Contrastive Learning for Sign Language Recognition and Translation

Shiwei Gan, Yafeng Yin*, Zhiwei Jiang, Kang Xia, Lei Xie and Sanglu Lu
State Key Laboratory for Novel Software Technology, Nanjing University, China
sw@smail.nju.edu.cn, {yafeng, jzw}@nju.edu.cn, xiakang@smail.nju.edu.cn, {lxie, sanglu}@nju.edu.cn

Abstract
There are two problems that widely exist in current end-to-end sign language processing architecture. One is the CTC spike phenomenon which weakens the visual representational ability in Continuous Sign Language Recognition (CSLR). The other one is the exposure bias problem which leads to the accumulation of translation errors during inference in Sign Language Translation (SLT). In this paper, we tackle these issues by introducing contrast learning, aiming to enhance both visual-level feature representation and semantic-level error tolerance. Specifically, to alleviate CTC spike phenomenon and enhance visual-level representation, we design a visual contrastive loss by minimizing visual feature distance between different augmented samples of frames in one sign video, so that the model can further explore features by utilizing numerous unlabeled frames in an unsupervised way. To alleviate exposure bias problem and improve semantic-level error tolerance, we design a semantic contrastive loss by re-inputting the predicted sentence into semantic module and comparing features of ground-truth sequence and predicted sequence, for exposing model to its own mistakes. Besides, we propose two new metrics, i.e., Blank Rate and Consecutive Wrong Word Rate to directly reflect our improvement on the two problems. Extensive experimental results on current sign language datasets demonstrate the effectiveness of our approach, which achieves state-of-the-art performance.

1 Introduction
Sign language research is committed to transferring sign video into natural language. Considering the unique grammatical rules and linguistic characteristics of sign language, current researchers mainly focus on two tasks: Continuous Sign Language Recognition (CSLR) and Sign Language Translation (SLT). CSLR aims at recognizing continuous signs as the corresponding gloss sequence, and it is a weakly labeled recognition task (i.e., source and target sequences have unequal lengths but the same order). Differently, SLT aims at translating sign language into spoken language, and it is a sequence-to-sequence (seq2seq) translation task (i.e., source and target sequences have unequal lengths and different orders). Due to the difference between these two tasks, researchers often adopt Connectionist Temporal Classification (CTC) loss [Graves et al., 2006] to address CSLR [Zhang et al., 2019] while adopting seq2seq architecture [Sutskever et al., 2014] to address SLT [Jin and Zhao, 2021].

Most current end-to-end sign language processing models can be simplified as Figure 1, and it consists of a visual module which mainly includes 1D, 2D, 3D CNN layers for mapping sign video into visual features, a temporal module which mainly includes RNN-like/transformer layers for exploring global context features. Besides, a unique CTC module is adopted to get the gloss sequence for CSLR, or a unique semantic module which mainly is RNN-like/transformer decoder is adopted to get the spoken sentence for SLT. However, there are two widely existing problems in this architecture: (1) For CSLR, CTC loss will bring the spike phenomenon [Graves et al., 2006], i.e., the model tends to classify frames as ‘blank’ tokens rather than meaningful gloss tokens, leading to the fact that only a few frames are useful for training visual module. Consequently, the representational ability of visual module is not fully exploited and further visual module may not provide efficient features during testing. (2) For SLT, teacher forcing (often used to train seq2seq model) leads to exposure bias which causes distribu-
tional shifts during testing [He et al., 2021], i.e., the model in training stage predicts the next word based on ground-truth words, but the model in testing stage needs to make the next prediction based on previous words generated by itself. Thus the probability distribution generated by model in testing will gradually deviate from that in training, which leads to performance degradation. Besides, video-gloss or video-sentence pairs in current sign language datasets are scarce to fully train each module in above architecture, which further exacerbates these two problems.

