CVT-SLR: Contrastive Visual-Textual Transformation for Sign Language Recognition with Variational Alignment

Jiangbin Zheng¹, Yile Wang¹,², Cheng Tan¹, Siyuan Li¹, Ge Wang¹, Jun Xia¹, Yidong Chen³, Stan Z. Li¹*

¹AI Lab, Research Center for Industries of the Future, Westlake University
²Institute for AI Industry Research (AIR), Tsinghua University
³School of Informatics, Xiamen University

{zhengjiangbin,wangyile,tancheng,lisiyuan,wangge,xiajun,Stan.ZQ.Li}@westlake.edu.cn
ydchen@xmu.edu.cn

Abstract

Sign language recognition (SLR) is a weakly supervised task that annotates sign videos as textual glosses. Recent studies show that insufficient training caused by the lack of large-scale available sign datasets becomes the main bottleneck for SLR. Most SLR works thereby adopt pre-trained visual modules and develop two mainstream solutions. The multi-stream architectures extend multi-cue visual features, yielding the current SOTA performances but requiring complex designs and might introduce potential noise. Alternatively, the advanced single-cue SLR frameworks using explicit cross-modal alignment between visual and textual modalities are simple and effective, potentially competitive with the multi-cue framework. In this work, we propose a novel contrastive visual-textual transformation for SLR, CVT-SLR, to fully explore the pre-trained knowledge of both the visual and language modalities. Based on the single-cue cross-modal alignment framework, we propose a variational autoencoder (VAE) for pre-trained contextual knowledge while introducing the complete pretrained language module. The VAE implicitly aligns visual and textual modalities while benefiting from pretrained contextual knowledge as the traditional contextual module. Meanwhile, a contrastive cross-modal alignment algorithm is designed to explicitly enhance the consistency constraints. Extensive experiments on public datasets (PHOENIX-2014 and PHOENIX-2014T) demonstrate that our proposed CVT-SLR consistently outperforms existing single-cue methods and even outperforms SOTA multi-cue methods. The source codes and models are available at https://github.com/binbinjiang/CVT-SLR.

1. Introduction

As a special visual natural language, sign language is the primary communication medium of the deaf community [19]. With the progress of deep learning [1, 17, 25, 39, 42], sign language recognition (SLR) has emerged as a multi-modal task that aims to annotate sign videos into textual sign glosses. However, a significant dilemma of SLR is the lack of publicly available sign language datasets. For example, the most commonly-used PHOENIX-2014 [23] and PHOENIX-2014T [2] datasets only include about 10K pairs of sign videos and gloss annotations, which are far from training a robust SLR system with full supervision as typical vision-language cross-modal tasks [34]. Therefore, data limitation that may easily lead to insufficient training or overfitting problems is the main bottleneck of SLR tasks.

The development of weakly supervised SLR has witnessed most of the improvement efforts focus on the visual module (e.g., CNN) [9, 10, 15, 29, 32, 33]. Transferring pre-trained visual networks from general domains of human actions becomes a consensus to alleviate the low-resource limitation. The mainstream multi-stream SLR framework extends the pretrained visual module with multi-cue visual information [3, 22, 24, 43, 48, 50], including global features and regional features such as hands and faces in independent streams. The theoretical support for this approach comes from sign language linguistics, where sign language utilizes multiple complementary channels (e.g., hand shapes, facial expressions) to convey information [3]. The multi-cue mechanism essentially exploits hard attention to key information, yielding the current SOTA performances. However, the multi-cue framework is complex (e.g., cropping multiple regions, requiring more parameters), and the fusion of multiple streams might introduce additional potential noise.

Another mainstream advanced solution is the single-cue
Our main contributions are as follows:

• A novel visual-textual transformation-based SLR framework is proposed, which introduces fully pretrained language knowledge for the first time and provides new approaches for other cross-modal tasks.

• New alignment methods are proposed for cross-modal consistency constraints: a) exploiting the special properties of the autoencoder to implicitly align visual and textual modalities; b) introducing an explicit contrastive cross-modal alignment method.

