SignLlama: Enhancing Gloss-free Sign Language Translation by Prioritizing Visual Features for LLMs

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

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

Yafeng Yin†
State Key Laboratory of Novel Software Technology, Nanjing University
Suzhou, Jiangsu, China
yafeng@nju.edu.cn

Zhiwei Jiang
State Key Laboratory of Novel Software Technology, Nanjing University
Suzhou, Jiangsu, China
jzw@nju.edu

Bowen Guo
State Key Laboratory of Novel Software Technology, Nanjing University
Suzhou, Jiangsu, China
602024720001@smail.nju.edu.cn

Lei Xie
State Key Laboratory of Novel Software Technology, Nanjing University
Nanjing, Jiangsu, China
lxie@nju.edu.cn

Sanglu Lu
State Key Laboratory of Novel Software Technology, Nanjing University
Nanjing, Jiangsu, China
sanglu@nju.edu.cn

Hongkai Wen
University of Warwick
Coventry, United Kingdom
hongkai.wen@warwick.ac.uk

arXiv:2608.09006v1 [cs.CV] 10 Aug 2026

Abstract

Large Language Models (LLMs) have achieved remarkable success across a wide range of tasks. However, fine-tuning LLMs for Gloss-Free Sign Language Translation (GFSLT) remains a challenge. In this paper, we investigate how to effectively adapt LLMs to the GFSLT task. We show that there are two key issues that need to be solved: (1) the inherent distributional gap between visual feature inputs and text feature inputs makes it difficult for LLMs to interpret visual inputs; and (2) existing approaches typically concatenate visual and textual features in an autoregressive framework, which leads to the model overemphasizing textual inputs and deprioritizing visual cues, as LLMs are pretrained predominantly on text-centric data. To address the first challenge, we propose a simple yet effective method named Filtered Pseudo-Gloss CTC Pretraining, which leverages filtered pseudo-gloss sequences generated from text sequences to supervise the training of the visual backbone. To tackle the second issue, we introduce a Visual-Prioritized Distillation training strategy. Specifically, we define a visual-only prediction path in which text inputs are masked, and the model is required to generate the target sequence relying solely on visual inputs. To guide this path, the outputs from the standard visual-textual prediction are then distilled into the visual-only prediction path, encouraging the model to prioritize visual features. Comprehensive experiments and qualitative analyses demonstrate the effectiveness of the proposed model. The proposed SignLlama achieves very competitive performance on multiple datasets for GFSLT tasks, without using any extra modalities or external sign language datasets for pretraining.

*Both authors contributed equally to this research.
†Yafeng Yin is the corresponding author.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

MM '26, Rio de Janeiro, Brazil

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2213-4/2026/11
https://doi.org/10.1145/3767308.3835002

CCS Concepts

• **Computing methodologies** → **Activity recognition and understanding**; *Natural language generation*; • **Human-centered computing** → *Accessibility technologies.*

Keywords

Sign Language Translation, Large Language Models

ACM Reference Format:
Shiwei Gan, Xiao Liu, Yafeng Yin, Zhiwei Jiang, Bowen Guo, Lei Xie, Sanglu Lu, and Hongkai Wen. 2026. SignLlama: Enhancing Gloss-free Sign Language Translation by Prioritizing Visual Features for LLMs. In *Proceedings of the 34th ACM International Conference on Multimedia (MM '26), November 10–14, 2026, Rio de Janeiro, Brazil.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3767308.3835002

1 Introduction

Current sign language (SL) understanding tasks mainly include Continuous SL Recognition (CSLR) [11] and SL Translation (SLT). CSLR aims to recognize a sequence of signs and convert it into the corresponding gloss sequence, while SLT [14, 38] focuses on translating sign sequences into spoken language. As SLT produces more natural and fluent language outputs, it has attracted increasing attention in recent research. The de facto architecture for current SLT tasks typically employs a 2D or 3D CNN-based backbone to extract visual features [12], followed by temporal modeling modules (*e.g.*, 1D CNNs or LSTMs) to capture dynamic changes across sign frames. Finally, a translation model (e.g., mBART [9, 10, 15], GPT-2 [44]) is used to generate the corresponding spoken language sentence. SLT research has branched into two distinct paths: gloss-based SLT (GBSLT) and gloss-free SLT (GFSLT). Previous state-of-the-art (SOTA) GBSLT models [4, 12] have emphasized that pretraining the visual encoder with gloss labels is critical for achieving strong SLT performance. Due to the reliance on gloss labels, SLT has increasingly shifted its focus towards GFSLT [15, 23, 44], which aims to boost SLT performance without gloss supervision.

Meanwhile Large Language Models (LLMs) have revolutionized the field of natural language processing, setting new benchmarks

across a broad spectrum of tasks [1, 16, 18, 29]. Benefiting from the vast pretraining datasets, LLMs demonstrate extraordinary generalization capabilities, and as a result, fine-tuning LLMs for downstream tasks has emerged as a prevailing trend [29, 33]. However, adapting LLMs for the GFSLT task remains understudied [17, 44], primarily due to the following two key challenges. **1)** There exists a fundamental distributional gap between textual features and SL video features. Textual inputs, which are the primary input modality during LLM pretraining, are typically tokenized into discrete, symbolic units, forming clearly segmented sequences. In contrast, SL video features are inherently continuous, with subtle variations between consecutive frames. This discrepancy in representation poses a significant challenge in effectively tokenizing SL videos and providing suitable inputs for LLMs, which is crucial for fine-tuning LLMs for GFSLT. **2)** Existing Vision-Language Models (*e.g.*, MiniGPT-4 [1]) and LLM-based SLT models [23] typically adopt a straightforward strategy by concatenating visual and textual features directly in an autoregressive manner. However, this approach often leads to the deprioritization of visual features and biases the model toward fitting the distribution of text tokens rather than visual ones [45], as LLMs are mainly pretrained on text-centric data, and visual inputs constitute an out-of-distribution modality.

To tackle the first challenge, a common approach in vision-language models (VLMs) is to align visual and textual features by employing a pre-trained image encoder and leveraging large-scale image-text datasets with objectives such as Image-Text Contrastive (ITC) loss and Image-Text Matching (ITM) loss [30, 47]. In addition, some works have explored using attention mechanisms [48], vector quantization [17], and pseudo glosses [15, 44] to tokenize SL videos. As for the second challenge, the effectiveness of concatenating visual and textual features in an autoregressive way remains underexplored. CAL [45] addresses this by focusing on reweighting text tokens based on their visual relevance, thereby enhancing the alignment between modalities. However, it overlooks the enhancement of visual representations themselves. In contrast, our approach aims to prioritize the contribution of visual features during prediction, ensuring that the model more effectively leverages visual information.

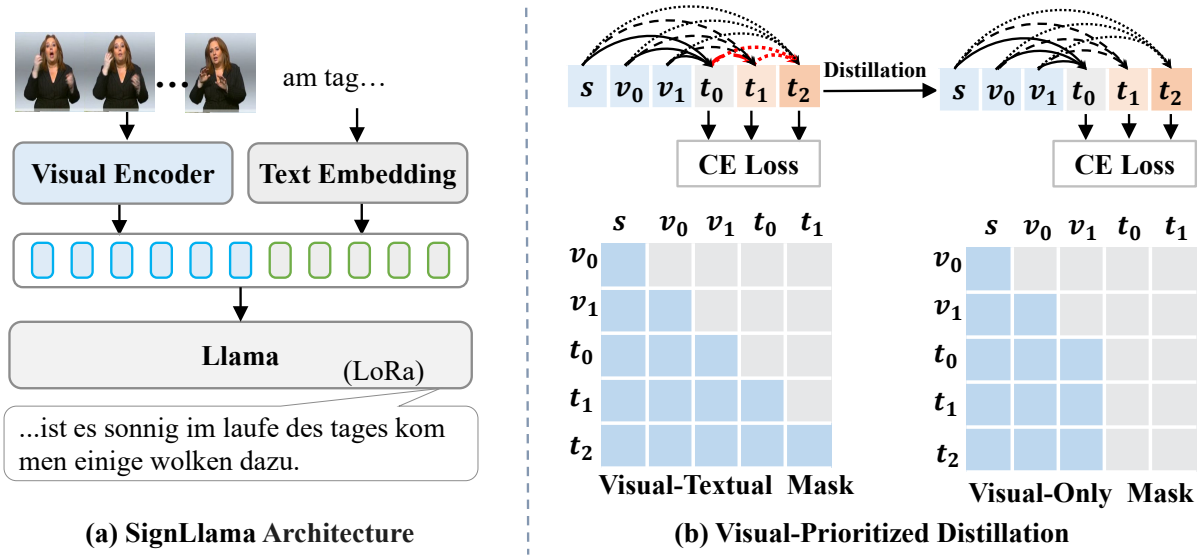
In this paper, we focus on the Gloss-Free SLT task by adopting open-source LLMs (e.g., the Llama family [40]). First, inspired by recent works [15, 17] and GBSLT approaches where CTC loss is commonly used to pretrain the visual encoder, we propose a simple yet effective method to bridge the distributional gap between video and textual features, namely filtered pseudo-gloss CTC Pretraining. Specifically, we generate filtered pseudo-gloss sequences from the target text by applying lemmatization, removing prepositions and conjunctions, and randomly dropping tokens. The resulting pseudo-gloss sequences are then used with CTC loss to pretrain the visual encoder. To further prioritize the model's reliance on visual features, we propose Vision-Prioritized Distillation (VPD). This training strategy distills knowledge from a 'Visual-Textual Prediction' path into a 'Visual-Only Prediction' path, enabling the LLMs to generate text with greater reliance on visual features. By prioritizing visual features, VPD can boost the ability of text generation from visual inputs, thereby improving the performance of GFSLT models. We make the following contributions:

