1. Overview of Supplementary Materials

Below we provide more details, experimental results, and discussion. More details are in the https://signllm.github.io project page.

2. Background Information

Here we expand on some of the nouns mentioned briefly:

Gloss: In the context of sign language, gloss refers to the process of providing a word-for-word translation of sign language into written or spoken language. It involves assigning a specific written or spoken word to each sign in order to facilitate communication and understanding between sign-language users and non-sign-language users. It generally represents a specific gesture or posture.

OpenPose: OpenPose[1] is a real-time multi-person keypoint detection library that uses computer vision techniques to identify and track human body movements. The output result is a video of the key point visualization and key point data stored in json format for 24 frames a second.

DensePose: DensePose[2] is a method that estimates dense correspondences between a 2D image and a 3D human model. It can be used to extract detailed information about the body posture, position, and movements of sign language users from 2D images or videos, stored or displayed as a dense map covering the entire body of a human being. Details can be found in the footnote links.

[1] https://github.com/CMU-Perceptual-Computing-Lab/openpose
[2] https://github.com/facebookresearch/detectron2/tree/DensePose

2.1. More Related Work

Here, we introduce the third step of sign language production: Pose2Video, which involves visualizing key points in a video rendering or converting it into a live person/model demonstration of sign language. We also give some basic concepts of RL for a better understanding.

Rendering of Conditional Input. Conditioning refers to the capacity of a generative model to manipulate its output based on our intentions. Previous instances of conditional input Generative Adversarial Networks (GANs) [27] have exhibited favorable performance in generating images [39, 64, 95, 108] and videos [54, 90, 93, 94, 96]. Numerous studies have also focused on generating human poses while considering various factors, including entire body [1, 11, 53, 56, 79, 85, 109], face [18, 47, 88, 102, 103, 106], and hand [50, 84, 98]. One particular application is human-style transfer [69], which involves replacing a person in a video with another individual while preserving their actions. This technique has also found extensive use in sign language production [11, 97, 107]. The key aspect lies in extracting keypoints to replicate movements [11, 91], utilizing tools such as OpenPose, i3D, and DensePose for common keypoint extraction [11, 62, 97, 107]. In our work, we do not care about Pose2video, we only present some qualitative results at the end of the paper and in the supplementary materials.

Reinforcement Learning. in the training or fine-tuning of large models is a common strategy. At the heart of reinforcement learning is the concept of a Markov Decision Process (MDP), an extension of Markov chains, which involves a finite set of states, a finite set of actions, state transition probabilities, and a reward function. The MDP delineates the interaction between an intelligent agent and the environment, wherein the agent chooses actions based on various states, and the environment imposes rewards or penalties on the agent based on the action and the current state, leading to a transition to the next state. An optimal policy is the mapping from state s to action a that maximizes the total expected return:

$$\pi^* = \arg\max_{\pi} \mathbb{E}[G_t | s_t = s, \pi] \tag{4}$$

where $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$, $0 \leq \gamma < 1$ is the discount factor, and $\mathbb{E}[]$ is the expectation operator. In LLMs, researchers often fine-tune models with reinforcement learning based on human feedback. Given that the SLP process aligns with the definition and can be reformed by the MDP, we simply simulate this concept to fine-tune our generation model. However, since the training scenario of sign language does not involve interaction with the environment, our reinforcement learning strategy is not a typical one, but rather only partially applied to component modules.

Prompt Template & Some Examples

Part I

I really want to learn how to say '{Text}' in sign language. Can you help me?
How would you express '{Text}' in sign language?
Can you show me how to say '{Text}' in sign language?
How do I say '{Text}' in sign language?
I am interested in mastering the sign language for '{Text}'.
What's the method to sign '{Text}'?
Can you show me how '{Text}' appears in sign language?
Could you tell me how '{Text}' is represented in sign language?

