Factorized Learning Assisted with Large Language Model for Gloss-free Sign Language Translation

Zhigang Chen[1,2], **Benjia Zhou**[3], **Jun Li**[1,2], **Jun Wan**[1,2,3†], **Zhen Lei**[1,2,4]
Ning Jiang[5], **Quan Lu**[5], **Guoqing Zhao**[5]
[1]MAIS, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[3]Macau University of Science and Technology, Macau, China
[4]CAIR, HKISI, Chinese Academy of Sciences, Hong Kong, China
[5]Mashang Consumer Finance, Chongqing, China
{chenzhigang2021, jun.wan}@ia.ac.cn

Abstract

Previous Sign Language Translation (SLT) methods achieve superior performance by relying on gloss annotations. However, labeling high-quality glosses is a labor-intensive task, which limits the further development of SLT. Although some approaches work towards gloss-free SLT through jointly training the visual encoder and translation network, these efforts still suffer from poor performance and inefficient use of the powerful Large Language Model (LLM). Most seriously, we find that directly introducing LLM into SLT will lead to insufficient learning of visual representations as LLM dominates the learning curve. To address these problems, we propose **F**actorized **L**earning **a**ssisted with **L**arge **L**anguage **M**odel (**FLa-LLM**) for gloss-free SLT. Concretely, we factorize the training process into two stages. In the visual initialing stage, we employ a lightweight translation model after the visual encoder to pre-train the visual encoder. In the LLM fine-tuning stage, we freeze the acquired knowledge in the visual encoder and integrate it with a pre-trained LLM to inspire the LLM's translation potential. This factorized training strategy proves to be highly effective as evidenced by significant improvements achieved across three SLT datasets which are all conducted under the gloss-free setting.

Keywords: Sign language translation, Large language model, Factorized learning

1. Introduction

Sign language is the primary form of communication for over 70 million deaf people worldwide. It is a visual language consisting of gestures, body movements, and expressions which has a unique linguistic structure. Therefore, it differs greatly from the natural spoken language. The study of sign language processing can bring great convenience between hearing and deaf people.

Unlike Neural Machine Translation (Bengio et al., 2000), which focuses on translating between different languages (e.g., English to Chinese), Sign Language Translation (SLT) is a cross-modal task that involves learning visual representations from sign language videos and generating corresponding spoken words. Previous SLT approaches (Camgoz et al., 2020; Zhou et al., 2021a; Chen et al., 2022a,b) have relied on gloss sequences to improve performance. Gloss refers to the transcription of signed languages sign-by-sign, where every sign has a unique identifier (Yin et al., 2021). Gloss sequences are utilized as the supervision for visual representation learning via performing Continuous Sign Language Recognition (CSLR). Due to the substantial manual labor and specialized linguistic expertise required for gloss annotation, it is challenging to construct large-scale datasets. Therefore, the existing datasets are relatively small in scale and domain-specific. Consequently, current gloss-based methods, while achieving good performance on certain specific test sets, suffer from limited generalizability and are unable to benefit from large, high-quality datasets without gloss an-

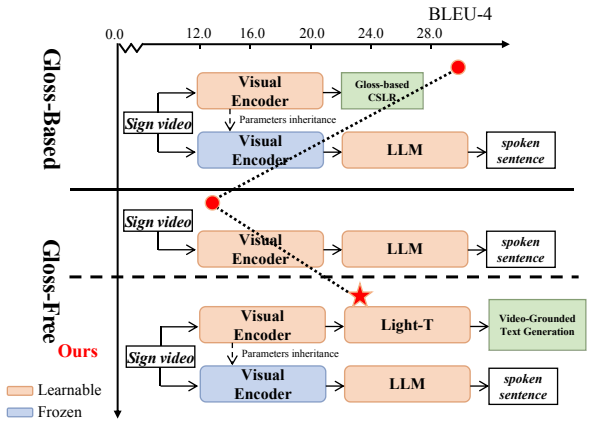

Figure 1: Different frameworks and performance of gloss-based and gloss-free methods with LLM. The first row shows the performance of the gloss-based method with LLM (Chen et al., 2022a). The second and third show the gloss-free method with LLM in our experiments. The BLEU-4 score is gotten on the PHOENIX-2014T test set.

[†]Corresponding author.

LREC-COLING 2024, pages 7071–7081

notations, such as the newly released large-scale SLT datasets like How2Sign (Duarte et al., 2021) and OpenASL (Shi et al., 2022).

In summary, the exploration of gloss-free methods is highly necessary, as it can significantly reduce annotation costs and contribute to the development of more reliable and general sign language translation systems.

Recently, there have been attempts (Camgoz et al., 2018; Li et al., 2020; Lin et al., 2023) to achieve gloss-free sign language translation by jointly training a visual network and translation network in an end-to-end manner. To enhance the performance of SLT, the intuitive approach is to employ larger and more powerful pre-trained large language models (LLM) like (Chen et al., 2022a) (first row in Figure 1). However, jointly training the visual encoder and LLM end-to-end without gloss annotations leads to significant performance degradation, as shown in the second row in Figure 1.