To tackle CTC spike phenomenon which may weaken visual representational ability, Hao et al. and Min et al. explicitly mentioned this issue and added extra constraints through knowledge distillation to enhance visual representation. Most other work implicitly tackled this issue by manually enhancing visual representational ability. They intended to design various visual modules by injecting extra clues, including local areas [Zhou et al., 2020], skeletons [Gan et al., 2021] and depth images [Tang et al., 2021]. As for exposure bias caused by teacher forcing, there are many attempts in Natural Language Processing (NLP) area while little work has paid attention in SLT. Besides, when considering data scarcity problem in CSLR and SLT, methods like back-translation [Zhou et al., 2021a], cross modality augmentation [Pu et al., 2020] and pre-trained models [Chen et al., 2022] had been proposed.

In this paper, we provide a simple unified framework by adopting contrastive learning to alleviate CTC spike phenomenon and exposure bias problem. The main intuitions are: (1) Although video-gloss or video-sentence pairs are limited and labeling each frame for fully training visual module is unrealistic, frames provided by datasets are rather a lot. Thus we design a visual contrastive loss to minimize feature distance of different augmented samples of frames in one sign video. Then the visual module can improve its representation by utilizing numerous unlabeled frames in an unsupervised way. (2) Distributional shift caused by teacher forcing can be alleviated by exposing model to its own mistakes. Thus we design a semantic contrastive loss by comparing semantic features of different inputs (ground-truth sequence and the predicted sequence). We make the following contributions:

- To alleviate CTC spike phenomenon in CSLR and enhance visual-level feature representation, we propose a visual contrastive loss to explore features by utilizing numerous unlabeled frames.
- To alleviate exposure bias in SLT and improve semantic-level error tolerance, we propose a semantic contrastive loss to expose model to its own translation mistakes by re-inputing predicted sequence into semantic module.
- Two new metrics, i.e., Blank Rate and Consecutive Wrong Word Rate are proposed to reflect our improvement on two problems.
- The extensive experiments demonstrate the superiority of proposed losses and our framework achieves state-of-the-art performance on CSLR and SLT.

2 Related Work

In this section, we review the related work on end-to-end CSLR, end-to-end SLT, and contrastive learning.

CSLR and SLT. CSLR aims at recognizing a sequence of signs to corresponding gloss sequence [Koller et al., 2017] while SLT aims to translate sign language to spoken language [Camgoz et al., 2018]. The unique grammatical rules of sign language lead to some differences between frameworks of two tasks. Nevertheless, both two frameworks contain a visual module for encoding sign videos, and a temporal module for exploring context features. In regard to the decoder module, CSLR models often adopt CTC module [Koller et al., 2019] which requires that source and target sequences have the same order. Differently, SLT [Camgoz et al., 2018] is limited to seq2seq architecture and often adopts a RNN-like/transformer decoder. However, it is worth mentioning that CSLR is not limited to CTC decoder and there is also some work choosing seq2seq architecture for CSLR [Huang et al., 2018]. Recently, Camgoz et al. [Camgoz et al., 2020b] combined CTC decoder and transformer decoder in one network, to achieve joint CSLR and SLT.

To alleviate CTC spike phenomenon explicitly or implicitly, existing work focused on designing various visual modules to enhance visual representation. Extra clues including face, hands [Camgoz et al., 2020a], skeletons [Gan et al., 2021] and depth images [Tang et al., 2021] were added in visual encoder. Besides, some work turned its attention to obtaining more training samples by back translation [Zhou et al., 2021a], cross modality augmentation [Pu et al., 2020] and pre-trained models [Chen et al., 2022]. While others were committed to adopting iterative training [Cui et al., 2017] or knowledge distillation [Hao et al., 2021; Min et al., 2021] to enhance representational ability. As for exposure bias problem [Feng et al., 2021], in NLP field, researchers utilized techniques from generative adversarial networks [Subramanian et al., 2017] and reinforcement learning [Shi et al., 2018], then proposed many training algorithms to replace teacher forcing training. However, in SLT, little work had paid attention to exposure bias and only beam search [Graves, 2012] which is the most related algorithm has been used in SLT [Camgoz et al., 2020a]. Besides, unlike NLP tasks that have millions of sentence pairs, SLT datasets only have thousands of sentences, which further exacerbates exposure bias problem. In this paper, to tackle CTC spike phenomenon and exposure bias problem, we provide a simple unified framework for CSLR and SLT, and introduce contrastive learning at visual and semantic levels.

