Efficient Sign Language Translation with a Curriculum-based Non-autoregressive Decoder

Pei Yu[1,2], **Liang Zhang**[1,2], **Biao Fu**[1,2], **Yidong Chen**[1,2]
[1]School of Informatics, Xiamen University, China
[2]Key Laboratory of Digital Protection and Intelligent Processing of Intangible Cultural Heritage of Fujian and Taiwan (Xiamen University), Ministry of Culture and Tourism, China
yupei@stu.xmu.edu.cn, ydchen@xmu.edu.cn

Abstract

Most existing studies on Sign Language Translation (SLT) employ AutoRegressive Decoding Mechanism (AR-DM) to generate target sentences. However, the main disadvantage of the AR-DM is high inference latency. To address this problem, we introduce Non-AutoRegressive Decoding Mechanism (NAR-DM) into SLT, which generates the whole sentence at once. Meanwhile, to improve its decoding ability, we integrate the advantages of curriculum learning and NAR-DM and propose a **C**urriculum-based **N**AR **D**ecoder (**CND**). Specifically, the lower layers of the CND are expected to predict simple tokens that could be predicted correctly using source-side information solely. Meanwhile, the upper layers could predict complex tokens based on the lower layers' predictions. Therefore, our CND significantly reduces the model's inference latency while maintaining its competitive performance. Moreover, to further boost the performance of our CND, we propose a mutual learning framework, containing two decoders, i.e., an AR decoder and our CND. We jointly train the two decoders and minimize the KL divergence between their outputs, which enables our CND to learn the forward sequential knowledge from the strengthened AR decoder. Experimental results on PHOENIX2014T and CSL-Daily demonstrate that our model consistently outperforms all competitive baselines and achieves $7.92/8.02\times$ speedup compared to the AR SLT model respectively. Our source code is available at https://github.com/yp20000921/CND.

1 Introduction

Sign language is the communication language of the deaf community. Sign language translation aims to transform sign language videos into spoken language sentences, which can greatly facilitate communication between hearing and deaf people. Therefore, sign language translation has attracted increasing attention [Camgoz *et al.*, 2020; Zhou *et al.*, 2021b; Cao *et al.*, 2022; Chen *et al.*, 2022a; Chen *et al.*, 2022b].

Model	Iter	BLEU4	Speedup
AR [Vaswani *et al.*, 2017]	T_y	23.24	1.00×
CMLMC [Huang *et al.*, 2021]	4	22.67	2.47×
GLAT [Qian *et al.*, 2021]	1	19.19	9.88×
DSLP [Huang *et al.*, 2022a]	1	22.28	9.88×

Table 1: BLEU scores and speed-up ratios of the AR model (Line 1) and three mainstream NAR models (Lines 2-4) on the test set of PHOENIX 2014T. **Iter** denotes the number of decoding steps and **Speedup** stands for the speed-up ratio compared to the AR model. $\mathbf{T_y}$ denotes the length of the target sentence.

Most existing studies employ a Transformer-based [Vaswani *et al.*, 2017] sequence-to-sequence framework with an AutoRegressive (AR) decoding mechanism [Camgoz *et al.*, 2020; Zhou *et al.*, 2021b; Zheng *et al.*, 2021; Cao *et al.*, 2022; Zheng *et al.*, 2023] for sign language translation. AR decoding mechanism aims to generate considered tokens in target sentences conditioned on previously generated tokens so that the target sentence will be predicted token by token. Although this decoding mechanism could generate fluent target sentences, it still suffers from two inherent flaws: **1)** obviously, this decoding mechanism leads to high inference latency [Gu *et al.*, 2018], which cannot meet the real-time requirement of sign language translation systems [Yin *et al.*, 2021]; **2)** this approach only focuses on modeling the forward (unidirectional) contextual information of the target sentences while ignoring their bidirectional contextual information [Zhou *et al.*, 2022].

In contrast, unlike AR decoding mechanism, the Non-AutoRegressive (NAR) decoding mechanism [Gu *et al.*, 2018; Huang *et al.*, 2021; Huang *et al.*, 2022b] can predict all target tokens in parallel. This decoding mechanism not only significantly reduces the inference latency, but also effectively models the bidirectional contextual information of the target sentences. Existing NAR decoding mechanisms can be roughly divided into two categories, i.e., iterative NAR decoding mechanism [Ghazvininejad *et al.*, 2019; Huang *et al.*, 2021] and fully NAR decoding mechanism [Gu and Kong, 2021; Qian *et al.*, 2021; Huang *et al.*, 2022b]. The iterative NAR decoding mechanism leverages multiple passes of decoding to get the final target sentence, whereas the fully NAR decoding mechanism generates the whole sen-

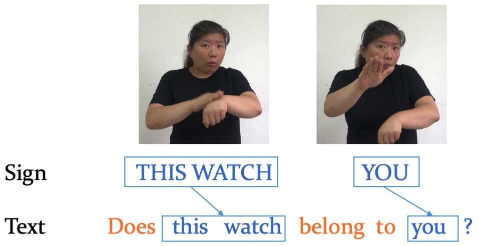

Figure 1: An example from CSL-Daily dataset, where **Sign** denotes sign language annotations and **Text** represents corresponding spoken sentence. We can observe that "this watch" and "you" can be predicted directly through the information of videos, while "does" and "belong to" can only be predicted correctly relying on the predicted target context (Notice that we have translated the Chinese into English for readability).