We conjecture it is due to the fact that: 1) the visual encoder was not pre-trained on the sign language dataset leads to poor modeling ability of temporal and spatial information in them. 2) LLM dominated the training process resulting in insufficient learning of visual representations. We provided a detailed analysis of the specific experiments conducted in Section 3. This phenomenon has made prior methods use small language models with random initialization resulting in poor performance. To cope with the above challenges and build a more realistic SLT system, we propose a **F**actorized **L**earning **a**ssisted with **L**arge **L**anguage **M**odel (termed **FLa-LLM**) for gloss-free SLT.

Specifically, as illustrated in the third row of Figure 1, we factorize the training process into two distinct stages, visual initialing stage and LLM fine-tuning stage. In the first stage, we introduce a lightweight translation model (Light-T) positioned after the visual encoder to pre-train the visual encoder using a video-grounded text generation task. This strategy can be seen as a soft visual-text alignment method that implicitly supervises the visual encoder with the assistance of a lightweight language translation model, enabling it to acquire language knowledge. While this stage compels the visual encoder to learn semantic visual representations, it may not yield satisfactory translation performance due to the limited strength of the lightweight translation model. To overcome this limitation, we introduce the Large Language Model (LLM) into the second stage namely LLM fine-tuning stage. we incorporate an LLM that has been pre-trained on extensive corpora using an unsupervised approach to enhance the translation performance. The parameters of the pre-trained visual encoder are all frozen to overcome its risk of being biased by the LLM. Finally, we successfully took advantage of LLM in gloss-free SLT and got a BLEU-4 score of 23.09.

In summary, our work makes the following significant contributions:

- We analyze the reason why directly training the visual encoder and LLM failed in gloss-free SLT and propose **FLa-LLM** to overcome this problem. To the best of our knowledge, this is the first successful attempt of LLM on gloss-free SLT.
- **FLa-LLM** method factorizes the training process into two distinct stages namely the visual initialing stage and LLM fine-tuning stage. This division helps mitigate the detrimental effects of the Large Language Model (LLM) on visual representation learning. Moreover, it allows us to leverage the LLM's assistance in SLT at a low cost, improving translation performance.
- Our approach greatly boosts the performance of the gloss-free SLT. Specifically, we improve the BLEU-4 score by a large margin of 1.65 on PHOENIX14T(Camgoz et al., 2018), 3.20 on CSL-Daily(Zhou et al., 2021a) and 1.63 on How2Sign(Duarte et al., 2021) compared with the previous state-of-the-art methods.

2. Related Work

Gloss-based SLT. Sign Language Translation (SLT) is proposed by (Camgoz et al., 2018) which intends to translate sign language videos into corresponding spoken sentences. Most SLT methods utilize gloss annotations for pre-training or as assisted supervision which we define as gloss-based SLT. Camgoz et al. (2020) used the transformer (Vaswani et al., 2017) architecture and jointly trained Continuous Sign Language Recognition (CSLR) and SLT. Zhou et al. (2021a) utilized gloss as an intermediary and translated spoken sentences back into sign language features to expand the translation corpus. Chen et al. (2022a) transferred a powerful large language model to the sign language domain and improved the performance of SLT significantly. Considering the special structure of sign language, Zhou et al. (2021b); Chen et al. (2022b) used a multi-cue network for detailed sign language modeling. Gloss-based SLT methods can achieve better performance, but the difficulty of labeling gloss leads to great limitations.

Gloss-free SLT. There are a few methods attempting to build a more realistic SLT system with the gloss-free setting. Camgoz et al. (2018) built an attention-based encoder-decoder framework for SLT. Li et al. (2020) learned hierarchical features of sign language via temporal semantic pyramid. Zhao et al. (2021) improved the accuracy and fluency of SLT by conditional sentence generation

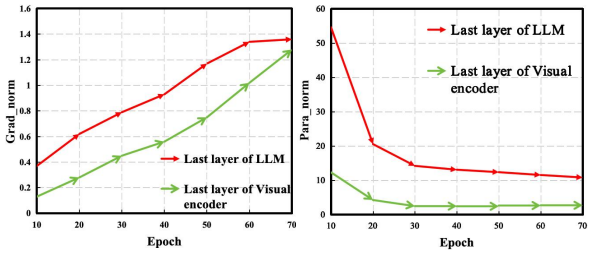

Figure 2: The grad norm and parameters norm of the last layer of the visual encoder and the last layer of the LLM when jointly training the visual encoder and LLM end-to-end.

and cross-modal reranking. Orbay and Akarun (2020) utilized adversarial, multi-task, and transfer learning to search for semi-supervised tokenization approaches. Yin et al. (2023) analyzed the role of gloss in SLT. Based on it, they proposed gloss attention which enables the model to keep its attention within video segments that have the same semantics locally as the gloss-based model. Lin et al. (2023) exploited the shared underlying semantics of signs and the corresponding spoken translation to improve the gloss-free SLT performance. Zhou et al. (2023) integrate contrastive language image pre-training with masked self-supervised learning to create pre-tasks that bridge the semantic gap between visual and textual representations and restore masked sentences. Due to the lack of the assistance of gloss, the above gloss-free methods basically did not use LLM resulting in poor performance of language generation. In addition, the huge computational cost caused by the large number of sign language video frames makes it difficult to perform end-to-end fine-tuning using LLM.

3. The Dominance of LLM in SLT

In this section, we analyze the idea presented in Section 1 that LLM dominates the SLT training process when jointly training the visual encoder and LLM end-to-end.