Contrastive learning. As one of the popular unsupervised learning algorithms, contrastive learning [Hadsell et al., 2006] is trained by minimizing the feature distance of different augmented views of the same input (‘positive pairs’), and maximizing feature distance of views from different inputs (‘negative pairs’). Methods like MOCO [He et al., 2020], SimCLR [Chen et al., 2020] usually adopted Siamese network [Chen and He, 2021], i.e., a weight-sharing network applied on two or more inputs, and they heavily relied on negative sample pairs to prevent collapsing problem (i.e., all outputs of network ‘collapse’ to a constant).
However, it may be hard to construct negative samples for sign videos. The reason can be included as: (i) In frame level, frames corresponding to transition actions or blur motions have high similarity and may appear in most sign videos, thus not all frames in one video are meaningful and many of them can not be labeled as specific classes. (ii) In video level, label sequences of different sign videos may overlap thus sign videos are not exactly ‘negative’ to each other (e.g., label ‘bab’ and label ‘caa’ have an overlap of token ‘a’). Recently, BYOL [Grill et al., 2020] proved that negative sample pairs are not necessary by adopting a momentum encoder. Moreover, SimSiam [Chen and He, 2021] further deleted the momentum encoder while using a stop-gradient operation. That is to say, we can design contrastive losses by stop-gradient operation without constructing negative pairs.

3 Method

In this section, we first give an overall look of our framework. Then, we simply describe CTC spike phenomenon in CSLR and propose a visual contrastive loss which enhances visual representation. After that, we describe the exposure bias problem in SLT and propose a semantic contrastive loss which enhances semantic-level error tolerance. Finally, we design joint loss to train our model for both CSLR and SLT.

3.1 Overall Framework

For a sign video \( f = \{f_i\}_{i=1}^{n} \) with \( n \) frames, the goal of CSLR is to learn a mapping from \( f \) to gloss sequence \( g = \{g_i\}_{i=1}^{m} \) with \( m \) glosses, and the goal of SLT is to translate \( f \) to spoken sentence \( w = \{w_i\}_{i=1}^{t} \) with \( t \) words. As shown in Figure 1, the process of CSLR and SLT can be simplified as follows.

Visual feature extraction. The visual module \( V \mathcal{E} \) processes frames to get visual features \( v = V \mathcal{E}(f) \) with \( m \) vectors, where \( v = \{v_i \in \mathbb{R}^{d_v}\}_{i=1}^{m} \) (\( m \leq n \)) and \( d_v \) is dimension of \( v_i \).

Temporal feature extraction. The temporal module \( T \mathcal{E} \) extracts the global context information from \( v \) to get \( u = T \mathcal{E}(v) \), where \( u = \{u_i \in \mathbb{R}^{d_t}\}_{i=1}^{m} \) and \( d_t \) is dimension of \( u_i \).

Recognition and translation. For CSLR, a fully connected layer with weight \( W_r \) and bias \( b_r \), and a softmax layer \( S \mathcal{M} \) are added to get the final probability matrix of recognized gloss sequence: \( \mathbf{R} = S \mathcal{M}(W_r u + b_r) \). For SLT, a RNN-like or transformer decoder \( S \mathcal{E} \) is needed to obtain the translation probability matrix of the \( i \)th step: \( \mathbf{Y}_i = S \mathcal{M}(W_y * g_{i-1} + b_y) \) where \( W_y, b_y \) are the weight and bias of the fully connected layer in semantic module.