• The proposed single-cue SLR framework not only outperforms existing single-cue baselines by a large margin but even surpasses SOTA multi-cue baselines.

2. Related Work

SLR tasks are generally divided into three categories [19]: finger-spelling recognition [18, 30, 31], isolated word recognition [26, 40, 41, 49] and continuous sign sentence recognition [8, 11, 12, 29, 33]. In the early days, SLR works mainly focused on lexical-level tasks, such as finger-spelling recognition and isolated word recognition. Nowadays, the more practical continuous sign sentence recogni-
tation task has become mainstream sign language research. In this work, the mentioned SLRs refer specifically to continuous sign sentence recognition.

The recent SLR works [8, 11, 15, 29] are summarized based on these three aspects: feature extraction, recognition, and alignment. Mostly, the components of feature extraction are composed of the visual module and the contextual module. The visual module encodes short-term spatial information, while the contextual module encodes long-term context information. Based on the extracted features, the classifiers can get a posterior probability for recognition. And the alignment module is required to find the proper alignment between clips and glosses to ensure an accurate training procedure.

A few studies extend the common SLR framework by incorporating multi-cue information [22, 24, 43, 44, 48]. The multi-cue features mainly include information from hand shapes, facial expressions, mouths, and poses. The current SOTA SLR works are based on multi-cue mechanisms, such as C²SLR [50]. Recently, some works show that explicitly enforcing the consistency between the visual and textual modules can also get comparable performance, although only single-cue features are considered. For example, VAC [28] treats the visual and textual modules as the student and teacher, respectively, and achieves knowledge distillation to align the visual and the textual modalities. Similarly, SMKD [15] achieves knowledge transfer by further sharing classifiers between different modalities. Note that SMKD is the current SOTA single-cue method.

It is worth mentioning that neural sign language translation [2, 3, 7, 46] is another popular sign language task. As a distinction, the neural sign language translation task aims to convert sign videos into equivalent spoken language translations, further focusing on semantic relationships and adjusting word order [2], while the SLR task only annotates sign videos into gloss sequences in the frame order.

3. Methods

As shown in Figure 2, our training pipeline is roughly divided into two steps. Step 1 is to pretrain a VAE network as the textual module. And step 2 is to transfer the off-the-shelf visual module (usually a publicly available CNN trained on Kinetics/ImageNet) and the pretrained textual module from step 1 into the CVT-SLR framework. A video-gloss adapter is used to bridge these two pretrained modules.

3.1. Textual Model Pretraining

Our VAE model is an asymmetric encoder-decoder architecture, where the encoder consists of self-attention layers, and the decoder consists of Bi-LSTM layers, as shown in the top part of Figure 2. To pretrain the VAE for our proposed SLR framework solely based on the gloss data from the PHOENIX-14 training set (excluding the development set and testing set), we construct a pseudo-translation task, Gloss2Gloss, which enables the VAE to learn the contextual semantic knowledge. Formally, it is assumed that the input gloss of the encoder is denoted as \( G \), and the generative gloss of the decoder is denoted as \( G^* \).

**Posterior Self-attention Encoder.** We employ the neural approximation, and reparameterization [21, 35] techniques to approximate the posterior self-attention encoder for generating the posterior distribution \( q_\phi(z|G) \), where \( \phi \) denotes the variational parameters of the encoder. The \( q_\phi(z|G) \) is a diagonal Gaussian distribution \( N(\mu, \text{diag}(\sigma^2)) \), and two single-layer MLPs are used to parameterize the mean \( \mu \) and the variance \( \sigma^2 \), respectively, as:

\[
z = \mu + \sigma \odot \epsilon,
\]

where \( \epsilon \) is a standard Gaussian noise, and \( \odot \) denotes an element-wise product.

**Generative Bi-LSTM Decoder.** Given the continuous latent variable \( z \), then the translation target \( G^* \) is generated from \( z \) as:

\[
p_\theta(G^*, z) = p_\theta(G^*|z)p_\theta(z),
\]

where \( \theta \) denotes the parameters of the generative Bi-LSTM decoder, and \( p_\theta(G^*|z) \) is the conditional distribution that models the generation procedure estimated via the decoder.