- We introduce a simple yet effective pretraining method, **Filtered Pseudo-gloss CTC Pretraining (FPG-CTC)**, which generates filtered pseudo-gloss sequences to supervise the visual encoder. We show that FPG-CTC can effectively discretize and align SL features, leading to substantial performance improvements on the GFSLT task.
- We propose a **Vision-Prioritized Distillation (VPD)** training method, which distills the 'Visual-Textual Prediction' into the 'Visual-Only Prediction' to enhance GFSLT performance. The proposed VPD can boost text generation and alleviate the exposure bias by prioritizing visual features.
- We contribute an array of pre-trained SignLlama models to the community, ranging from 1 billion to 13 billion parameters. Extensive experiments on public SL datasets demonstrate the effectiveness of our models, which achieve very competitive performance on the GFSLT task.

2 Related Work

LLMs for Visual Understanding. Open-source LLMs have revolutionized a wide range of downstream tasks, such as image understanding [3, 21, 26, 43] and video understanding [46]. These methods typically use a pretrained visual backbone (e.g., ViT [7]) to extract image or video features, which are then concatenated with textual inputs and fed into LLMs for text generation. The training of these models typically follows a two-stage paradigm [22]. First, they are tuned on large-scale video-text data to align the visual features with textual features. The second stage involves instruction tuning with data such as videos with descriptions and question-answering pairs. Such a paradigm allows the visual backbone to effectively tokenize images or videos and provide suitable inputs for LLMs. Although these methods have achieved promising results, transferring these pretrained backbones directly to SL tasks remains challenging. Unlike visual understanding tasks that often focus on scene-level changes, SL tasks involve much finer-grained visual cues, such as subtle hand movements, hand shapes, and facial expressions, which are often overlooked in general visual understanding. Moreover, a key limitation lies in the common practice of concatenating video and text features for autoregressive fine-tuning, which often leads the model to over-rely on textual features, deprioritize visual information, and ultimately degrade model performance.

Sign Language Tasks. Current approaches in CSLR [5, 11, 55] and SLT [13, 27, 41, 53] have established a de facto architecture, which typically involves employing a visual module to capture visual features, followed by a temporal module (*e.g.*, 1D CNNs, LSTM, or transformer layers) to model temporal dependencies. For CSLR tasks, a CTC loss is applied to compute the probabilities over all possible alignment paths. For SLT tasks, an additional translation model is employed to generate textual sequences from video features. Consequently, current CSLR research primarily focuses on improving the modeling of SL-related visual features and temporal relationships by designing various visual backbones (*e.g.*, VAC [31], TwoStream [4], SignGraph [11]and MixSignGraph [15]) and temporal modules [51, 54]. Benefiting from CSLR, gloss-based SLT [12, 13] leverages pretrained backbones from the CSLR task and incorporates a translation model [28, 34] to generate text sequences. These



(a) SignLlama Architecture (b) Visual-Prioritized Distillation

Figure 1: The proposed SignLlama with Visual-Prioritized Distillation

pretrained backbones effectively align SL video features with gloss sequences, functioning as a tokenizer for SL videos. However, in GFSLT, a key challenge lies in how to effectively tokenize SL videos and align visual features with textual features to provide suitable inputs for LLMs, which is crucial for fine-tuning LLMs for SLT. Existing GFSLT models typically adopt attention mechanisms [48] or contrastive language-image pretraining [20, 23] to enhance alignment and improve performance. Recently, SignLLM [17] attempted to address this by training two discrete codebooks to align sign video features with text. There are also some works focusing on designing pseudo gloss [9, 15] to pretrain their backbone, while our methods provides a simpler, more straightforward, and more effective way to pretrain SL backbones. Nonetheless, these methods primarily focus on how to effectively tokenize SL video inputs. In addition, recent GFSLT models incorporating LLMs also typically adopt the same strategy, *i.e.*, , concatenating video and textual features as input for autoregressive fine-tuning. However, this approach tends to prioritize textual features over visual ones. In this paper, we aim to develop a simple and effective training paradigm for effectively tokenizing SL video, along with a strategy that enhances the GFSLT model performance by prioritizing visual features.

3 Method

Overall Framework. For an SL video with θ frames $f = \{f_i\}_{i=1}^{\theta}$, the goal of GFSLT is to generate $t = \{t_i\}_{i=1}^{\varsigma}$ with ς words based on input f. As illustrated in Figure 1, our model consists of a visual encoder and a Llama model. Note that the text embedding module is part of Llama. We show it explicitly here for clarity. The visual encoder $\mathcal{VE}$ processes frames to get visual features $v{=}\mathcal{VE}(f)$ with m vectors, where $v = \{v_i \in \mathbb{R}^{d_l}\}_{i=1}^{m}$ $(m \leq \theta)$ and d_l is the input dimension of Llama. The tokenizer converts the target text into n token IDs $\{u_i\}_{i=1}^{n}$, and the text embedding module maps them to text embeddings $e = \{e_i \in \mathbb{R}^{d_l}\}_{i=1}^{n}$. $(n \geq \varsigma)$. During training stage, the model predicts token $\hat{u}_i$ based on v and previous ground truth tokens: $p(\hat{u}_i|v, \{u_j\}_{j=0}^{i-1})$. During inference, the model predicts token $\hat{t}_i$ based on v and previous predicted tokens: $p(\hat{u}_i|v, \{\hat{u}_j\}_{j=0}^{i-1})$.

3.1 Filtered Pseudo-gloss CTC Pretraining

For GBSLT [4, 15], the visual encoder is typically trained with the CSLR task, in which CTC loss is used to optimize the visual encoder to learn semantic segmentation and alignment with gloss sequences g. A translation model is then integrated and fine-tuned for SLT, taking the output of the encoder as its input. Let R denote the recognition network (with the visual encoder as a subcomponent) and T denote the translation network, while Θ_R and Θ_T denote the parameters of the recognition and translation networks, respectively. $\mathcal{L}_{CTC}$ and $\mathcal{L}_{CE}$ denote the CTC loss and the cross-entropy loss. The hyperparameters α and β are used to balance the two objectives. The training process can be formalized as:

$$\min_{\Theta_R} \mathcal{L}_{CTC}(R(f), g) \tag{1}$$

$$\min_{\Theta_R, \Theta_T} (\alpha \mathcal{L}_{CTC}(R(f), g) + \beta \mathcal{L}_{CE}(T(v), t)) \tag{2}$$

For gloss-free SLT, the absence of gloss annotations makes it challenging to pretrain the visual encoder. Unlike previous pseudo gloss based GFSLT models that adopt contrastive learning for visual backbone pretraining, we aim to develop a simple yet effective method that enables the visual encoder to effectively tokenize SL videos and align visual features with textual representations. We draw inspiration from previous work [15] and propose the Filtered Pseudo-gloss CTC Pretraining (FPG-CTC) method. Specifically, we generate a pseudo-gloss sequence g_p from the target text by applying lemmatization and filtering out prepositions and conjunctions. Considering that certain words in the text may not be explicitly expressed in the corresponding SL video, we further apply random dropping to remove a fixed proportion of words, resulting in the

final pseudo-gloss sequence g_p. The FPG-CTC training process can be formalized as:

$$\min_{\Theta_R} \mathcal{L}_{CTC}(R(f), g_p) \tag{3}$$

$$\min_{\Theta_R,\Theta_T} (\alpha\mathcal{L}_{CTC}(R(f), g_p) + \beta\mathcal{L}_{CE}(T(v), t)) \tag{4}$$

3.2 Visual-Prioritized Distillation Training for LLM Fine-Tuning

After obtaining visual features, a widely adopted approach is to concatenate visual and textual features directly, followed by next-token prediction, which we refer to as 'Visual-Textual Prediction', formalized as: $p(\hat{t}_i|v, \{t_j\}_{j=0}^{i-1})$. However, this training paradigm suffers from a critical limitation: visual deprioritization. LLMs tend to prioritize fitting the distribution of text tokens over that of the newly introduced visual tokens. This is because LLMs are pretrained on text-centric data, and visual inputs constitute an out-of-distribution modality, leading to the deprioritization of visual information during generation. During inference, the model no longer has access to ground-truth tokens $\{t_j\}_{j=0}^{i-1}$ and instead conditions on its own predictions: $p(\hat{t}_i|v, \{\hat{t}_j\}_{j=0}^{i-1})$. This mismatch between training and testing leads to cumulative errors. When the model is overly reliant on generated text, errors in early predictions can quickly accumulate, leading to degraded performance.