3.2 Visual-Level Contrastive Learning for CTC Spike Phenomenon

CTC spike phenomenon. In CSLR, CTC loss tackles the temporal classification problem for unsegmented sign video, by maximizing probability of all possible paths which align frames with labeled gloss sequence \( g \) in probability matrix \( \mathbf{R} \). Ideally, it is necessary to label each frame with one specific class. However, considering the meaningless frames and same glosses occurring consecutively, CTC introduces ‘blank’ token and further defines a many-to-one map \( B \) which simply removes all repeated tokens and blanks from the path (e.g., in Figure 2, \( B(\text{PathB}) = B(\text{a} \rightarrow \text{a}) = \text{a} \)). Thus CTC loss can be formalized as follows:

\[
\mathcal{L}_{CTC}(\mathbf{R}, g) = - \sum_{\pi \in B^{-1}(g)} \log(p(\pi))
\]

where \( p(\pi) = \prod_{i=1}^{m} \mathbf{R}^{\pi_i}_{i} \), \( \pi \) denotes one of alignment paths (i.e., \( B(\pi) = g \)) and \( \mathbf{R}^{\pi_i}_{i} \) is the probability of the \( i \)th token \( \pi_i \).

However, CTC loss will lead to spike phenomenon. As shown in Figure 2, the model tends to classify frames as ‘blank’ tokens when it cannot confidently distinguish gloss boundaries in video, because predicting ‘blank’ tokens is a much safer choice for minimizing loss \( \mathcal{L}_{CTC} \) [Min et al., 2021]. For example, \( B(\text{PathA}) = B(\text{aaab}) = \text{ab} \) and \( B(\text{PathB}) = B(\text{a} \rightarrow \text{a}) = \text{a} \). Hence CTC ‘prefers’ \( \text{PathA} \) than \( \text{PathB} \), which causes that only a few frames contribute to final results. Due to that most frames are classified as ‘blank’ tokens during the training, the visual module is not fully trained and cannot provide effective sign features on the test set.

Visual-level contrastive learning. To tackle spike phenomenon, our intuitive thought is to enhance visual representational ability by training the visual module with more data. However, unlike image classification task (1.28 million labeled images used by ResNet) or NLP task (4.5 million sentence pairs used by Transformer), the data pairs (video-gloss pairs or video-sentence pairs) in current sign language tasks are rather limited, and increasing samples by labeling each frame manually to train visual module separately is unrealistic. But unlabeled frames provided by sign videos are
Next, as shown in Figure 3, a projection MLP head $\mathcal{PAO}_{vi}$ and a prediction MLP head $\mathcal{PRE}_{vi}$ are attached to map visual features to the space where the contrastive loss is applied.

After getting $p^i$ and $z^i$, our similarity function $S$ calculates negative cosine similarity of each $p^i$ and $z^i$ in frame level, as follows. Here, $\|\cdot\|_2$ is $\ell_2$-norm.

$$S(p^i, z^i) = \frac{-p^i \cdot z^i}{\|p^i\|_2 \cdot \|z^i\|_2}$$  \hspace{1cm} (6)

When using $S$ to calculate similarity of $p^i$ and $z^i$, we get can the averaged loss of all frames, as follows.

$$S(p^i, z^i) = \frac{1}{m} \sum_{i=1}^{m} S(p^i, z^i)$$  \hspace{1cm} (7)

Finally, following SimSiam [Chen and He, 2021], we further get $p^h=\mathcal{PRE}_{vi}(z^h)$ in a symmetrized way and define a symmetrized visual contrastive loss $L_{ViCo}$ as follows,

$$L_{ViCo} = \frac{S(p^*, \text{detach}(z^h)) + S(p^h, \text{detach}(z^*))}{2}$$  \hspace{1cm} (8)

where $\text{detach}(\cdot)$ function denotes a stop-gradient operation which means that $z^h$ and $\hat{z}^*$ are treated as a constant instead of a variable with gradient.