Part II

How is "So we're going to go up and down; let's switch hands, down and up; down and up." denoted in sign language?
Can you elucidate how And just let those fingers relax. looks in sign language?
Can you elucidate how 'You do a full knot with both strands or a square knot with that.' materializes in sign language?
How do I say And I also use memory wire. with sign language?
I really want to learn how Now together you're going to go opposite. is said in sign language. Can you help?
How do I articulate "It's real easy to actually get your fingers to lead, so try not to let them do that." using sign language?
I am intrigued to learn the sign language for 'Let the wrist do all the leading.'.
I am wondering how "Don't let the fingers take over, let the wrist do all the guiding." appears in sign language.

Part III

Ich möchte wirklich lernen wie man '{Text}' in Gebärdensprache sagt. Können Sie mir helfen?
Wie würden Sie '{Text}' in Gebärdensprache ausdrücken?
Können Sie mir zeigen wie man '{Text}' mit Gebärdensprache sagt?
Wie sage ich '{Text}' in Gebärdensprache?
Könnten Sie mir sagen wie '{Text}' in Gebärdensprache dargestellt wird?
Mich interessiert wie man '{Text}' in Gebärdensprache sagt.
Können Sie die Gebärdensprache für '{Text}' demonstrieren?
Ich möchte erfahren wie '{Text}' in Gebärdensprache übersetzt wird.
Was ist die Gebärdensprache für '{Text}'?

Part IV

'regen und schnee lassen an den alpen in der nacht nach im norden und nordosten fallen hier und da schauer sonst ist das klar' Wie stellt man das in Gebärdensprache dar?
Können Sie die Gebärdensprache für 'am donnerstag regen in der nordhälfte in der südhälfte mal sonne mal wolken ähnliches wetter dann auch am freitag' demonstrieren?
Mich interessiert, wie vom nordmeer zieht ein kräftiges tief heran und bringt uns ab den morgenstunden heftige schneefälle zum teil auch gefrierenden regen in Gebärdensprache aussieht.
Wie wird sonnig geht es auch ins wochenende samstag ein herrlicher tag mit temperaturen bis siebzehn grad hier im westen in Gebärdensprache dargestellt?
Wie würden Sie deutschland liegt morgen unter hochdruckeinfluss der die wolken weitgehend vertreibt gebärden?
Können Sie mir zeigen, wie am sonntag im nordwesten eine mischung aus sonne und wolken mit einigen zum teil gewittrigen schauern in Gebärdensprache aussieht?
Wie sieht die Gebärdensprache für örtlich schauer oder gewitter die heftig sein können aus?
Was ist die Gebärdensprache für und zum wochenende wird es dann sogar wieder ein bisschen kälter?
Was ist die Gebärdensprache für in der südhälfte weht der wind schwach sonst schwach bis mäßig richtung küsten frisch und stark böig?
Ich möchte wissen, wie man 'am freitag ruhiges trockenes wetter vor allem im norden ist es recht freundlich ähnliches wetter am samstag nur im norden vereinzelt etwas schnee oder gefrierender sprühregen' gebärdet.

Table 7. We provide two templates for sign language as a reference, and {Text} is where the video oral dialogue is inserted.

Name	Language	Vocab.	Duration (h)	Signers	Multiview	Transcription	Gloss	Pose	Depth	Speech	Prompt	Compress
Video-Based CSL [36]	CSL	178	100	50	✗	✓	✗	✓	✓	✗	✗	✗
SIGNUM [92]	GSL	450	55	25	✗	✓	✓	✗	✗	✗	✗	✗
RWTH-Phoenix-2014T [14]	GSL	3k	11	9	✗	✓	✓	✗	✗	✗	✗	✗
Public DGS Corpus [32]	GSL	–	50	327	✓	✓	✓	✓	✗	✗	✗	✗
BSL Corpus [76]	BSL	5k	–	249	✗	✓	✓	✗	✗	✗	✗	✗
NCSLGR [59]	ASL	1.8k	5.3	4	✓	✓	✓	✗	✗	✗	✗	✗
How2Sign [20]	ASL	16k	79	11	✓	✓	✓	✓	✓	✓	✗	✗
Prompt2Sign (ours)	Multilingual	40k	200	40	✓	✓	✓	✓	✓	✓	✓	✓

Table 8. **Dataset Details:** PROMPT2SIGN uses tools to automate the acquisition and processing of sign language videos on the web, is a better dataset that is efficient (a higher level of preprocessing, standardized and more models available), and lightweight (average reduction of 80% in space usage). Languages included: American Sign Language (ASL), German Sign Language (GSL, Alias DGS), Swiss German Sign Language (DSGS), French Sign Language of Switzerland (LSF-CH), Italian Sign Language of Switzerland (LIS-CH), Argentine Sign Language (Lengua de Señas Argentina, LSA), Korean Sign Language (KSL), and Turkish Sign Language (TSL).