To encourage the LLMs to prioritize visual features, we propose a novel training paradigm termed Visual-Prioritized Distillation. Specifically, we perform two separate forward passes through the LLM: one for Visual-Textual Prediction, where visual and textual features are concatenated as input; and another for Visual-Only Prediction, where the textual inputs are masked out using a 'Visual-Only Mask' (as illustrated in Figure 1 (b)), and the model is required to generate the full target sequence relying solely on visual inputs. To ensure that the 'Visual-Only Prediction' path can converge and perform well, we distill the logits from the 'Visual-Textual Prediction' path into the 'Visual-Only Prediction' one. This distillation guides the learning process of the visual-only prediction path, encouraging it to align with the output distribution of the visual-textual prediction path while learning to rely exclusively on visual features. The training loss can be formalized as:

$$\begin{aligned}\mathcal{L}_{VPD} &= \mathcal{L}_T^{vt} + \mathcal{L}_T^{v} + \gamma\mathcal{L}_{KD} \\ &= \sum_{i=1}^{n} \mathcal{L}_{CE}(T(v, \{t_j\}_{j=0}^{i-1}), t_i) \\ &+ \sum_{i=1}^{n} \mathcal{L}_{CE}(T(v), t_i) + \lambda \mathrm{KL}(T(v,t)||T(v))\end{aligned} \tag{5}$$

where γ is a hyperparameter and n denotes the number of tokens. $\mathcal{L}_T^{vt}$, $\mathcal{L}_T^{v}$ are cross-entropy losses of 'Visual-Textual Prediction' and 'Visual-Only Prediction', and $\mathcal{L}_{KD}(T(v,t), T(v))$ denotes the distillation loss.

Training and Inference for GFSLT. The final training process of our model begins with pretraining the visual backbone using pseudo-gloss supervision, as described below.

$$\mathcal{L}_R = \min_{\Theta_R} \mathcal{L}_{CTC}(R(f), g_p) \tag{6}$$

The entire model is then fine-tuned using the VPD objective together with the FPG-CTC loss.

$$\min_{\Theta_R,\Theta_T} (\alpha\mathcal{L}_R + \beta(\mathcal{L}_T^{vt} + \mathcal{L}_T^{v} + \lambda\mathcal{L}_{KD})) \tag{7}$$

During inference, the visual encoder takes the SL video as input and extracts SL video features, which are then fed into the Llama model. The Llama model follows the standard autoregressive process to generate the target text sequence. Neither pseudo-gloss supervision nor the VPD training procedure is required during inference.

4 Theoretical Insights into VPD

The difference between VPD and Knowledge Distillation. Although VPD shares a superficial similarity with knowledge distillation (KD), it is conceptually distinct in both purpose and design. The key novelty lies in how VPD prioritizes visual features between visual-only and visual-textual paths, mitigating the tendency of the AR model to over-rely on textual cues and underutilize visual information, which is a critical bottleneck that has not been addressed in prior work. Thus, VPD is not merely an adaptation of KD, but a novel mechanism that explicitly prioritizes visual features in SLT.

The theoretical intuition of VPD. Here, we also provide a brief theoretical intuition of VPD at the gradient level. Let v denote the visual input encoded by parameters θ_v, t the optional textual input, $T(v,t)$ the teacher (visual-textual path) with logits z_{vt} and softmax probabilities p_{vt}, and $S(v)$ the student (visual-only path) with logits z_v and softmax probabilities p_v. The simplified VPD loss is:

$$\mathcal{L} = \underbrace{\mathcal{L}_{vt}}_{\text{visual-textual CE}} + \underbrace{\mathcal{L}_{v}}_{\text{visual-only CE}} + \gamma \underbrace{\mathcal{L}_{KD}}_{\text{distillation (KL)}}, \tag{8}$$

where

$$\mathcal{L}_{vt} = -\sum_i y_i \log p_{vt,i}, \tag{9}$$

$$\mathcal{L}_{v} = -\sum_i y_i \log p_{v,i}, \tag{10}$$

$$\mathcal{L}_{KD} = \mathrm{KL}(p_{vt} \,\|\, p_v) = \sum_i p_{vt,i} \log \frac{p_{vt,i}}{p_{v,i}}. \tag{11}$$

Where y denote the one-hot target distribution. We are interested in the gradient of $\mathcal{L}$ with respect to θ_v:

$$\frac{\partial \mathcal{L}}{\partial \theta_v} = \underbrace{\frac{\partial \mathcal{L}_{vt}}{\partial \theta_v} + \frac{\partial \mathcal{L}_{v}}{\partial \theta_v}}_{\text{CE gradients}} + \gamma \sum_i (p_{v,i} - p_{vt,i}) \frac{\partial z_{v,i}}{\partial \theta_v}. \tag{12}$$

Intuitively, when the teacher assigns a higher probability to the correct class, the KL term provides an additional corrective gradient that pushes p_v toward p_{vt}, supplementing the supervision provided by the visual-only CE loss. This provides an additional training signal to the visual encoder, encouraging the visual-only path to produce predictions closer to those of the visual-textual path.

5 Experiments

Datasets. We evaluate our model on several publicly available SLT datasets, including the widely used Phoenix14T and CSL-Daily, as well as two large-scale ASL datasets: How2Sign and OpenASL. (1) **Phoenix14T** [2] is a German Sign Language dataset annotated

Table 1: Effects of our training strategy in Phoenix14T.

FPG-CTC	VPD	TEST				
		ROUGE	BLEU1	BLEU2	BLEU3	BLEU4
		20.09	18.12	8.56	6.46	5.46
✓		49.93	49.22	36.53	28.61	23.47
✓	✓	52.81	52.54	40.96	32.87	26.74

Table 2: Effects of VPD loss in Phoenix14T.

$\mathcal{L}_T^{vt}$	$\mathcal{L}_T^{v}$	$\mathcal{L}_{KD}$	TEST				
			ROUGE	BLEU1	BLEU2	BLEU3	BLEU4
✓			49.93	49.22	36.53	28.61	23.47
	✓		49.72	49.75	36.51	28.30	23.25
✓	✓		51.08	50.16	38.08	29.51	23.82
✓	✓	✓	52.81	52.54	40.96	32.87	26.74

with both glosses and translations. It consists of 7096 training, 519 validation, and 642 test samples from 9 signers, with a vocabulary of 1066 glosses and 2877 German words. (2) **CSL-Daily** [53] is a Chinese SL dataset containing 18401, 1077, and 1176 videos for training, validation, and testing from 10 signers. It provides 2000 glosses and 2343 words for translation. (3) **How2Sign** [8] is a large-scale American Sign Language (ASL) dataset comprising over 80 hours of multi-view video and multimodal data. We use only the frontal-view RGB videos, including 31128, 1741, and 2322 samples for training, validation, and testing, respectively. (4) **OpenASL** [37] is another large ASL dataset with over 280 hours of video from more than 200 signers. It includes 96476 training, 997 validation, and 999 test samples.

Data Preprocessing. Following prior work [11, 15, 31], we adopt the same preprocessing pipeline to ensure fair comparisons. During training, we apply standard data augmentations, including resizing frames to 256×256 pixels, random cropping to 224×224 pixels, random horizontal flipping with a probability of 0.5, and random temporal scaling within ±20%. During inference, frames are resized to 256×256 and center-cropped to 224×224.

Implementation Details. Our architecture consists of following key components: (1) *Visual Encoder*: In our baseline setting, we choose PoolFormer [49] with 1D CNN layers used in previous work [11, 31]. (2) *Llama*: We adopt the Llama-3.2-1B model[1] in our baseline setting. We choose the base pretrained model rather than instruction-tuned variants (*e.g.*, Llama-3.1-1B-Instruct), as the effects of prompt engineering are beyond the scope of this work. Instead, our focus is on improving the performance of base LLMs in the context of gloss-free SLT. For LoRA used in Llama, we apply LoRA [19] with a rank of 4 and a scaling factor ('LoRA alpha') of 32. LoRA adapters are inserted into the 'q-proj', 'v-proj', and 'o-proj' layers. In addition, we fully fine-tune both the input embedding layer and the output layer in Llama, which are included in the list of trainable modules. (3) *Training Setting*: We train the model using

[1]https://huggingface.co/meta-llama/Llama-3.2-1B

Table 3: Effect of pseudo-gloss. POS means Part-Of-Speech.