tence through only one decoding pass. Therefore, the fully NAR decoding mechanism has a higher decoding speed than the iterative NAR decoding mechanism. However, the NAR decoding mechanisms cannot focus on modeling the forward sequential information [Shao *et al.*, 2019] of the target sentences, which results in their inferior performances compared to the AR decoding mechanism. To exhibit how AR and NAR decoding mechanisms perform differently on sign language translation dataset, we report the BLEU scores and speed-up ratios of the AR model and three mainstream NAR models on the test set of the popular benchmark dataset PHOENIX 2014T [Camgoz *et al.*, 2018] in Table 1. As shown in Table 1, the inference speed of the AR model is significantly lower than that of the models with the NAR decoding mechanisms. Meanwhile, GLAT and DSLP using a fully NAR decoding mechanism have the highest inference speed. Moreover, we can also observe that the AR model obtains superior translation performance compared to other NAR models.

In this paper, to reduce the inference latency of the sign language translation model while maintaining its competitive decoding quality, we integrate curriculum learning into the NAR decoder and propose a Curriculum-based NAR Decoder (CND), which adopts fully NAR decoding mechanism. Specifically, we expect the lower layers of the decoder to predict simple tokens that can be predicted correctly relying on source-side information solely. Meanwhile, with the help of the predictions at the lower layers, the upper layers of our CND are required to predict complex tokens that can be predicted based on the context of target sentence. In this way, our CND can predict the target sentence from easy to hard. For example, as shown in Figure 1, "this watch" and "you" can be predicted directly by leveraging the visual features of sign videos, while "does" and "belong to" can only be predicted correctly relying on the context of target sentence.

Furthermore, to improve our CND's capacity to model the forward contextual information of the target sentence, we propose a mutual learning framework that contains two decoders, i.e., an AR decoder and an NAR decoder (CND). Specifically, we jointly train the AR decoder and our CND and minimize the KL divergence between their outputs, which enables the AR decoder and CND to promote each other. In this way, our CND could learn forward sequential knowledge from the AR decoder, while the AR decoder could learn bidirectional contextual knowledge from our CND. Unlike conventional knowledge distillation [Kim and Rush, 2016; Gu *et al.*, 2018] used in NAR machine translation, the AR decoder in our framework could be further strengthened to provide our CND with better guidance.

Our contributions are summarized as follows:

- We propose a Curriculum-based NAR Decoder (CND) that generates the whole target sentence through only one decoding step, which considerably reduces the inference latency of the model while maintaining its competitive decoding quality.
- We propose a mutual learning framework that enables the AR decoder and our CND to promote each other and further improves the performance of our CND.
- The experimental results on PHOENIX2014T and CSL-Daily demonstrate that our model consistently outperforms all competitive baselines and achieves about 7.92/8.02 times speed-up compared to the AR model.

2 Methodology

In this section, to better illustrate our method, we first simply introduce our base model (i.e., the transformer-based autoregressive sign language model) in Section 2.1 and the fully NAR decoding mechanism in Section 2.2. Then, we describe our Curriculum-based NAR Decoder in detail (Section 2.3). Finally, we give a detailed description of our proposed mutual learning framework (Section 2.4).

2.1 The Base Model

As shown in Figure 2(a), we employ the Transformer-based autoregressive sign language translation model as our base model, which consists of a visual features extractor and a Transformer [Vaswani *et al.*, 2017]. Given a sign language video $X = \{x_1, x_2, ..., x_{T_x}\}$ and corresponding spoken language sentence $Y = \{y_1, y_2, ..., y_{T_y}\}$, our base model aims to model the conditional probability $p(Y|X)$. Specifically, following previous works [Camgoz *et al.*, 2020; Zhou *et al.*, 2021a; Zhang *et al.*, 2023], we first utilize a pretrained CNN-based visual module [Hao *et al.*, 2021] as our visual features extractor to extract the visual features $V \in R^{T_x \times d}$. Then we feed V into the encoder of the Transformer to model the correlations among the visual features and obtain better visual contextual representation V'. Finally, through the causal masking mechanism, the decoder of the Transformer generates each token in the target sentence conditioned on V' and the tokens previously generated. In this way, the conditional probability $p(Y|X)$ can be formulated as

$$p(Y|X) = \prod_{t=1}^{T_y+1} p(y_t|y_{0:t-1}, X) \qquad (1)$$

where y_0 and y_{T_y+1} denote the start and end of the target sentences (i.e., $\langle bos \rangle$ and $\langle eos \rangle$), respectively.

Obviously, the AR decoding mechanism focuses on modeling the forward contextual information of the target sentence

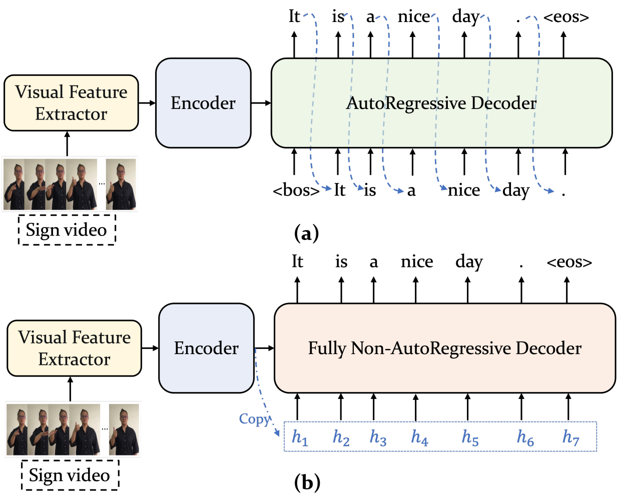

Figure 2: The overview of (a) the Transformer-based AR sign language translation model and (b) the Transformer-based fully NAR sign language translation model.

while ignoring its backward contextual information. Meanwhile, during inference, this decoding mechanism requires generating target sentence token by token, which leads to high inference latency.