3. More Details of Prompt2Sign

3.1. Dataset Modalities

In comparison to the previous datasets, we possess numerous additional advantageous attributes and a larger scale. As with the previous dataset work, we extracted everything automatically except speech/text. But we've added some automated channel tools that go deeper than that.

Prompt Word Templates. We constructed 120 English templates and 210 prompt word templates generated by GPT4 for other languages (with 30 templates for each language), which were randomly associated with the script data to form a part of our dataset. Some examples are in Table 7 above. This prompt word data is needed for the future development of large language models of sign language because we were able to develop a model with understands more complex, natural human conversational inputs by using prompt word data.

Data Enhancement. In our multilingual sign language production tasks, we found that the model underperformed in low-resource sign languages. For these sign languages, where it's inherently difficult to increase the data volume, we use tools to rewrite lines or prompt words and can obtain several times more data to enhance the robustness of the trained model in low-resource data.

Multiview. Our multiple perspectives depend on the original video, and it is worth noting that if the researchers cannot guarantee that the newly acquired perspectives are all positive, then the model will generally be contaminated.

Depth Data. Our depth depends on whether the raw data video has relevant support, we believe that this is generally not needed, as most work uses lifting work to obtain 3D key points, rather than high-cost professional equipment.

Speech. Some of our audio comes from raw data and some comes from Google's text-to-audio tool.

Compress. It refers to whether the data set has been compressed in a special way to make it easy to use.

3.2. Pose Information

Necessity of Uniform Standards. If there is a mismatch between any of these components in SLP [11, 71, 73, 82, 94, 95] or SLT [4, 7, 15, 43], it can lead to complex challenges. For instance, if the results of pose recognition cannot be used as training data, the results of SLR cannot be used for model testing, or if the results of sign language generation cannot be used as conditional input [11, 97, 107], which have troubled many novice researchers in the field.

Data Format Conversion.

- **How to extract key points?** We extracted two-dimensional (2D) frontal human pose information from videos of different resolutions, including upper body pose information of the body and hands, through OpenPose [9]. Includes 8 upper body key points. 21 keypoints in each hand, which is a total of 42 hand keypoints. These two parts add up to fifty keypoints, each of which has three XYZ messages, or 150 numbers.

Then in steps "json (2D keypoints) to h5", "h5 to txt (3D keypoints)", and "txt to compressed data":

- **How to complete "json to h5"?** We successively obtain a json number in a folder (a frame of pose information, 50 key points, 150 numbers), and then read all the json numbers in a folder into the key name of an h5 (h5 is a format of numpy) file, multiple folders form multiple build names, and finally form an h5 file.
- **How to complete "h5 to txt"?** We read each key name of h5 in turn (the original folder name), create the corresponding folder, each folder generates 5 txt files, the last one is the result, the first 4 txt stores the intermediate variable. This is the part of 2D to 3D, and the key formula 3 in the text is the formula of this part. Additionally, we read the relevant data and delete the unqualified data such as NaN, 0, or replace it with the average median. Finally, we condensed the data to about 1/5 of the original, which comes from the processing of the ASL part.
- **How to complete "txt to compressed data"?** We read the fifth txt file of each folder in turn, the number of

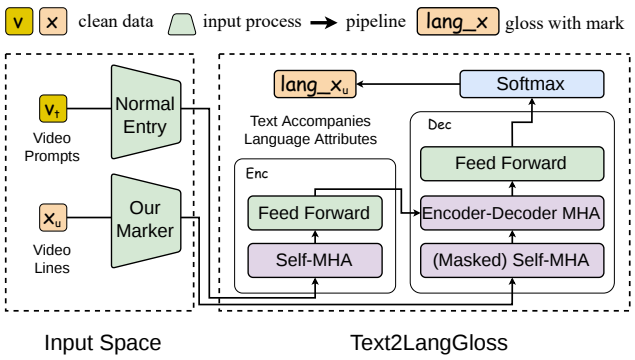

Figure 6. We enhance Text2Gloss [71] with a marker to generate the Gloss with linguistic properties. The $\boldsymbol{v}_t$ (v_t) and $\boldsymbol{x}_u$ (x_u) represent data types and abstract representations.