Model	ROUGE	BLEU1	BLEU4
w/o lemmatizer	50.19	51.62	24.74
w/o stopword	50.00	50.04	24.16
POS-only	33.00	31.41	18.34
Ours	52.81	52.54	26.74

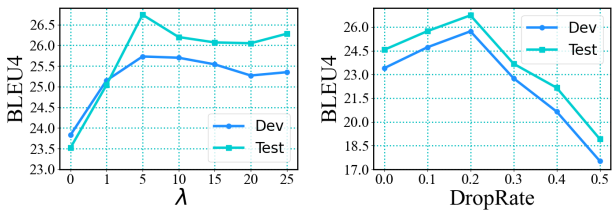


Figure 2: Effect of different λ in VPD and different drop rates in FPG-CTC

the Adam optimizer with a weight decay of 0.0001 for 50 epochs on three GeForce RTX 3090 GPUs. The initial learning rate is set to 1e-4 for the recognition model and 1e-5 for the translation model, and is decayed by a factor of 0.5 at epochs 10, 20 and 30. The batch size is set to 3. For FPG-CTC pretraining, the random dropout rate is set to 0.2. The loss weights α, β in Equation 7 are set to 1, and λ is set to 5. To ensure the model can be trained end-to-end within 24GB of GPU memory, the entire model is trained in half-precision.

Evaluation Metrics. To evaluate our model, we adopt the ROUGE-L F1 Score [24], BLEU-1,2,3,4 [32] and BLEURT [36] for gloss-free SLT, which are common metrics in existing work [4, 13].

6 Ablation Study

Following previous work [2, 4, 11], we perform ablation studies on the Phoenix14T dataset to verify the effectiveness of our proposed SignLlama model.

Effect of the Proposed Training Strategies. To validate the effectiveness of our proposed FPG-CTC pretraining and VPD training strategies, we compare our approach against baselines that are trained via direct fine-tuning. The results are shown in Table 1. The direct fine-tuning baseline model performs poorly, achieving only 5.46 BLEU-4 on the 'Test' sets. In contrast, applying FPG-CTC significantly boosts performance, demonstrating that our pseudo-gloss CTC pretraining effectively trains the visual encoder and facilitates better alignment between visual and textual features. Combining FPG-CTC with VPD yields the best results, which suggests that prioritizing visual features leads to better performance.

Effect of VPD Loss. The proposed VPD loss comprises three components: $\mathcal{L}_T^{vt}$, $\mathcal{L}_T^{v}$, and $\mathcal{L}_{KD}$. As shown in Table 2, training with $\mathcal{L}_T^{vt}$ alone (*i.e.*, standard autoregressive learning with both visual and textual inputs) achieves BLEU-4 scores of 23.47 on the test set. In comparison, using only $\mathcal{L}_T^{v}$, which relies solely on visual inputs, yields slightly lower BLEU-4 score of 23.25. When the distillation loss $\mathcal{L}_{KD}$ is introduced, the model achieves substantial performance

Table 4: Effect of different Llama pretrained model.

Model	TEST		
	ROUGE	BLEU1	BLEU4
Llama2 7B	52.61	52.14	26.26
Llama2 13B	52.73	52.54	26.54
Llama3.2 1B	52.81	52.54	26.74
Llama3.2 3B	52.15	52.61	26.97
Llama3.1 8B	52.98	52.55	26.85
Llama3 8B	52.40	52.24	26.82

gains, reaching 26.74 BLEU-4 on the test set. These results highlight the effectiveness of VPD in enhancing visual feature utilization via KD, thereby improving overall translation quality.

Effect of Pseudo Gloss. To evaluate the effect of different preprocessing steps on FPG-CTC, we conduct ablation studies, as shown in Table 3. Removing either the lemmatization or stopword filtering results in a small drop in performance. Using only POS information significantly degrades performance, showing that part-of-speech information alone is insufficient. These results indicate that our approach benefits from the combined preprocessing steps of lemmatization and stopword removal, while the overall trends remain consistent across different settings.

Effect of the Hyperparameter λ in VPD. We introduce a hyperparameter λ in the VPD loss to control the contribution of the distillation term. To analyze its impact, we conduct experiments with varying values of λ. As illustrated in Figure 2, the best BLEU-4 performance is observed when λ is set to 5. Based on this observation, we set $\lambda = 5$ in all experiments for better performance.

Effect of Drop Rate in FPG-CTC. We apply random dropping during pseudo-gloss generation in FPG-CTC and further investigate the impact of varying dropout rates. As shown in Figure 2, model performance improves with moderate dropout rates but degrades when the dropout rate is too high, as excessive word removal hinders effective alignment learning between SL videos and pseudo-gloss sequences during FPG-CTC pretraining. Accordingly, we use a drop rate of 0.2 in all experiments.

Effect of Different Llama Sizes. Llama is a family of LLMs ranging from 1B to 405B parameters. Given our limited GPU resources, we scale our model from the 1B model to larger variants up to 13B, including both the Llama 2 and Llama 3 series. As shown in Table 4, interestingly, increasing the size of the pretrained model yields only marginal improvements. This may be attributed to the fact that emergent behaviors often associated with LLMs tend to manifest at larger model scales. **This conclusion is consistent with the findings in [50].** Since scaling the Llama model size does not yield significant improvements, we report results using the Llama 3.2 1B model in all subsequent comparisons, unless otherwise specified.

Training and Inference Speed. To evaluate VPD's impact on training time, we conduct experiments on three GeForce RTX 3090 GPUs with a batch size of 3, and limit the maximum training sample to

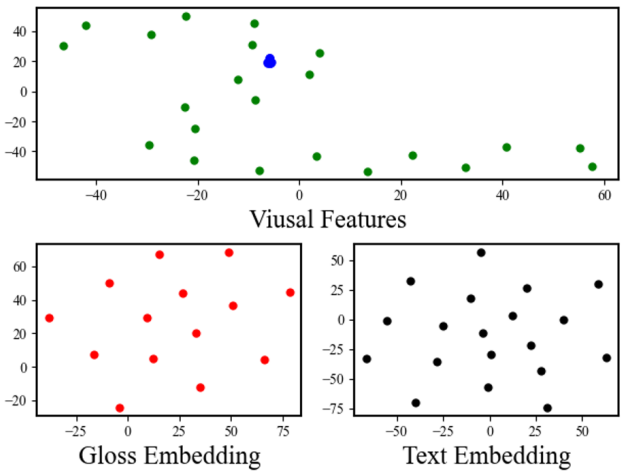


Figure 3: Visualization of feature distributions using t-SNE [42]. Blue and green points represent the visual features without/with FPG-CTC. Red and black points represent the corresponding gloss and text sequence embeddings of Llama.

Table 5: Training speed.

Model	Training Time
w VPD	22min/epoch
w/o VPD	20 min/epoch

Table 6: Model inference speed.

Inference(s)	LLama Model Size			
	1B	3B	8B	13B
250 frames	0.85	2.14	2.55	4.23

280 frames on Phoenix14T. As shown in Table 5, VPD introduces only a slight overhead, increasing the training time per epoch from 20 minutes to 22 minutes. This demonstrates that VPD brings negligible additional cost while providing performance gains. Here, we also report the inference latency on a single NVIDIA RTX 3090 GPU with approximately 250 frames for one video. As shown in Table 6, even the 13B model maintains reasonable inference efficiency, while the 1B version achieves 0.85s/video with one 3090 GPU.

7 Qualitative Results

FPG-CTC Visualization. We visualize the visual feature distribution of one test sample from Phoenix14T, trained with and without FPG-CTC, alongside the corresponding gloss/text sequence embeddings from Llama. As shown in Figure 3, the gloss/text inputs are tokenized into discrete tokens. In contrast, the visual feature distribution trained without FPG-CTC exhibits a non-discriminative and continuous pattern. When trained with FPG-CTC, the visual features are better tokenized and exhibit a more discriminative distribution, which closely resembles that of the textual input. This demonstrates that FPG-CTC effectively tokenizes SL videos and provides suitable inputs for the LLMs.

VPD Visualization. To investigate the effect of VPD, we visualize one of the attention heads from the final layer of the Llama model (other heads and layers exhibit similar patterns) under models trained with and without VPD. As shown in Figure 4, 's' denotes the start-of-sequence token, 'v0–v11' represent visual tokens, and

Table 7: Comparison of SLT performance on Phoenix14T datasets.

Gloss free SLT	Phoenix14T									
	DEV					TEST				
	ROUGE	BLEU1	BLEU2	BLEU3	BLEU4	ROUGE	BLEU1	BLEU2	BLEU3	BLEU4
SignCL [47]						49.04	49.76	36.85	29.97	22.74
Sign2GPT[44]	-	-	-	-	-	48.90	49.54	35.96	28.83	22.52
GFSLT-VLP-SignCL [47]	-	-	-	-	-	49.04	49.76	36.85	29.97	22.74
LLaVA-SLT [23]	-	-	-	-	-	50.44	51.20	37.51	29.39	23.43
FLa-LLM [6]	-	-	-	-	-	45.27	46.29	35.33	28.03	23.09
C2RL [5]	-	-	-	-	-	50.96	52.81	40.20	32.20	26.75
SignLLM [17]	44.49	46.88	36.59	29.91	25.25	47.23	45.21	34.78	28.05	23.40
MixSignGraph [15]	51.71	51.07	37.97	29.98	24.87	51.14	50.01	38.04	29.95	24.02
SignDINO [9]	52.36	53.64	38.65	30.49	25.62	52.75	52.13	39.64	33.73	25.46
SignLlama 1B	52.53	52.13	38.88	30.61	25.73	52.81	52.54	40.96	32.87	26.74

Table 8: Comparison of SLT performance on CSL-Daily datasets.