2.2 Fully NAR Decoding Mechanism

Unlike AR decoding mechanism that generates target tokens one by one, the fully NAR decoding mechanism predicts all target tokens through only one decoding step (see Figure 2(b)). Specifically, following [Ghazvininejad *et al.*, 2019], the encoder first predicts the length of the target sentence $\hat{T}_y$ using a linear layer. Then, the decoder input $H^{(0)}$ of length $\hat{T}_y$ is generated through the source-side soft copy mechanism [Wei *et al.*, 2019; Qian *et al.*, 2021]. Finally, $H^{(0)}$ is fed into the fully NAR decoder to generate the whole sentence through one decoding pass. In this situation, the conditional probability $p(Y|X)$ mentioned above can be expressed as

$$p(Y|X) = p_L(T_y|X) \cdot \prod_{t=1}^{T_y} p(y_t|X) \tag{2}$$

Compared to AR decoder, the fully NAR decoder has two significant characteristics: 1) the fully NAR decoder has a high decoding speed because it could generate the whole target sentence at once; 2) since the fully NAR decoder does not implement the causal mask mechanism, it can effectively model the bidirectional contextual information of the target sentence. However, the fully NAR decoder cannot focus on modeling the forward sequential information of the target sentence, which results in its inferior performance compared to the AR decoder.

2.3 Curriculum-Based NAR Decoder (CND)

To effectively improve the inference speed of the sign language translation model and maintain its competitive performance, we introduce curriculum learning into the fully NAR decoder and propose a Curriculum-based NAR Decoder (CND). As shown in Figure 3, our CND consists of N stacked identical Transformer decoder layers. Unlike vanilla fully NAR decoder [Gu *et al.*, 2018] that only makes predictions at its top layer, our CDN predicts simple tokens at the lower layers while generating complex tokens at the upper layers.

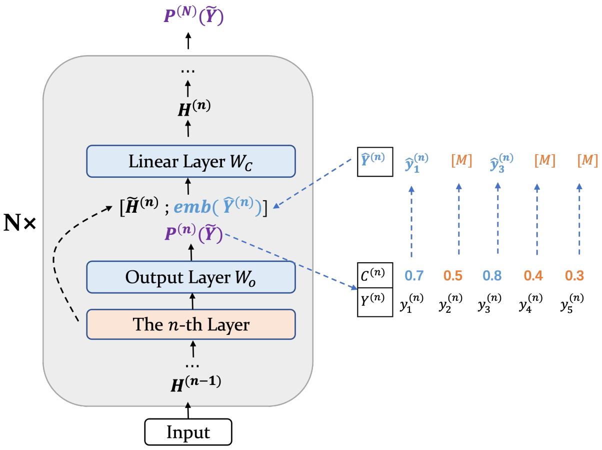

Figure 3: The overview of our proposed CND, where the grey areas depict the specific calculations at layer n.

Specifically, we first obtain the predicted length of target sentence $\hat{T}_y$. To achieve this, a special $[length]$ token is concatenated to the input of the encoder, and the encoder is trained to predict $\hat{T}_y$ as the output of the $[length]$ token. Then, following [Qian *et al.*, 2021], we employ the input of the encoder to generate the input of our CND $H^{(0)} = \{h_1^{(0)}, h_2^{(0)}, ..., h_{\hat{T}_y}^{(0)}\}$ through an attention-based soft copy mechanism. Finally, we feed $H^{(0)}$ into our CND and require it to predict corresponding tokens at each layer.

In particular, considering the n-th layer of our CND, we first obtain its hidden state $\tilde{H}^{(n)}$ by

$$\tilde{H}^{(n)} = \text{Layer}_{\text{dec}}^{(n)}(H^{(n-1)}, V') \tag{3}$$

where $\text{Layer}_{\text{dec}}^{(n)}$ and $H^{(n-1)}$ denote the n-th layer of our CND and the output of the $(n-1)$-th layer, respectively.

After that, we pass $\tilde{H}^{(n)}$ into a linear output layer to acquire the predicted tokens $Y^{(n)}$ and corresponding confidences $C^{(n)}$ by

$$P^{(n)}(\tilde{Y}) = \text{softmax}(W_o \tilde{H}^{(n)}) \tag{4}$$

$$Y^{(n)}, C^{(n)} = (\text{arg}) \max_{\omega} P^{(n)}(\tilde{Y} = \omega) \tag{5}$$

where W_o denotes the the weight matrix of the output layer. Note that the linear output layer is shared among all the layers of our CND.