lines in the txt file represents the number of frames of the folder corresponding to the video. We read a line of txt (150 numbers, separated by Spaces, a frame of information), plus a space, and then add a count value (the current line divided by the total number of lines, representing the progress bar), add a space after the count value, Then add the second line txt and continue to repeat the above. Then we put a txt (video information, the total number of numbers in it = 151 $\times$ video frames) into a line of content, in turn, tens of thousands of videos are all stored in our standard format.

3.3. More Details of the Data

Details of Processing. Firstly, we obtain the original video from the internet. As mentioned in the main text, this part still needs to be done manually, but a script can be written to speed up the process. Firstly, preliminary preprocessing can be done through scripts written by oneself or OpenASL [78] scripts . Secondly, the dialogue of the video is transcribed into text, videos are processed using OpenPose, and then used as input for our tool. Finally, enters the language mode corresponding to the data by setting the model to start training.

Time and Cost of Dataset Processing Among all the data processing steps, the most time-consuming step is 2Dto3D, a GPU can process 1000 clips after 10 hours, and can process 50-80 hours of How2Sign data in about half a month (there is no 80 after editing). Improving the performance of a single card does not make it much faster, which may be caused by multithreading concurrency restrictions.

3.4. Prompt2LangGloss.

As shown in Figure 2 (Left), our proposed enhancement of this model involves appending an additional language attribute to each Text word during the reading and tokenizing stages. For instance, a traditional gloss token "<xxx>" can be transformed into "<ASL_xxx>", thus introducing a layer of conditional input $f_u = E_{T2LG}(x_u|x_{1:U})$ into SLP based on Eq. 1: $lg_{w+1} = D_{T2LG}(lg_w|lg_{1:w-1}, f_{1:\mathcal{U}})$. This figure is a supplement to the main paper text.

During this process, we can see in detail how our components operate and what the encoder-decoder structure is. In the main text, we noticed that more complex prompt words act as noise relative to the text we actually want to translate. Therefore, we conducted some experiments in Table 11 to verify whether this impact could be reduced or eliminated.

4. More Experiments

4.1. Extensibility & Visual Study

Subsequently, we provide an overview and comparison of motion capture techniques and novel visual models. Our objective is to advocate for adopting motion capture technology as a replacement for traditional visual methods in sign language rendering; they can reduce the finger-missing problems mentioned in the main text of the paper. Before that, we need to introduce some background:

4.1.1 Motion & Visual Method Introduction

SMPL skeleton system: The SMPL [51] (Skinned Multi-Person Linear) skeleton system is a parametric model that represents human body shape and pose. It is commonly used in computer graphics and animation. In the context of sign language, the SMPL skeleton system can be utilized to model and animate sign language movements and gestures.

VMD files and OpenMMD: VMD (Vocaloid Motion Data) files and OpenMMD (Open-source MikuMikuDance) refer to specific file formats and software tools used in character animation. VMD files contain motion data and are commonly used in the MikuMikuDance software for animating virtual characters. OpenMMD is an open-source implementation that allows users to create and modify character animations. In the context of sign language, VMD files and OpenMMD can be utilized to animate virtual characters performing sign language gestures or movements.

Keypoint driven model: A key point driven model is a computational model or algorithm that relies on the detection and tracking of specific key points, landmarks, or features in order to analyze and interpret data or generate desired outputs. In the final pose-to-video stage of sign language rendering, the generation of realistic human videos from keypoints is essential. This can be accomplished through either motion capture or purely visual methods. In the following sections, we will evaluate the strengths and limitations of each approach. In the context of sign language, a keypoints-driven model can be used to analyze and interpret sign language movements based on the detection and tracking of key points on the signer's body, such as hand positions, facial expressions, and body postures. It is our tentative exploration in this work.