Gloss-free SLT	CSL-Daily									
	DEV					TEST				
	ROUGE	BLEU1	BLEU2	BLEU3	BLEU4	ROUGE	BLEU1	BLEU2	BLEU3	BLEU4
GASLT [48]	-	-	-	-		20.35	19.90	9.94	5.98	4.07
GFSLT [52]	35.16	37.60	23.30	14.89	9.92	35.42	37.69	23.28	14.93	9.88
GFSLT-VLP [52]	36.44	39.20	25.02	16.35	11.07	36.70	39.37	24.93	16.26	11.00
SignCL [47]						48.92	47.47	32.53	22.62	16.16
Sign2GPT[44]	-	-	-	-	-	42.36	41.75	28.73	20.60	15.40
SignLLM [17]	39.18	42.45	26.88	17.90	12.23	39.91	39.55	28.13	20.07	15.75
GFSLT-VLP-SignCL [47]	-	-	-	-	-	48.92	47.47	32.53	22.62	16.16
FLa-LLM [6]	-	-	-	-	-	37.25	37.13	25.12	18.38	14.20
C2RL [5]	-	-	-	-	-	48.21	49.32	36.28	27.54	21.61
LLaVA-SLT [23]	-	-	-	-	-	51.26	52.15	36.24	26.47	20.42
MixSignGraph [15]	49.16	49.98	36.42	26.89	20.43	49.93	50.24	36.91	27.54	20.78
SignLlama	51.62	52.12	41.23	33.23	24.83	51.43	52.38	39.78	33.91	24.76

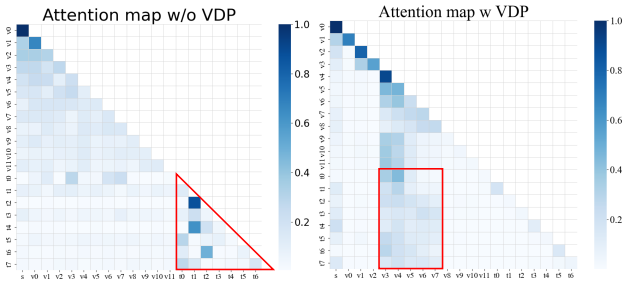


Figure 4: Visualization of the attention weights of Llama.

't0–t7' indicate text tokens. The model trained without VPD primarily relies on previous text tokens for next-token prediction, with limited attention to visual inputs. In contrast, the model trained with VPD allocates significantly more attention to the visual tokens. This demonstrates that VPD effectively prioritizes visual features for LLMs, thereby improving translation performance.

8 Comparisons

For a fair comparison, we only compare with **image-based GFSLT methods** that do not use **external SL datasets** for pretraining.

Evaluation on Phoenix14T Dataset. As shown in Table 7, we compare the GFSLT performance of our model with existing approaches on the Phoenix14T dataset. Our model achieves excellent results, surpassing most of previous methods and improving the BLUE4 score by 1.28 points over SignDINO [9], reaching a 52.81 ROUGE-L score and 26.74 BLEU-4 score on the test set.

Evaluation on CSL-Daily Dataset. As shown in Table 8, we also evaluate the GFSLT performance of our model on the CSL-Daily dataset. To better support Chinese translation, we replace the original Llama with the Chinese-Llama-2-1.3B model. As shown in the table, our model achieves strong performance with a BLEU-4 score of 24.76, achieving competitive performance.

Evaluation on How2Sign Dataset. How2Sign is a larger American SL dataset, only provides text labels. As shown in Table 9, we compare the GFSLT performance of our proposed model with existing

Table 9: Comparison of GFSLT performance on How2Sign.

Gloss-free SLT	How2Sign									
	DEV					TEST				
	ROUGE	BLEU1	BLEU2	BLEU3	BLEU4	ROUGE	BLEU1	BLEU2	BLEU3	BLEU4
YouTube-SLT [41]	-	-	-	-	-	-	14.96	5.11	2.26	1.22
YouTube-SLT-P [41]	-	-	-	-	-	-	37.82	24.13	16.92	12.39
SSVP-SLT [35]	-	-	-	-	-	25.70	30.20	16.70	10.50	7.00
C^2RL [5]	-	-	-	-	-	27.81	29.81	18.99	13.27	9.66
FLa-LLM [6]	-	-	-	-	-	27.81	29.81	18.99	13.27	9.66
GloFE-VN [25]	12.98	15.21	7.38	4.07	2.37	12.61	14.94	7.27	3.93	2.24
SLT-IV [39]	-	35.20	20.62	13.25	8.89	-	34.01	19.3	12.18	8.03
MixSignGraph [15]	25.41	26.82	16.70	11.48	8.36	25.71	26.65	16.55	11.68	8.69
SignLlama 1B	30.34	30.96	17.32	12.24	9.31	30.26	30.51	19.95	16.92	9.89

Table 10: Comparison of SLT performance on OpenASL.

Gloss-free SLT	OpenASL									
	DEV					TEST				
	ROUGE	BLEU1	BLEU2	BLEU3	BLEU4	ROUGE	BLEU1	BLEU2	BLEU3	BLEU4
OpenASL [37]	25.31	24.35	14.94	10.72	8.39	24.83	23.87	14.08	9.90	7.54
GloFE-VN [25]	21.37	21.06	12.34	8.68	6.68	21.75	21.56	12.74	9.05	7.06
MixSignGraph [15]	25.41	26.82	16.70	11.48	8.36	25.71	26.65	16.55	11.68	8.69
C^2RL [5]	-	-	-	-	-	31.36	31.46	21.85	16.58	13.21
SignLlama 1B	35.28	35.11	26.44	19.02	16.59	37.74	37.37	25.97	17.70	15.16

models. Our method, incorporating FPG-CTC and VPD, achieves a BLEU4 score of 14.47 on the test set, outperforming the previous SOTA model.

Evaluation on OpenASL Dataset. Similar to How2Sign, the OpenASL dataset is a large-scale benchmark with a large vocabulary and text-only annotations. Our model achieves excellent performance across all evaluation metrics on both the development and test sets. Compared to the previous best-performing model, C2RL [5], our model achieves substantial improvements, particularly in BLEU scores. On the test set, it improves BLEU-4 from 13.21 to 15.16, and BLEU-1 from 31.46 to 37.37, indicating more accurate and fluent generation.

9 Conclusion

In this paper, we focus on fine-tuning large language models (LLMs) for gloss-free sign language translation (GFSLT). We reveal two key challenges in this setting: (1) the inherent distributional gap between SL video features and textual features, which makes alignment difficult; and (2) the common practice of simply concatenating visual and textual inputs in autoregressive training, which causes the model to overemphasize textual cues, deprioritize visual information, and further amplify exposure bias during inference. To address these challenges, we propose two key contributions: Filtered Pseudo-Gloss CTC Pretraining (FPG-CTC), which leverages automatically generated pseudo-gloss sequences to supervise the training of the visual backbone; and Visual-Prioritized Distillation (VPD), a training strategy designed to encourage prioritizing visual features. Extensive experiments across multiple benchmark datasets demonstrate the effectiveness of our approach. Our model achieves competitive results on the gloss-free SLT task.

10 Limitations And Discussions

Here, we list some potential ideas that can be further explored to improve performance. First, our FPG-CTC approach generates pseudo-gloss sequences using simple rule-based NLP techniques. More discussion of effective pseudo-gloss generation strategies can be further investigated in future work. Second, VPD requires dual forward passes for visual-text prediction and visual-only prediction, increasing training cost. More efficient approaches to enhance visual feature utilization are worth investigating. In addition, we highlight the potential negative social impacts. Our model, like other deep learning approaches, requires significant computational resources, which could raise issues related to energy use and environmental impact. Moreover, since our method is data-driven, it may inherit potential biases from the training data. Careful dataset selection and balanced data are important to reduce such risks.

Scaling to Larger Models. Due to hardware limitations, we scale the LLM from 1B up to 13B in our experiments. In future work, we plan to explore larger pretrained LLMs, as emergent capabilities observed at greater scales may lead to further gains in GFSLT.

Acknowledgments

This work is supported in part by National Natural Science Foundation of China under Grant Nos. 62172208, 92467202, 62272216; Key Projects of Jiangsu Provincial Basic Research Program under Grant No. BK20243040; JiangSu Natural Science Foundation under Grant No. BK20251989. This work is partially supported by Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (No. JYB2025XDXM118); the "111 Center" (No. B26023); Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

[1] Kirolos Ataallah, Xiaoqian Shen, Eslam Abdelrahman, Essam Sleiman, Deyao Zhu, Jian Ding, and Mohamed Elhoseiny. 2024. MiniGPT4-Video: Advancing Multimodal LLMs for Video Understanding with Interleaved Visual-Textual Tokens. *arXiv preprint arXiv:2404.03413* (2024).

