SeRV: Semantic-Aligned Residual Vector Quantization for American Sign Language Generation

Anonymous ACL submission

Abstract

American Sign Language (ASL) generation remains challenging due to limited paired text-ASL motion data and the difficulty of learning motion representations both precise for reconstruction and predictable from linguistic input. Existing methods rely on motion tokenizers optimized for reconstruction, without explicit semantic supervision from paired text. As a result, the learned tokens remain limited in supporting semantically consistent and fine-grained ASL motion generation. To address this limitation, we propose **SeRV** (**Se**mantic-Aligned **R**esidual **V**ector Quantization), a semantic-aligned RVQ tokenizer for ASL generation. SeRV learns a semantically structured residual token space by combining sentence-level motion-text alignment with token-level text-conditioned supervision. Building on this tokenizer, a Hierarchical GPT predicts residual motion tokens in a coarse-to-fine manner, generating structurally coherent and semantically aligned 3D ASL motion. We further construct a large-scale reconstructed 3D ASL motion–text benchmark by recovering paired 3D motion from YouTube-ASL videos. Experiments across 375 hours of ASL video show that SeRV achieves state-of-the-art pose accuracy on both How2Sign and YouTube-ASL datasets, while producing semantically consistent 3D ASL motion directly from text.

1 Introduction

American Sign Language (ASL) is the primary communication method of Deaf and Hard-of-Hearing (DHH) communities, but the lack of fluent signers in the United States creates substantial communication barriers. To bridge this gap, recent AI research has focused on Sign Language Translation (SLT) (Jang et al., 2025; Li et al., 2025; Wong et al., 2024) and Sign Language Generation (SLG) (Zuo et al., 2025; Yu et al., 2024; Fang et al., 2025; Yin et al., 2024). Some SLG methods formulate the task as visual content generation (Saunders et al., 2022; Fang et al., 2023), which often underrepresents the fine-grained motion patterns that carry critical linguistic information. Motivated by the linguistic nature of sign language, tokenization-based SLG has recently attracted growing interest (Yin et al., 2024; Zuo et al., 2025), where continuous sign motions are converted into discrete tokens. However, many existing methods still rely on an external sign dictionary and a retrieval step to assemble motion (Zuo et al., 2025), which limits generation to a fixed vocabulary rather than producing sign motion directly from text. A more fundamental limitation lies in the tokenizer itself. ASL meaning is carried by fine-grained hand articulation, where small differences distinguish different signs. Existing tokenizers are trained purely for reconstruction, so the token space has no alignment to the meaning of the conditioning text. The token space is thus hard to predict from text: two signs with distinct meanings can map to nearly the same tokens, and the generator must learn a tangled text-to-token mapping, producing motion that looks natural but conveys the wrong sign. Current 3D ASL resources (Duarte et al., 2021; Yu et al., 2024) are also limited in scale, vocabulary, and motion diversity.

In this work, we address these limitations by redesigning the motion tokenizer and downstream generator for 3D ASL generation. We propose **SeRV** (**Se**mantic-Aligned **R**esidual **V**ector Quantization), which reshapes the residual quantization hierarchy with linguistic supervision rather than reconstruction alone, anchoring fine-grained hand articulation to text-relevant semantics so that these residual codes become predictable from language. To be specific, SeRV is built on a Residual Vector-Quantized Variational AutoEncoder (RVQ-VAE) (Lee et al., 2022; Guo et al., 2024), where ASL motion is represented by a coarse-to-fine hierarchy of residual codebooks. Each level progressively refines the motion representation from

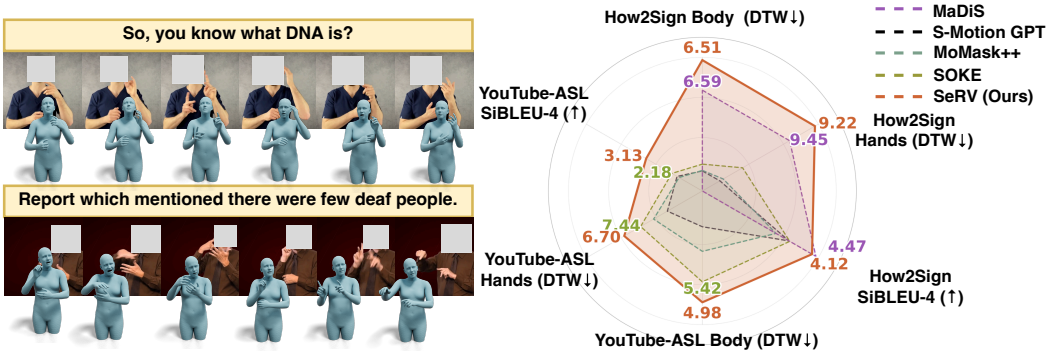

Figure 1: *Left*: ASL generated by SeRV from text for two example sentences, rendered as 3D avatars alongside the reference signer. Right: comparison with state-of-the-art methods on six metrics across How2Sign and YouTube-ASL datasets (body/hand DTW-JPE, lower better; SiBLEU-4, higher better; all axes oriented so larger area = better). SeRV achieves the best result on five of six metrics. **Note MaDiS (Zuo et al., 2026) is concurrent work reporting How2Sign only; its YouTube-ASL axes are drawn at the center to indicate results are unavailable, not zero.**

coarse global structure to fine-grained hand and upper-body articulations, reducing approximation error while preserving motion detail. SeRV trains the encoder with two semantic alignment objectives: a sentence-level alignment on the base-level (coarsest) tokens, and a token-level motion–text alignment on the fully quantized latent. This drives coarse tokens to encode sentence-level semantics while finer residuals refine articulatory details. Building on this token space, we further propose Hierarchical GPT, an RVQ-aware autoregressive generator that mirrors SeRV's coarse-to-fine hierarchy. At each residual level, it predicts tokens conditioned on per-token text features and the previously decoded lower-level motion tokens. Unlike flat single-sequence prediction (Guo et al., 2025a), which inflates sequence length and weakens the residual hierarchy, or two-stage schemes (Guo et al., 2024) that decouple coarse and fine prediction, this progressive design preserves the hierarchy and produces structurally coherent, semantically consistent ASL motion.

To support scalable token learning, we further expand ASL training resources by reconstructing paired 3D sign motion from YouTube-ASL (Uthus et al., 2023) videos using a state-of-the-art motion reconstruction pipeline (Sun et al., 2024), followed by quality-based cleaning and filtering. To the best of our knowledge, this results in the largest reconstructed 3D ASL motion-text benchmark to date, providing richer supervision for token learning. As illustrated in Figure 1, SeRV generates 3D ASL motions with faithful hand articulation and achieves the best result on five of six reported metrics across How2Sign and YouTube-ASL. Note MaDiS (Zuo et al., 2026) is concurrent work released shortly before our submission; as its code and checkpoints are not publicly available, we cannot evaluate it on the larger YouTube-ASL benchmark under the same protocol. Our contributions are summarized as follows:

- We propose **SeRV**, a semantic-aligned RVQ framework for text-to-3D ASL generation. SeRV combines sentence-level and token-level motion–text alignment to learn a residual token space that is precise for reconstruction and predictable from text, and instantiates a hierarchy-aware autoregressive generator to synthesize ASL motion following the learned coarse-to-fine token hierarchy.
- We construct the largest reconstructed 3D ASL motion-text benchmark to date by reconstructing 3D sign motion from YouTube-ASL videos and curating it at scale, providing broad signer and linguistic diversity for training generative ASL models.
- Extensive experiments on How2Sign and YouTube-ASL show that SeRV substantially outperforms prior text-to-3D ASL generation methods, achieving state-of-the-art pose accuracy on both benchmarks and strong sign-token consistency.