### 3.3 Semantic-Level Contrastive Learning for Exposure Bias

**Exposure bias.** In SLT, seq2seq architecture (more precisely auto-regressive architecture) is commonly trained via Maximum Likelihood Estimation (MLE). As shown in Figure 4, taking the RNN-like decoder as an example, the MLE-based training is optimized by maximizing the likelihood of the next word $y_{i}$ based on its previous ground-truth word $w_{i-1}$ (called teacher forcing) [He et al., 2021], as follows. Here, $\theta$ denotes the model parameters, $\theta^*$ denotes updated parameters, and $h_{i-1}$ is the $(i-1)th$ hidden state of semantic module and $u$ denotes the temporal features.

$$\theta^* = \arg \max_\theta \prod_{i=1}^{\ell} P_{\theta}(y_i|w_{i-1}, h_{i-1}, u)$$  \hspace{1cm} (9)

While during the inference stage, the model needs to predict $Y^*_i$ (i.e., probability of word $y_i$) based on prefix word $y_{i-1}$ generated by model itself.

$$Y^*_i = P_{\theta^*}(y_i|y_{i-1}, h_{i-1}, u)$$  \hspace{1cm} (10)

Due to the exposure to ground-truth data during training, the model may over-rely on right prefix words. While during inference, the wrong predicted word $y_{i-1}$ could exacerbate error propagation along the generated sequence, and the target probability distributions generated by the model will be incrementally distorted (called exposure bias). Exposure bias will lead to huge performance degradation especially when the sequence distribution of training samples is insufficient to cover the sequence distribution of testing samples.

**Semantic-level contrastive learning.** To tackle the exposure bias problem, the most straightforward solution is scheduled sampling [Bengio et al., 2015]. Scheduled sampling decides whether to use ground-truth word $w_{i-1}$ or predicted word $y_{i-1}$ with a probability of $\alpha$ at the $i$th step during training. In this way, the model can weaken the dependence on ground-truth words by exposing model to its own mistakes. However, it is non-trivial to apply scheduled sampling directly for transformer model, since that different from RNN-like decoder, transformer generates the $i$th word based on the whole prefix sequence $\{y_j\}_{j=0}^{i-1}$ instead of the last word $y_{i-1}$. Besides, scheduled sampling training is a sequential process along time thus hinders parallel training. In this paper, following the same idea of exposing the model to its own mistakes, we introduce semantic-level contrastive learning by comparing semantic features of different inputs (the ground-truth sequence input and the predicted sequence input) in semantic module (i.e., decoder module) to alleviate exposure...
bias. When compared with scheduled sampling, our method can be used for both RNN-like decoder and transformer decoder, and easily trained in parallel.

To construct the semantic contrastive loss, we first need to get translated sentence \( y \) based on ground-truth sentence \( w \). Specifically, for temporal features \( w = \{w_t\}_{t=1}^m \), the semantic module \( \mathcal{S} \) takes \( u \) and \( w \) as inputs to get semantic features (i.e., hidden states) \( h^w \),

\[
h^w = \mathcal{S}(u, w) \tag{11}\]

Then, a fully connected layer and a softmax function \( \mathcal{S} \) are adopted to get the possibility matrix \( \mathbf{Y} \) of translated sentence and further get the translated sentence \( y \) through an argmax operation,

\[
\begin{align}
\mathbf{Y} & = \mathcal{S}(W_y * h^w + b_y) \\
y & = \argmax_2 \mathbf{Y}
\end{align} \tag{12}\]

After that, as shown in Figure 5, we re-input the predicted sequence \( y \) into semantic module to get another semantic features \( h^y \), as follows. Here, re-inputing predicted sequence \( y \) can be treated as re-inputing augmented ground-truth \( w \), and can also be regarded as a form of scheduled sampling.

\[
h^y = \mathcal{S}(u, y) \tag{13}\]

