Improving Non-Autoregressive Neural Machine Translation via Modeling Localness

Yong Wang[†*] **Xinwei Geng**
[†]Tencent Corporation, China
{seaywang,xwgeng2014}@gmail.com

Abstract

Non-autoregressive translation (NAT) models, which eliminate the sequential dependencies within the target sentence, have achieved remarkable inference speed, but suffer from inferior translation quality. Towards exploring the underlying causes, we carry out a thorough preliminary study on the attention mechanism, which demonstrates the serious weakness in capturing localness compared with conventional autoregressive translation (AT). In response to this problem, we propose to improve the localness of NAT models by explicitly introducing the information about surrounding words. Specifically, temporal convolutions are incorporated into both encoder and decoder sides to obtain localness-aware representations. Extensive experiments on several typical translation datasets show that the proposed method can achieve consistent and significant improvements over strong NAT baselines. Further analyses on the WMT14 En⇒De translation task reveal that compared with baselines, our approach accelerates the convergence in training and can achieve equivalent performance with a reduction of 70% training steps.

1 Introduction

Based on the encoder-decoder framework (Cho et al., 2014; Sutskever et al., 2014), neural machine translation (NMT) (Bahdanau et al., 2015; Gehring et al., 2017; Vaswani et al., 2017) has achieved tremendous success in the past several years due to its excellent performance. Currently, state-of-the-art NMT systems are built in an autoregressive manner, which generates target tokens one by one from continuous representations summarized by the encoder. However, with the constraint of output tokens conditioned on previously generated ones, autoregressive translation (AT) inevitably suffers from serious latency during decoding, which becomes a bottleneck of inference speed.

Towards accelerating the inference process, non-autoregressive neural machine translation (NAT) (Gu et al., 2018; Ghazvininejad et al., 2019; Du et al., 2021; Huang et al., 2022) has been proposed to break the above bottleneck. Instead of sequential generation in autoregressive translation, NAT models output the entire target sentence at once. Unfortunately, removing sequential dependencies within the target sentence brings NAT models with serious weakness in capturing the highly multimodal distribution of target sentences (Gu et al., 2018). Accordingly, NAT models suffer from two kinds of incoherent translations, including *repetitive translations* and *incomplete translations* (Wang et al., 2019), which leads to inferior translation performance.

Since both types of translation errors do not commonly appear in AT models, several works (Li et al., 2019; Wei et al., 2019) have proposed to leverage a well-trained AT model to enhance the training of NAT models. Inspired by this consideration, we empirically carry out a thorough study to present the weakness of NAT by investigating the distinction between NAT and AT models. Specifically, we inspect the attention mechanism on two translation tasks and reveal that, in contrast to autoregressive models, NAT models lack the ability of either modeling the localness (Yang et al., 2018; Ding et al., 2020) or producing localness-aware representations.

Motivated by this observation, we propose to improve non-autoregressive machine translation via explicitly modeling localness. Specifically, we incorporate multi-layer temporal convolutions (MTC) into both encoder and decoder sides to enhance the ability to model localness-aware representations in NAT models. To validate the effectiveness of our approach, we implement this method on two advanced NAT models, namely conditional masked language model (CMLM) (Ghazvininejad et al., 2019) and Vanilla-NAT (Gu et al., 2018).

[*] Work done while at Microsoft Corporation.

Proceedings of the 29th International Conference on Computational Linguistics, pages 5217–5226
October 12–17, 2022.

Extensive experiments on typical translation benchmarks demonstrate that our proposed approach can significantly and consistently improve the translation quality by up to 1.0 BLEU points over a series of strong NAT baselines. Further analyses reveal that our approach enhances the ability to generate localness-aware representations. In addition, the analysis on the WMT14 En⇒De task shows that compared with the original CMLM model, our approach accelerates the convergence in training and achieves comparable performance with a reduction of 70% training steps.

In summary, the contributions of this work are detailed as follows:

- Our study demonstrates the necessity of explicitly modeling localness for non-autoregressive machine translation models.
- We propose a simple yet effective approach to enhance the ability to generate localness-aware representations in NAT models and extensive experiments validate the effectiveness and universality of our approach.
- Further analyses reveal that our approach benefits the translation for long sentences and accelerates the convergence in training.