2 Related Work

Sign Language Generation (SLG). Early SLG methods often adopt cascaded pipelines, bridging text and sign motion via glosses, skeletons, or pose sequences (Stoll et al., 2020; Zelinka and Kanis,

2020). These intermediate representations provide linguistic or structural guidance but require costly annotations and risk information bottlenecks or error accumulation (Saunders et al., 2021; Zuo et al., 2025, 2026). Others generate continuous sign poses or videos directly from text via recurrent networks (Saunders et al., 2020, 2021), Transformers (Saunders et al., 2022; Qi et al., 2024; Fang et al., 2025), GANs (Stoll et al., 2020), or diffusion models (Fang et al., 2023). However, operating in a continuous output space makes long-horizon generation harder to stabilize and offers limited discrete structure for scalable autoregressive modeling. Recent SLG methods tokenize 3D sign motion into discrete codes. Early efforts use pose-level VQ-VAE (Xie et al., 2024b) or VQ-GAN-style video tokenizers (Esser et al., 2021; Xie et al., 2024a). More recent work targets 3D motion: SignAvatars (Yu et al., 2024) pairs a large-scale 3D holistic sign dataset with a VQ-VAE codebook; T2S-GPT (Yin et al., 2024) uses dynamic vector quantization for adaptive-length codes; SOKE (Zuo et al., 2025) tokenizes upper-body, left-hand, and right-hand separately and generates multilingual signs via a pretrained LM with dictionary retrieval; MaDiS (Zuo et al., 2026) combines part-wise tokens with masked diffusion language modeling and tri-level cross-modal pretraining; and SignViP (Wang et al., 2025) extends tokenization to sign video generation with multi-condition tokens.

ASL Datasets. How2Sign (Duarte et al., 2021) offers 80+ hours of in-studio continuous ASL with aligned RGB, depth, and English transcripts. YouTube-ASL (Uthus et al., 2023) scales to 984 hours of in-the-wild video from 2,500+ signers, broadening vocabulary and visual diversity.

To address these limitations, we construct a large-scale reconstructed 3D ASL motion–text benchmark and propose **SeRV**, a semantic-aligned RVQ tokenizer that learns residual tokens both precise for reconstruction and predictable from text, enabling fine-grained articulation control for ASL generation.

3 Methodology

The overview of our framework is shown in Figure 2 and Figure 3. Our method has two stages: **SeRV** (**Se**mantic-Aligned **R**esidual **V**ector Quantization), a text-aligned RVQ tokenizer, and Hierarchical GPT, a coarse-to-fine token generator. SeRV converts continuous 3D ASL motion into residual discrete tokens while aligning the motion representations with the paired transcript. Hierarchical GPT then predicts these residual tokens from text in a coarse-to-fine autoregressive order, and the predicted tokens are decoded by the learned motion decoder to reconstruct the final 3D ASL motion.

3.1 Preliminaries of RVQ-VAE

Given a sign motion sequence $\mathbf{S}$, a motion encoder $\mathcal{E}$ produces a continuous latent sequence $\mathbf{z} = \mathcal{E}(\mathbf{S}) \in \mathbb{R}^{T' \times C}$, with temporal length T' and feature dimension C. A codebook $\mathcal{C} = \{\mathbf{e}_m\}_{m=0}^{K-1}$ is a learnable set of K embedding vectors; each latent vector is quantized to its nearest code, and the selected index serves as a discrete token. We adopt residual vector quantization with L codebooks $\{\mathcal{C}_l\}_{l=0}^{L-1}$ (Guo et al., 2024; Lee et al., 2022). Starting from $\mathbf{r}^{(0)} = \mathbf{z}$, stage l computes $\mathbf{q}^{(l)} = Q^{(l)}(\mathbf{r}^{(l)})$ and updates the residual $\mathbf{r}^{(l+1)} = \mathbf{r}^{(l)} - \mathbf{q}^{(l)}$, where $Q^{(l)}(\cdot)$ is nearest-neighbor quantization with $\mathcal{C}_l$. The selected index $k_t^{(l)}$ at position t is the discrete motion token. The final quantized latent is $\hat{\mathbf{z}} = \sum_{l=0}^{L-1} \mathbf{q}^{(l)}$, decoded to $\hat{\mathbf{S}} = \mathcal{D}(\hat{\mathbf{z}})$ by the motion decoder $\mathcal{D}$. The RVQ-VAE is trained with a reconstruction loss and a per-level embedding constraint:

$$\mathcal{L}_{\text{RVQ}} = \mathcal{L}_{\text{rec}} + \beta \sum_{l=0}^{L-1} \left\| \mathbf{r}^{(l)} - \text{sg}\left[\mathbf{q}^{(l)}\right] \right\|_2^2, \quad (1)$$

where $\mathcal{L}_{\text{rec}} = \|\mathbf{S} - \hat{\mathbf{S}}\|_1$, $\text{sg}[\cdot]$ is the stop-gradient operator, and β weights the embedding constraint.

3.2 SeRV Model Architecture

Unlike a reconstruction-only RVQ-VAE, SeRV adds semantic supervision from paired transcripts during tokenizer training. As illustrated in Figure 2, a frozen T5 encoder (Raffel et al., 2020) provides both sentence-level and token-level text features from the same transcript. We align the sentence-level embedding with the level-0 quantized latent $\mathbf{q}^{(0)}$, a coarse content representation of the ASL motion, and use the token-level features in a motion–text late interaction loss over quantized motion latents. These losses inject global and token-level textual supervision into the tokenizer.

Global-Level Sign–Text Alignment. We align $\mathbf{q}^{(0)}$ with the paired text representation so that the learned ASL motion tokens retain text-relevant semantics. Given the paired transcript $\mathbf{c}$, a frozen pretrained T5 encoder (Raffel et al., 2020) $\mathcal{E}\text{text}$

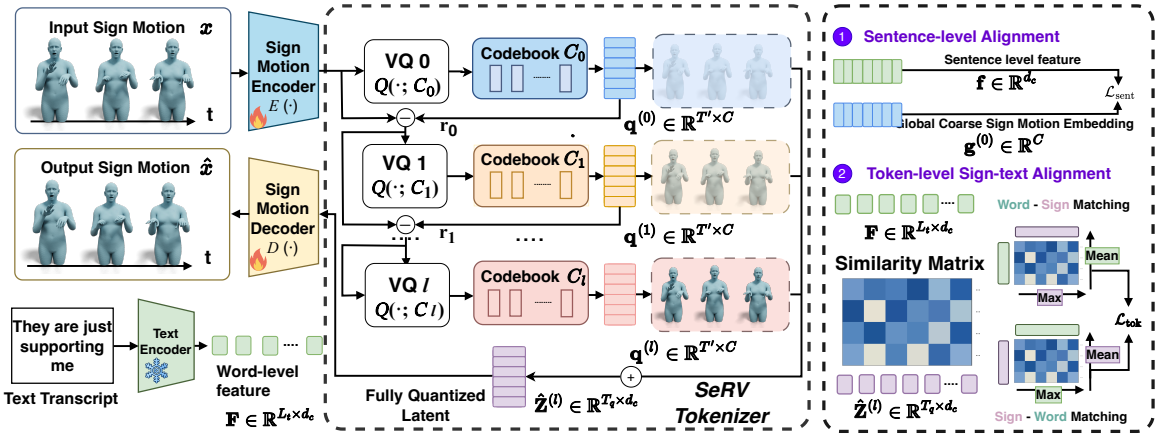

Figure 2: Overview of SeRV. We introduce a global-level loss $\mathcal{L}_{\text{global}}$ on the level-0 motion representation and a token-level loss $\mathcal{L}_{\text{tok}}$ on the fully quantized latent sequence. Together with the standard RVQ tokenizer objective $\mathcal{L}_{\text{RVQ}}$, these objectives encourage the learned discrete motion tokens to preserve motion fidelity while retaining text-relevant semantics.