Next, similar to visual module in Figure 3, a projection MLP head \( \mathcal{PR}_e \) and a prediction MLP head \( \mathcal{PR}_s \) are added to get \( \rho^w \) and \( \xi^w \). Besides, the similarity function \( S \) is adopted to calculate the distance between \( \rho^w \) and \( \xi^y \).

\[
\begin{align}
\rho^w & = \mathcal{PR}_e(h^w), \\
\rho^y & = \mathcal{PR}_s(h^y) \\
\xi^w & = \mathcal{PR}_s(h^w)
\end{align} \tag{14-15}\]

Finally, we also get \( \xi^y = \mathcal{PR}_s(h^y) \), and define a symmetric semantic contrastive loss \( \mathcal{L}_{\mathcal{Sc}Co} \) as follows,

\[
\mathcal{L}_{\mathcal{Sc}Co} = \frac{S(\xi^w, \text{detach}(\rho^w)) + S(\xi^y, \text{detach}(\rho^w))}{2} \tag{16}\]

### 3.4 Joint Loss for CSLR and SLT

To optimize our model with proposed contrastive losses, we need to infer our model \( \mathcal{M} \) twice based on different inputs. One contains soft augmented video \( f_s \) and labeled word sequence \( w \). The other one contains hard augmented video \( f^h \) and predicted word sequence \( y \), where \( y = \arg\max \mathbf{Y} \).

\[
\begin{align}
\mathbf{R}, \mathbf{Y}, (p^s, z^s), (\rho^w, \xi^w) & = \mathcal{M}(f^s, w) \tag{17} \\
\mathbf{R}', \mathbf{Y}', (p^h, z^h), (\rho^h, \xi^h) & = \mathcal{M}(f^h, y)
\end{align} \tag{18}\]

After getting \( (p^s, z^s, p^h, z^h, \xi^w, \rho^w, \rho^h, \xi^h) \), we can calculate contrastive losses \( \mathcal{L}_{\mathcal{Vi}Co}, \mathcal{L}_{\mathcal{Sc}Co} \) according Equation 8, Equation 16, respectively. Meanwhile, we get two recognition probability matrices \( \mathbf{R}, \mathbf{R}' \) and two translation probability matrices \( \mathbf{Y}, \mathbf{Y}' \). To achieve the goal of CSLR and SLT jointly, we employ CTC loss \( \mathcal{L}_{\mathcal{CT}C} \) and cross entropy loss \( \mathcal{L}_{\mathcal{CE}} \) to train our model, as follows,

\[
\begin{align}
\mathcal{L}_{\text{soft}} & = \mathcal{L}_{\mathcal{CT}C}(\mathbf{R}, g) + \mathcal{L}_{\mathcal{CE}}(\mathbf{Y}, w) \tag{19} \\
\mathcal{L}_{\text{hard}} & = \mathcal{L}_{\mathcal{CT}C}(\mathbf{R}', g) + \mathcal{L}_{\mathcal{CE}}(\mathbf{Y}', w) \tag{20}
\end{align}
\]

Finally, we get the joint loss \( \mathcal{L} \) with a balance weight \( \alpha \), where \( \alpha \) is a hyperparameter, as follows,

\[
\mathcal{L} = \mathcal{L}_{\text{soft}} + \mathcal{L}_{\text{hard}} + \alpha(\mathcal{L}_{\mathcal{Vi}Co} + \mathcal{L}_{\mathcal{Sc}Co}) \tag{21}\]
Training setting. To train our model, we use the following settings for CSLR and SLT. We adopt Adam optimizer with a weight decay of 0.0001 to train our model for 70 epochs on 2 GeForce RTX 3090 GPUs. The initial learning rate is 0.0001 with a decay factor of 0.5 and the batch size is set to 6.

Evaluation setting. To evaluate our model, we adopt Word Error Rate (WER) as CSLR performance metric, while adopting ROUGE-L F1-Score [Lin, 2004] and BLEU-1,2,3,4 [Papineni et al., 2002] as SLT performance metrics. Those metrics have been often used to measure the quality of recognition and translation in the existing work.