The grad norm represents the rate of change of the parameter while the parameter norm represents the magnitude of the change. They can reflect which part of the training process is more active. We chose the last layer of LLM to represent the LLM module and the last layer of the visual encoder to represent the visual encoder. As shown in Figure 2, we visualized the grad norm and parameters norm of these two layers when jointly training the visual encoder and LLM end-to-end. The grad norm of the last layer of LLM was always greater than the last layer of the visual encoder. At the same time, the parameter norm of the last layer of LLM changed more drastically than the last layer of the visual encoder. It indicates that the main update of the model lies in the LLM module i.e. LLM dominates the SLT training process. It will lead to suppression of visual encoder training and thus failure to learn good visual representations of sign language. The results of Table 4 experiments similarly prove this statement.

4. Method

4.1. Overview

SLT aims to translate a sign video $V = (I_1, I_2, ...I_T)$ with T frames into the corresponding spoken sentence $S = (w_1, w_2, ...w_U)$ with U words. In this work, we focus on a gloss-free solution that doesn't require gloss annotations. As illustrated in Figure 3, the training process is factorized into two stages. In the visual initialing stage (Section 4.2), the objective is to facilitate the learning of semantic visual knowledge by the visual encoder from downsampled videos. The visual features are then mapped into a textual embedding space using a Visual-Language Adapter (VL-Adapter). We construct a lightweight translation model (Light-T) to perform video-grounded text generation pre-training. Subsequently, in the LLM fine-tuning stage (Section 4.3), we retain the pre-trained visual encoder from the visual initialing stage to extract sign-wise features of input videos. Then the features are passed into an LLM-Adapter and LLM to generate corresponding spoken sentences.

4.2. Visual Initialing

Since sign language possesses special spatial properties, the visual initialing of the visual encoder on sign language datasets is necessary. With the gloss-free setting, only the spoken sentences can be used as text supervision. Therefore, we construct a visual encoder followed by a lightweight translation model (Light-T) to perform visual initialing by a video-grounded text generation task.

Video Downsampling. The number of frames in a sign language video is greater than the number of words in the corresponding sentence, which has a lot of redundant information. Therefore, we downsample a $(T \times 3 \times H \times W)$ input sign language video into $(T/4 \times 3 \times H \times W)$ to reduce computational cost without performance degradation.

Visual Encoder. The visual encoder consists of a vision backbone and a local temporal module. ResNet18 (He et al., 2016) is chosen as our vision backbone. The downsampled sign video is fed into the visual backbone frame by frame to get frame-wise features. A complete sign language token is often expressed by several continuous frames. Therefore, we designed a temporal module to capture the local timing information within the sign language video. The temporal module consists of one

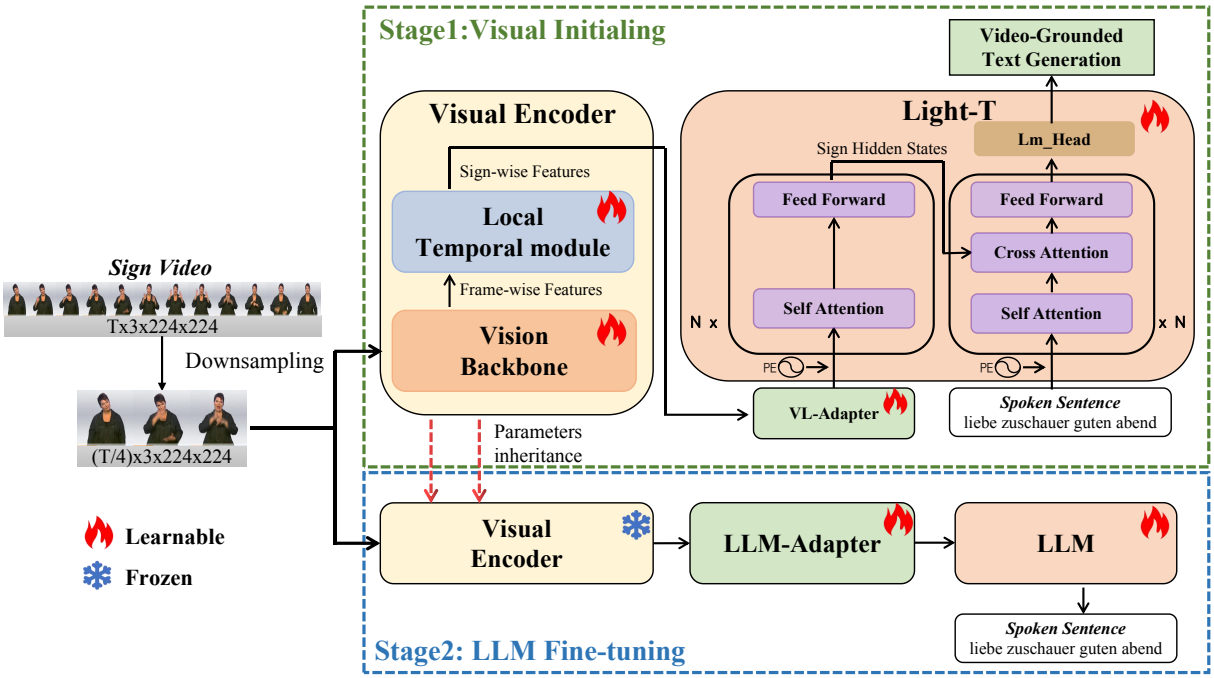

Figure 3: The overall framework of our proposed method. LLM represents the large language model.