produces token-level text features $\mathbf{F} = \mathcal{E}\text{text}(\mathbf{c}) \in \mathbb{R}^{L_t \times d_c}$, where L_t is the tokenized text length and d_c is the T5 hidden dimension. We use the average hidden state over non-padded tokens as the sentence-level representation $\mathbf{f} \in \mathbb{R}^{d_c}$. On the motion side, we summarize $\mathbf{q}^{(0)}$ into a global motion embedding $\mathbf{g}^{(0)}$ using temporal mean pooling. The motion and text embeddings are then projected into a shared semantic space and ℓ_2-normalized. For a mini-batch of size B, we apply a symmetric global-level alignment loss $\mathcal{L}_{\text{global}} = \frac{1}{2}\big(\mathcal{L}^{\text{global}}_{m\to t} + \mathcal{L}^{\text{global}}_{t\to m}\big)$. Let $s^{(0)}_{ij} = \text{sim}\big(\hat{\mathbf{g}}^{(0)}_i, \hat{\mathbf{f}}_j\big)$ denote the cosine similarity between the i-th motion embedding and the j-th text embedding. The two directional losses are then defined as

$$\mathcal{L}^{\text{global}}_{m\to t} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp\big(s^{(0)}_{ii}/\tau\big)}{\sum_{j=1}^{B}\exp\big(s^{(0)}_{ij}/\tau\big)}, \quad (2)$$

$$\mathcal{L}^{\text{global}}_{t\to m} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp\big(s^{(0)}_{ii}/\tau\big)}{\sum_{j=1}^{B}\exp\big(s^{(0)}_{ji}/\tau\big)}. \quad (3)$$

Here, $\hat{\mathbf{g}}^{(0)}_i$ and $\hat{\mathbf{f}}_i$ denote the projected and ℓ_2-normalized level-0 motion and text embeddings, and τ is the temperature. The matched motion–text pair is treated as the positive pair, while the other samples in the mini-batch serve as negatives.

Token-Level Sign–Text Alignment. Sentence level contrastive loss provides global alignment but leaves token-level text–motion interactions implicit. We therefore introduce a late-interaction loss (Yao et al., 2021) that averages token-wise maximum similarities between text tokens and quantized ASL motion latents. For text features, we reuse the T5 token hidden states $\mathbf{F} \in \mathbb{R}^{L_t \times d_c}$; for motion features, we use the fully quantized latent $\hat{\mathbf{z}} = \sum_{l=0}^{L-1} \mathbf{q}^{(l)} \in \mathbb{R}^{T' \times C}$. Separate token-level projection heads map them into a shared normalized space, producing $\tilde{\mathbf{F}} \in \mathbb{R}^{L_t \times d_a}$ and $\tilde{\mathbf{Z}} \in \mathbb{R}^{T' \times d_a}$. For each ASL motion–text pair, we compute a token-wise cosine similarity matrix $\mathbf{A} \in \mathbb{R}^{L_t \times T'}$, where $\mathbf{A}_{t,m} = \text{sim}(\tilde{\mathbf{F}}_t, \tilde{\mathbf{Z}}_m)$ measures the similarity between the t-th text token and the m-th motion latent. Let Ω_F and Ω_Z denote valid text-token and motion-latent positions. We define the late-interaction scores as $s_{F\to Z} = \frac{1}{|\Omega_F|}\sum_{u\in\Omega_F} \max_{v\in\Omega_Z} \mathbf{A}_{u,v}$ and $s_{Z\to F} = \frac{1}{|\Omega_Z|}\sum_{v\in\Omega_Z} \max_{u\in\Omega_F} \mathbf{A}_{u,v}$. Each valid token is therefore matched to its closest token in the other modality.

Given a mini-batch of size B, let $s^{m\to t}_{ij}$ and $s^{t\to m}_{ij}$ denote the motion-to-text and text-to-motion scores between the motion sequence of sample i and the text sequence of sample j. We apply a symmetric token-level InfoNCE loss $\mathcal{L}_{\text{tok}} = \frac{1}{2}(\mathcal{L}^{\text{tok}}_{m\to t} + \mathcal{L}^{\text{tok}}_{t\to m})$. The two directional losses are defined as

$$\mathcal{L}^{\text{tok}}_{m\to t} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(s^{m\to t}_{ii}/\tau')}{\sum_{j=1}^{B}\exp(s^{m\to t}_{ij}/\tau')}, \quad (4)$$

$$\mathcal{L}^{\text{tok}}_{t\to m} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(s^{t\to m}_{ii}/\tau')}{\sum_{j=1}^{B}\exp(s^{t\to m}_{ji}/\tau')}. \quad (5)$$

Here, τ' is the token-level temperature.

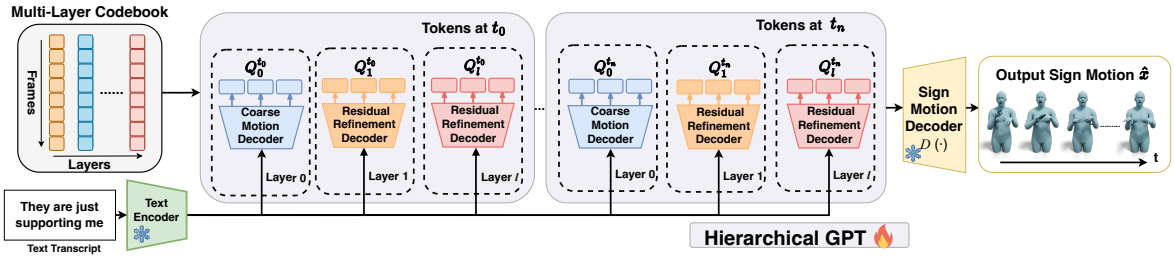

Figure 3: Hierarchical GPT for coarse-to-fine ASL motion generation. Given the text feature from the frozen condition encoder, the generator autoregressively predicts SeRV tokens over time and RVQ levels. For each time step, a coarse decoder first predicts the level-0 token, while residual decoders progressively predict higher-level tokens conditioned on the temporal history, the generated coarser tokens, and the text feature. The predicted multi-level tokens are then decoded by the shared motion decoder to reconstruct the ASL motion sequence.

Training Objective. The overall training objective of SeRV is defined as

$$\mathcal{L}_{SeRV} = \mathcal{L}_{\text{RVQ}} + \lambda_{\text{global}}\mathcal{L}_{\text{global}} + \lambda_{\text{tok}}\mathcal{L}_{\text{tok}}. \quad (6)$$

Here, $\mathcal{L}_{\text{RVQ}}$ denotes the standard RVQ tokenizer objective, $\mathcal{L}_{\text{global}}$ aligns the coarse level-0 motion representation with sentence-level text semantics, while $\mathcal{L}_{\text{tok}}$ encourages fine-grained interactions between text tokens and quantized ASL motion latents. The coefficients λ_{global} and λ_{tok} balance the two semantic alignment objectives.

3.3 Hierarchical GPT

Hierarchy-aligned generation. We propose Hierarchical GPT, an RVQ-aware autoregressive generator that predicts SeRV tokens over both time and quantization depth. We reuse the frozen text encoder and denote its output as $\mathbf{f} = \mathcal{E}_{\text{cond}}(\mathbf{c})$, where $\mathbf{c}$ is the input text. Let $\mathbf{k}_{1:T_q}^{0:L-1}$ be the multi-level token hierarchy produced by SeRV, where L is the number of RVQ levels, T_q is the token sequence length, and k_t^l is the token at time t and level l. Following the text-aligned hierarchy learned by SeRV, the generator predicts lower-level coarse tokens before higher-level residual tokens:

$$p(\mathbf{k}_{1:T_q}^{0:L-1} \mid \mathbf{f}) = \prod_{t=1}^{T_q}\prod_{l=0}^{L-1} p\Big(k_t^l \mid \mathbf{k}_{<t}^{0:L-1}, \mathbf{k}_t^{0:l-1}, \mathbf{f}\Big), \quad (7)$$

where $p(\cdot)$ is the categorical distribution over codebook entries, $\mathbf{k}_{<t}^{0:L-1}$ denotes all tokens before time t, and $\mathbf{k}_t^{0:l-1}$ denotes coarser tokens already generated. This factorization preserves temporal causality and coarse-to-fine dependencies.