2 Background

Autoregressive Neural Machine Translation In recent years, autoregressive models, which predict the target sentence sequentially conditioned on translation history, have achieved extraordinary success on machine translation. Specifically, given a source sentence $\mathbf{x} = \{x_1, \ldots, x_I\}$, a standard encoder-decoder autoregressive framework (Cho et al., 2014; Bahdanau et al., 2015) optimizes the conditional probability of a target sentence $\mathbf{y} = \{y_1, \ldots, y_J\}$, namely:

$$P(\mathbf{y}|\mathbf{x};\theta) = \prod_{j=1}^{J} P(y_j|y_{<j}, \mathbf{x};\theta), \quad (1)$$

where $y_{<j}$ indicates the partial translation and θ is a set of trainable parameters. The encoder-decoder framework can be implemented as different choices of architectures, such as recurrent neural network (Bahdanau et al., 2015), convolutional neural network (Gehring et al., 2017) and Transformer (Vaswani et al., 2017). The typical training objective is to maximize log-likelihood on a set of training examples $D = \{[\mathbf{x}^m, \mathbf{y}^m]\}_{m=1}^{M}$:

$$\mathcal{L}(\theta) = \arg\max_{\theta} \sum_{m=1}^{M} \log P(\mathbf{y}^m|\mathbf{x}^m;\theta). \quad (2)$$

In the inference, the conditional dependency on translation history produces the autoregressive property, which predicts the token sequentially based on previous output tokens in the target. This nature of sequential processing results in high latency in translation.

Non-Autoregressive Neural Machine Translation In contrast, Gu et al. (2018) proposed to accelerate the decoding process by generating target tokens in parallel. In practice, by breaking the probabilistic factorization, the prediction is modeled as a product of the probability, which is independent for each token:

$$P(\mathbf{y}|\mathbf{x};\theta) = P(J|\mathbf{x}) \prod_{j=1}^{J} P(y_j|\mathbf{x};\theta), \quad (3)$$

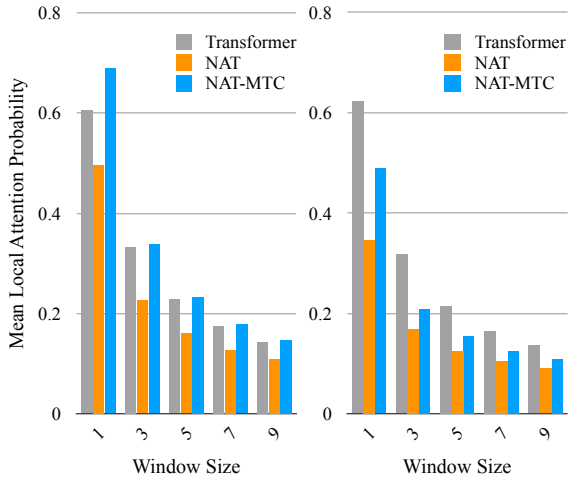
where $P(J|\mathbf{x})$ indicates an auxiliary length predictor, which is used to determine the translation length. NAT models employ an identical encoder as the conventional Transformer architecture, while the decoder is distinct from the original one as it avoids the utilization of causal masks in the self-attention mechanism.

However, due to the lack of explicit dependency within targets, NAT models suffer from serious multi-modality problem (Gu et al., 2018) and significantly degenerate the translation performance. This issue inevitably causes NAT models to suffer from repeated and incomplete translations.

3 Preliminary Study

In this section, we conduct a thorough empirical study on the attention mechanism to observe the distinction between AT and NAT models. In practice, a good probabilistic distribution of attention weights suggests a good alignment between source and target words, and usually leads to more accurate translation (Bahdanau et al., 2015; Luong et al., 2015; Li et al., 2019). Inspired by this observation, we systematically investigate the cross-attention weights on both WMT14 En⇒De and WMT16 En⇒Ro translation tasks.