temporal convolutional layer, one batch normalization layer, and one Relu layer. The frame-wise features are passed through the temporal module to get sign-wise features $F = (f_1, f_2, ...f_N)$ where $N = T/4$ with a size of $(T/4 \times C)$. The above operation can be formulated as:

$$f_{1:N} = \text{VisualEncoder}(I_{1:T}). \quad (1)$$

VL-Adapter. After the visual encoder, we build a VL-Adapter using an MLP with one hidden layer. The sign-wise features from visual space $R^{N\times C}$ are mapped into textual space $R^{N\times D}$ by the VL-Adapter as follows:

$$g_{1:N} = \text{VL-Adapter}(f_{1:N}). \quad (2)$$

Light-T. We pick transformer (Vaswani et al., 2017) as our lightweight translation model. It contains a text encoder and a text decoder which are composed of several transformer layers. The input features are added with a positional encoding (PE) as $\hat{g}_n = g_n + PE(n)$. The text encoder models the global timing information of the input as follows:

$$h_{1:N} = \text{TextEncoder}(\hat{g}_{1:N}). \quad (3)$$

Meanwhile, the corresponding spoken sentence $S = (w_1, w_2, ...w_U)$ is tokenized into $S' = (o_1, o_2, ...o_L)$ by the same tokenizer from the LLM we will use next. Then it is passed through the word embedding layer (WEL) and a positional encoding (PE) layer as:

$$z_i = \text{WEL}(o_i) + \text{PE}(i). \quad (4)$$

The text decoder takes word embeddings alone with sign hidden states $h_{1:N}$ as input to generate a predicted sentence one word at a time:

$$y_i = \text{TextDecoder}(z_{1:i-1}, h_{1:N}). \quad (5)$$

A language modeling head (Lm_Head) is plugged after the text decoder to calculate the conditional probabilities as follows:

$$p(o_i|o_{1:i-1}, V) = (\text{softmax}(\text{Lm\_Head}(y_i)))_{o_i}. \quad (6)$$

Training. We train the model using the video-grounded text generation objective, which aims to generate spoken sentences corresponding to the input videos. We use the ground truth spoken sentences to calculate the cross-entropy loss and optimize the entire network:

$$\mathcal{L}_{CE} = -\sum_{i=1}^{L} \log p(o_i|o_{1:i-1}, V). \quad (7)$$

After the above workflow, we finished initializing the visual encoder on the sign language datasets. The well-initialized visual encoder is now capable of extracting text-oriented features from sign videos. Next, we will take advantage of LLM to generate more approximate and fluent spoken sentences.

4.3. LLM Fine-tuning

Now we present how the proposed method can exploit the potential of LLM in Gloss-free SLT. In general, we keep the pre-trained visual encoder and plug it into an LLM-Adapter and an LLM. During

training, the visual encoder is frozen while the other modules are fine-tuned.

LLM Selection. The LLM selection follows two standards. Firstly, the selected LLM should be an encoder-decoder architecture because SLT is a translation-type downstream task. Secondly, LLM pre-trained on multilingual corpus is preferred because different datasets have spoken sentences in various languages, such as German in PHOENIX14T (Camgoz et al., 2018) and Chinese in CSL-Daily (Zhou et al., 2021a). Under these two criteria, we choose MBart (Liu et al., 2020) as our LLM. MBart is a sequence-to-sequence denoising auto-encoder pre-trained on large-scale monolingual corpora in many languages. It has a standard transformer (Vaswani et al., 2017) architecture with 12 layers of the encoder and 12 layers of the decoder. MBart primarily intends for translation tasks and has been proven to significantly improve the performance of SLT (Chen et al., 2022a). Although the number of parameters in MBart is only 680M, to the best of our knowledge, it is the largest language model in the SLT domain.

Fine-tuning.

The word embedding layer of the MBart's encoder is replaced by an LLM-Adapter. It is simply implemented as a fully connected MLP with one hidden layer whose output dimension fits with the LLM. We use the well-initialized visual encoder to extract the sign-wise features of the input sign videos and feed them into the LLM-Adapter to generate sign embeddings. Then the MBart's encoder takes sign embeddings as input to generate hidden states. The corresponding spoken sentences are tokenized by a tokenizer and project one-hot vectors into dense text embeddings via MBart's pre-trained word embedding layer. MBart's decoder takes the hidden states alone with the text embeddings to generate predicted sentences one word at a time. During fine-tuning, we freeze the visual encoder and optimize other modules using sequence-to-sequence cross-entropy loss as shown in Equation 7.

5. Experiments

5.1. Datasets and Evaluation Metrics

Datasets. The experiments are performed on PHOENIX14T (Camgoz et al., 2018), CSL-Daily (Zhou et al., 2021a) and How2Sign (Duarte et al., 2021). PHOENIX14T is a German Sign Language (DGS) dataset taken from a TV broadcast whose topic focuses on weather forecasts. It contains 7096, 519, and 642 video-gloss-text pairs in train, dev, and test set, respectively. CSL-Daily is a Chinese Sign Language (CSL) dataset that contains 18401, 1077, and 1176 video-gloss-text pairs in train, dev, and test set, respectively. It is recorded in the laboratory whose topic focuses on daily life. How2Sign is an American Sign Language (ASL) dataset that contains 31164, 1740, and 2356 video-text pairs in train, dev, and test set, respectively. It is recorded in the laboratory and focuses on instructional topics corresponding to various categories. The proposed method is compared with state-of-the-art methods on three datasets and conducted ablation analysis on PHOENIX14T. We report all the results on the test set.

