1. Abstract of Materials

In the supplementary material, we have provided additional details and content that complements the existing sections of the article. It is structured in a manner that aligns with the original order of the sections and expands upon the methods, details, experimental content, theoretical analysis, qualitative assessments, and further demonstrations. By reviewing the supplementary material, readers can gain a better understanding of our proposed methods, clarify any misconceptions, and achieve a clearer comprehension. More details are available at https.

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. A Gloss 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 each frame.

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.

[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 stage of sign language production that was omitted in the main text (Pose2Video, which involves visualizing key points in a video rendering or converting it into a live person/model demonstration of sign language) and some basic concepts of reinforcement learning (some reinforcement learning knowledge is added here 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) [22] have exhibited favorable performance in generating images [31, 53, 82, 93] and videos [44, 77, 80, 81, 83]. Numerous studies have also focused on generating human poses while considering various factors, including entire body [1, 9, 43, 47, 67, 72, 94], face [17, 38, 75, 86, 87, 90], and hand [41, 71, 85]. One particular application is human-style transfer [56], 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 [9, 84, 92]. The key aspect lies in extracting keypoints to replicate movements [9, 78], utilizing tools such as OpenPose, i3D, and DensePose for common keypoint extraction [9, 51, 84, 92]. 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:

$$\pi^* = \arg\max_{\pi} \mathbb{E}[G_t | s_t = s, \pi] \tag{11}$$

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?
What is the sign language for '{Text}'?

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?
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 8. Each of the four sections is a template of an ASL cue word, which is the result of embedding the line into the ASL template. The template of the cue word in German sign language, the result of embedding the line into the GSL template.

Name	Language	Vocab.	Duration (h)	Signers	Modalities							
					Multiview	Transcription	Gloss	Pose	Depth	Speech	Prompt	Compress
Video-Based CSL [29]	CSL	178	100	50	✗	✓	✗	✓	✓	✗	✗	✗
SIGNUM [79]	GSL	450	55	25	✗	✓	✓	✗	✗	✗	✗	✗
RWTH-Phoenix-2014T [12]	GSL	3k	11	9	✗	✓	✓	✗	✗	✗	✗	✗
Public DGS Corpus [25]	GSL	–	50	327	✓	✓	✓	✓	✗	✗	✗	✗
BSL Corpus [63]	BSL	5k	–	249	✗	✓	✓	✗	✗	✗	✗	✗
NCSLGR [48]	ASL	1.8k	5.3	4	✓	✓	✓	✗	✗	✗	✗	✗
How2Sign [19]	ASL	16k	79	11	✓	✓	✓	✓	✓	✓	✗	✗
Prompt2Sign (ours)	ASL&GSL	21k	90	20	✓	✓	✓	✓	✓	✓	✓	✓

Table 9. **Summary of previous mainstream sign language datasets:** Our PROMPT2SIGN, which uses tools to automate the acquisition and processing of sign language videos on the web, is an evolving data set that is efficient and lightweight without the previous shortcomings. Languages such as Chinese Sign Language and British Sign Language will also be included in the future.

3. More Details of Prompt2Sign

3.1. Dataset Modalities

Our dataset, as of now, is the largest bilingual dataset, highly preprocessed, as shown in Table 9. 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. 120 English prompt word templates and 30 German prompt word templates are generated by GPT4, with some examples in Table 8 above.

Data Enhancement. With tools that rewrite lines or prompt words, users can obtain several times more data to enhance the robustness of the trained model.

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, most work using 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 [9, 58, 60, 69, 81, 82] or SLT [3, 6, 13, 35], 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 [9, 84, 92], 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 [8]. 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 skels (Standard Pose Storage)":

- **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 of the data. Finally, we condensed the data to about 1/5 of the original, this data comes from the processing of ASL part.
- **How to complete "txt to skels"?** We read the fifth txt file of each folder in turn, the number of 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 (a video information, the total number of numbers in it = 151* video frames) into a line of content, in turn, tens of thousands of videos are all stored in our standard format.

input : Three arrays Xx, Xy and Xw of size $T \times n$, a structure array, a sigma value for noise, a random number generator and a percentile value

output: Lines array, rootsx, rootsy, rootsz arrays, anglesx, anglesy, anglesz arrays, Yx, Yy, Yz arrays