Meanwhile, we keep the $\lceil \hat{T}_y \cdot \frac{n}{N} \rceil$ tokens with the highest confidence in $Y^{(n)}$ and replace the others using a special $[mask]$ token. Then we get the updated predictions $\hat{Y}^{(n)} = \{\hat{y}_1^{(n)}, \hat{y}_2^{(n)}, ..., \hat{y}_{\hat{T}_y}^{(n)}\}$ where

$$\hat{y}_t^{(n)} = \begin{cases} y_t^{(n)} & \text{if } t \in K^{(n)}; \\ [mask] & \text{otherwise;} \end{cases} \tag{6}$$

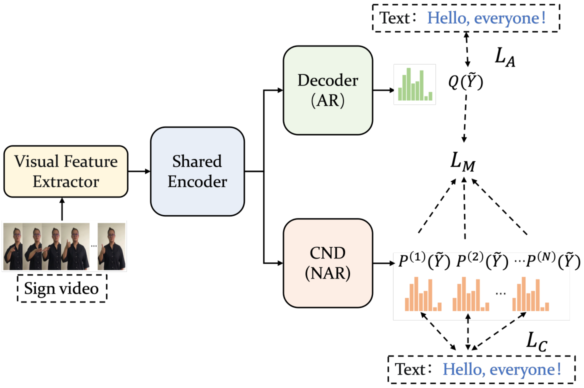

Figure 4: Illustration of our mutual learning framework. The framework contains two decoders, i.e., an AR decoder and an NAR decoder (CND).

where $K^{(n)}$ stands for the set of the indexes of the most confident $\lceil \hat{T}_y \cdot \frac{n}{N} \rceil$ tokens in $Y^{(n)}$.

Finally, we obtain the output of the n-th layer $H^{(n)}$ by concatenating the embedding of $\hat{Y}^{(n)}$ with $\tilde{H}^{(n)}$ and utilizing a linear layer to further process it, as follows:

$$H^{(n)} = W_c[\tilde{H}^{(n)}; \text{emb}(\hat{Y}^{(n)})] \tag{7}$$

where W_c denotes the weight matrix of the linear layer.

Note that $H^{(n)}$ is also the input of the $(n+1)$-th layer. In this way, the lower layers provide confident predictions to the upper layers, which promotes the prediction of complex tokens at the upper layers.

In addition, we can notice that the number of tokens to be predicted ($\lceil \hat{T}_y \cdot \frac{n}{N} \rceil$) is increased linearly layer by layer, and the top layer will predict all the target tokens. Therefore, our CND is able to generate the target sentence from the bottom up with increasing difficulty. It is worth noting that we have also considered using different functions (e.g., logarithmic or exponential) other than the linear function to calculate the prediction ratio for each layer, and we will illustrate it in Section 3.6.

Due to the fact that the predictions at the top layer of our CND are generated based on all the predictions at the lower layers, we use the top layer's output as our final prediction during inference.

2.4 Mutual Learning

To further boost the performance of our CND and integrate both AR and NAR decoding mechanisms, we propose a mutual learning framework. As shown in Figure 4, our framework contains two decoders, i.e., an AR decoder and our CND. In this framework, we jointly train the AR decoder and our CND and minimize the distance of their outputs. Therefore, our final training objective $\mathcal{L}$ contains three components: AR loss $\mathcal{L}_\mathcal{A}$, CND loss $\mathcal{L}_\mathcal{C}$, and mutual learning loss $\mathcal{L}_\mathcal{M}$.

$$\mathcal{L} = \mathcal{L}_\mathcal{C} + \alpha\mathcal{L}_\mathcal{A} + \beta\mathcal{L}_\mathcal{M} \tag{8}$$

where α and β are hyperparameters.

AR loss $\mathcal{L}_\mathcal{A}$. We employ vanilla Transformer [Vaswani *et al.*, 2017] decoder as our AR decoder. Specifically, we first feed the shifted ground-truth tokens into our AR decoder and obtain the hidden states of its top layer $H^{(AR)}$. Then, we send $H^{(AR)}$ into the linear output layer to get the probability distribution

$$Q(\tilde{Y}) = \text{softmax}(W_o H^{(AR)}) \tag{9}$$

Finally, we use Cross-Entropy loss as our AR loss $\mathcal{L}_\mathcal{A}$

$$\mathcal{L}_\mathcal{A} = \sum_{t=1}^{T_y} \text{CE}(q_t, y_t) \tag{10}$$

where q_t denotes the t-th element of $Q(\tilde{Y})$, corresponding to the predicted probability distribution of the t-th token of the target sentence, and CE$(\cdot)$ denotes the calculation of Cross-Entropy.

CND loss $\mathcal{L}_\mathcal{C}$. Since our CND makes predictions at every layer, the $\mathcal{L}_\mathcal{C}$ is the sum of the Cross-Entropy losses in each layer.

$$\mathcal{L}_\mathcal{C} = \sum_{i=1}^{N} \sum_{t \in K^{(i)}} \text{CE}(p_t^{(i)}, y_t) \tag{11}$$

where $p_t^{(i)}$ denotes the t-th token's probability distribution produced by equation (4). In this way, our CND could predict simple tokens at the lower layers, which builds a solid foundation for predicting complex tokens at the upper layers.

Mutual learning loss $\mathcal{L}_\mathcal{M}$. We use KL divergence to measure the gap between the outputs of the two decoders. Since our CND makes predictions at every layer, we calculate the KL divergence between the distribution predicted by each layer of the CND and that predicted by the top layer of the AR decoder.

$$\mathcal{L}_\mathcal{M} = \frac{1}{2} \sum_{i=1}^{N} \sum_{t \in K^{(i)}} (KL(q_t || p_t^{(i)}) + KL(p_t^{(i)} || q_t)) \tag{12}$$

In this way, our CND is able to learn forward (unidirectional) sequential knowledge from the AR decoder, while the AR decoder could learn bidirectional contextual knowledge from our CND.