**Variational Optimization.** Both the generative model \( p_\theta(G^*, z) \) and the posterior inference model \( q_\phi(z|G) \) are incorporated tightly in an end-to-end manner, and we can apply a standard back-propagation to optimize the gradient of the variational lower bound:

\[
\mathcal{L}_{\text{VAE}}(\theta, \phi; G) = -\text{KL}(q_\phi(z|G)||p_\theta(z)) + \mathbb{E}_{q_\phi(z|G)}[\log p_\theta(G^*|z)] \leq \log p_\theta(G^*) ,
\]

where KL denotes the Kullback-Leibler divergence and the prior \( p_\theta(z) \) denotes a standard Gaussian distribution. Practically, the first KL term of Eq. (3) can be written as:

\[
\mathcal{L}_{\text{KL}} = -\frac{1}{2}(\log \sigma^2 - \sigma^2 - \mu^2 + 1).
\]

Since our main target is a Gloss2Gloss translation task, we employ a cross-entropy loss to maximize the generative gloss probability as the second term of Eq. (3):

\[
\mathcal{L}_{\text{gloss2gloss}} = \text{CrossEntropy}(G, G^*). 
\]

Therefore, the overall optimization objective of textual model pretraining is a joint loss of \( \mathcal{L}_{\text{KL}} \) and \( \mathcal{L}_{\text{gloss2gloss}} \) as:

\[
\mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{KL}} + \mathcal{L}_{\text{gloss2gloss}}.
\]

3.2. Training CVT-SLR Framework

3.2.1 Training Pipeline

In the CVT-SLR framework, we sequentially concatenate the pretrained visual CNN module and the pretrained textual VAE module through a video-gloss adapter, as shown
in Figure 2. The pretrained ResNet18 model [17] is selected as the frame-wise feature extractor, which is trained on large-scale human action datasets. For the VAE module, although Gaussian noise is introduced in the reparameterization process in step 1, we do not add any noise after migrating the VAE to the SLR framework in step 2. In this way, the reparameterization module of the VAE degenerates into two common linear layers, i.e., $\epsilon$ is removed in Eq. (1).

Formally, given a sign video with $T$ RGB frames $\mathcal{F} = \{F_t\}_{t=1}^{T} \in \mathbb{R}^{T \times H \times W \times 3}$, the visual module, which is composed of a stack of 2D-CNN layers and a global average pooling layer, first extracts spatial features $v = \{v_t\}_{t=1}^{T} \in \mathbb{R}^{T \times d}$. Then the textual module further extracts temporal features $s = \{s_t\}_{t=1}^{T} \in \mathbb{R}^{T \times d}$. Finally, the alignment module utilizes CTC to compute the conditional probability of the gloss label sequence $p(\mathcal{G}|\mathcal{F})$ based on temporal features $s$, where $\mathcal{G} = \{G_t\}_{t=1}^{N}$ is a generated sign gloss sequence with $N$ glosses. A shared classifier is used between the visual module and the textual module to generate spatial and temporal probability, respectively. And the proposed contrastive cross-modal alignment algorithm further enhances the consistency across different modalities.

### 3.2.2 Video-Gloss Adapter

To maximize the preservation of both pretrained visual and textual knowledge, we use a video-gloss adapter to connect spatial-temporal features, as shown in Figure 2. The video-gloss adapter is simply implemented as a fully-connected MLP with two hidden layers. We take the CNN output probabilities as the input of the adapter where the final fully connected layer is initialized using the weights of the pretrained word embedding from the pretrained VAE [7].

Since both the input and output are in the same textual form during the textual model pretraining in step 1, the VAE plays an implicit cross-modal alignment role after being transferred to the CVT-SLR framework in step 2, which is close to the effect of consistency. Therefore, the video-gloss adapter is essentially a mediator for activating the consistency between the spatial and temporal modalities.