Coarse-to-fine decoding. We implement Eq. 7 with a coarse token decoder followed by residual token decoders. At each time step t, the coarse decoder predicts the level-0 token k_t^0 from the previous token history and the text feature $\mathbf{f}$, capturing the coarse text-conditioned motion pattern. The coarse decoder is a causal Transformer built with LLaMA-style blocks, including RMSNorm, SwiGLU, and rotary positional embeddings (RoPE) (Touvron et al., 2023).

For each higher level $l > 0$, a residual decoder predicts k_t^l conditioned on the previous token history $\mathbf{k}_{<t}^{0:L-1}$, the coarser tokens $\mathbf{k}_t^{0:l-1}$ already generated at the current time step, and the text feature $\mathbf{f}$. The embeddings of these coarser tokens are fed to the residual decoder, allowing higher levels to refine the current coarse representation rather than generate the motion state from scratch. Since residual decoders mainly model refinement details, we use fewer Transformer layers while retaining attention layers for fine-grained articulation modeling. This decoding process follows the coarse-to-fine hierarchy learned by SeRV.

Training objective. Let $\hat{\mathbf{p}}_t^l$ denote the predicted token distribution at level l and time step t, and let k_t^l be the corresponding ground-truth token from the frozen SeRV tokenizer. The generator is trained with a level-wise cross-entropy loss:

$$\mathcal{L}_{\text{HGPT}} = \sum_{l=0}^{L-1} \lambda_l \frac{1}{T_q}\sum_{t=1}^{T_q} \text{CE}(\hat{\mathbf{p}}_t^l, k_t^l), \quad (8)$$

where T_q is the token length, λ_l is the loss weight for level l, and $\text{CE}(\cdot,\cdot)$ denotes cross-entropy.

4 Performance Evaluation

4.1 Experimental Methodology

Datasets. We evaluate on two ASL datasets, How2Sign (Duarte et al., 2021) and YouTube-ASL (Uthus et al., 2023). For How2Sign, we use

Table 1: Comparison with state-of-the-art methods on How2Sign and YouTube-ASL. Lower is better for DTW-JPE, and higher is better for SiBLEU-4.

Method	How2Sign			YouTube-ASL		
	Body↓	Hands↓	SiBLEU-4↑	Body↓	Hands↓	SiBLEU-4↑
S-MotionGPT* (Jiang et al., 2023)	12.41	13.74	2.32	8.95	10.12	1.84
MoMask++* (Guo et al., 2025b)	9.06	11.34	1.53	6.54	8.35	1.67
SOKE† (Zuo et al., 2025)	7.75	10.08	2.37	5.42	7.44	2.18
MaDiS (Zuo et al., 2026)	6.59	9.45	**4.47**	–	–	–
Ours	**6.51**	**9.22**	4.12	**4.98**	**6.70**	**3.13**

Notes. Body and Hands are DTW-JPE errors; Hands denotes the average over left and right hands. * denotes general motion-generation baselines adapted to sign motion generation, and † denotes methods using external sign dictionaries. How2Sign results for prior methods are taken from reported results (Zuo et al., 2025, 2026). YouTube-ASL results are obtained under the same SMPL-X representation, split, and evaluation protocol. MaDiS is not reported on YouTube-ASL because its official code and checkpoints are unavailable.

its SMPL-X motion annotations and the official train/validation/test split. For YouTube-ASL, we reconstruct 3D SMPL-X signing motion from these videos and filter out low-quality clips with unreliable reconstructions or text-motion misalignment. For both datasets, we represent sign motion as sequences of SMPL-X parameters covering upper-body and hand joints. Full preprocessing, filtering criteria, and dataset statistics are provided in Appendix A.2.

Evaluation metrics & Implementation. Since generated sign sequences may differ in length from the ground truth, we follow prior 3D sign language generation work and use Dynamic Time Warping over Joint Position Error (DTW-JPE) (Zuo et al., 2025; Cai et al., 2019) to measure sequence-level motion distance. DTW temporally aligns the generated and reference sequences before computing joint-position errors. We report DTW-JPE for the body, left hand, and right hand, and compute Hand Avg. by averaging the left- and right-hand errors. For tokenizer reconstruction, where the generated motion is frame-aligned with the input, temporal alignment is unnecessary; we instead report PA-MPJPE, the Mean Per-Joint Position Error after rigid (Procrustes) Alignment. Following MaDiS (Zuo et al., 2026), we also report SiBLEU-4 to measure token-level consistency in the sign-token space. We discretize both generated and ground-truth motions with the same frozen evaluator tokenizer and compute BLEU-4 over the resulting token sequences. Formal definitions of all metrics are provided in Appendix A.3.

We train the framework in two stages. SeRV uses $L = 3$ RVQ levels, latent dimension $C = 512$, and codebook size $K = 512$ per level. Hierarchical GPT uses a hidden dimension of 1024, 9 Transformer layers, and 16 attention heads. Full details are in Appendix A.4.

4.2 Comparison with State-of-the-Art Methods

We compare our method against both general motion-generation models adapted to sign motion and recent sign-specific generation methods. The general motion generators include S-MotionGPT (Jiang et al., 2023), an autoregressive token-based model, and MoMask++ (Guo et al., 2025b), a masked generative token model, both adapted to 3D sign motion under our SMPL-X representation. The sign-specific methods include SOKE (Zuo et al., 2025), a strong autoregressive baseline with part-wise motion tokenization and external dictionary retrieval, and MaDiS (Zuo et al., 2026), a recent masked diffusion model for sign generation. Since MaDiS is not open-sourced, we are unable to evaluate it on YouTube-ASL, and we therefore report only its How2Sign results from the original paper. As shown in Table 1, general motion generators adapted to sign perform substantially worse than sign-specific methods, confirming that ASL generation requires modeling tailored to its fine-grained articulatory structure. On YouTube-ASL, our method achieves the best results on all metrics, lowering the body and hand DTW-JPE of the strongest baseline, SOKE, by 8.1% and 9.9%, and raising SiBLEU-4 from 2.18 to 3.13. On How2Sign, our model attains the lowest body and hand errors among all methods; MaDiS reports a higher SiBLEU-4 (4.47 vs. 4.12), although, as noted above, it cannot be assessed on the larger YouTube-ASL benchmark due to the lack of a public release. Notably, our model surpasses SOKE on every metric without relying on an external sign dictionary, indicating that the improvements stem from the learned semantic-aligned token space rather than from retrieval. These results demonstrate the benefit of semantic-aligned residual tokenization for ASL generation.