Figure 7. **Extensibility Presentation:** We used five motion capture models and five sign language rendering models to show the final production effect.

4.1.2 Comparison of Motion and Visual

Extensibility Study. In Figure 7, the first line of it is obtained either directly or indirectly by reading our SIGNLLM output sequence through motion capture[3] [6, 12] software or models, while the second line of the image comes from the commonly used Pose2Vid [35, 54, 67, 90, 93, 105] or Pose2Img [39, 64, 95, 108] models. The broad scope of our model becomes apparent from the initial two statements. Subsequently, the next four lines present sign language demonstration videos created using either direct or indirect input of keypoints (some videos sourced from the project website). It is important to note that SMPLer-X and Avatar are utilized solely for demonstrative purposes in this context. Taking DeepMotion and VMD as instances, our model exhibits the capability to operate within a broader scope by utilizing keypoints as input, rather than relying solely on visual methods. This advancement provides the potential for more precise sign-language demonstrations. Details can be found in the footnote links.

[3]DeepMotion; Plask.ai; Avatar; OpenMMD

	SSIM ↑	Hand SSIM ↑	Similarity ↑	F2FD ↓
Vid2Vid [94]	0.743	0.582	78.42	27.86
ControlNet [105]	0.817	0.646	82.11	25.47
Motion Capture	**0.826**	**0.687**	**81.29**	**22.71**

Table 9. **Visual Study:** SSIM: Comparison of image structure similarity between the generated image and the condition graph extracted from the Ground Truth. Similarity: Extract the similarity percentage of keypoints between the generated video and the input action. F2FD: The degree of difference between frames.

Visual Study. We explored the influence of different forms on performance as shown in Table 9, current existing motion capture models do not fully support our keypoints format, and there may be some loss in certain transmission processes. Therefore, our primary focus is evaluating the presentation effect of motion capture models in sign language. Taking DeepMotion as an example, it is a deep learning-based method that drives models in a software environment using keypoints. In previous work, the comparison between rendered results and GroundTruth was measured using the structural similarity index (SSIM). However, since driving models do not have a specific GroundTruth, our comparison is based on the visualized keypoints extracted, which may introduce some errors but generally remain below 1%, providing a sufficient basis for simple comparisons. The percentage similarity refers to the comparison of extracted sequence numbers. Additionally, the difference between frames focuses on the smoothness of the video, as motion capture models do not exhibit the flickering issue common in generative models, resulting in smaller differences between consecutive frames. While the software can output a higher number of frames for enhanced results, we set the frame rate to 24 frames per second for fair comparisons. In conclusion, we believe that introducing motion capture-related techniques, models, or methods holds great promise in the final rendering stage of sign language.

4.2. Model Parameter Study

As shown in Table 10, we have investigated the optimal parameter settings under different circumstances to provide further discussion and guide future researchers in their training. This includes the optimal results of our primary model parameters, architecture, and various learning rates or other parameters. The experimental results were derived by evaluating the performance of the Text to Pose function from the SIGNLLM-40M-Base model to the SIGNLLM-1B-Large model on the ASL part of PROMPT2SIGN dataset. In general, we find that (1) The optimal values of the parameters conform to the scaling law [34, 41], according to

Key name	Values	Note
BSLP method	{**ASL**, GSL}	Choose before training
Vocabulary size	{1k, 4k, 7k, **16k**}	Case-sensitive
Batch size	{**8**, 16, 32}	Adjust according to configuration
Learning Rate (LR)[4]	{5e-2, **1e-3**, 5e-3}	Training initial value
Loss mode	{**MSE**, RL, L1, L2, LV}	Adjust according to situation
Max_sent_length	{300, **400**}	Input is usually less than maximum
Priority Learning Channel	{False, **True**}	Use with RL Loss
Dropout	{**0**, 0.1, 0.2, 0.3}	Adjust according to situation
# Layers (encoder-decoder)	{2-2, 4-4, **8-8**}	Not necessarily correspond
Embed dim	{512, 1024, **2048**}	Adjust for the amount of data
FFN dim	{2048, 4096, **8192**}	Must equal to 4*hidden size
# Attention heads	{4, 8, **16**}	Adjust for the amount of data

Table 10. **Hyperparameters Space:** Optimal choices are marked in **bold**, while defaults are underlined. The default values come from the SIGNLLM-120M-Base-M (ASL).