### 3.2.3 CTC Alignment

Due to the spatial-temporal nature of sign videos, glosses have a one-to-many mapping to video frames but share the same ordering. Unlike spoken language texts, the continuous gloss annotations are chronologically consistent with the sign videos. Concretely, for an input sign video $\mathcal{F}$ and the corresponding generated gloss sequence $\mathcal{G}$, we use CTC to compute $p(\mathcal{G}|\mathcal{F})$ by marginalizing over all possible $\mathcal{F}$ to $\mathcal{G}$ alignments:

$$p(\mathcal{G}|\mathcal{F}) = \sum_{\pi \in B} p(\pi|\mathcal{F}), \quad (7)$$

where $\pi$ denotes a path and $B$ is the set of all viable paths corresponding to $\mathcal{G}$. The probability $p(\pi|\mathcal{F})$ is computed by the textual module. The CTC loss is then formulated as:

$$\mathcal{L}_{\text{ctc}} = -\log p(\mathcal{G}^*|\mathcal{F}), \quad (8)$$

where $\mathcal{G}^*$ is the ground truth gloss sequence.

### 3.2.4 Contrastive Alignment

Aligning both visual and textual modalities help improve SLR [28]. To enforce alignment constraints across modalities, we propose a contrastive alignment loss. In addition, shared classifiers are also used as an improvement technique between visual and textual features [15].
<table>
<thead>
<tr>
<th>Groups</th>
<th>Models</th>
<th>Dev (%)</th>
<th>Test (%)</th>
<th>Cues</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>DEL/INS</td>
<td>WER</td>
<td>DEL/INS</td>
</tr>
<tr>
<td>Group 1</td>
<td>SubUNet [9]</td>
<td>14.6/4.0</td>
<td>40.8</td>
<td>14.3/4.0</td>
</tr>
<tr>
<td></td>
<td>Staged-Opt [9]</td>
<td>13.7/7.3</td>
<td>39.4</td>
<td>12.7/7.5</td>
</tr>
<tr>
<td></td>
<td>Align-iOpt [33]</td>
<td>12.6/2.6</td>
<td>37.1</td>
<td>13.0/2.5</td>
</tr>
<tr>
<td></td>
<td>DPD+TEM [47]</td>
<td>9.5/3.2</td>
<td>35.6</td>
<td>9.3/3.1</td>
</tr>
<tr>
<td></td>
<td>SFL [29]</td>
<td>-</td>
<td>23.7</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>DNF [11]</td>
<td>7.8/3.5</td>
<td>23.8</td>
<td>7.8/3.4</td>
</tr>
<tr>
<td></td>
<td>VAC [28]</td>
<td>7.9/2.5</td>
<td>21.2</td>
<td>8.4/2.6</td>
</tr>
<tr>
<td></td>
<td>CMA [32]</td>
<td>7.3/2.7</td>
<td>21.3</td>
<td>7.3/2.4</td>
</tr>
<tr>
<td></td>
<td>SFL [29]</td>
<td>-</td>
<td>21.9</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>VL-SLT [7]</td>
<td>7.9/2.5</td>
<td>21.2</td>
<td>8.4/2.6</td>
</tr>
<tr>
<td></td>
<td>SMKD [15]</td>
<td>7.3/2.7</td>
<td>21.3</td>
<td>7.3/2.4</td>
</tr>
<tr>
<td>Group 2</td>
<td>DNF [11]</td>
<td>7.3/3.3</td>
<td>23.1</td>
<td>6.7/3.3</td>
</tr>
<tr>
<td></td>
<td>STMC [48]</td>
<td>7.7/3.4</td>
<td>21.1</td>
<td>7.4/2.6</td>
</tr>
<tr>
<td></td>
<td>C2SLR [50]</td>
<td>-</td>
<td>20.5</td>
<td>-</td>
</tr>
<tr>
<td>Group 3</td>
<td>Ours$_1$ (w/o VAE+Contra)</td>
<td>7.1/3.0</td>
<td>21.1</td>
<td>7.3/2.9</td>
</tr>
<tr>
<td></td>
<td>Ours$_2$ (w/ VAE)</td>
<td>6.5/2.4</td>
<td>20.2</td>
<td>6.3/2.2</td>
</tr>
<tr>
<td></td>
<td>Ours$_3$ (w/ Contra)</td>
<td>6.7/2.7</td>
<td>20.4</td>
<td>6.4/2.5</td>
</tr>
<tr>
<td></td>
<td>Ours$_4$ (w/ VAE+Contra)</td>
<td>6.4/2.6</td>
<td><strong>19.8</strong></td>
<td>6.1/2.3</td>
</tr>
</tbody>
</table>