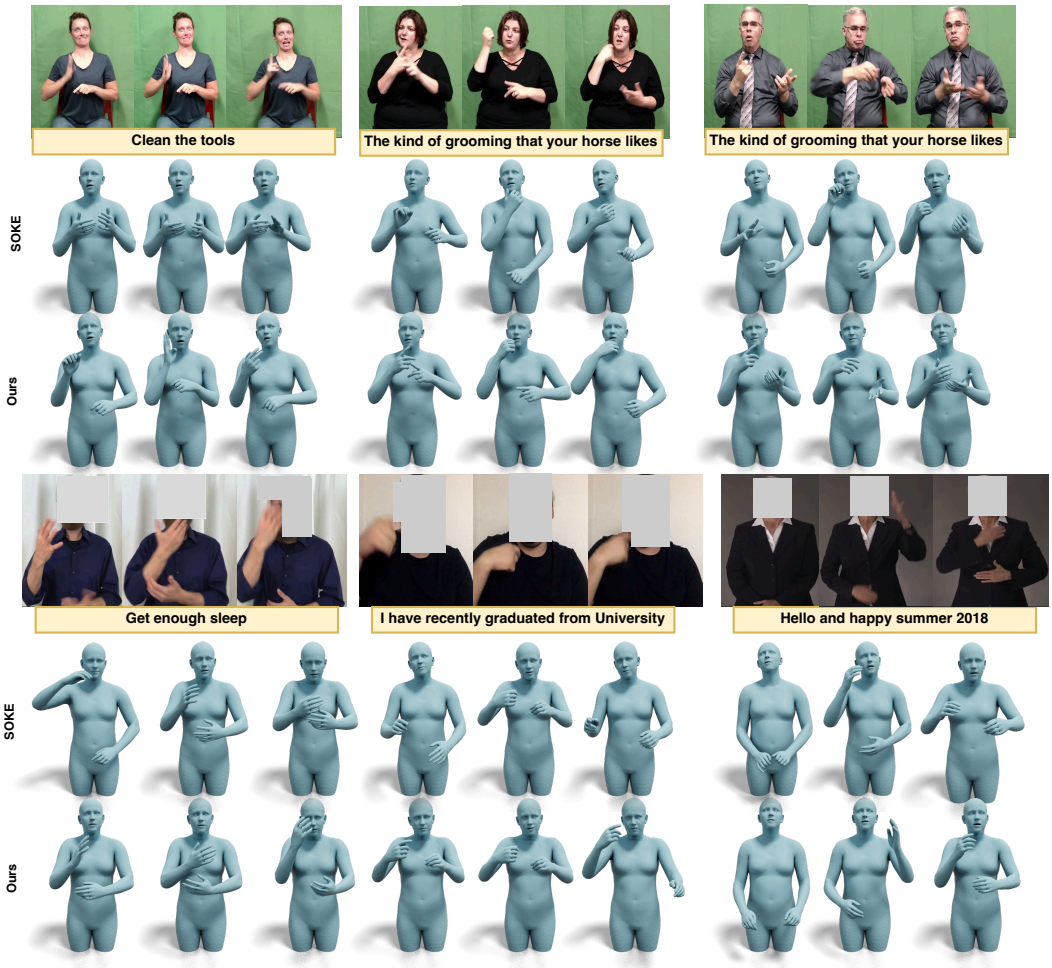

Figure 4: Qualitative comparison of text-to-3D ASL generation. For each of six sentences, we show the reference signer video (top) and the 3D motion generated by SOKE and by our method (SeRV). The top three examples are from How2Sign; the bottom three are from YouTube-ASL (faces masked for privacy).

Performance of Tokenizer Reconstruction. We further evaluate whether the proposed tokenizer preserves the input 3D motion. As shown in Table 2, SeRV achieves the lowest reconstruction errors on both How2Sign and YouTube-ASL. Compared with the standard VQ-VAE and the part-wise Multi-VAE tokenizer used in SOKE, SeRV consistently reduces body and hand reconstruction errors, indicating that the proposed semantic alignment does not compromise motion fidelity. The gains are especially important for downstream generation, as a more accurate residual token space provides cleaner targets for predicting both coarse body motion and fine-grained hand articulation.

Table 2: Reconstruction performance of different motion tokenizers on How2Sign and YouTube-ASL.

Method	How2Sign(PA-MPJPE↓)		YouTube-ASL(PA-MPJPE↓)	
	Body↓	Hands↓	Body↓	Hands↓
VQ-VAE	24.07	7.96	19.53	4.45
Multi-VAE	19.37	6.65	16.96	3.62
SeRV	**16.23**	**6.13**	**13.91**	**3.31**

4.3 Qualitative Comparison

We provide qualitative comparisons in Figure 4. For each input transcript, we show the 3D signing motion generated by SOKE and SeRV. Compared with SOKE, SeRV produces more accurate hand configurations, especially for the fine-grained handshapes that carry important linguistic information; for instance, in the *"Clean tools"* example, the baseline outputs leave the hands in a weakly articulated, near-static pose, whereas our method reproduces the intended handshape. Our method also yields more plausible upper-body posture with fewer self-contact artifacts.

4.4 Ablation Studies

Text–sign retrieval analysis. We further evaluate whether SeRV learns a more text-aligned token space through bidirectional text–sign retrieval on

Table 3: Per-batch bidirectional text–sign retrieval analysis with batch size 64. Higher is better for all recall metrics.

Dataset	Tokenizer	Direction	R@1↑	R@5↑	R@10↑
How2Sign	RVQ	T → S	11.61	37.28	59.04
	SeRV		14.52	46.97	73.21
	RVQ	S → T	6.73	27.05	47.10
	SeRV		8.42	34.08	58.41
YouTube-ASL	RVQ	T → S	7.55	24.23	38.38
	SeRV		9.73	31.50	49.01
	RVQ	S → T	4.37	17.58	30.62
	SeRV		5.64	22.85	39.10

Table 4: Effect of SeRV tokenizer objectives on How2Sign and YouTube-ASL. We report DTW-JPE (lower is better) and SiBLEU-4 (higher is better). All variants use the same generator.

Tokenizer Objective	How2Sign			YouTube-ASL		
	Body↓	Hands↓	SiB4.↑	Body↓	Hands↓	SiB4.↑
Baseline (RVQ)	6.82	9.63	3.20	5.39	7.20	2.56
+ $\mathcal{L}_{\text{global}}$	6.67	9.50	3.53	5.24	7.01	2.62
+ $\mathcal{L}_{\text{tok}}$	6.53	9.31	3.80	5.15	6.85	2.87
+ $\mathcal{L}_{\text{global}}$ + $\mathcal{L}_{\text{tok}}$	**6.51**	**9.22**	**4.12**	**4.98**	**6.70**	**3.13**

both How2Sign and YouTube-ASL. Text inputs are represented by frozen T5 global embeddings, while sign motions are represented by pooled latents from the frozen tokenizer encoder. Under the same per-batch retrieval protocol with batch size 64, SeRV improves Recall@1, Recall@5, and Recall@10 in both Text→Sign and Sign→Text directions, as shown in Table 3. This suggests that semantic supervision improves cross-modal alignment in the learned sign-token space.

Effect of SeRV Tokenization. We first examine whether semantic supervision improves learned motion tokens. We compare a standard RVQ tokenizer trained only with reconstruction losses, a variant with sentence-level motion–text alignment on the level-0 quantized latent, a variant with token-level late-interaction alignment on the fully quantized latent sequence, and the full SeRV tokenizer with both objectives. All variants use the same generator architecture. As shown in Table 4 and Figure 6 in Appendix A.6, on both How2Sign and YouTube-ASL, semantic supervision consistently improves generation accuracy, with larger gains on hand motion, and raises SiBLEU-4, indicating better token-level consistency with the input text. Combining sentence-level and token-level alignment performs best across all metrics, suggesting that global semantic alignment and token-level late interaction provide complementary supervision for learning text-predictable motion tokens.

Table 5: Ablation on generator architectures on How2Sign and YouTube-ASL. All variants use the same SeRV tokenizer. Lower is better for DTW-JPE.

Generator	How2Sign (DTW-JPE↓)		YouTube-ASL (DTW-JPE↓)	
	Body	Hands	Body	Hands
Flat AR	7.73	10.41	7.59	9.25
Two-stage	7.21	9.86	6.92	8.33
Hierarchical GPT	**6.51**	**9.22**	**4.98**	**6.70**

Effect of Hierarchical Generation. We evaluate whether the proposed generator benefits from explicitly modeling the residual token hierarchy. We compare three generation designs using the same SeRV tokenizer. The flat autoregressive baseline flattens all residual tokens into a single sequence and predicts them left-to-right. The two-stage baseline first predicts coarse tokens and then predicts the remaining residual tokens in a separate refinement stage. Our Hierarchical GPT predicts residual tokens in a coarse-to-fine order while conditioning each level on previously generated lower-level tokens. As shown in Table 5, on both How2Sign and YouTube-ASL, Hierarchical GPT consistently outperforms both the flat autoregressive (Flat AR) and two-stage baselines on body and hand motion, suggesting that explicitly preserving the residual token hierarchy is important for accurate ASL motion generation.

