprint arXiv:2608.09006*, 2026a.

Shiwei Gan, Lichen Wang, Xiao Liu, Yafeng Yin, Kuizhuang Liu, Sanglu Lu, and Lei Xie. Sign language question answering: A new task, benchmark, and baseline for sign language understanding. *arXiv preprint arXiv:2607.27826*, 2026b.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10696–10706, 2022.

Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Hezhen Hu, Wengang Zhou, and Houqiang Li. Hand-model-aware sign language recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 1558–1566, 2021.

Li Hu. Animate anyone: Consistent and controllable image-to-video synthesis for character animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8153–8163, 2024.

Jie Huang, Wengang Zhou, Houqiang Li, and Weiping Li. Attention-based 3d-cnns for large-vocabulary sign language recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(9):2822–2832, 2018.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Chin-Yew Lin and Franz Josef Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd annual meeting of the association for computational linguistics (ACL-04)*, pages 605–612, 2004.

Xiao Liu, Shiwei Gan, Yafeng Yin, Bowen Guo, Zhiwei Jiang, Shunmei Meng, Lei Xie, and Sanglu Lu. Signpr: A progressive vector-quantized diffusion framework for sign language production. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2198–2208, 2026a.

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*, 2026b.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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, 2002.

Fan Qi, Yu Duan, Huaiwen Zhang, and Changsheng Xu. Signgen: End-to-end sign language video generation with latent diffusion. In *European Conference on Computer Vision*, pages 252–270. Springer, 2024.

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

Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. Adversarial training for multi-channel sign language production. *arXiv preprint arXiv:2008.12405*, 2020a.

Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. Everybody sign now: Translating spoken language to photo realistic sign language video. *arXiv preprint arXiv:2011.09846*, 2020b.

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

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, 2021a.

Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. Mixed signals: Sign language production via a mixture of motion primitives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1919–1929, 2021b.

Ben Saunders, Necati Cihan Camgoz, and Richard 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*, pages 5141–5151, 2022.

Bowen Shi, Diane Brentari, Gregory Shakhnarovich, and Karen Livescu. Open-domain sign language translation learned from online video. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6365–6379, 2022.

Tongkai Shi, Lianyu Hu, Fanhua Shang, Jichao Feng, Peidong Liu, and Wei Feng. Pose-guided fine-grained sign language video generation. In *European Conference on Computer Vision*, pages 392–409. Springer, 2024.

Stephanie Stoll, Necati Cihan Camgöz, Simon Hadfield, and Richard Bowden. 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, 2018.

Stephanie Stoll, Armin Mustafa, and Jean-Yves Guillemaut. There and back again: 3d sign language generation from text using back-translation. In *2022 International Conference on 3D Vision (3DV)*, pages 187–196. IEEE, 2022.

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*, pages 5630–5638, 2022.

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, 2025.

Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description. *arXiv preprint arXiv:2210.02399*, 2022.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Junke Wang, Yi Jiang, Zehuan Yuan, Bingyue Peng, Zuxuan Wu, and Yu-Gang Jiang. Omnitokenizer: A joint image-video tokenizer for visual generation. *Advances in Neural Information Processing Systems*, 37:28281–28295, 2024.

Fangyun Wei and Yutong Chen. Improving continuous sign language recognition with cross-lingual signs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23612–23621, 2023.

Pan Xie, Qipeng Zhang, Peng Taiying, Hao Tang, Yao Du, and Zexian Li. G2p-ddm: Generating sign pose sequence from gloss sequence with discrete diffusion model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 6234–6242, 2024.

Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

Aoxiong Yin, Tianyun Zhong, Li Tang, Weike Jin, Tao Jin, and Zhou Zhao. Gloss attention for gloss-free sign language translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2551–2562, 2023.

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*, pages 1316–1325, 2021.

Ronglai Zuo, Rolandos Alexandros Potamias, Evangelos Ververas, Jiankang Deng, and Stefanos Zafeiriou. Signs as tokens: A retrieval-enhanced multilingual sign language generator. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23806–23816, 2025.

Table 9: Comparison of **pose back-translation** performance. We additionally report DTW-PA-MPJPE, computed by applying DTW to the per-frame Procrustes-aligned joint position error sequence to obtain a sequence-level distance robust to temporal warping.