Protocol. Our experiments follow *Gloss-free Sign2Text* protocol proposed by (Lin et al., 2023). It requests a direct translation from sign language videos to the corresponding spoken sentences without gloss assistance through the entire framework.

Evaluation Metrics. Following (Zhou et al., 2021a; Chen et al., 2022b; Lin et al., 2023; Yin et al., 2023), we adopt ROUGE-L (Lin, 2004) and BLEU (Papineni et al., 2002) to evaluate SLT performance.

5.2. Implementation Details

Our model is implemented using the Pytorch framework (Paszke et al., 2019). The experiments are conducted on NVIDIA GeForce RTX 3090 GPUs.

Network setting. We choose ResNet18 (He et al., 2016) pretrained on ImageNet (Deng et al., 2009) as visual backbone. The local temporal module uses a combination of Conv1D-BN-Relu layers. The Light-T has 3 transformer layers for both encoder and decoder. Each layer has an attention head of 8, a hidden size of 512, and a feed-forward dim of 2048. Our LLM is initialized with the official release of MBart-large-cc25. The model and corresponding tokenizers are trimmed using the translation corpus of the target SLT train set to save GPU memory.

Training and Inference. In the visual initialing stage, the model is trained using SGD optimizer (Robbins and Monro, 1951) with 0.9 momentum and a batch size of 8 across 2 GPUs. The learning rate is set to 1×10^{-2} with a cosine annealing schedule (Loshchilov and Hutter, 2016). In the LLM fine-tuning stage, the model is trained using Adam optimizer (Kingma and Ba, 2014) with a batch size of 16 on 1 GPU. The learning rate is set to 1×10^{-5} for the LLM and 1×10^{-3} for the LLM-Adapter layer with a cosine annealing schedule. We employ cross-entropy loss with a label smoothing of 0.2 in both stages. During inference, we use beam search strategy (Wu et al., 2016) with a beam size of 5.

https://huggingface.co/facebook/MBart-large-cc25

Method	Gloss-Free	Rouge-L	BLEU-1	BLEU-2	BLEU-3	BLEU-4
SLRT (Camgoz et al., 2020)	×	-	46.61	33.73	26.19	21.32
STMC-T (Zhou et al., 2021b)	×	46.65	46.98	36.09	28.70	23.65
SignBT (Zhou et al., 2021a)	×	49.54	50.80	37.75	29.72	24.32
MMTLB (Chen et al., 2022a)	×	52.65	53.97	41.75	33.84	28.39
TS-SLT (Chen et al., 2022b)	×	**53.48**	**54.90**	**42.43**	**34.46**	**28.95**
NSLT (Camgoz et al., 2018)	✓	31.80	32.24	19.03	12.83	9.58
SLRT-GF* (Camgoz et al., 2020)	✓	31.10	30.88	18.57	13.12	10.19
TK-SLT (Orbay and Akarun, 2020)	✓	36.28	37.22	23.88	17.08	13.25
TSPNet (Li et al., 2020)	✓	34.96	36.10	23.12	16.88	13.41
CSGCR (Zhao et al., 2021)	✓	38.85	36.71	25.40	18.86	15.18
GASLT (Yin et al., 2023)	✓	39.86	39.07	26.74	21.86	15.74
GFSLT-VLP (Zhou et al., 2023)	✓	42.49	43.71	33.18	26.11	21.44
FLa-LLM(ours)	✓	**45.27**	**46.29**	**35.33**	**28.03**	**23.09**
Improvement		+2.78	+2.58	+2.15	+1.92	+1.65

Table 1: Experimental results on PHOENIX14T dataset. * denotes methods reproduced by (Yin et al., 2023). We bold the best results in the gloss-based setting and gloss-free setting. Improvement represents comparisons with the previous best gloss-free result.

Method	Gloss-Free	Rouge-L	BLEU-1	BLEU-2	BLEU-3	BLEU-4
SLRT† (Camgoz et al., 2020)	×	36.74	37.38	24.36	16.55	11.79
SignBT (Zhou et al., 2021a)	×	49.31	51.42	37.26	27.76	21.34
MMTLB (Chen et al., 2022a)	×	53.25	53.31	40.41	30.87	23.92
TS-SLT (Chen et al., 2022b)	×	**55.72**	**55.44**	**42.59**	**32.87**	**25.79**
NSLT† (Camgoz et al., 2018)	✓	34.54	34.16	19.57	11.84	7.56
TSPNet* (Li et al., 2020)	✓	18.38	17.09	8.98	5.07	2.97
GASLT (Yin et al., 2023)	✓	20.35	19.90	9.94	5.98	4.07
GFSLT-VLP (Zhou et al., 2023)	✓	36.44	**39.37**	24.93	16.26	11.00
FLa-LLM(ours)	✓	**37.25**	37.13	**25.12**	**18.38**	**14.20**
Improvement		+0.81	-2.24	+0.19	+2.12	+3.20

Table 2: Experimental results on CSL-daily dataset. * denotes methods reproduced by (Yin et al., 2023). † denotes methods reproduced by (Zhou et al., 2021a). We bold the highest scores in the gloss-based setting and gloss-free setting. Improvement represents comparisons with the previous best gloss-free result.