3 Experiment

3.1 Datasets and Evaluation Metrics

We evaluate our approach on two popular benchmark datasets for sign language translation task, i.e., PHOENIX 2014T [Camgoz *et al.*, 2018] and CSL-Daily [Zhou *et al.*, 2021a].

- PHOENIX 2014T is a German sign language translation dataset recorded from weather forecast news, including sign language videos from 9 signers, sign gloss annotations and German translations which are all segmented into parallel sentences. The dataset includes 7,096/519/642 continuous sign language videos in train/dev/test splits.

Category	Model	Iter	PHOENIX 2014T				Latency	Speedup
			Dev		Test			
			ROUGE	BLEU4	ROUGE	BLEU4		
AR	AR [Vaswani *et al.*, 2017]	N	49.33	23.12	49.48	23.24	188.47ms	1.00×
Iterative NAR	CMLM [Ghazvininejad *et al.*, 2019]	4	49.83	22.15	50.33	22.45	76.30ms	2.47×
	CMLMC [Huang *et al.*, 2021]	4	49.88	22.42	50.51	22.67	76.30ms	2.47×
Fully NAR	GLAT [Qian *et al.*, 2021]	1	47.73	19.88	47.10	19.19	19.07ms	9.88×
	DSLP [Huang *et al.*, 2022a]	1	50.42	22.36	50.38	22.28	19.07ms	9.88×
Ours	CND	1	51.86	22.95	51.08	22.92	23.79ms	7.92×
	CND+Mutual Learning	1	**52.99**	**24.32**	**53.58**	**24.71**	23.79ms	7.92×

Table 2: Experimental results on the development and test sets of PHOENIX 2014T. Note that we report the test scores of the best checkpoint on the development set. In addition to reporting the scores of BLEU4 and ROUGE, we also report the sentence-level inference latency as well as the speedup ratio of the test set. Best results are highlighted in **bold**.

Category	Model	Iter	CSL-Daily				Latency	Speedup
			Dev		Test			
			ROUGE	BLEU4	ROUGE	BLEU4		
AR	AR [Vaswani *et al.*, 2017]	N	44.16	15.92	43.93	15.88	197.28ms	1.00×
Iterative NAR	CMLM [Ghazvininejad *et al.*, 2019]	4	47.35	14.96	47.49	14.47	77.84ms	2.53×
	CMLMC [Huang *et al.*, 2021]	4	47.33	14.71	47.56	15.01	77.84ms	2.53×
Fully NAR	GLAT [Qian *et al.*, 2021]	1	44.25	11.48	44.33	11.27	19.46ms	10.14×
	DSLP [Huang *et al.*, 2022a]	1	46.65	13.56	46.54	13.73	19.46ms	10.14×
Ours	CND	1	48.85	14.94	48.41	14.84	24.59ms	8.02×
	CND+Mutual Learning	1	**50.74**	**16.49**	**51.05**	**16.61**	24.59ms	8.02×

Table 3: Experimental results on the development and test sets of CSL-Daily. We use the same experimental settings as in Table 2. Best results are highlighted in **bold**.

- CSL-Daily is a recently released Chinese sign language translation dataset. The content of the corpus focuses on the daily life of the deaf community and covers a wide range of topics, including family life, medical care, school life, bank service, shopping, social contact and so on. The dataset includes 18,401/1,077/1,176 continuous sign language videos in train/dev/test splits. There are 10 signers participating in the video recording work.

Moreover, following the previous studies [Camgoz *et al.*, 2018; Camgoz *et al.*, 2020; Chen *et al.*, 2022a; Chen *et al.*, 2022b], we evaluate the performance of our model using the BLEU [Papineni *et al.*, 2002] and ROUGE [1] [Lin, 2004].

3.2 Implementation Details

Our model is developed based on PyTorch and all the experiments are run on 1 Titan RTX GPU.

Before training, we employ Xavier initialization to initialize the parameters of all the networks with a gain of 1.0. Then we train all the models for 60 epochs using Adam ($\beta_1 = 0.9$, $\beta_2 = 0.998$) [Kingma and Ba, 2014] with a Linear Warm-up Scheduler [Goyal *et al.*, 2017], where the peak learning rate is set to 5e-4 and warm-up step is 8K.

The hyper-parameters α and β are set to 0.5 and 1, respectively. The encoder/decoder layer number N is set to 5. For NAR inference, we follow [Ghazvininejad *et al.*, 2019] and set the length beam to 5 and for AR inference the beam size is set to 5.

3.3 Baseline Models

We compare our model with the following Transformer-based sign language translation models:

AR model. Autoregressive sign language translation model.

Iterative NAR models. We choose two strong iterative non-autoregressive models, i.e., CMLM [Ghazvininejad *et al.*, 2019] and CMLMC [Huang *et al.*, 2021], and reproduce their results on the two sign language translation datasets.

Fully NAR models. We select two competitive fully non-autoregressive models, i.e., GLAT [Qian *et al.*, 2021] and DSLP [Huang *et al.*, 2022a], and reproduce their performance on the two sign language translation datasets.