5 Conclusion

We present SeRV, a semantic-aligned RVQ tokenizer for text-to-3D ASL generation. Unlike reconstruction-only tokenizers, SeRV uses sentence-level and token-level motion–text alignment to learn residual tokens that are both precise and text-predictable. Building on SeRV, Hierarchical GPT predicts residual tokens over time and quantization depth while preserving the coarse-to-fine hierarchy. To support scalable training and evaluation, we construct a large-scale reconstructed 3D ASL motion-text benchmark from YouTube-ASL. Experiments on How2Sign and YouTube-ASL show state-of-the-art performance, with consistent gains in body and hand motion accuracy; ablations confirm the contributions of both semantic alignment and hierarchy-aligned generation.

6 Limitations

Despite the effectiveness of our current framework, several limitations remain. First, the current autoregressive architecture still generates tokens se-

quentially, which limits inference parallelism and makes long-sequence generation inefficient. Future work could explore masked, semi-autoregressive, or diffusion-based token generation to improve decoding speed while preserving temporal coherence. Second, although semantic alignment improves hand motion, our current objective does not explicitly supervise linguistic handshape correctness, palm orientation, or sign-level articulatory categories. Since hand motion carries much of the linguistic information in sign language, future models should introduce hand-specific objectives, higher-resolution hand tokenization, or dedicated hand refinement modules. Third, our current representation does not sufficiently model facial expression and emotion, which are essential non-manual components of sign language. Incorporating facial expression parameters, affective cues, and emotion-aware supervision may lead to more expressive and natural signing avatars. Finally, our current setting relies only on text as the input modality. Future work could extend the framework to multi-modal conditions, such as speech, prosody, glosses, reference videos, or signer-specific style prompts, to support more controllable and expressive sign generation.

References

Xingyu Cai, Tingyang Xu, Jinfeng Yi, Junzhou Huang, and Sanguthevar Rajasekaran. 2019. Dtwnet: A dynamic time warping network. *Advances in neural information processing systems*, 32.

Amanda Duarte, Shruti Palaskar, Lucas Ventura, Deepti Ghadiyaram, Kenneth DeHaan, Florian Metze, Jordi Torres, and Xavier Giro-i Nieto. 2021. How2sign: a large-scale multimodal dataset for continuous american sign language. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2735–2744.

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

Sen Fang, Chen Chen, Lei Wang, Ce Zheng, Chunyu Sui, and Yapeng Tian. 2025. Signllm: Sign language production large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6622–6634.

Sen Fang, Chunyu Sui, Xuedong Zhang, and Yapeng Tian. 2023. Signdiff: Learning diffusion models for american sign language production. *arXiv preprint arXiv:2308.16082*.

Chuan Guo, Inwoo Hwang, Jian Wang, and Bing Zhou. 2025a. Snapmogen: Human motion generation from expressive texts. *Advances in Neural Information Processing Systems*, 38:99939–99955.

Chuan Guo, Inwoo Hwang, Jian Wang, and Bing Zhou. 2025b. Snapmogen: Human motion generation from expressive texts. *arXiv preprint arXiv:2507.09122*.

Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. 2024. Momask: Generative masked modeling of 3d human motions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1900–1910.

Youngjoon Jang, Haran Raajesh, Liliane Momeni, Gül Varol, and Andrew Zisserman. 2025. Lost in translation, found in context: Sign language translation with contextual cues. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8742–8752.

Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. 2023. Motiongpt: Human motion as a foreign language. *Advances in Neural Information Processing Systems*, 36:20067–20079.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. 2022. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11523–11532.

Zecheng Li, Wengang Zhou, Weichao Zhao, Kepeng Wu, Hezhen Hu, and Houqiang Li. 2025. Uni-sign: Toward unified sign language understanding at scale. *arXiv preprint arXiv:2501.15187*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. 2019. Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*.

Rolandos Alexandros Potamias, Jinglei Zhang, Jiankang Deng, and Stefanos Zafeiriou. 2025. Wilor: End-to-end 3d hand localization and reconstruction in-the-wild. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12242–12254.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text

transformer. *Journal of machine learning research*, 21(140):1–67.

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

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

Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. 2022. 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.

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

Qingping Sun, Yanjun Wang, Ailing Zeng, Wanqi Yin, Chen Wei, Wenjia Wang, Haiyi Mei, Chi-Sing Leung, Ziwei Liu, Lei Yang, and Zhongang Cai. 2024. Aios: All-in-one-stage expressive human pose and shape estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1834–1843.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Dave Uthus, Garrett Tanzer, and Manfred Georg. 2023. Youtube-asl: A large-scale, open-domain american sign language-english parallel corpus. *Advances in Neural Information Processing Systems*, 36:29029–29047.

Cong Wang, Zexuan Deng, Zhiwei Jiang, Yafeng Yin, Fei Shen, Zifeng Cheng, Shiping Ge, Shiwei Gan, and Qing Gu. 2025. Advanced sign language video generation with compressed and quantized multi-condition tokenization. *arXiv preprint arXiv:2506.15980*.

Ryan Wong, Necati Cihan Camgoz, and Richard Bowden. 2024. Sign2gpt: Leveraging large language models for gloss-free sign language translation. In *International Conference on Learning Representations*.

Pan Xie, Taiying Peng, Yao Du, and Qipeng Zhang. 2024a. Sign language production with latent motion transformer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3024–3034.

Pan Xie, Qipeng Zhang, Peng Taiying, Hao Tang, Yao Du, and Zexian Li. 2024b. 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.

Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. 2021. Filip: Fine-grained interactive language-image pre-training. *arXiv preprint arXiv:2111.07783*.

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

Zhengdi Yu, Shaoli Huang, Yongkang Cheng, and Tolga Birdal. 2024. Signavatars: A large-scale 3d sign language holistic motion dataset and benchmark. In *European Conference on Computer Vision*, pages 1–19. Springer.

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

Ronglai Zuo, Rolandos Alexandros Potamias, Qi Sun, Evangelos Ververas, Jiankang Deng, and Stefanos Zafeiriou. 2026. Madis: Taming masked diffusion language models for sign language generation. *arXiv preprint arXiv:2601.19577*.

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

A Appendix

A.1 Comparison with Existing 3D Sign Resources

Recent 3D sign language resources have made important progress toward avatar-based sign language understanding and generation. Early 3D ASL resources such as How2Sign provide aligned videos, depth, and English transcripts, but their scale and signer coverage remain limited for training high-capacity text-to-motion generators. More recent SLG systems further curate SMPL-X poses for existing continuous sign language benchmarks, including How2Sign, CSL-Daily, and Phoenix-2014T, enabling token-based 3D sign generation across multiple languages. In parallel, SignAvatars introduces a large-scale holistic 3D sign

language dataset with SMPL-X, MANO, and 2D/3D keypoint annotations, covering both isolated and continuous signs with multiple prompt types. However, existing resources are either relatively limited in ASL scale, constructed from studio-recorded benchmarks, or designed for general multi-prompt/multilingual 3D sign production rather than large-scale paired text-to-3D ASL generation from in-the-wild videos.

Table 6 summarizes the comparison. Our processed YouTube3D dataset contains 236,537 ASL–text pairs from 8,119 in-the-wild videos, totaling 295.1 hours of signing motion.