To verify whether our method can effectively deal with CTC spike phenomenon and exposure bias problem, we propose two new metrics: Blank Rate (BR) and Continuous wrong Word Rate (CWWR). For CSLR (as the top part shown in Figure 6), suppose both model A and model B have the same WER (12.5%), while model A predicts fewer ‘blank’ tokens than model B. To measure how many ‘blank’ tokens are in recognized sequence, we introduce the BR metric. BR is an indicator that directly reflects the severity of spike phenomenon and can be formulated as follows,

$$BR = \frac{\#Blank}{\#TotalR}$$  \hspace{1cm} (22)

where $\#Blank$ denotes the number of ‘blank’ tokens in recognized sequence and $\#TotalR$ denotes the total number of tokens in recognized sequence.

For SLT (as the bottom part shown in Figure 6), model B is easier to make wrong predictions given wrong prefix words, compared with model A. Therefore we introduce the CWWR metric which measures how many consecutive wrong words (≥2) are in translated sequence. CWWR is an indicator that directly reflects how much the model relies on previously predicted words in translation and can be formulated as follows,

$$CWWR = \frac{\#CW}{\#TotalT}$$  \hspace{1cm} (23)

where $\#CW$ is number of consecutive wrong words and $\#TotalT$ is total number of words in translated sequence.

Effects of the beam width. Beam search is the most common algorithm used to alleviate exposure bias problem [Zhou et al., 2021a]. Here, to set a proper beam width for beam search, we test recognition performance and translation performance under different beam widths ranging from 1 to 9 and 10 to 50. According to Figure 7, we set beam width of CTC decoder to 6 for lower WER in CSLR, and set beam width to 2 for higher BLEU-4 in SLT.
Effects of weights $\alpha$. To balance the different losses in Equation 21, we test model performance under different $\alpha$ ranging from 0.1 to 2. According to Table 2, the model achieves the best performance when $\alpha$ is set to 1. Thus we set $\alpha$ to 1 in this paper.

Effects of training strategies. In the early training stage where word predictions may be noisy, the model directly trained with contrastive losses can be difficult to converge. To verify whether training strategies affect model performance, we compare different training strategies and show the ROUGE scores on Phoenix14T dev set. As shown in Figure 8, as for the fine-tuning strategy, the model converges much more stable in the early stage, while the model needs more epochs for fine-tuning. As for our training strategy, during 1 to 5 epochs of the training stage, the training process is unstable and the ROUGE oscillates around 0. But the model can quickly converges after 10 epochs. Thus the model is trained with our training strategy in this paper.

Effects of proposed contrastive losses. As shown in Table 3, results demonstrate that the proposed losses contribute to higher performance for CSLR and SLT. Specially, visual contrastive loss $L_{ViCo}$ improves the recognition performance (WER drops by 2.8%), which verifies that visual representation ability can be enhanced with proposed $L_{ViCo}$. The semantic contrastive loss $L_{SeCo}$ improves the translation performance (ROUGE increases by 5.65%) which indicates that the model can benefit by exposing model to its own mistakes.

To further verify whether the proposed losses can tackle CTC spike phenomenon and exposure bias problem, we show results on proposed BR and CWWR in Table 3. From this table, we get the following conclusions: (1) CTC spike phenomenon does widely exist in CSLR task, and up to 67.72% of tokens are predicted as ‘blank’. (2) ‘Blank’ token still exists with proposed visual contrastive loss, but $L_{ViCo}$ can effectively alleviate CTC spike phenomenon, i.e., BR drops by 10.25% when adopting $L_{ViCo}$. (3) As for translation, exposure bias problem also exists with proposed $L_{SeCo}$. However, CWWR drops by 3.88% when $L_{SeCo}$ is adopted, which verifies the effectiveness of semantic contrastive loss.