Table 1. Performance comparison (%) on PHOENIX-14 dataset. DEL/INS: deletion error and insertion error. The best results and SOTA baseline for each group are marked as **bold** and *underlined*.

In previous SLR works, cross-modal alignment only focuses on positive samples [15, 48]. Inspired by contrastive learning [5, 16, 45], we construct both positive and negative samples in the same mini-batch and implement a contrastive cross-modal alignment method to ensure that similar features are closer while different features are farther apart. Given that the normalized spatial features from CNN as $S_{\text{logits}} \in \mathbb{R}^{B \times T \times d}$, and the normalized temporal features from VAE as $V_{\text{logits}} \in \mathbb{R}^{B \times T \times d}$, where $B$ denotes the number of samples. Then we compute pair matrices firstly as:

$$S^{\text{2V}}_{\text{pair}} = S_{\text{logits}} \times V_{\text{logits}}^T,$$

$$V^{\text{2S}}_{\text{pair}} = V_{\text{logits}} \times S_{\text{logits}}^T.$$  

Taking $S^{\text{2V}}_{\text{pair}}$ or $V^{\text{2S}}_{\text{pair}}$ as the input matrix denoted as $\mathcal{P}$ as an example, $\mathcal{P}[i, j](0 \leq i, j < B)$ represents the similarity value between the visual feature of the $i$-th batch and the textual feature of $j$-th batch. The visual and textual features from the same input instances are positive samples, and the features from other different input instances are negative samples. Hence, in the matrix $\mathcal{P}$, the diagonal similarity values are from positive sample pairs, and the rest values are from negative sample pairs. To make the loss computation differentiable and easy, we convert the computation of positive and negative pairs into a binary classification task [34]. The value $i$ corresponding to the $i$-th row in the matrix $\mathcal{P}$ is a positive label. Therefore, the corresponding labels of $\mathcal{P}$ are $\text{Labels} = \{0, 1, \cdots, i, \cdots, B\}$, then the contrastive alignment loss related to $S^{\text{2V}}_{\text{pair}}$ as:

$$L_{\text{align}} = \text{CrossEntropy}(\text{Softmax}(S^{\text{2V}}_{\text{pair}}), \text{Labels}).$$

Similarly, the $V^{\text{2S}}_{\text{pair}}$ temporal-to-spatial alignment loss as:

$$L_{\text{align}} = \text{CrossEntropy}(\text{Softmax}(V^{\text{2S}}_{\text{pair}}), \text{Labels}).$$

Therefore, the complete contrastive alignment loss as:

$$L_{\text{align}} = \frac{1}{2}(L_{\text{align}} + L_{\text{align}}).$$

Overall, the objective of CVT-SLR is jointly optimized CTC loss $L_{\text{ctc}}$ and contrastive alignment loss $L_{\text{align}}$ as:

$$L_{\text{CTV-SLR}} = L_{\text{ctc}} + L_{\text{align}}.$$  

4. Experiments

4.1. Settings

Datasets. We use PHOENIX-2014 [23] and PHOENIX-2014T [2] as datasets, where RGB videos and their corresponding annotations are provided. PHOENIX-2014 is a German SLR dataset about weather forecasts with a vocabulary size of 1,081, which is divided into three parts: 5,672
<table>
<thead>
<tr>
<th>Groups</th>
<th>Models</th>
<th>WER Dev(%)</th>
<th>Test(%)</th>
<th>Cues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group 1</td>
<td>SFL [29]</td>
<td>25.1</td>
<td>26.1</td>
<td>video</td>
</tr>
<tr>
<td></td>
<td>CNN+LSTM+HMM [22]</td>
<td>24.5</td>
<td>26.5</td>
<td>video</td>
</tr>
<tr>
<td></td>
<td>SLT [3]</td>
<td>24.9</td>
<td>24.6</td>
<td>video</td>
</tr>
<tr>
<td></td>
<td>FCN [8]</td>
<td>23.3</td>
<td>25.1</td>
<td>video</td>
</tr>
<tr>
<td></td>
<td>SMKD [15]</td>
<td>20.8</td>
<td>22.4</td>
<td>video</td>
</tr>
<tr>
<td>Group 2</td>
<td>CNN+LSTM+HMM [22]</td>
<td>24.5</td>
<td>25.4</td>
<td>video+mouth</td>
</tr>
<tr>
<td></td>
<td>CNN+LSTM+HMM [22]</td>
<td>22.1</td>
<td>24.1</td>
<td>video+mouth+hand</td>
</tr>
<tr>
<td></td>
<td>SLT [3]</td>
<td>24.6</td>
<td>24.5</td>
<td>video+text</td>
</tr>
<tr>
<td></td>
<td>STMC [48]</td>
<td>19.6</td>
<td>21.0</td>
<td>video+hand+face+pose</td>
</tr>
<tr>
<td></td>
<td>C^2SLR [50]</td>
<td>20.2</td>
<td>20.4</td>
<td>video+keypoints</td>
</tr>
<tr>
<td>Group 3</td>
<td>Ours (w/o VAE+Contra)</td>
<td>21.8</td>
<td>22.0</td>
<td>video</td>
</tr>
<tr>
<td></td>
<td>Ours (w/ VAE)</td>
<td>20.1</td>
<td>20.4</td>
<td>video</td>
</tr>
<tr>
<td></td>
<td>Ours (w/ Contra)</td>
<td>21.0</td>
<td>21.5</td>
<td>video</td>
</tr>
<tr>
<td></td>
<td>Ours (w/ VAE+Contra)</td>
<td><strong>19.4</strong></td>
<td><strong>20.3</strong></td>
<td>video</td>
</tr>
</tbody>
</table>

Table 2. Performance comparison (%) on PHOENIX14T dataset. The best results and SOTA baseline for each group are marked as **bold** and *underlined*, respectively.

instances for training, 540 for validation, and 629 for testing, respectively. And PHOENIX-2014T is an extension of PHOENIX-2014 with a vocabulary size of 1,085, which is also divided into three parts: 7,096 instances for training, 519 for validation, and 642 for testing, respectively.

**Evaluation Metric.** Word error rate (WER) is the most widely-used metric to evaluate the SLR performance, which measures the number of necessary insertions, substitutions, and deletions in the recognized sentences when compared to the reference sentences. WER is an edit distance, indicating the least number of operations of substitutions (\#sub), insertions (\#ins), and deletions (\#del) to transform the predicted sentences into the reference sequences (\#reference):

\[
\text{WER} = \frac{\#\text{del} + \#\text{ins} + \#\text{sub}}{\#\text{reference}}.
\]

(15)

**Implementation Details.** The videos are resized to 256 x 256 and then cropped to 224 x 224. During SLR training, we use random crop and horizontal flips (50%) for data augmentation. During SLR testing, we only adopt center cropping. Beam Search is used for gloss generation. Our visual module has the same configuration as [15]. The VAE module contains two multi-head self-attention layers with 4 heads and two Bi-LSTM layers with 2 x 512 dimensional hidden states sequentially. Adam [20] optimizer is used to pretrain our VAE network with a batch size of 16, an initial learning rate of 1e-4. And AdamW [27] optimizer is used to train our SLR framework with a batch size of 8, a learning rate of 1e-4, and a weight decay of 1e-4. And 1 NVIDIA A100 80GB GPU is used.