A.2 Data Preparation

Existing 3D ASL datasets (Duarte et al., 2021) remain limited in scale, signer diversity, vocabulary coverage, and recording duration, making it difficult to train high-capacity generative models that generalize well. To support scalable 3D ASL generation, we build a reconstructed 3D motion–text dataset from YouTube-ASL (Uthus et al., 2023), a large-scale in-the-wild ASL video corpus with English text annotations. For privacy protection and to respect the rights of video owners, YouTube-ASL releases YouTube video IDs rather than raw video files. We therefore download the available videos using the released IDs. In total, 61,153 available video clips are successfully downloaded and used as the initial pool for 3D motion reconstruction.

Following prior work on in-the-wild 3D sign motion reconstruction (Zuo et al., 2025; Sun et al., 2024; Potamias et al., 2025), we recover expressive 3D signing motion from RGB videos, including upper-body pose, hand articulation, and facial expression. Each reconstructed motion sequence is represented as $\mathbf{S} \in \mathbb{R}^{T \times d}$, where T denotes the sequence length and $d = 133$ is the dimensionality of the SMPL-X motion parameters, consisting of 3D rotations for 11 upper-body joints and 30 hand joints, plus 10 facial expression coefficients. Each reconstructed ASL motion sequence is paired with its corresponding text transcript.

To improve data quality, we apply a multi-stage filtering pipeline. First, we use MediaPipe-based person detection (Lugaresi et al., 2019) to remove clips containing multiple visible signers, since multi-signer clips make it ambiguous which person should be paired with the transcript. Second, we discard clips in which no reliable signer can be detected. Third, after 3D reconstruction, we filter out samples whose reconstruction confidence is below a predefined threshold, which removes clips with severe reconstruction failures, poorly tracked hands, or unreliable body estimates. The remaining samples are normalized into a consistent body representation and coordinate system.

After downloading, reconstruction, filtering, and normalization, our processed YouTube3D dataset contains 236,537 ASL–text pairs from 8,119 videos, totaling 295.1 hours of signing motion. This pipeline converts large-scale in-the-wild ASL videos into structured paired 3D motion–text data, providing broader signer diversity, richer motion variation, and larger linguistic coverage for training both the semantic-aligned tokenizer and the hierarchical autoregressive generator.

A.3 Evaluation Metric Definitions

We provide formal definitions of the metrics used in Section 4.1.

Notation. Let $\hat{\mathbf{S}} = \{\hat{\mathbf{p}}_i\}_{i=1}^{\hat{T}}$ denote a generated sign motion sequence and $\mathbf{S} = \{\mathbf{p}_j\}_{j=1}^{T}$ the corresponding reference, where each frame contains J 3D joints and $\mathbf{p}^{(k)} \in \mathbb{R}^3$ is the k-th joint position. The per-frame joint position error between generated frame i and reference frame j is

$$d(i,j) = \frac{1}{J} \sum_{k=1}^{J} \left\| \hat{\mathbf{p}}_i^{(k)} - \mathbf{p}_j^{(k)} \right\|_2. \tag{9}$$

DTW-JPE. Since the generated and reference sequences may differ in length, we align them with dynamic time warping. Let Π be the set of monotonic warping paths $\pi = \{(i,j)\}$ satisfying the standard boundary, monotonicity, and continuity constraints. DTW selects the path with minimum cumulative cost,

$$\pi^\star = \arg\min_{\pi \in \Pi} \sum_{(i,j)\in\pi} d(i,j), \tag{10}$$

and DTW-JPE is the cost along the optimal path normalized by its length:

$$\text{DTW-JPE} = \frac{1}{|\pi^\star|} \sum_{(i,j)\in\pi^\star} d(i,j). \tag{11}$$

We compute Eq. (9) over the body, left-hand, and right-hand joint sets separately, and report Hand Avg. as the mean of the left- and right-hand errors.

PA-MPJPE. For tokenizer reconstruction the output is frame-aligned with the input ($\hat{T} = T$), so

Table 6: Comparison with existing 3D sign language motion resources. "Text paired" indicates whether spoken-language text is paired with the 3D motion.

Dataset	Lang.	Clips / Pairs	Duration	Text Paired
How2Sign-3D (Duarte et al., 2021)	ASL	35K	80+h	✓
SOKE Curated 3D Poses (Zuo et al., 2025)	ASL / CSL / DGS	35K / 20K / 8K	–	✓
SignAvatars (Yu et al., 2024)	Multi-SL	70K	117h	✓
Reconstructed YouTube-ASL (Ours)	ASL	236,537	295.1h	✓

no temporal warping is required. Before computing the error, we apply a rigid Procrustes alignment that estimates the optimal rotation $\mathbf{R}$, translation $\mathbf{t}$, and scale s minimizing the residual between the two frames:

$$(\mathbf{R}^\star, \mathbf{t}^\star, s^\star) = \arg\min_{\mathbf{R},\mathbf{t},s} \sum_{k=1}^{J} \left\| s\,\mathbf{R}\,\hat{\mathbf{p}}_t^{(k)} + \mathbf{t} - \mathbf{p}_t^{(k)} \right\|_2^2. \tag{12}$$

PA-MPJPE is then the per-joint error averaged over all frames after alignment:

$$\text{PA-MPJPE} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{J}\sum_{k=1}^{J} \left\| s^\star \mathbf{R}^\star \hat{\mathbf{p}}_t^{(k)} + \mathbf{t}^\star - \mathbf{p}_t^{(k)} \right\|_2. \tag{13}$$

SiBLEU-4. Let $\mathcal{T}(\cdot)$ be a frozen evaluator tokenizer mapping a motion sequence to a discrete token sequence. Denote the generated and reference token sequences as $\hat{\mathbf{u}} = \mathcal{T}(\hat{\mathbf{S}})$ and $\mathbf{u} = \mathcal{T}(\mathbf{S})$. SiBLEU-4 is the BLEU-4 score between them:

$$\text{SiBLEU-4} = \text{BP} \cdot \exp\left(\sum_{n=1}^{4} w_n \log p_n\right), \tag{14}$$

where p_n is the modified n-gram precision over token sequences, $w_n = \frac{1}{4}$ are uniform weights, and BP is the brevity penalty

$$\text{BP} = \begin{cases} 1, & c > r, \\ \exp(1 - r/c), & c \le r, \end{cases} \tag{15}$$

with c and r the total lengths of the generated and reference token sequences, respectively. A higher SiBLEU-4 indicates better token-level consistency.

A.4 Implementation details.

We train the framework in two stages: SeRV tokenizer training followed by Hierarchical GPT training. In the first stage, the SeRV tokenizer is trained on the training split with the RVQ reconstruction objective and two semantic alignment objectives: the global-level motion–text alignment loss and the token-level motion–text late-interaction loss. Unless otherwise specified, we use $L = 3$ residual quantization levels, a latent dimension of $C = 512$, and a codebook size of $K = 512$ at each level. The motion encoder downsamples the input sequence by a factor of 4, so each discrete token corresponds to four motion frames.

For semantic supervision, we use a frozen T5 encoder (Raffel et al., 2020) to encode the paired transcript. The mean-pooled T5 hidden states are used as sentence-level text embeddings for the global alignment loss, where they are aligned with the temporally pooled level-0 quantized motion representation. The token-level T5 hidden states are used in the late-interaction loss, where they are matched with the fully quantized motion latent sequence. The SeRV training objective is $\mathcal{L}_{\text{SeRV}} = \mathcal{L}_{\text{RVQ}} + \lambda_{\text{global}}\mathcal{L}_{\text{global}} + \lambda_{\text{tok}}\mathcal{L}_{\text{tok}}$. We optimize the tokenizer with AdamW (Loshchilov and Hutter, 2017) for 500 epochs using a batch size of 128, an initial learning rate of 4×10^{-4}, a weight decay of 0.1, and cosine learning-rate decay.