Method	Phoenix14T						CSL-Daily					
	BLEU1	BLEU2	BLEU3	BLEU4	ROUGE	DTW↓	BLEU1	BLEU2	BLEU3	BLEU4	ROUGE	DTW↓
Ground-Truth	30.99	18.36	12.83	9.87	31.02	0	31.54	19.32	13.54	9.91	32.73	0
ProTran [Saunders et al., 2020c]	17.96	8.99	5.64	4.07	20.97	16.80	14.85	8.25	6.31	4.85	17.02	19.57
Adversarial [Saunders et al., 2020a]	17.70	8.96	5.72	4.18	21.15	16.43	14.69	8.13	6.55	4.26	16.94	18.81
MDN [Saunders et al., 2021a]	18.06	9.30	6.06	4.52	21.44	14.16	16.17	10.52	7.87	5.66	18.41	15.20
MoMP [Saunders et al., 2021b]	20.55	10.98	7.02	5.14	20.75	12.31	19.27	11.37	7.95	5.94	22.16	12.53
SignGen [Qi et al., 2024]	10.42	2.42	0.89	0.38	12.68	44.60	9.36	1.86	0.41	0.02	11.33	43.95
LLMCond	**22.70**	**11.26**	**7.04**	**5.59**	**23.39**	**9.41**	**22.56**	**13.96**	**9.35**	**6.52**	**23.83**	**8.78**

A More Comparison

Comparison with text-to-pose SLP models. Although our goal is text-to-video generation, we evaluate the accuracy of the generated signs by extracting pose sequences from the generated videos using HRNet and measuring their correctness. We follow ProTran [Saunders et al., 2020c] and train a SLT model [Camgoz et al., 2020] for pose back-translation.

As shown in Table 9, we compare our method with existing text-to-pose sign language production models. LLMCond consistently outperforms previous SLP methods (*e.g.*, ProTran [Saunders et al., 2020c]) across all evaluation metrics. Notably, the SLP baselines generate pose sequences directly from text, making their outputs naturally compatible with our pose back-translation evaluation. In comparison, LLMCond must first produce videos from which poses are then extracted, introducing potential errors that could influence the evaluation. Nevertheless, these results further demonstrate LLMCond's ability to maintain semantic fidelity.

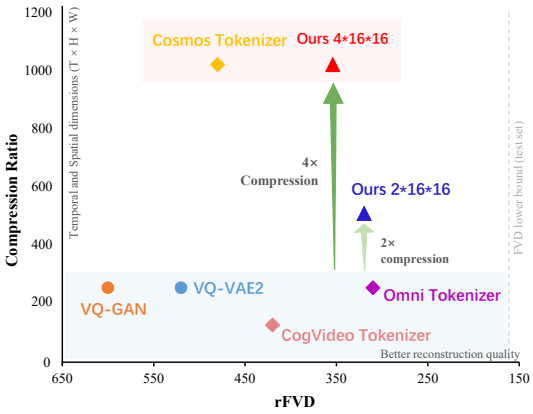


Figure 7: Compression Ratio and rFVD Comparison across Different Tokenizers on Phoenix14T.

Comparison of Video Tokenizer. We further compare our video tokenizer, SignVQ, with representative image/video tokenizers, including VQ-GAN [Esser et al., 2021], VQ-VAE2 [Razavi et al., 2019], Cosmos Tokenizer [Agarwal et al., 2025], Omni Tokenizer [Wang et al., 2024], and CogVideo Tokenizer [Hong et al., 2022]. As shown in Figure 7, we present the trade-off between compression ratio and reconstruction quality measured by rFVD. Compared with VQ-GAN and VQ-VAE2, SignVQ achieves higher compression while maintaining strong reconstruction and generation performance. Relative to Omni Tokenizer and CogVideo Tokenizer, it achieves up to 1024× compression within an acceptable rFVD range. Compared with Cosmos Tokenizer, our method attains higher-quality reconstruction at similar compression levels. Overall, these results demonstrate that our SignVQ tokenizer achieves a favorable balance between compactness and fidelity.

B User Study

In addition to objective metrics (*i.e.*, back-translation accuracy and FVD), we conducted a user study to evaluate the semantic comprehensibility and motion naturalness of the generated sign sequences. We randomly sampled 10 non-overlapping videos from each dataset (Phoenix14T and CSL-Daily), obtaining 20 unique videos in total. Each video was 2–8 seconds long, resized to 256×256 at 25 fps, and shown alongside its ground-truth reference. Ten participants took part in a within-subject design.

For every dataset, each participant rated randomized, method-blind videos from our method and a baseline, presented in random order per video. For each video, raters scored two criteria on an overall 10-point rating scale: (i) semantic consistency with the associated text and (ii) motion naturalness. Video order and method labels were randomized for every participant, and no additional instructions were provided.

As shown in Figure 8, ground-truth videos receive the highest ratings on both datasets, forming the upper bound. Our method (LLMCond) obtains higher scores than SignGAN, with an average improvement of roughly +3.5 points on Phoenix14T and +3 points on CSL-Daily. These results confirm that LLMCond produces more semantically faithful and motion-accurate sign sequences than the baseline.