Explore random masking in visual level. Inspired by MAE [He et al., 2022], we add random masking in hard augmentation $H_A$ to further explore the possible improvement of visual representation ability. As shown in Figure 9, as the mask rate increases, BR decreases and CSLR performance further improves (i.e., WER decreases). However, when mask rate further increases, it is difficult for the visual module to capture valid features through visual contrastive loss since that sign-related features could be masked.

Explore random padding in semantic level. Semantic module benefits from exposing possible errors during training. However, as the model converges, the predicted sentence $y$ tends to be fixed (i.e., almost unchanged), which leads to that the model could not generalize well. Thus we further replace $y$ with $y_{pad}$ to manually add errors in re-input sequence $y$, where we randomly replace words in $y$ with ‘PAD’ tokens to get $y_{pad}$. As shown in Figure 10, as the random padding rate increases, translation performance increases first and then decreases, which demonstrates that manually adding a small proportion of random errors can enhance the generalization ability of semantic module. But when padding rate further increases, the distribution of $y_{pad}$ will deviate far from that of testing sequence, leading to performance degradation.

Qualitative Analysis. To verify whether our model can alleviate CTC spike phenomenon and exposure bias problem, we show a recognized example and translated example in Figure 11. Recognized results and translated results indicate that the proposed contrastive losses can effectively alleviate CTC spike phenomenon and exposure bias problem, further improve model performance with lower BR and CWWR.

4.4 Comparisons

Evaluation on Phoenix14T dataset. As shown in Table 4 and 6, we compare our model with existing models on both CSLR performance and SLT performance. We provide both the performance on validation set (i.e., “DEV”) and test set (i.e., “TEST”). On CSLR performance, our model achieves a comparable performance (21.8% WER on test set) with the state-of-the-art (SOTA) approaches which inject extra clues
Table 4: Comparison with SLT performance on Phoenix14T (F: face, M: mouth, H: hands, S: skeleton, P: pretraining on other datasets).

Table 5: Comparison of SLT performance on CSL-daily dataset.

Table 6: Comparison of CSLR performance on Phoenix14T dataset.

Table 7: Comparison of CSLR performance on CSL-daily dataset.

like local areas, skeletons or pre-trained models. Furthermore, our model improves the CSLR performance and outperforms the SOTA model on test set, i.e., 20.1% WER, when adopting random masking and random padding. On SLT performance, our model still outperforms current models.

Evaluation on CSL-daily dataset. We also provide CSLR and SLT performance comparisons with other methods on CSL-daily dataset. As shown in Table 5 and Table 7. The WER of our model on dev and test set is 26.0% and 25.3%, respectively, and our model outperforms existing models by a large margin on CSLR performance. Furthermore, the BLEU4 of our model on the dev, test set is 21.79%, 21.81%, which outperforms current models.

Evaluation on Phoenix14 dataset. We compare our model with existing models of CSLR performance on Phoenix14 dataset. As shown in Table 8, our model which does not use any extra clues or pre-trained models, achieves SOTA results compared with models trained with extra clues or pre-trained models.

5 Conclusion

In this paper, we introduce contrastive learning to tackle two problems that widely exist in sign language tasks: CTC spike phenomenon in CSLR task and exposure bias in SLT task. Specifically, to tackle CTC spike phenomenon, we design a visual contrastive loss to enhance visual module by learning features from numerous unlabeled frames. To tackle exposure bias problem, we design a semantic contrastive loss to enhance semantic module by exposing the model to its own mistakes. Besides, we propose two new metrics i.e., Blank Rate and Consecutive Wrong Word Rate to directly reflect our improvement on two problems. The experimental results on public sign language datasets demonstrate the effectiveness of our method.

Acknowledgments

This work is supported by Jiangsu Provincial Key Research and Development Program under Grant BE2020001-4; National Natural Science Foundation of China under Grant Nos. 62172208, 62272216, 61832008. This work is partially supported by Collaborative Innovation Center of Novel Software Technology and Industrialization.
References


[He et al., 2022] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022.