In the second stage, Hierarchical GPT is trained to autoregressively predict the residual token hierarchy produced by the frozen SeRV tokenizer. Given frozen T5 text features, the generator predicts tokens over time and RVQ depth in a coarse-to-fine order. This preserves the residual hierarchy learned by SeRV instead of flattening all residual tokens into a single sequence.

Coarse Motion Decoder. The level-0 decoder predicts coarse motion tokens from the text condition and previous temporal history. Since the level-0 RVQ code carries the dominant motion structure, this decoder is responsible for generating the coarse text-conditioned signing trajectory. We implement it as a causal Transformer with LLaMA-style blocks, including RMSNorm, SwiGLU feed-forward layers, rotary positional embeddings, and prefix-based conditioning on the T5 text features. The T5 token features are projected to the generator

hidden dimension and prepended as conditioning tokens, allowing the decoder to autoregressively predict coarse motion tokens while attending to the transcript representation.

Residual Refinement Decoder. Higher-level tokens are predicted by residual refinement decoders. Each residual decoder predicts refinement tokens conditioned on the text feature, previous temporal tokens, and the lower-level tokens already generated at the current time step. Rather than generating motion from scratch, these decoders refine the representation produced by lower RVQ levels. Concretely, the level-1 decoder is conditioned on the level-0 token embeddings, while the level-2 decoder is conditioned on the additive coarse context formed by level-0 and level-1 token embeddings. This design follows the additive structure of RVQ, where later codebooks encode residual details over the coarse motion state.

The generator uses a hidden dimension of 1024, 16 attention heads, a dropout rate of 0.1, and a maximum token length of 100. The coarse motion decoder contains 9 Transformer layers, while the two residual refinement decoders contain 6 and 3 layers, respectively. We train the generator for 150 epochs with AdamW, using a batch size of 16, a learning rate of 2×10^{-4}, a weight decay of 0.01, and cosine learning-rate decay. At inference time, tokens are generated in the same coarse-to-fine order until an EOS token is predicted or the maximum length is reached. We use autoregressive sampling with a temperature of 1.0. The predicted multi-level token streams are decoded by the frozen SeRV decoder into continuous SMPL-X motion features, which are then converted into joints or vertices for evaluation. All experiments are conducted on three NVIDIA L40S GPUs.

A.5 Additional Qualitative Results for Tokenizer Reconstruction

A.6 Additional Qualitative Results on Ablation Study

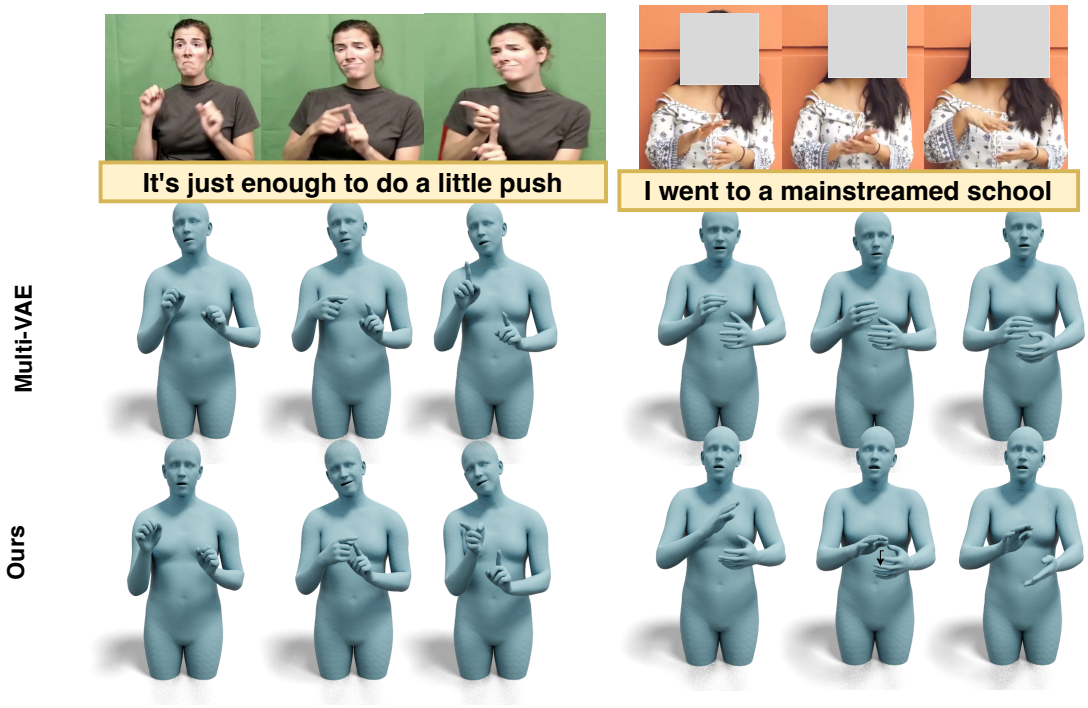

Figure 5: Qualitative comparison of tokenizer reconstruction results. Our tokenizer preserves motion details more effectively, especially for hand articulations.

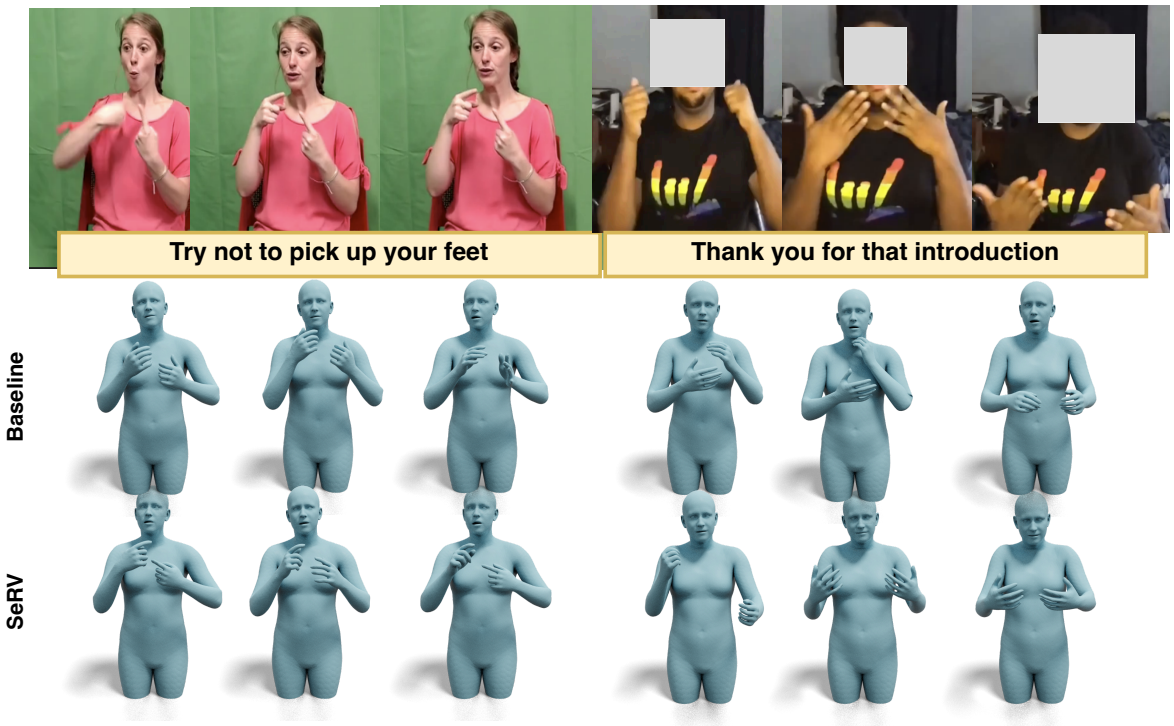

Figure 6: Qualitative comparison of generated 3D ASL motions on two examples. For each example, the top row shows the baseline and the bottom row shows SeRV. With semantic-aligned RVQ, SeRV better captures hand information and produces more faithful signing motions.