5.3. Comparison with State-of-the-art Methods

Results on PHOENIX14T dataset. We compare our method with state-of-the-art gloss-based and gloss-free SLT approaches in Table 1. With the gloss-free setting, our method achieves a significant breakthrough in all metrics compared to the previous methods. In particular, we get an outstanding BLEU-4 improvement of 1.65 on the test set compared with the previous state-of-the-art method GFSLT-VLP (Zhou et al., 2023). Surprisingly, our approach is fairly comparable to some gloss-based approaches, such as SLRT (Camgoz et al., 2020), STMC-T (Zhou et al., 2021b) and SignBT (Zhou et al., 2021a). The performance of our method is still far from MMTLB (Chen et al., 2022a) and TS-SLT (Chen et al., 2022b), which also utilizes the LLM capability to enhance SLT. However, they rely heavily on gloss for visual and linguistic pre-training with great limitations.

Results on CSL-Daily dataset. Table 2 shows the comparisons between our method and other state-of-the-art methods on the CSL-Daily dataset. When compared with other gloss-free methods, our method achieves a substantial improvement with a margin of 3.20 in BLEU-4 which is 29.09% higher than the previous state-of-the-art method GFSLT-VLP (Zhou et al., 2023). However, there is a big gap between our method and the gloss-based methods. This may be due to the size of the sign word's vocabulary. CSL-daily has more than 2K sign words' vocabulary size resulting in more reliance on glosses.

Results on How2Sign dataset. In Table 3, our method is compared with other state-of-the-art methods on the How2Sign dataset. The performance of our method is substantially better than TF-H2S (Alvarez et al.) and GloFE-VN (Lin et al., 2023). However, we only surpass SLT-IV (Tarrés et al., 2023) on BLEU-3 and BLEU-4 while falling behind on BLEU-1 and BLEU-2. Higher BLEU-3

Method	Gloss-Free	Rouge-L	BLEU-1	BLEU-2	BLEU-3	BLEU-4
TF-H2S (Alvarez et al.)	✓	-	17.40	7.69	3.97	2.21
SLT-IV (Tarrés et al., 2023)	✓	-	**34.01**	**19.30**	12.18	8.03
GloFE-VN (Lin et al., 2023)	✓	12.61	14.94	7.27	3.93	2.24
FLa-LLM(ours)	✓	**27.81**	29.81	18.99	**13.27**	**9.66**
Improvement		+15.20	-4.20	-0.31	+1.09	+1.63

Table 3: Experimental results on How2Sign dataset. We bold the highest scores. Improvement represents comparisons with the previous best gloss-free result.

Factorized	R	B1	B2	B3	B4
×	32.52	31.96	21.96	16.32	12.90
✓	**45.27**	**46.29**	**35.33**	**28.03**	**23.09**

Table 4: Effect of the proposed factorized learning strategy. The first row represents end-to-end joint training of the visual encoder and LLM.

VIS	LFS	R	B1	B2	B3	B4
×	✓	17.33	17.64	10.51	7.37	5.62
✓	×	38.67	39.09	28.20	21.83	17.69
✓	✓	**45.27**	**46.29**	**35.33**	**28.03**	**23.09**

Table 5: Effect of each stage. VIS represents the visual initialing stage and LFS means the LLM fine-tuning stage. The first row represents freezing the vision backbone and fine-tuning the other modules.

and BLEU-4 indicate our model has better short phrase generating ability which possibly gains from LLM.

5.4. Ablation Studies

Our ablation experiments are performed on the PHOENIX14T test set since it is the most widely used benchmark for SLT. Note that we use R to represent ROUGE-L and B1-B4 to represent BLEU1-BLEU4.

5.4.1. Ablation on Factorized Learning

Effect of Factorized Learning Strategy. We first verify the effectiveness of our proposed factorized learning strategy. The most straightforward approach is to compare our factorized learning with the end-to-end joint training of the visual encoder and LLM. As shown in Table 4, our factorized learning strategy substantially outperforms the end-to-end training approach. This may be because the LLM dominates during end-to-end training as mentioned in Section 3, resulting in weak supervision for the visual encoder.

Effect of Each Stage. In Table 5, we verify the contribution of each stage in the proposed FLa-LLM method. The first row of the Table 5 means we freeze the vision backbone which is only pre-trained on ImageNet and trained the local temporal module with the LLM-Adapter and LLM end-to-end. It leads to a very poor result without initialing the visual encoder which demonstrates the importance of visual initialing on the sign language datasets. The second row shows the performance of the visual encoder and Light-T i.e. the visual initialing stage performance. The visual initialing stage focuses on the visual encoder resulting in fair performance. Based on sufficient initialization of the visual encoder, we successfully take advantage of the LLM and yield better results as shown in the third row.