**4.2. Main Results**

Table 1 and Table 2 show our comparisons with two-group different baselines, where Group 1 includes single-cue baselines and Group 2 includes multi-cue baselines. Ours\textsubscript{number} denote our trained CVT-SLR framework, which have only two variables, “VAE” and “Contra”. Note that “w/o (w/) VAE” denotes the VAE module in CVT-SLR framework without (with) transferred pretrained parameters, and “w/o (w/) Contra” denotes training CVT-SLR framework without (with) contrastive alignment loss.

**Evaluation on PHOENIX-14.** Table 1 shows our methods compared to other methods on PHOENIX-14. The optimal WER scores of our single-cue CVT-SLR (Ours\textsubscript{4}) on the development set and testing set are 19.8% and 20.1%, respectively. And Ours\textsubscript{4} achieves the best performance among video-based only/single-cue methods in Group 1 (Ours\textsubscript{4} vs. SMKD). Furthermore, while no extra cues are used, our CVT-SLR achieves the best result among the models trained with extra cues (i.e., multi-cue methods) in Group 2 (Ours\textsubscript{4} vs. C^2SLR).

From within-group comparisons in Group 3, we find that the case, Ours\textsubscript{1}, which neither migrates the pretrained VAE parameters nor uses the contrastive alignment constraint, performs the worst. Nevertheless, relative to other groups, it still performs well. Based on Ours\textsubscript{1}, by only configuring the pretrained VAE parameters (Ours\textsubscript{2}) or the contrastive alignment constraints (Ours\textsubscript{3}), the CVT-SLR framework can also gain gains, and the pretrained VAE brings more obvious gains (+0.9%/+1.1% vs. +0.7%/+0.7%). Overall, Group 3 demonstrates that both the prior language knowledge from the pretrained textual module and the contrastive constraints through cross-modal alignment play important enhancement roles for our SLR framework.

**Evaluation on PHOENIX-14T.** In Table 2, we evaluate the CVT-SLR on PHOENIX-14T. We can see that our optimal model (Ours\textsubscript{4}) in this dataset also achieves the best per-
Table 3. Ablation study of pretrained modules on PHOENIX-14 (Pre-Visual: pretrained visual module, Pre-Language: pretrained textual module). The best results are bolded.

<table>
<thead>
<tr>
<th>#</th>
<th>Pre-Visual</th>
<th>Pre-Textual</th>
<th>WER</th>
<th>Dev(%)</th>
<th>Test(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>✓</td>
<td>✓</td>
<td>89.2</td>
<td>88.5</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>30.8</td>
<td>31.4</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>✓</td>
<td>✓</td>
<td>20.4</td>
<td>20.7</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>✓</td>
<td>✓</td>
<td>19.8</td>
<td>20.1</td>
<td></td>
</tr>
</tbody>
</table>

Table 4. Ablation study of varied weights of the CTC loss and the alignment loss on PHOENIX-14. The best results are bolded.

<table>
<thead>
<tr>
<th>#</th>
<th>CTC Weight</th>
<th>Align Weight</th>
<th>WER</th>
<th>Dev(%)</th>
<th>Test(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0</td>
<td>0</td>
<td>20.2</td>
<td>20.3</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1.0</td>
<td>1.0</td>
<td>20.0</td>
<td>20.2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1.0</td>
<td>10.0</td>
<td>19.8</td>
<td>20.1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1.0</td>
<td>100.0</td>
<td>20.1</td>
<td>20.6</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1.0</td>
<td>1000.0</td>
<td>20.5</td>
<td>20.9</td>
<td></td>
</tr>
</tbody>
</table>

4.3. Ablation Study

We study the effectiveness of the proposed pretrained modules and alignment loss weights in our framework.

4.3.1 Pretrained Modules

To illustrate the importance of the pretrained modules in our CVT-SLR framework, we conduct the ablation studies of both the pretrained visual module (CNN module) and the pretrained textual module (VAE module) on PHOENIX-14, as verified in Table 3.