[2] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. 2018. Neural sign language translation. In *CVPR*. 7784–7793.

[3] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechu Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. 2023. MiniGPT-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478* (2023).

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

[5] Zhigang Chen, Benjia Zhou, Yiqing Huang, Jun Wan, Yibo Hu, Hailin Shi, Yanyan Liang, Zhen Lei, and Du Zhang. 2025. C 2 RL: Content and Context Representation Learning for Gloss-free Sign Language Translation and Retrieval. *IEEE Transactions on Circuits and Systems for Video Technology* (2025).

[6] Zhigang Chen, Benjia Zhou, Jun Li, Jun Wan, Zhen Lei, Ning Jiang, Quan Lu, and Guoqing Zhao. 2024. Factorized Learning Assisted with Large Language Model for Gloss-free Sign Language Translation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. 7071–7081.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ICLR* (2021).

[8] 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*. 2735–2744.

[9] Shiwei Gan, Xiao Liu, Yafeng Yin, Nan Liu, Kuizhuang Liu, Desibieer Tuerdaken, Zhiwei Jiang, Lei Xie, Sanglu Lu, and Hongkai Wen. 2026. Learning Effective Sign Features without Text for Gloss-free Sign Language Translation. (2026).

[10] Shiwei Gan, Lichen Wang, Xiao Liu, Yafeng Yin, Kuizhuang Liu, Sanglu Lu, and Lei Xie. 2026. Sign Language Question Answering: A New Task, Benchmark, and Baseline for Sign Language Understanding. *arXiv preprint arXiv:2607.27826* (2026).

[11] Shiwei Gan, Yafeng Yin, Zhiwei Jiang, Hongkai Wen, Lei Xie, and Sanglu Lu. 2024. SignGraph: A Sign Sequence is Worth Graphs of Nodes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13470–13479.

[12] Shiwei Gan, Yafeng Yin, Zhiwei Jiang, Kang Xia, Lei Xie, and Sanglu Lu. 2023. Contrastive learning for sign language recognition and translation. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*. 763–772.

[13] Shiwei Gan, Yafeng Yin, Zhiwei Jiang, Lei Xie, and Sanglu Lu. 2021. Skeleton-Aware Neural Sign Language Translation. In *MM*. 4353–4361.

[14] Shiwei Gan, Yafeng Yin, Zhiwei Jiang, Lei Xie, and Sanglu Lu. 2023. Towards Real-Time Sign Language Recognition and Translation on Edge Devices. In *Proceedings of the 31st ACM International Conference on Multimedia*. 4502–4512.

[15] Shiwei Gan, Yafeng Yin, Zhiwei Jiang, Lei Xie, Sanglu Lu, and Hongkai Wen. 2025. MixSignGraph: A Sign Sequence is Worth Mixed Graphs of Nodes.

[16] Peter Gaskell, Jana Pavlasek, Tom Gao, Abhishek Narula, Stanley Lewis, and Odest Chadwicke Jenkins. 2024. MBot: A modular ecosystem for scalable robotics education. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 18294–18300.

[17] Jia Gong, Lin Geng Foo, Yixuan He, Hossein Rahmani, and Jun Liu. 2024. Llms are good sign language translators. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18362–18372.

[18] Michael Guevarra, Indronil Bhattacharjee, Srijita Das, Christabel Wayllace, Carrie Demmans Epp, Matthew E Taylor, and Alan Tay. 2025. An LLM-Guided Tutoring System for Social Skills Training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 29643–29645.

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

[20] Peiqi Jiao, Yuecong Min, and Xilin Chen. 2024. Visual alignment pre-training for sign language translation. In *European Conference on Computer Vision*. Springer, 349–367.

[21] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*. PMLR, 19730–19742.

[22] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. 2023. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355* (2023).

[23] Han Liang, Chengyu Huang, Yuecheng Xu, Cheng Tang, Weicai Ye, Juze Zhang, Xin Chen, Jingyi Yu, and Lan Xu. 2024. LLaVA-SLT: Visual Language Tuning for Sign Language Translation. *arXiv preprint arXiv:2412.16524* (2024).

[24] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[25] Kezhou Lin, Xiaohan Wang, Linchao Zhu, Ke Sun, Yi Yang, et al. 2023. Gloss-Free End-to-End Sign Language Translation. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

[26] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems* 36 (2023), 34892–34916.

[27] Xiao Liu, Shiwei Gan, Yafeng Yin, Bowen Guo, Zhiwei Jiang, Shunmei Meng, Lei Xie, and Sanglu Lu. 2026. SignPR: A Progressive Vector-Quantized Diffusion Framework for Sign Language Production. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2198–2208.

[28] Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics* 8 (2020), 726–742.

[29] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2024. Video-ChatGPT: Towards Detailed Video Understanding via Large Vision and Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*.

[30] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2024. VideoGPT+: Integrating Image and Video Encoders for Enhanced Video Understanding. *arxiv* (2024). https://arxiv.org/abs/2406.09418

[31] Yuecong Min, Aiming Hao, Xiujuan Chai, and Xilin Chen. 2021. Visual alignment constraint for continuous sign language recognition. In *ICCV*. 11542–11551.

[32] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*. 311–318.

[33] Long Qian, Juncheng Li, Yu Wu, Yaobo Ye, Hao Fei, Tat-Seng Chua, Yueting Zhuang, and Siliang Tang. 2024. Momentor: Advancing Video Large Language Model with Fine-Grained Temporal Reasoning. arXiv:2402.11435 [cs.CV]

[34] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[35] Phillip Rust, Bowen Shi, Skyler Wang, Necati Cihan Camgöz, and Jean Maillard. 2024. Towards privacy-aware sign language translation at scale. *arXiv preprint arXiv:2402.09611* (2024).

[36] Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. BLEURT: Learning Robust Metrics for Text Generation. In *Proceedings of ACL*.

[37] Bowen Shi, Diane Brentari, Gregory Shakhnarovich, and Karen Livescu. 2022. Open-Domain Sign Language Translation Learned from Online Video. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 6365–6379.

[38] Ozge Mercanoglu Sincan, Necati Cihan Camgoz, and Richard Bowden. 2023. Is context all you need? scaling neural sign language translation to large domains of discourse. In *Proceedings of the IEEE/CVF international conference on computer vision*. 1955–1965.

[39] Laia Tarrés, Gerard I Gállego, Amanda Duarte, Jordi Torres, and Xavier Giró-i Nieto. 2023. Sign language translation from instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5624–5634.

[40] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[41] 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 (2023), 29029–29047.

[42] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[43] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Song XiXuan, et al. 2024. Cogvlm: Visual expert for pretrained language models. *Advances in Neural Information Processing Systems*

37 (2024), 121475–121499.

[44] Ryan Wong, Necati Cihan Camgoz, and Richard Bowden. 2024. SIGN2GPT: leveraging large language models for gloss-free sign language translation. In *ICLR 2024: The Twelfth International Conference on Learning Representations*.

[45] Xin Xiao, Bohong Wu, Jiacong Wang, Chunyuan Li, Haoyuan Guo, et al. 2024. Seeing the image: Prioritizing visual correlation by contrastive alignment. *Advances in Neural Information Processing Systems* 37 (2024), 30925–30950.

[46] Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid. 2023. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10714–10726.

[47] Jinhui Ye, Xing Wang, Wenxiang Jiao, Junwei Liang, and Hui Xiong. 2024. Improving Gloss-free Sign Language Translation by Reducing Representation Density. In *NeurIPS*.

[48] Aoxiong Yin, Tianyun Zhong, Li Tang, Weike Jin, Tao Jin, and Zhou Zhao. 2023. Gloss attention for gloss-free sign language translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2551–2562.

[49] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. 2022. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10819–10829.

[50] Biao Zhang, Garrett Tanzer, and Orhan Firat. 2024. Scaling sign language translation. *Advances in neural information processing systems* 37 (2024), 114018–114047.

[51] Jihai Zhang, Wengang Zhou, and Houqiang Li. 2014. A threshold-based hmm-dtw approach for continuous sign language recognition. In *ICIMCS*. 237–240.

[52] Benjia Zhou, Zhigang Chen, Albert Clapés, Jun Wan, Yanyan Liang, Sergio Escalera, Zhen Lei, and Du Zhang. 2023. Gloss-free sign language translation: Improving from visual-language pretraining. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 20871–20881.

[53] Hao Zhou, Wengang Zhou, Weizhen Qi, Junfu Pu, and Houqiang Li. 2021. Improving Sign Language Translation with Monolingual Data by Sign Back-Translation. In *CVPR*. 1316–1325.

[54] Qidan Zhu, Jing Li, Fei Yuan, and Quan Gan. 2024. Multiscale temporal network for continuous sign language recognition. *Journal of Electronic Imaging* 33, 2 (2024), 023059–023059.

[55] Ronglai Zuo and Brian Mak. 2022. C2SLR: Consistency-Enhanced Continuous Sign Language Recognition. In *CVPR*. 5131–5140.