Rate	Time	R	B1	B2	B3	B4
100%	17.90h	44.55	45.68	35.01	27.82	22.96
50%	9.85h	44.60	46.22	35.12	27.90	23.04
25%	4.75h	**45.27**	**46.29**	**35.33**	**28.03**	**23.09**
12.5%	3.55h	40.77	42.42	31.62	24.68	20.02

Table 6: Effect of downsampling rate. The second column represents the time required to complete the visual initialing stage.

5.4.2. Ablation on Visual Initializing

Effect of Downsampling Rate. We show the effect of different downsampling rates on the training time and model performance in Table 6. When the downsampling rate is not lower than 25%, it has little impact on the model performance while significantly reducing the training time. Therefore, we choose a sampling rate of 25% to ensure model performance and save training time.

Effect of Light-T. We investigate whether the scale of the Light-T in the visual initialing stage affects the final LLM fine-tuning results. As shown in Table 7, the transformer scale has little impact on final performance. This indicates that the main focus during the visual initialing stage is on the visual encoder. The visual encoder can get a good sign language representation ability after initialing regardless of the transformer scale connected.

Effect of Initialing Time. We train various visual encoders with different epochs in the initialing stage and use them to do LLM fine-tuning in the same setting. Figure 4 shows the results of the visual initialing stage and the LLM fine-tuning stage with different initialing epochs. LLM fine-tuning can sig-

Size	Settings	Params	B4
Tiny	(1,4,256,1024)	3.61M	22.52
Small	(2,4,512,2048)	18.25M	22.36
Base	(3,8,512,2048)	25.61M	**23.09**
Large	(4,8,1024,4096)	124.66M	22.49

Table 7: Effect of translation network scale. The (1,4,256,1024) in the second column represents that the transformer has 1 hidden layer, 4 attention heads, a hidden size of 256, and a feed-forward dim of 2048. Params represents the number of model parameters.

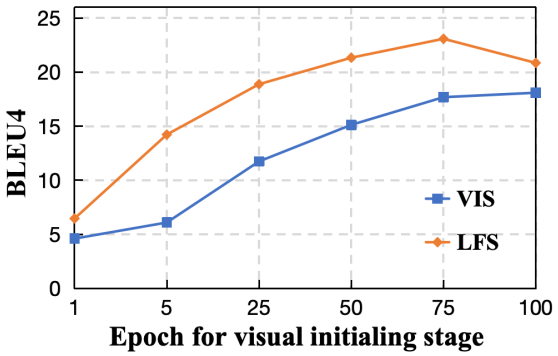

Figure 4: Effect of initialing time. VIS represents the visual initialing stage and LFS represents the LLM fine-tuning stage.

nificantly improve the model translation capability in all cases. When the initialing epoch is 100, the performance of LLM fine-tuning decreases compared to epoch 75, probably due to model overfitting. Therefore, we choose the model with epoch 75 for fine-tuning in other experiments.

5.4.3. Ablation on LLM Fine-tuning

Effect of Input Features. In Table 8, we investigate the effect of different input features of the LLM-Adapter layer. We select three features as shown in Figure 3, namely frame-wise features, sign-wise features, and sign hidden states. The best result is obtained by using sign-wise features as input for LLM fine-tuning. This may be due to the fact that sign-wise features contain both spatial representations and local timing information of sign language videos.

Effect of Frozen Blocks. The visual encoder with valid initialization is frozen during the LLM fine-tuning stage. We examined the effect of freezing different parts of the visual encoder in Table 9. It can be seen that fine-tuning the visual encoder during the LLM fine-tuning stage hurt the performance substantially. This may be due to that LLM dominates the training process and disrupts the initialization of the visual encoder. It also indicates

Features	R	B1	B2	B3	B4
Frame-wise	41.59	42.54	31.72	24.82	20.30
Sign-wise	**45.27**	**46.29**	**35.33**	**28.03**	**23.09**
Hidden states	45.16	45.41	34.74	27.47	22.60

Table 8: Effect of different input features of LLM. The different features are shown in Figure 3.

VB	TM	R	B1	B2	B3	B4
×	×	40.72	40.74	29.79	22.65	17.86
✓	×	39.86	41.64	31.48	24.86	20.40
✓	✓	**45.27**	**46.29**	**35.33**	**28.03**	**23.09**

Table 9: Effect of freezing different parts of the visual encoder. VB means visual backbone. TM represents the local temporal module. ✓ means freezing the module while × means no freezing.

LLM	R	B1	B2	B3	B4
MBart w/o pre	40.18	37.18	26.99	20.54	16.19
MT5-Base w/o pre	22.71	18.02	12.21	9.17	7.39
MBart w/ pre	**45.27**	**46.29**	**35.33**	**28.03**	**23.09**
MT5-Base w/ pre	41.06	41.96	31.20	24.24	19.71

Table 10: Effect of different LLMs. W/o, w/, and pre means without, with, and pretraining, respectively.

that the visual encoder already has a sufficient visual representation of sign language after the visual initializing stage.

Effect of Different LLMs. In order to verify the robustness of our training strategy and to select the most suitable LLM for sign language, we perform the fine-tuning stage using different LLMs. We select two popular multilingual unsupervised pre-training models which are MBart (Liu et al., 2020) and MT5-Base (Xue et al., 2021). Table 10 shows the fine-tuning results of these two LLMs with random initialization and after pre-training. It can be seen that unsupervised pre-training on a large-scale corpus can significantly improve the performance of LLM fine-tuning on sign language datasets. In addition, MBart performs better compared to MT5-Base, so we chose MBart as our default LLM.