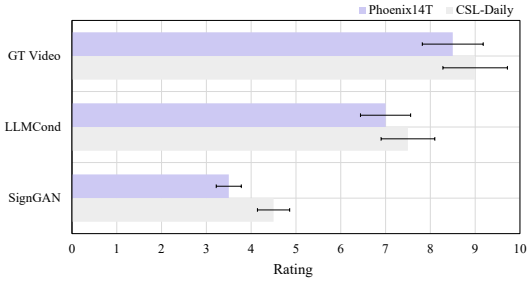


Figure 8: Average user ratings across semantic consistency and motion naturalness on Phoenix14T and CSL-Daily.

C Architecture Details

SignVQ. As shown in Figure 9, the encoder of SignVQ is composed of seven 3D ResNet blocks arranged in a spatial–temporal–spatial order. The first four blocks are spatial 3D ResNet blocks (kernel size $1 \times 3 \times 3$) that progressively downsample the feature maps along the height and width dimensions, resulting in an overall $16\times$ reduction in spatial resolution. They are followed by two temporal ResNet blocks ($3 \times 1 \times 1$), which further downsample the video by a factor of 4 along the temporal dimension. Finally, an additional spatial 3D ResNet block ($1\times3\times3$) is employed to further reduce the spatial resolution and produce a compact latent representation that preserves high-level spatial semantics. The decoder mirrors this spatial–temporal–spatial structure using De3D (3D deconvolutional) ResNet blocks. Starting from the quantized latents, a spatial De3D block operates at the coarsest resolution, two temporal De3D blocks upsample the sequence in time, and the final four spatial De3D blocks upsample the feature maps in space to reconstruct the full-resolution video.

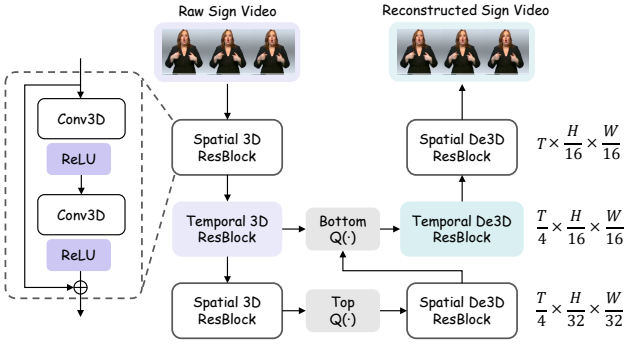


Figure 9: The architecture of the proposed SignVQ tokenizer. Left: encoder that downsamples the sign video into top and bottom discrete latents. Right: decoder that upsamples these latents with De3D ResBlocks to reconstruct the video.

LLM Conditioner. The LLM Conditioner is initialized with the pretrained weights of the Llama 3.2 1B model from HuggingFace, to fully leverage the rich linguistic knowledge of large language models. We apply LoRA adaptation with a rank of 64 and a scaling factor (α) of 32.

VQ-Diffusion. Following the open-sourced VQ-Diffusion, we adopt the pretrained VQ-Diffusion weights on ImageNet [4] for initialization.

Reference Image Encoder. We adopt a ResNet-18 backbone as the reference image encoder to extract reference image embeddings. During training, the reference frame is randomly selected from the input frame sequence, while during inference, it is fixed as the first frame of the generated sign language video.

Condition2Visual Alignment. Before the condition embeddings are fed into the VQ-Diffusion model, a Condition2Visual Mapper is employed to align feature dimensions between the LLM

[4] https://github.com/microsoft/VQ-Diffusion/tree/main

Conditioner and the diffusion model. The Condition2Visual Mapper consists of two MLP layers with a ReLU activation in between. We omit its description from the main paper for clarity of presentation.

Video Length Prediction. We train a lightweight model to predict the target video length based on the text embeddings. The Video Length Predictor takes the output embeddings from the Llama embedding layer as input and consists of four Transformer encoder layers followed by an MLP layer to estimate the final video length. We omit its description from the main paper for clarity of presentation.

Training Details. The entire implementation is based on PyTorch 2.1.
(1) SignVQ. The SignVQ module is trained on a single NVIDIA A6000 GPU with 48 GB for 50 epochs using the Adam optimizer with an initial learning rate of 1×10^{-4}. A MultiStepLR scheduler is applied with a decay factor of 0.5 at epochs 1, 3, 10, 30, and 35.
(2) LLM Conditioner and VQ-Diffusion. The LLM Conditioner and VQ-Diffusion modules are trained jointly using Adam [Kingma, 2014] with an initial learning rate of 1×10^{-5}, a batch size of 3, and 20k training steps per stage. The learning rate follows a MultiStepLR schedule with a decay factor of 0.5 at epochs 3, 10, and 30. All experiments are conducted on four NVIDIA A6000 GPUs. For generation, we follow the standard VQ-Diffusion framework [Gu et al., 2022] with classifier-free guidance [Ho and Salimans, 2022] (guidance scale set to 3.5) and use 100 denoising steps during inference.
(3) Back-Translation Model. To evaluate the semantic consistency of the generated sign language videos, we adopt the SLT model [Gan et al., 2025] to translate sign language videos (*i.e.*, the video back-translation model) into texts. The video back-translation model is trained using a single NVIDIA RTX A6000 GPU. An Adam optimizer [Kingma, 2014] is utilized with a learning rate of 1e-3, which decays by a factor of 0.5 every 10 epochs, and the batch size is set to 8.