A Datasets

Details of the Datasets. We provide detailed information on the five datasets used in our paper in Table 11. The datasets we downloaded may slightly differ from the official versions described in their respective papers. We also provide the token vocabulary obtained by the Llama tokenizer and the vocabulary in FPG-CTC of processed text for reference.

B Model Details

Our FPG-CTC module generates a pseudo-gloss sequence from the target text by applying lemmatization and removing function words such as prepositions and conjunctions. Since certain words in the text may not be explicitly expressed in the corresponding SL video, we further introduce random dropping to remove a fixed proportion of tokens, producing the final pseudo-gloss labels. The NLTK toolkit is used for pseudo-gloss preprocessing steps.

C Ablation Study on Hyperparameters

Effect of Different λ. In the main paper, we illustrated the impact of different λ values on our model's performance using a line chart (Figure 2). Here, we provide the detailed results in Table 12. As shown, when $\lambda = 0$ (i.e., without the VPD loss), the model achieves the lowest performance. Increasing λ leads to notable improvements across ROUGE and BLEU metrics, peaking around $\lambda = 5$ to $\lambda = 10$. Therefore, we set λ to 5 in our final model to balance performance and stability.

Effect of Different Drop Rates. In the main paper, we illustrated the impact of different drop rates in FPG-CTC using a line chart (Figure 2). Here, we provide detailed quantitative results in Table 13. The model performance improves as the drop rate increases from 0 to 0.2, reaching the best results at a drop rate of 0.2 across all ROUGE and BLEU metrics. However, further increasing the drop rate beyond 0.2 leads to significant performance degradation, as excessive word dropping hinders effective alignment learning.

D Ablation Study on Proposed Module

Effect of FPG-CTC.. We evaluate the effectiveness of our proposed FPG-CTC module by comparing the SLT performance on four benchmark datasets with and without FPG-CTC pretraining. As shown in Table 14, incorporating FPG-CTC consistently yields significant improvements across all datasets and evaluation metrics. For instance, on the Phoenix14T dataset, ROUGE and BLEU-4 scores improve from 20.09/5.46 to 52.81/26.74 on the test set, indicating that the pseudo-gloss supervision helps align sign video features with text features more effectively.

Additionally, we report the CSLR performance based on the pseudo-gloss sequences generated by FPG-CTC (right side of Table 14). Although the pseudo-gloss labels are not ground truth and the resulting WERs are suboptimal, they still demonstrate a reasonable level of recognition performance, suggesting that the pseudo sequences preserve coherent structure and temporal alignment. Overall, these results demonstrate that FPG-CTC offers strong supervisory signals for improving gloss-free SLT, and lays a solid foundation for high-quality translation without relying on ground-truth gloss annotations.

E Visualization

SLT Qualitative Results. Figure 5 presents a qualitative analysis of our SignLlama model on the gloss-free SLT task, with examples selected from the test sets of Phoenix14T, CSL-Daily, How2Sign, and OpenASL. The results show that SignLlama, when equipped with both FPG-CTC and VPD, produces the most accurate translations. In contrast, the end-to-end fine-tuning model performs worse, highlighting the effectiveness of our proposed FPG-CTC and VPD.

Visualization of Attention Maps of Llama. Here, we provide more attention maps from the last layer of Llama models trained with and without VPD. As shown in Figure 6 and 7, VPD significantly enhances the model's ability to prioritize relevant visual features, thereby improving its overall accuracy.

Table 11: Details of datasets used in our paper. Voc: vocabulary size. PG: pseudo gloss.

Dataset	Token Voc in Text	PG Voc in FPG-CTC	Video Samples		
			Train	Test	Validation
Phoenix14T	2121	30,01	7,096	642	519
CSL-Daily	7277	2377	18,401	1,176	1,077
How2Sign	9534	15347	30,904	2328	1713
OpenASL	19,250	25,402	96,476	975	966

Table 12: Effect of different λ in Equation 7.

λ	DEV					TEST				
	ROUGE	BLEU1	BLEU2	BLEU3	BLEU4	ROUGE	BLEU1	BLEU2	BLEU3	BLEU4
0	50.96	50.68	35.98	28.75	23.83	51.08	50.16	38.08	29.51	23.52
1	51.03	51.73	38.07	29.26	25.15	51.41	50.50	39.16	31.27	25.04
5	52.53	52.13	38.88	30.61	25.73	52.81	52.54	40.96	32.87	26.74
10	52.10	51.21	38.79	29.97	25.70	52.61	51.91	39.72	31.61	26.20
15	51.13	51.16	38.65	29.77	25.54	51.31	49.87	37.31	29.32	26.07
20	51.71	51.66	38.36	29.52	25.27	51.48	49.75	37.29	29.33	26.05
25	51.57	51.87	38.51	29.62	25.35	51.20	50.11	37.58	29.58	26.28

Table 13: Effect of different drop rate in FPG-CTC.

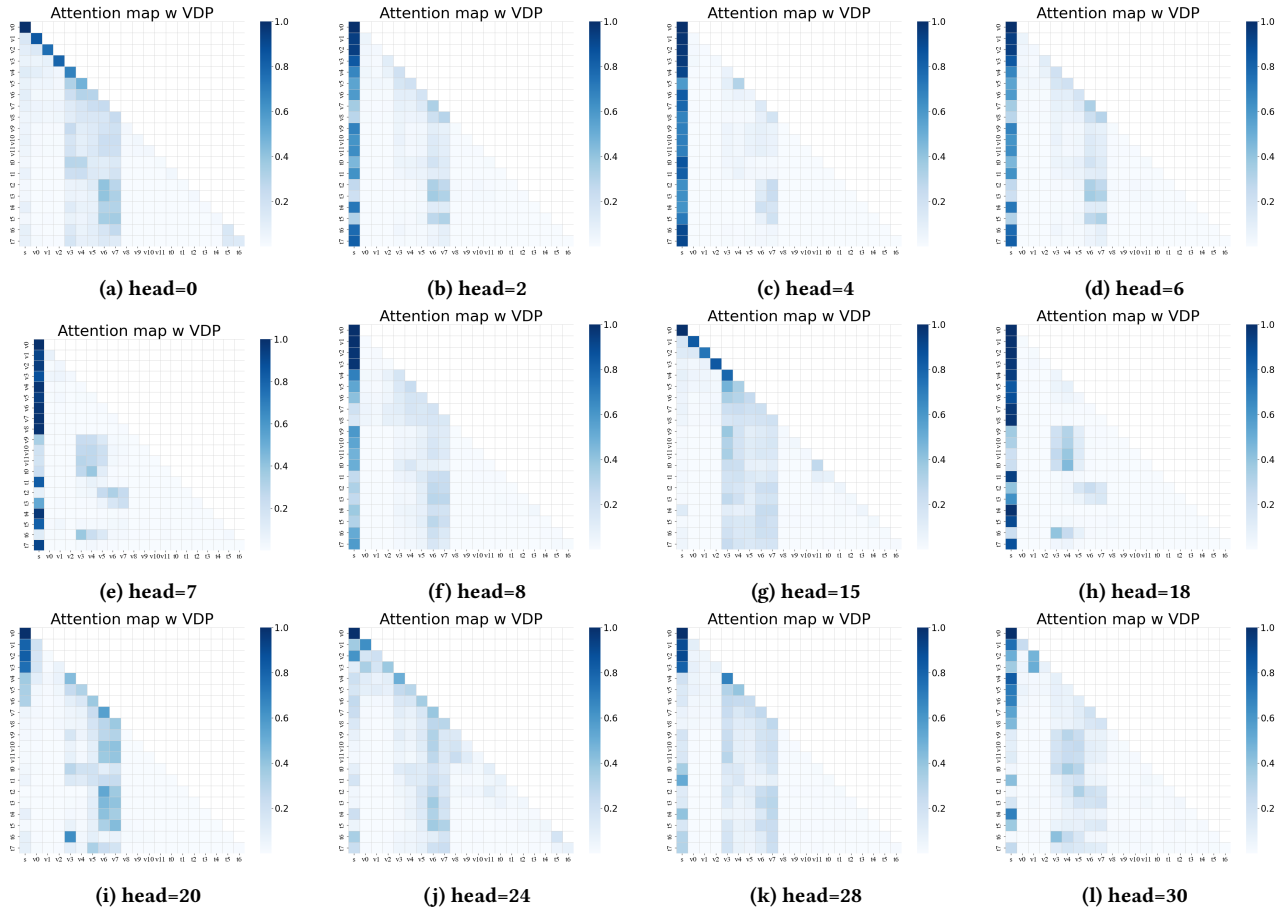
DropRate	DEV					TEST					Dev		Test	
	ROUGE	BLEU1	BLEU2	BLEU3	BLEU4	ROUGE	BLEU1	BLEU2	BLEU3	BLEU4	WER	Del/Ins	WER	Del/Ins
0	48.77	47.53	34.57	26.56	23.40	49.05	48.28	35.59	27.68	24.56	69.55	53.67/ 5.23	69.35	53.76/ 4.65
0.1	50.35	48.63	36.03	28.06	24.72	50.31	49.39	36.88	28.94	25.74	72.95	63.67/ 0.44	72.77	63.03/ 0.45
0.2	52.53	52.13	38.88	30.61	25.73	52.81	52.54	40.96	32.87	26.74	78.77	74.77/ 0.15	78.95	73.68/0.18
0.3	48.65	47.62	36.14	28.02	22.76	48.86	48.33	35.39	27.12	23.68	88.28	86.95/ 0.03	88.36	86.99/ 0.01
0.4	45.80	44.97	33.65	25.70	20.64	45.87	46.13	33.14	25.25	22.16	94.60	94.25/ 0.00	94.12	93.73/0.01
0.5	42.24	41.76	29.99	22.23	17.53	41.64	42.75	29.51	21.73	18.92	94.77	94.43/ 0.00	94.63	94.17/0.01

Table 14: Effect of proposed FPG-CTC for SLT. Besides, we also show the CSLR performance based on pseudo gloss sequence obtained by FPG-CTC in the right part.