Overall, both pretrained modules are useful. Comparing #1 and #3, we find that pretrained visual features account for a larger determinant of performance, which is consistent with our reason for retaining the pretrained visual knowledge. Surprisingly, comparing #1 and #2, we find that without pretrained parameters of the visual module, i.e., using only the pretrained textual module, our CVT-SLR performance is also greatly improved. Notably, this is a new observation that has not been explored in previous SLR works extensively. These comparisons demonstrate that the pretrained VAE module has learned sufficient language knowledge to improve the SLR task. Comparing #3 with #4 further confirms this point.

To verify the effectiveness of the pretrained modules for the training convergence speed, training loss curves corresponding to Table 3 are shown in Figure 3. Specifically, comparing #1 and #2 (or #3 and #4), configuring a pretrained textual module can promote faster convergence and reach a stable state earlier. Similarly, comparing #1 and #3 (or #2 and #4), a pretrained visual module can accelerate convergence speed compared to the pretrained textual module. As can be seen from the loss curves, the degrees of convergence are positively correlated with the final evaluation performances of WER scores.

4.3.2 Alignment Loss Weights

Training CVT-SLR to recognize glosses is optimized jointly by CTC loss and contrastive alignment loss. To explore the effect of loss weights, we conduct an ablation study of varied weights of the CTC loss and the proposed contrastive alignment loss, as shown in Table 4. For simplicity, we set the weights of CTC loss to a fixed value of 1.0. In Table 4, #1, without contrastive alignment loss (weight=0), performs worse than the models with contrastive alignment in an optimal configuration (#2, #3), which also illustrates the gain effect of the proposed contrastive alignment. As shown in #2-#5 in Table 4, we keep increasing the weight of the alignment loss from 1.0 to 1000.0. As the weight of the contrastive loss increases, the performance starts to rise, reach-
4.4. Qualitative Analysis

We randomly select four examples from the test set of PHOENIX-14 for qualitative analysis. Each example displays an alignment matrix, a series of saliency maps, and generated glosses in Figure 4. An alignment matrix shows alignment relationships between visual and textual features, while a saliency map shows the highest activations.

Visualization on Cross-modal Alignment Matrices. From these four alignment matrices in Figure 4, we can see that the reference and inferred glosses are almost the same with quite low WER scores, and the performances are relatively perfect. Hence, the highlighted areas of the alignment matrices are concentrated near the diagonal, which means that our contrastive alignment algorithm plays a crucial role in aligning visual and textual features.

Visualization on Saliency Maps. We visualize the key parts of the sign video frames in focus by using Grad-CAM [36] as shown in Figure 4. The results show that, for well-performing inferences, significant spatial features are mainly focused on the face and hand regions in the saliency maps. This observation is consistent with the sign linguistic knowledge [3, 19], where sign language conveys information mainly relying on hand shapes and facial expressions.

5. Conclusion

In this work, we propose a novel contrastive visual-textual transformation framework for SLR, called CVT-SLR, which further alleviates insufficient training by fully exploring the pretrained knowledge of both the visual and language modalities. We introduce prior language knowledge into the single-cue CVT-SLR for the first time via a VAE-based pretrained textual module. Furthermore, we propose new methods for cross-modal consistency constraints. One method takes advantage of the properties of the autoencoder to implicitly align the visual and textual modalities. While another method introduces a contrastive cross-modal alignment for explicit consistency. To prove the proposed CVT-SLR framework, extensive quantitative experiments and qualitative analysis are conducted, showing that our single-cue CVT-SLR not only outperforms single-cue baselines by a large margin but also surpasses SOTA multi-cue methods.

Overall, we hope our CVT-SLR can inspire the SLR community to mine pretrained modules from new aspects and design a more efficient SLR framework for practical usage. In future work, we plan to introduce a large-scale pretrained language model to guide textual module learning and further explore the potential of prior contextual linguistic knowledge for SLR tasks.

Acknowledgement

We thank the anonymous reviewers for their constructive and helpful reviews. This work was supported by the National Key R&D Program of China (Project No. 2022ZD0115100), the National Natural Science Foundation of China (Project No. U21A20427), the Competitive Research Fund (Project No. WU2022A009) from the Westlake Center for Synthetic Biology and Integrated Bioengineering.