Set T as number of rows and n as number of columns of Xx;
Initialize arrays lines, rootsx, rootsy, rootsz, anglesx, anglesy, anglesz;
Set rootsx as first column of Xx;Set rootsy as first column of Xy;
Set rootsz as array of zeros with size T;
Add noise to rootsx, rootsy, rootsz arrays;
Initialize arrays Yx, Yy, Yz as arrays of zeros with size $T \times n$;
Set first column of Yx as rootsx; Set first column of Yy as rootsy; Set first column of Yz as rootsz;
for *each bone in structure* **do**
    Add empty list to lines;
    for *each row t in range T* **do**
        Compute length L using equation 1;
        Append L to lines;
    end
end
for *each line in lines* **do**
    Calculate max L as percentile of the line list;
    Assign math.log(max L) to lines array;
end
for *each bone in structure* **do**
    Assign a, b, line as elements of the bone;
    for *each row t in range T* **do**
        Compute rotation angles anglex, angley, anglez using equation 2;
        if *any of anglex, angley, anglez is not finite* **then**
            Set anglex, angley, anglez as 0.0;
        end
        if *anglez ¡ 0.0* **then**
            Set anglez as -anglez;
        end
        Add 0.001 to anglez;
        Normalize anglex, angley, anglez;
        Assign anglex, angley, anglez to anglesx[t, iBone], anglesy[t, iBone], anglesz[t, iBone];
        Compute new 3D coordinates Yx, Yy, Yz using equation 3;
    end
end
Reshape rootsx, rootsy, rootsz as arrays of size T × 1;
Return lines, rootsx, rootsy, rootsz, anglesx, anglesy, anglesz, Yx, Yy, Yz;

Algorithm 1: The core formula and code of 2D to 3D can complement the text

3.3. More Details of the Data

Details of Acquisition and 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. What kind of sign language model you want to train requires corresponding corpus. Firstly, preliminary preprocessing can be done through scripts written by oneself or OpenASL [66] 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, whose core code is shown in Algorithm 1, RTX3060 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.

Language	Format	Example	BLEU	ROUGE
(ASL)	Reference (Text) Prediction	The number one loss for these birds, is flight. The birds couldn't lose the flight.	20.21	62.86
(GSL)	Reference (Text) Prediction	Hoher luftdruck sorgt bei uns morgen für viel sonnenschein. Morgen viel sonnenschein wegen hohem luftdruck.	10.1	56.43
(ASL)	Prompt Reference (LangGloss) Prediction	What is the sign language for 'When does this take place?'. ASL_When ASL_does ASL_this ASL_take ASL_place? When does it takes up?	40.00	40.00
(GSL)	Prompt Reference (LangGloss) Prediction	Wie gebärdet man 'am freundlichsten wird es am meer' zeigen? GSL_Am GSL_freundlichsten GSL_wird GSL_es GSL_am GSL_meer. Am meer wird es am freundlichsten sein.	57.14	73.4

Table 10. **Concrete Cases Study:** We select some sample examples for readers to understand better. When using the prompt, we referred to the intermediate text, which would have made the measurement more accurate. Although there is still a problem that the accuracy is reduced due to the loss in the process of prompt-word to text. Red and Blue mark similarities in the text in ASL and GSL, respectively.

	DEV SET		TEST SET	
Approach:	BLEU-4 ↑	ROUGE ↑	BLEU-4 ↑	ROUGE ↑
Baseline [68]	16.34	48.42	15.26	48.10
Stoll et al. [58]	20.23	55.41	19.10	54.55
T (no prompt)	**22.45**	**53.56**	**20.12**	**51.37**
Ours	**23.10**	**58.76**	**22.05**	**56.46**

Table 11. **Prompt Ablation Study:** We added a row of comparative data to the table, T represents a model that uses Text2LangGloss, but does not use Prompt as input this time.

4. More Experiments

4.1. Concrete Cases Study

We show specific production examples in both MLSF and Text2LangGloss Settings and in both ASL and GSL in Table 10. They include reference inputs and the results we produce and translate, as well as some intermediate results. Multiple examples show that not only do we perform well with multiple language generation, but there are exciting results with prompt word generation. The case of the table is adjusted for the sake of aesthetics, and the calculation result is based on the actual situation.