Dataset	Model	Dev					Test					Dev		Test	
		ROUGE	BLEU1	BLEU2	BLEU3	BLEU4	ROUGE	BLEU1	BLEU2	BLEU3	BLEU4	WER	Del/Ins	WER	Del/Ins
Phoenix14T	w/o FPG-CTC	20.11	17.22	8.21	6.18	5.23	20.09	18.12	58.56	6.46	5.46	-	-/-	-	-/-
	w/ FPG-CTC	52.53	52.13	38.88	30.61	25.73	52.81	52.54	40.96	32.87	26.74	78.62	72.57/00.22	78.32	72.23/00.24
CSL-Daily	w/o FPG-CTC	36.51	34.54	23.72	14.655	9.11	35.31	33.65	19.34	12.67	8.27	-	-/-	-	-/-
	w/ FPG-CTC	51.62	52.12	41.23	33.23	24.83	51.43	52.38	39.78	33.91	24.76	69.75	56.79/1.60	69.77	55.89/1.79
How2Sign	w/o FPG-CTC	19.45	24.15	13.56	7.90	5.15	19.15	24.47	12.98	7.81	5.14	-	-/-	-	-/-
	w/ FPG-CTC	37.34	37.96	22.32	17.24	14.31	39.26	40.51	26.95	18.92	14.47	70.62	69.57/00.22	70.32	68.23/00.24
OpenASL	w/o FPG-CTC	13.13	12.69	4.57	2.39	1.96	12.68	11.27	4.87	2.45	1.98	-	-/-	-	-/-
	w/ FPG-CTC	35.28	35.11	26.44	19.02	16.59	37.74	37.37	25.97	17.70	15.16	82.57	69.21/0.65	82.85	69.19/0.69

example(a)	Phoenix14T dataset
Groundtruth	und nun die wettervorhersage für morgen sonntag den zwölften juli (And now the weather forecast for tomorrow, Sunday, July 11th)
End-To-End	und nun die wettervorhersage für morgen mittwoch den fünfundzwanzigsten august (and now the weather forecast for tomorrow, Wednesday the twenty-fifth of August)
+ FPG-CTC	und nun die wettervorhersage für morgen sonntag den elften juli (And now the weather forecast for tomorrow, Sunday, July 12th)
+ VPD	und nun die wettervorhersage für morgen sonntag den zwölften juli (And now the weather forecast for tomorrow, Sunday, July 11th)
example(b)	CSL-daily dataset
Groundtruth	是， 他这几天很忙没有时间休息。 (Yes, he has been very busy these days and has no time to rest.)
End-To-End	他最近很忙 He's been very busy lately.
+ FPG-CTC	他这几天很忙没有时间休息。 (He has been very busy these days and has no time to rest.)
+ VPD	是， 他这几天很忙没有时间休息。 (Yes, he has been very busy these days and has no time to rest.)
example(c)	How2Sign dataset
Groundtruth	So, we've got to find a way to get to the exit.
End-To-End	There is a way to the exit.
+ FPG-CTC	we have found a way to get to the exit.
+ VPD	we've got to find a way to get to the exit.
example(d)	OpenASL dataset
Groundtruth	We are thankful to be here, and also to the Saudi club for their help.
End-To-End	We thanks for their help.
+ FPG-CTC	We are thankful to be here, and thanks for their help.
+ VPD	We are thankful to be here, and also thanks for the club's help.

Figure 5: SLT Qualitative results on Phoenix-2014T, CSL-Daily, How2Sign and OpenASL.

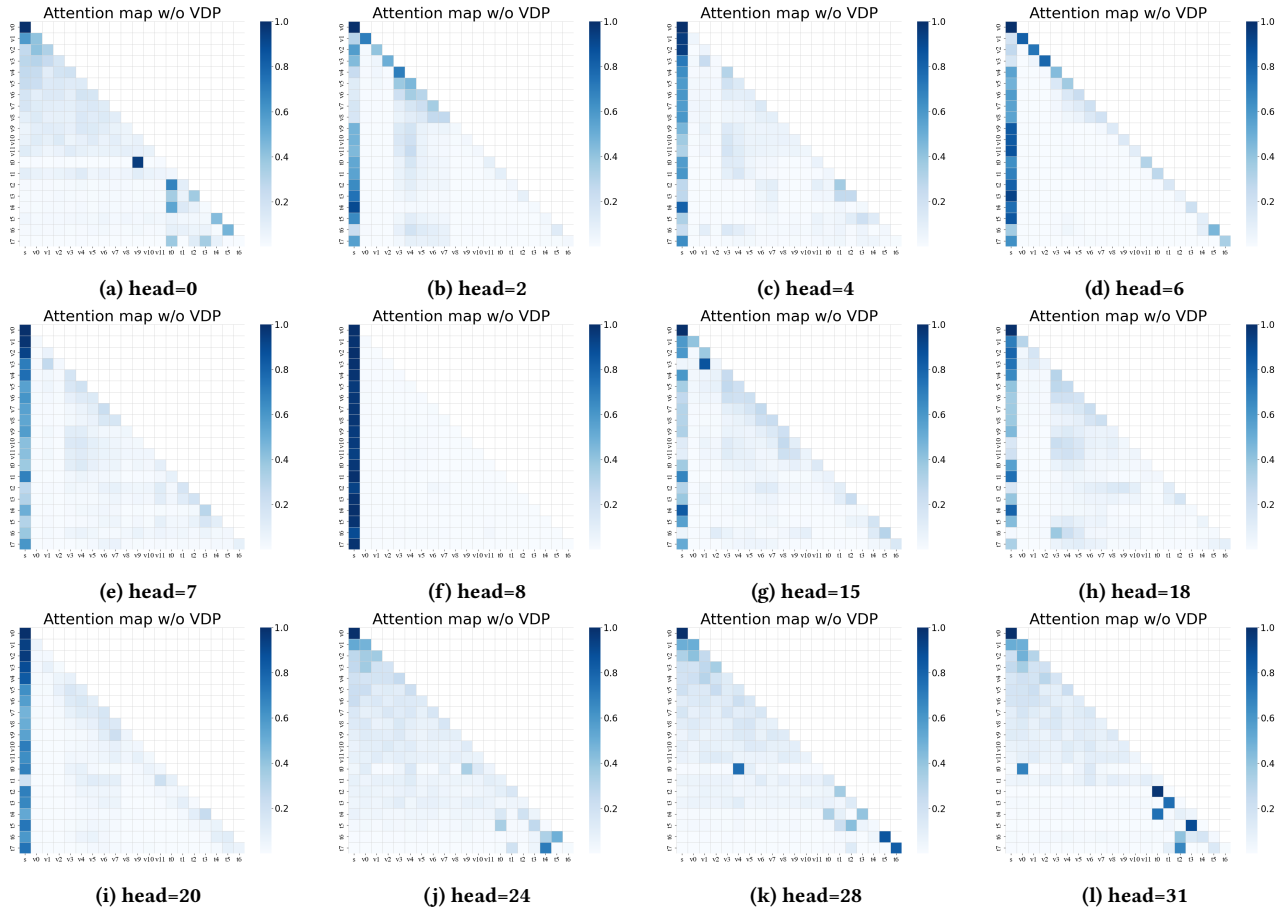


(a) head=0 (b) head=2 (c) head=4 (d) head=6

(e) head=7 (f) head=8 (g) head=15 (h) head=18

(i) head=20 (j) head=24 (k) head=28 (l) head=30

Figure 6: Visualization of attention maps in the last layer of Llama with VPD, showing that the attention is mainly focused on visual tokens.



(a) head=0 (b) head=2 (c) head=4 (d) head=6

(e) head=7 (f) head=8 (g) head=15 (h) head=18

(i) head=20 (j) head=24 (k) head=28 (l) head=31

Figure 7: Visualization of attention weight in the last layer of Llama without VPD, showing that the attention is mainly focused on text tokens.