3.4 Main Results

Results on PHOENIX 2014T. As illustrated in the Table 2, our model consistently outperforms all baselines on PHOENIX 2014T dataset. First, compared with two iterative NAR models, i.e., CMLM and CMLMC, our CND obtains improvements of 0.47 and 0.25 BLEU4 points respectively and a much higher decoding speed ($7.92\times$ vs. $2.47\times$)

[1] https://github.com/neccam/slt/blob/master/signjoey/metrics.py

on the test set of the PHOENIX 2014T. Second, compared with two fully NAR models, i.e., GLAT and DSLP, our CND achieves an average performance gain of 2.18 BLEU4 score while maintaining a comparable speed (9.88× vs. 7.92×). Third, compared with the AR model, our CND achieves about 7.92 times speed-up while maintaining a competitive performance with a decrease of 0.32 BLEU4 points. These experimental results demonstrate that our curriculum learning is effective in improving the decoding capability of the fully NAR decoder. Further, our mutual learning framework brings a large performance gain of 1.79 points to our CND, which enables our model to outperform the AR model by 1.47 BLEU4 points on the test set of PHOENIX 2014T, which suggests that our mutual learning framework could enhance our CND's capacity to model the forward contextual information of the target sentence.

Results on CSL-Daily. As shown in Table 3, firstly, compared with two fully NAR models, our CND obtains an average improvement of 2.34 BLEU4 points and a comparable speed (8.02× vs. 10.14×). Secondly, our CND has a much higher decoding speed (8.02× vs. 2.53×) than the two iterative NAR models. In terms of performance, our CND surpasses CMLM by 0.37 BLEU4 points and achieves a similar performance to the CMLMC (14.84 vs. 15.01 BLEU4). Third, compared with the AR model, our CND obtains about 8.02 times speed-up with a performance decrease of 1.04 BLEU4 points. Further, our mutual learning framework enables our CND to outperform the AR model by 0.73 BLEU4 points with 8.02 times speed up on the test set of CSL-daily.These results completely demonstrate the effectiveness and efficiency of our CND and mutual learning framework.

3.5 Ablation Study

To further comprehend the contributions of different components of our model, we conduct extensive ablation studies on PHOENIX 2014T dataset. Our experimental details and results are shown in Table 4.

w/o Lower Layers Mutual Learning. In this variant, we only minimize the KL divergence between the distribution predicted by the top layer of the CND and that predicted by the top layer of the AR decoder. From Line 3 in Table 4, we observe that this variant will lead to a performance drop of 1.19 BLEU4 points, which illustrates that the lower layers of the CND require more forward sequential knowledge of the AR decoder than the top layer.

w/o $\mathcal{L}_{\mathcal{M}}$. As shown in Lines 5 and 6 in Table 4, if we remove the $\mathcal{L}_{\mathcal{M}}$ in equation (8), the performance of our CND and AR will drop by 1.71 and 2.98 BLEU4 points respectively, which demonstrates the effectiveness of bidirectional distillation between the AR decoder and the CND in our mutual learning framework.

Mutual Learning → Online Unidirectional Distillation. In this variant, we replace the bidirectional distillation in the mutual learning framework with the unidirectional distillation from the AR decoder to our CND. From Lines 7-8 we can observe a significant drop in performance of both AR and CND (3.01 and 1.25 BLEU4). This result indicates that the AR decoder is enhanced through learning bidirectional contextual knowledge from the CND, and the strengthened AR decoder provides the CND with more effective sequential information to better guide it.

lines	model	BLEU4	ROUGE
ours			
1	CND+Mutual Learning	**24.71**	**53.58**
2	AR+Mutual Learning	**26.60**	**53.50**
w/o Lower Layers Mutual Learning			
3	CND+Mutual Learning	23.52	52.36
4	AR+Mutual Learning	26.24	53.26
w/o $\mathcal{L}_{\mathcal{M}}$			
5	CND+Mutual Learning	22.97	50.92
6	AR+Mutual Learning	23.62	48.94
Mutual Learning → Online Unidirectional Distillation			
7	CND+Mutual Learning	23.46	51.78
8	AR+Mutual Learning	23.59	49.01
Mutual Learning → Offline Unidirectional Distillation			
9	student (CND)	23.06	51.12
10	teacher (AR)	23.24	49.48
w/o Mutual Learning			
11	CND	22.92	51.08
12	AR	23.24	49.48
w/o Mutual Learning & Curriculum Learning			
13	CND	22.28	50.38

Table 4: Ablation study of our model on the development set of PHOENIX 2014T.

Mutual Learning → Offline Unidirectional Distillation. We replace our mutual learning framework with conventional sequence-level knowledge distillation, which is a kind of offline unidirectional distillation. The results in Line 9 of Table 4 show that this variant leads to significant performance degradation in CND (1.65 BLUE4). This is due to the fact that the frozen AR teacher cannot provide valid guidance for our CND, which proves that our mutual learning framework is superior to conventional sequence-level knowledge distillation.

w/o Mutual Learning & Curriculum Learning. To demonstrate the efficacy of our curriculum learning, we remove it from the CND. As shown in the Lines 11 and 13 of Table 4, this variant causes a significant performance decline of 0.64 BLEU4 points, which confirms that our curriculum learning could effectively improve the decoding capability of fully NAR decoder.