6. Conclusion

In this paper, we propose a factorized learning strategy to transfer LLM for gloss-free SLT. In the visual initialing stage, we use a lightweight translation model to pre-train the visual encoder without gloss supervision. In the LLM fine-tuning stage, we freeze the well-initialized visual encoder and fine-tune a powerful LLM to adapt to the downstream SLT task. By splitting the training into two stages, we avoid performance degradation and utilize LLM in a resource-friendly situation. Our method significantly boosts the performance of gloss-free SLT on several datasets.

7. Limitations

Our proposed method has two main drawbacks. First, though the factorized learning strategy can avoid performance degradation, it is more cumbersome compared to end-to-end training. A more ideal approach would be adding additional constraints on the visual encoder to the end-to-end framework. Second, in the fine-tuning stage, we fine-tune all parameters of the large language model which limits the scale of our LLM. In future work, we will investigate parameter-efficient fine-tuning methods such as Lora (Hu et al., 2022) and Prefix-Turning (Li and Liang, 2021).

8. Acknowledgements

This work was supported by the National Key Research and Development Program of China under Grant 2021YFE0205700, Beijing Natural Science Foundation JQ23016, the External cooperation key project of Chinese Academy Sciences 173211KYSB20200002, the Science and Technology Development Fund of Macau Project 0123/2022/A3, and 0070/2020/AMJ, Open Research Projects of Zhejiang Lab No. 2021KH0AB07, and CCF-Zhipu AI Large Model Project 202219.

9. Bibliographical References

Patricia Cabot Alvarez, Xavier Giró Nieto, and Laia Tarrés Benet. Sign language translation based on transformers for the how2sign dataset.

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2000. A neural probabilistic language model. *Advances in neural information processing systems*, 13.

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

Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. 2020. Sign language transformers: Joint end-to-end sign language recognition and translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10023–10033.

Yutong Chen, Fangyun Wei, Xiao Sun, Zhirong Wu, and Stephen Lin. 2022a. A simple multi-modality transfer learning baseline for sign language translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5120–5130.

Yutong Chen, Ronglai Zuo, Fangyun Wei, Yu Wu, Shujie LIU, and Brian Mak. 2022b. Two-stream network for sign language recognition and translation. In *Advances in Neural Information Processing Systems*.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

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.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

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

Dongxu Li, Chenchen Xu, Xin Yu, Kaihao Zhang, Benjamin Swift, Hanna Suominen, and Hongdong Li. 2020. Tspnet: Hierarchical feature learning via temporal semantic pyramid for sign language translation. *Advances in Neural Information Processing Systems*, 33:12034–12045.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597.

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

Kezhou Lin, Xiaohan Wang, Linchao Zhu, Ke Sun, Bang Zhang, and Yi Yang. 2023. Gloss-free end-to-end sign language translation. *arXiv preprint arXiv:2305.12876*.

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*, 8:726–742.

Ilya Loshchilov and Frank Hutter. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

Alptekin Orbay and Lale Akarun. 2020. Neural sign language translation by learning tokenization. In *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pages 222–228. IEEE.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Herbert Robbins and Sutton Monro. 1951. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

Bowen Shi, Diane Brentari, Greg Shakhnarovich, and Karen Livescu. 2022. Open-domain sign language translation learned from online video. In *EMNLP*.

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*, pages 5624–5634.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

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*, pages 2551–2562.

Kayo Yin, Amit Moryossef, Julie Hochgesang, Yoav Goldberg, and Malihe Alikhani. 2021. Including signed languages in natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7347–7360.

Jian Zhao, Weizhen Qi, Wengang Zhou, Nan Duan, Ming Zhou, and Houqiang Li. 2021. Conditional sentence generation and cross-modal reranking for sign language translation. *IEEE Transactions on Multimedia*, 24:2662–2672.

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 (ICCV)*, pages 20871–20881.

Hao Zhou, Wengang Zhou, Weizhen Qi, Junfu Pu, and Houqiang Li. 2021a. Improving sign language translation with monolingual data by sign back-translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1316–1325.

Hao Zhou, Wengang Zhou, Yun Zhou, and Houqiang Li. 2021b. Spatial-temporal multi-cue network for sign language recognition and translation. *IEEE Transactions on Multimedia*, 24:768–779.

10. Language Resource References

Camgoz, Necati Cihan and Hadfield, Simon and Koller, Oscar and Ney, Hermann and Bowden, Richard. 2018. *RWTH-PHOENIX-2014-T*. PID https://www-i6.informatik.rwth-aachen.de/ koller/RWTH-PHOENIX-2014-T/.

Duarte, Amanda and Palaskar, Shruti and Ventura, Lucas and Ghadiyaram, Deepti and DeHaan, Kenneth and Metze, Florian and Torres, Jordi and Giro-i-Nieto, Xavier. 2021. *How2sign*. PID https://how2sign.github.io/.

Zhou, Hao and Zhou, Wengang and Qi, Weizhen and Pu, Junfu and Li, Houqiang. 2021. *CSL-Daily*. PID https://ustc-slr.github.io/datasets/2021_csl_daily/.