D More Qualitative Results

We present additional qualitative examples to further demonstrate the effectiveness of our proposed method. As illustrated in Figure 10 and Figure 11, we provide more generated samples from both the Phoenix14T and CSL-Daily datasets.

E Limitations and Broader impacts

In this paper, we propose a novel end-to-end SLVG model, LLMCond, which achieves notable performance improvements. Nevertheless, our framework still has two limitations that warrant further investigation. The first limitation lies in its data-driven nature. Since the model learns patterns directly from the training data, it may inherit dataset biases and occasionally produce uncontrolled or inconsistent video content. Ensuring careful dataset curation and maintaining a balanced data distribution are therefore crucial to mitigating such issues. The second limitation lies in generalization. Although our model performs well on benchmark datasets, its robustness to varying lighting conditions and diverse signing styles remains to be further validated.

This paper presents work whose goal is to advance the field of machine learning for sign language video generation. By improving end-to-end generation of sign language videos, our work has the potential to support the development of more accessible communication technologies for the Hard-of-Hearing community. At the same time, we acknowledge that sign language technologies may inherit biases present in training data, including variations across signers, regions, and signing styles. While our work focuses on modeling and representation learning rather than deployment, responsible use of such systems should consider data diversity, fairness, and appropriate human oversight. Overall, we believe the primary impact of this work is positive, contributing to more inclusive and efficient sign language research, and we do not foresee significant negative societal consequences beyond those commonly associated with machine learning research.



Figure 10: More generated cases of Phoenix14T dataset.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



Figure 11: More generated cases of CSL-Daily dataset.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly summarize the main contributions of the paper, including the proposal of LLM-based conditioner and the end-to-end SLVG models. (see Section 1).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We include a dedicated "Limitations and Broader impacts " section in the appendix of main paper, where we discuss practical challenges, and the potential negative social impacts including high resource demands, data bias, and model reliability in real-world applications. (see Section E).

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: Our work focuses on empirical model design and evaluation for sign language video generation. It does not include theoretical results that require formal assumptions or proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide comprehensive details of the model architecture, training settings, and evaluation protocol in Section 4. Specifically, we describe the datasets used, data preprocessing steps, model configurations, hyperparameters, and optimization strategies. We also report implementation details such as training schedules, batch sizes, and evaluation metrics to ensure reproducibility. Furthermore, additional model details and implementation specifications are provided in the supplementary materials. We will also release the code to further facilitate reproducibility.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The paper and supplementary material provide sufficient implementation details, training settings, and experimental configurations to facilitate reproduction of the main results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper provides comprehensive training and testing details, including data splits, model architecture, hyperparameters, learning rate, optimizer, and training epochs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: While we report the main experimental results, we do not include error bars or statistical significance tests due to limited computational resources. In future work, we plan to provide more extensive statistical analysis over multiple random seeds or cross-validation splits.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper specifies the type of computing resources used, including GPU model (e.g., A6000 GPU), memory (e.g., 48GB) (See Section C).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms in every respect with the NeurIPS Code of Ethics. We have ensured transparency, reproducibility, and fairness in our methodology and reporting.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Our work discusses both potential positive and negative societal impacts.(See Section E)

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The model weights used in this work are trained solely on publicly available datasets, and we do not release any new datasets or pretrained models that pose a high risk for misuse. Therefore, no additional safeguards are necessary.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We used publicly available datasets and pretrained model weights, all of which are properly cited in the paper. The licenses and terms of use for these assets were carefully reviewed and respected. We have also specified the sources, versions, and relevant URLs where appropriate.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: This paper does not release any new datasets, models, or code assets. All experiments are based on publicly available datasets and pretrained models, which are properly cited.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [Yes]

Justification: Appendix B provides the user study protocol, participant instructions, evaluation criteria, and compensation details.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification: The user study underwent an equivalent internal ethics review in accordance with applicable institutional requirements.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLMs are a core component of our method, serving as a condition generator that transforms text into structured signals for sign language video generation. Their role is explicitly described in both the main paper and supplementary materials.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.