	DEV SET		TEST SET	
Approach:	BLEU-4 ↑	ROUGE ↑	BLEU-4 ↑	ROUGE ↑
Stoll *et al*. [81]	16.34	48.42	15.26	48.10
Baseline [71]	20.23	55.41	19.10	54.55
Ours	**23.10**	**58.76**	**22.05**	**56.46**
Δ *Acc.*	***+ 14.2%***	***+ 6.0%***	***+ 15.4%***	***+ 3.5%***

Table 11. **Prompt Channel Accuracy:** We investigate the Prompt2LangGloss channel information loss by German SLP.

which we should increase the number of parameters by four times when the data is increased by four times. There is no significant difference between the 120M model and the 40M model prediction without too much increase in data volume, and there is also a larger magnitude. (2) When we use Prompt2LangGloss, our data equals the sum of two single language versions. But at this time, their performance mainly depends on the data of a single language, which is a special case: Although they share parameters, the LangGloss has distinguished enough of the sign language pose corresponding to the input text, they do not enjoy the bonus of shared parameters.

Prompt Fine-Tuning and User Study. By employing prompt words as input for the text channel and using the original text or the original gloss as input for the gloss channel (*i.e*., usage of Prompt2LangGloss), we can develop a model with understanding prompts competency. This approach aims to translate natural language into objective text/gloss before inputting it into the model. In reality, users might question, *"How do you demonstrate 'the sky is blue' in sign language?"*, rather than directly inputting *"the sky is blue"*. This training strategy gives the model a degree of Understanding more complex input text ability. We compared the Prompt2LangGloss channel with a tokenizer to the previous Text2Gloss approach in terms of prompt usage. The experiments indicated that the impact of the tokenizer in the Prompt2LangGloss channel is relatively small and can be overcome through better training. As presented in Table 11, it underscores the effectiveness of our mode in reducing semantic information loss in the channel.

[4]LR is getting smaller and smaller over time, approaching a set value.

5. More Discussion

Discussion on The Dataset Range. We have cited the sources of our publicly available data, and some of the more popular works were not considered due to their limited accessibility and potential usage restrictions. Additionally, while there are other multilingual datasets available [29, 33, 55, 101], they may not possess the same level of comprehensiveness as ours. Like [29] and [101], they are papers that translate two types of sign language videos into spoken language (SLT), while our work is from spoken language to videos (SLP). We aim to develop the multilingual SLP method, and our dataset has more diverse application scenarios than them (*e.g*., one is Bible translation[5] [29], the other[6] [101] is cross SLT, but we are comprehensive scenarios and SLP. Some are multilingual dictionary datasets [33, 55]). We are a very beneficial supplement to previous work, which is quite different from previous work.

Discussion on The Dataset Errors. We handle issues related to NaN, zero, and missing data by applying deletion or replacement techniques and our tool simplifies certain calibration stages in comparison to previous 2D to 3D tools, which may introduce some errors. The substituted data is derived using median or mean values, resulting in minuscule errors. Within the vast parts of dataset, these errors typically fall within the range of 0.5% to 0.7% (We conducted a random sampling of results and obtained a ratio of 87 out of 17,549 to 47 out of 6,685). Moreover, our processing steps involving normalization greatly diminish such errors. Hence, we have reasonable grounds to assert that the data error is minimal enough to be bearable. In addition, we consider reducing these errors before release. And some of the data sets have some potential problems, and the number of our data sets is based on the final release.

Discussion on The Prompt2Pose Task? If existing methods want to use long text prompts as input, they must obtain more data to achieve better results. However, sign language data is very scarce. We need to propose a new Prompt2LangGloss method that can efficiently translate based on the scarcity of sign language. Therefore, it is necessary to propose Prompt2Pose as an independent task, as this module has many application scenarios even in a wider range of action synthesis (*e.g*., the robot can understand human commands more efficiently, so that the LLM to summarize the Prompt is not required). Compared to using LLM to simplify and summarize everyone's input, creating an efficient end-to-end sign language model is very valuable, similar to the popular Retrieval Augmented Generation (RAG), but for sign language.