3.6 Effect of the Curriculum Function

In our CND, we introduce curriculum learning into the fully NAR decoder and expect it to make predictions from easy to hard. Specifically, We use a linear function (i.e., $\lceil T_y \cdot \frac{n}{N} \rceil$) as curriculum function to calculate the number of tokens to be predicted at each layer. In addition, we also utilize three different kinds of functions in our curriculum. We present the variation of the prediction ratio per layer with these functions in Figure 5 and the corresponding performance of the model in Table 5.

As shown in Table 5, the linear function achieves the best results compared to the other three functions (Line 1). Besides, we can also observe that the results of the linear, logarithmic and quadratic functions are close to each other, with no clear superiority or inferiority. However, the exponen-

Functions	Numbers	BLEU4	ROUGE
Linear	$\lceil T_y \cdot \frac{n}{N} \rceil$	**22.92**	**51.08**
Logarithmic	$\lceil T_y \cdot \frac{\log(n+1)}{\log(N+1)} \rceil$	22.89	50.76
Exponential	$\lceil T_y \cdot \frac{e^n}{e^N} \rceil$	22.77	50.41
Quadratic	$\lceil T_y \cdot \frac{n^2}{N^2} \rceil$	22.86	50.26

Table 5: Experimental results of CNDs with different curriculum functions where **Numbers** denotes the number of tokens to be predicted at the n-th layer.

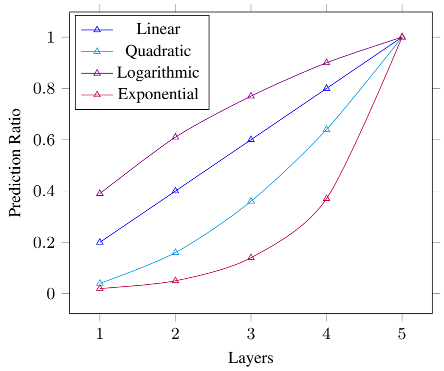

Figure 5: The variation of the prediction ratio per layer with four different functions.

tial function leads to a slight decrease in the performance (Line 3). According to Figure 5, the linear, logarithmic, and quadratic functions grow relatively smoothly throughout, while the exponential function sharply increases in the later stages. We speculate that the smooth increment of the prediction ratio leads to superior performance of the model. However, if the prediction ratio is small at the lower layers and grows too fast at the upper layers, the lower layers could not provide enough contextual information to the upper, making it more difficult to predict complex tokens at the upper layers.

4 Related Work

Sign Language Translation. Sign language translation aims to transform sign language videos into spoken languages. Most existing approaches formulate this task as a neural machine translation(NMT) problem [Chen *et al.*, 2022a] and utilize seq2seq structure with attention mechanism. [Camgoz *et al.*, 2020] firstly applied Transformer [Vaswani *et al.*, 2017] to sign language translation task and utilized sign gloss as intermediate supervision to regularize the Transformer encoder. [Zhou *et al.*, 2021a] proposed a two-stage back translation method to solve the problem of data scarcity. [Zhou *et al.*, 2021b] enhanced the visual feature modeling of sign language translation by exploiting multi-cue features in sign language video frames. [Chen *et al.*, 2022a] applied visual and linguistic pretraining approaches to sign language translation model. However, all these approaches use autoregressive decoding mechanism, which cannot meet the real-time requirement of sign language translation systems. Therefore, our work mainly focuses on how to effectively reducing the inference latency of sign language translation model.

Non-autoregressive Decoding Mechanisms. Most existing NAR decoding mechanisms can be roughly divided into iterative NAR decoding mechanism [Ghazvininejad *et al.*, 2019; Huang *et al.*, 2021] and fully NAR decoding mechanism [Huang *et al.*, 2022b; Qian *et al.*, 2021; Li *et al.*, 2022]. [Ghazvininejad *et al.*, 2019; Ghazvininejad *et al.*, 2020; Huang *et al.*, 2021] applied Conditional Mask Language Model (CMLM) to NAR translation task and modeled the dependency between observable and masked target tokens. However, iterative decoding limits the efficiency of NAR decoding, so fully NAR, which requires only one decoding pass, is now becoming the focus of research. [Qian *et al.*, 2021] proposed a glancing sampling strategy for CMLM that kept more visible tokens at the early stage of training and gradually reduced the number of them. [Huang *et al.*, 2022b] introduced a Directed Acyclic Graph structure into NAR Transformer and demonstrated that introducing the left-to-right dependencies can effectively improve the performance of the fully NAR model. [Huang *et al.*, 2022a] provided supervision for each layer of the decoder, significantly improving the performance of the NAR model. Unlike previous work, our proposed CND integrates the strengths of curriculum learning and the fully NAR decoding mechanism, which generates the target sentence from the bottom up with increasing difficulty.

Knowledge Distillation between AR and NAR. Most Existing studies on NAR leveraged sequence-level knowledge distillation [Kim and Rush, 2016; Gu *et al.*, 2018] to bridge the performance gap between NAR and AR model. Meanwhile, [Zhou *et al.*, 2022] demonstrated that transferring bidirectional contextual knowledge from NAR decoder to AR decoder could effectively improve the performance of AR decoder. Motivated by their studies, we propose a mutual learning framework that allows AR and NAR decoders to promote each other.