4.2. Prompt Fine-Tuning and 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, we were able to develop a model with fundamental understanding prompts competency. The aim of this approach is to translate natural language into objective text/Gloss before inputting them 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"*. We have elaborated on the relevant details in the main text, and in the attached material table, we have added ablation experiments for prompt words. Because when we use prompt words as input and run Text2LangGloss, both of these behaviors will affect the model's performance. As shown in Table 11, the impact of markers is not particularly significant, and the main factor that reduces model performance is the introduction of noise.

4.3. Extensibility & Visual Study

Subsequently, we present a brief overview of motion capture techniques and novel visual models. Following that, readers can observe our qualitative assessment demonstrations of scalability and our straightforward comparative analysis. We intend to promote motion capture technology to replace traditional visual methods in the field of sign language rendering. Before that, we need to introduce some supplementary information:

4.3.1 Motion & Visual Method Introduction

SMPL skeleton system: The SMPL [42] (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 of the MikuMikuDance software 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.

Key point 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.

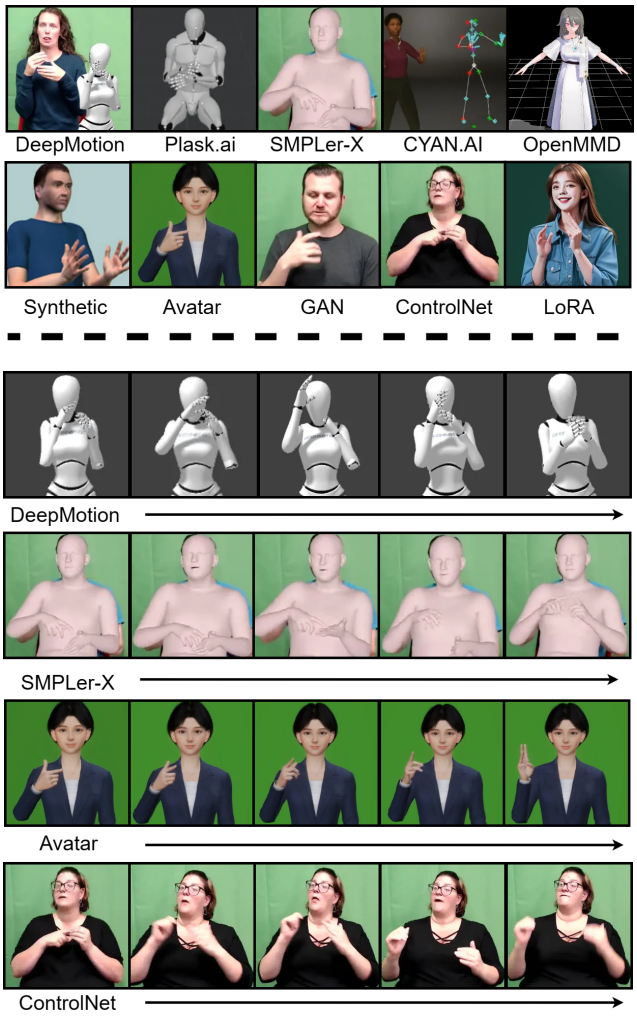

Figure 7. **Extensibility Presentation:** We used five motion capture models and five sign language rendering models to represent the final effect of production.

4.3.2 Comparison of Motion and Visual

Extensibility Study. In Figure 7, the first line of it is obtained either directly or indirectly by reading our SIGN-LLM output sequence through motion capture[3] [5, 10] software or models, while the second line of the image comes from the commonly used Pose2Vid [27, 44, 55, 77, 80, 89] or Pose2Img [31, 53, 82, 93] 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.

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

	SSIM ↑	Hand SSIM ↑	Similarity ↑	F2FD ↓
Vid2Vid [81]	0.743	0.582	78.42	27.86
ControlNet [89]	0.817	0.646	82.11	25.47
Motion Capture	**0.826**	**0.687**	**81.29**	**22.71**

Table 12. **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 12, 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 lies in 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 the introduction of motion capture-related techniques, models, or methods holds great promise in the final rendering stage of sign language.

4.4. Discussion on 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 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.