[5]https://aclanthology.org/2023.findings-emnlp.664/
[6]https://ieeexplore.ieee.org/document/9878501

Mode	Function	Address	Enc-Dec	Prompt	Feature	Note
M	multilingual SLP	text2pose	Multiple	No	More efficient/stable	Language is easy to add or subtract
P	multilingual SLP	text2gloss	Single	Allow	Understand complex input	Greater potential for development

Table 12. **The difference between the two modes:** M and P represent MLSF and Prompt2LangGloss, respectively. Adress represents which traditional step has been innovated, while Features represents the ability of the mode to focus more on.

Why do we need to have two models? The reasons for designing two modes are as follows:

1. **Different Usage Scenarios:**
   - MLSF targets direct text translation needs
   - Prompt2LangGloss addresses sign language-related Q&A and instruction needs
2. **Research Purpose Level:**
   - Provides two different paradigms for researchers to reference; Demonstrates two different approaches for multilingual sign language generation
   - Proves that existing/LLM models can be transformed into multilingual models through different approaches; These two directions will be the mainstream approaches in the future
3. **Practical Significance:**
   - Provides solutions for different application scenarios
   - Enables sign language models to serve different needs more flexibly

Therefore, the core reason for using two separate modes is: they address different needs in sign language generation tasks while providing diverse technical paradigm references for the research community.

Why can't the two modes merge into one? Because the two modes cannot coexist in different AB stages. Let's imagine a scenario where you're going to a distant place, with two stages, A and B, occurring in sequence. If you have three paths to choose from in Stage A and only one path in Stage B, you will ultimately have 3×1 choices. If you have only 1 path in stage A and 3 paths in stage A, overall, you still have 1×3 choices.

Therefore, they cannot coexist because 3×3 is a meaningless and more complex choice: If you want to implement 9 languages (9 choices), you only need to modify either the first stage or the second stage into nine paths. There's no need to modify both stages separately, as this would make the model unnecessarily complex. To implement multilingual sign language production in a model, only one stage of AB needs to develop multiple paths (languages).

The difference of Two Modes MLSF dynamically adds encoders, which can avoid semantic confusion and maximize its convenience (*e.g.*, a general model can execute multilingual SLP tasks that were impossible for researchers in the past. It saves significant development time, potentially twice the effort, ten times the return). Prompt2LangGloss focuses on improving the ability to understand complex inputs, which is complementary to the MLSF. It will have great prospects with the data volume increase (*e.g.*, ChatGPT rarely mixes languages when speaking in a specific language). Moreover, LangGloss can be used without choosing a language as a valuable feature. Therefore, both approaches have their focus, input type, and user cases, summarized in Table 12.

Discussion on Multilingual SLP Task. As mentioned in Table 3 of the main text, most lesser-known sign languages basically have no baseline data for sign language production. This may be due to various reasons, and some datasets don't even have baseline data for sign language recognition. In this situation, it's difficult for us to replicate work from many years ago, and it's challenging to obtain the performance level of sign language production models for these languages. Therefore, for transformer-based or deep learning-based sign language production, we indeed proposed the first baseline for these languages without needing to compare with any previous models (of course, in most cases, these models don't exist).

Discussion on Similar Name. We noticed that a work [26] has introduced a large language model to translate sign language videos into spoken text. Their proposed framework is also called SignLLM. Since our work focuses on converting spoken text to sign language videos, which is the opposite direction, there is no conflict between the two approaches despite sharing the same name.

Discussion on The Significance of Our Dataset. Sign language is different from most motion recognition fields, requiring many complex annotations, and video datasets generally have a low level of processing. This work is also very time-consuming and labor-intensive, even more challenging than creating an original video dataset with more than a dozen languages. What we need is high-quality data. If each researcher needs to process thousands of hours of data before starting experiments, many researchers will lose enthusiasm, which happens frequently in this field. Therefore, our new Prompt2Sign work is very necessary.