5 Conclusion and Future Work

In this paper, we propose a Curriculum-Based Non-autoregressive Decoder (CND) and a mutual learning framework for sign language translation. Our CND integrates the strengths of curriculum learning and fully NAR decoding mechanisms, which not only significantly reduces the inference latency of the model but also maintains its competitive performance. Meanwhile, our mutual learning framework effectively leverages the advantages of both AR and NAR decoding mechanisms, enabling our CND to learn the forward sequential knowledge from the strengthened AR decoder and obtain significant performance gain. The experimental results on PHOENIX2014T and CSL-Daily show that our model consistently outperforms all the competitive baselines and achieves more than 7.92 times speed-up compared to the AR model.

In the future, we plan to further optimize our mutual learning framework to enable more significant improvements in both AR and NAR decoders.

Acknowledgments

We thank all the anonymous reviewers for their insightful and valuable comments. This work was supported by National Natural Science Foundation of China "Research on Neural Chinese Sign Language Translation Methods Integrating Sign Language Linguistic Knowledge" (No. 62076211) and Central Leading Local Project "Fujian Mental Health Human-Computer Interaction Technology Research Center" (No. 2020L3024).

Contribution Statement

Pei Yu and Liang Zhang make equal contributions and share co-first authorship. Meanwhile, Yidong Chen is the corresponding author of this paper.

References

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

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

[Cao *et al.*, 2022] Yong Cao, Wei Li, Xianzhi Li, Min Chen, Guangyong Chen, Long Hu, Zhengdao Li, and Kai Hwang. Explore more guidance: A task-aware instruction network for sign language translation enhanced with data augmentation. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2679–2690, Seattle, United States, July 2022. Association for Computational Linguistics.

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

[Chen *et al.*, 2022b] Yutong Chen, Ronglai Zuo, Fangyun Wei, Yu Wu, Shujie Liu, and Brian Mak. Two-stream network for sign language recognition and translation. *arXiv preprint arXiv:2211.01367*, 2022.

[Ghazvininejad *et al.*, 2019] Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6112–6121, 2019.

[Ghazvininejad *et al.*, 2020] Marjan Ghazvininejad, Omer Levy, and Luke Zettlemoyer. Semi-autoregressive training improves mask-predict decoding. *arXiv preprint arXiv:2001.08785*, 2020.

[Goyal *et al.*, 2017] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[Gu and Kong, 2021] Jiatao Gu and Xiang Kong. Fully non-autoregressive neural machine translation: Tricks of the trade. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 120–133, 2021.

[Gu *et al.*, 2018] Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*, 2018.

[Hao *et al.*, 2021] Aiming Hao, Yuecong Min, and Xilin Chen. Self-mutual distillation learning for continuous sign language recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11303–11312, 2021.

[Huang *et al.*, 2021] Xiao Shi Huang, Felipe Perez, and Maksims Volkovs. Improving non-autoregressive translation models without distillation. In *International Conference on Learning Representations*, 2021.

[Huang *et al.*, 2022a] Chenyang Huang, Hao Zhou, Osmar R Zaïane, Lili Mou, and Lei Li. Non-autoregressive translation with layer-wise prediction and deep supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10776–10784, 2022.

[Huang *et al.*, 2022b] Fei Huang, Hao Zhou, Yang Liu, Hang Li, and Minlie Huang. Directed acyclic transformer for non-autoregressive machine translation. In *Proceedings of the 39th International Conference on Machine Learning, ICML 2022*, 2022.

[Kim and Rush, 2016] Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, 2016.

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

[Li *et al.*, 2022] Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. Elmer: A non-autoregressive pre-trained language model for efficient and effective text generation. *arXiv preprint arXiv:2210.13304*, 2022.

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

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

[Qian *et al.*, 2021] Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu, and Lei Li.

Glancing transformer for non-autoregressive neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1993–2003, 2021.

[Shao *et al.*, 2019] Chenze Shao, Yang Feng, Jinchao Zhang, Fandong Meng, Xilin Chen, and Jie Zhou. Retrieving sequential information for non-autoregressive neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3013–3024, Florence, Italy, July 2019. Association for Computational Linguistics.

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

[Wei *et al.*, 2019] Bingzhen Wei, Mingxuan Wang, Hao Zhou, Junyang Lin, and Xu Sun. Imitation learning for non-autoregressive neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1304–1312, 2019.

[Yin *et al.*, 2021] Aoxiong Yin, Zhou Zhao, Jinglin Liu, Weike Jin, Meng Zhang, Xingshan Zeng, and Xiaofei He. Simulslt: End-to-end simultaneous sign language translation. In *ACM Multimedia*, 2021.

[Zhang *et al.*, 2023] Biao Zhang, Mathias Müller, and Rico Sennrich. SLTUNET: A simple unified model for sign language translation. In *The Eleventh International Conference on Learning Representations*, 2023.

[Zheng *et al.*, 2021] Jiangbin Zheng, Yidong Chen, Chong Wu, Xiaodong Shi, and Suhail Muhammad Kamal. Enhancing neural sign language translation by highlighting the facial expression information. *Neurocomputing*, 464:462–472, 2021.

[Zheng *et al.*, 2023] Jiangbin Zheng, Yile Wang, Cheng Tan, Siyuan Li, Ge Wang, Jun Xia, Yidong Chen, and Stan Z Li. Cvt-slr: Contrastive visual-textual transformation for sign language recognition with variational alignment. *arXiv preprint arXiv:2303.05725*, 2023.

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

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

[Zhou *et al.*, 2022] Chulun Zhou, Fandong Meng, Jie Zhou, Min Zhang, Hongji Wang, and Jinsong Su. Confidence based bidirectional global context aware training framework for neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2878–2889, 2022.