Specifically, given a source sentence $\mathbf{x} = \{x_1, \ldots, x_I\}$, for the j-th token in the target sentence $\mathbf{y} = \{y_1, \ldots, y_J\}$, the attention probability



(a) WMT14 En⇒De (b) WMT16 En⇒Ro

Figure 1: Mean local attention probability on WMT14 En-De and WMT16 En-Ro translation tasks, varied by the window size.

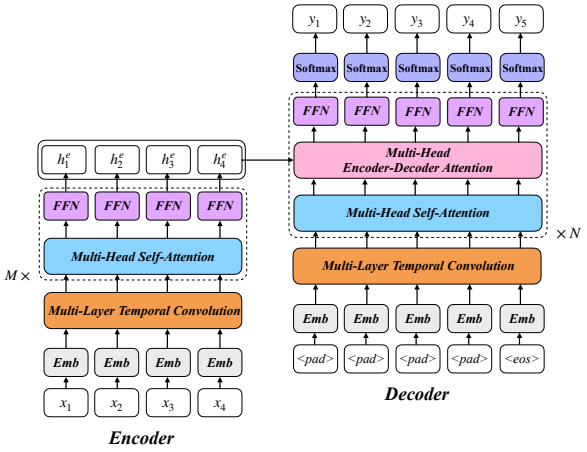


Figure 2: The framework of the proposed model. The left and right figures represent the encoder and decoder respectively. Distinct from conventional NAT models, our model incorporates multi-layer temporal convolutions to model localness for both encoder and decoder.

is computed as $\mathbf{p}_j = \{p_{j1}, p_{j2}, \ldots, p_{jI}\}$. To quantify the extent of centralization on a window, we define a measure γ_j, termed as local attention probability (LAP):

$$\gamma_j = \text{avg}(\mathbf{p}_j(f_k(\arg\max \mathbf{p}_j))), \quad (4)$$

where k is the window size, $\text{avg}(\cdot)$ is the average function, and $f_k(\cdot)$ indicates choosing surrounding indexes based on the central index (i.e., $\arg\max \mathbf{p}_j$). For instance, for $\mathbf{p}_j = [0.2, 0.5, 0.2, 0.1]$ and $k = 3$, $f_k(\cdot)$ returns the indexes $[1, 2, 3]$ and $\gamma_j = (0.2 + 0.5 + 0.2)/3 = 0.3$. Particularly noted that the number of surrounding indexes involved is less than k when the window exceeds the bounds of the sequence. For a target sentence, LAP is computed as $\gamma^m = \frac{1}{J}\sum_j \gamma_j^m$. We average γ^m on all translated sentences to obtain mean local attention probability (MLAP): $\bar{\gamma} = \frac{1}{M}\sum_m \gamma^m$. As seen, MLAP evaluates attention weights on a fixed window within source words, which is considered as local information.

We conduct this study on top of the advanced CMLM model and compare MLAP with the last layer and all heads of the encoder on three models: 1) Transformer (AT model), 2) NAT, and 3) NAT-MTC. As shown in Figure 1, the results reveal two key points. First of all, as the window size increases, the computation of MLAP includes more words. The decrease of MLAP indicates that attentions between target and source words usually focus on neighbor words within the source sentence. Secondly, compared with Transformer, the NAT model holds a smaller MLAP, which demonstrates that it lacks the ability of modeling localness effectively. However, our approach corrects this phenomenon effectively (shown by the bar with color blue vs. the bar with color orange).

4 Approach

In this section, we elaborate the proposed framework of improving NAT models via explicitly modeling localness. First of all, convolutional neural networks benefit from effectively capturing local information and have achieved remarkable success in computer vision (Krizhevsky et al., 2012; He et al., 2016). Inspired by this observation, we augment NAT models with temporal convolutions to enhance localness-aware representations. As shown in Figure 2, we stack temporal convolutions on top of the representations of word embedding. Let $\epsilon \in \mathbb{R}^{|V|\times d}$ denote trainable word embedding. The input sentence $\mathbf{x} = \{x_1, \ldots, x_I\}$ is represented as $\mathbf{E} = \{\mathbf{E}_1, \ldots, \mathbf{E}_I\} = \{\epsilon[x_1]+p_1, \ldots, \epsilon[x_I]+p_I\}$, where p_i is a learnable positional embedding. The incorporated multi-layer temporal convolutions encode the localness-aware representations and the representations of the localness-aware encoder are calculated by:

$$\mathbf{H}_{\text{TC}} = \text{ENC}_{\text{TC}}(\mathbf{E}), \quad (5)$$

$$\mathbf{H}_{\text{SAN}} = \text{ENC}_{\text{SAN}}(\mathbf{H}_{\text{TC}}), \quad (6)$$

where $\text{ENC}_{\text{TC}}(\cdot)$ represents multi-layer temporal convolutions, and $\text{ENC}_{\text{SAN}}(\cdot)$ indicates self-attention networks.

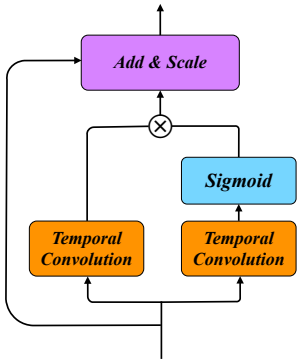


Figure 3: The architecture of temporal convolutions. The add operation indicates the residual connection. The sigmoid and multiplicative boxes demonstrate gated linear units.

Temporal Convolutional Structure In this paper, we employ gated temporal convolutional networks to modeling localness, which is shown in Figure 3. Let $\mathbf{h}^l_{i:j}$ refer to the concatenation of words $\{\mathbf{h}^l_i, \ldots, \mathbf{h}^l_j\}$. A temporal convolution operation over a sequence involves a *filter kernel* $\{W, W_g\} \in \mathbb{R}^{d \times kd}$ and *bias* $\{b, b_g\} \in \mathbb{R}^d$. Following gated linear units (GLU) (Dauphin et al., 2017), the feature $\mathbf{h}'^{l+1}_{(i+j)/2}$ is generated by mapping $\mathbf{h}^l_{i:j}$ to a vector with dimension d:

$$\mathbf{h}'^{l+1}_{(i+j)/2} = (W \cdot \mathbf{h}^l_{i:j} + b) \otimes \sigma(W_g \cdot \mathbf{h}^l_{i:j} + b_g),$$

where $(i + j)/2$ is derived by i and j, and $\otimes$ denotes point-wise multiplication. The gate $\sigma(\cdot)$ is the sigmoid function and it controls the relevance of current context to the inputs.

Inspired by previous works (Dou et al., 2018; Bapna and Firat, 2019), we propose to combine the localness-aware representations with previous representations. Specifically, conditioned on the input latent representations, we employ a residual connection (He et al., 2016) to generate representations: $\mathbf{h}^{l+1}_i = (\mathbf{h}'^{l+1}_i + \mathbf{h}^l_i) \times s$, where we normalize the output by a factor $s = \sqrt{0.5}$ to lower the variance of the sum.

Incorporating into Transformer As a feature extractor, the encoder in NAT models is dealing with a more sophisticated task and takes a more important role than the decoder regarding the translation quality (Guo et al., 2020b). Therefore, it is natural to incorporate multi-layer temporal convolutions into the encoder. Besides, compared with the conventional decoder in autoregressive models, NAT models avoid the causal mask (Gu et al., 2018) in self-attention networks (SAN). Therefore, an identical temporal convolutional structure can be introduced into the decoder accordingly. In our implementation, we use multi-layer stacked temporal convolutions to model localness-aware representations. The presented approach is model-agnostic and can be applied in different NAT models. In this paper, we mainly implement the proposed method on top of advanced CMLM (Ghazvininejad et al., 2019) and Vanilla-NAT (Gu et al., 2018) models.

5 Experiments

5.1 Setup

Datasets We conduct experiments on two typical benchmark datasets: WMT14 English-German (En-De)[1] and WMT16 English-Romanian (En-Ro)[2], which consist of 4.0M and 610K sentence pairs respectively. We strictly follow the dataset settings as previous works (Gu et al., 2018; Lee et al., 2018). Specifically, for the WMT14 En-De translation task, we use newstest2013 and newstest2014 as the validation and test set respectively. For the WMT16 En-Ro task, newsdev2016 and newstest2016 are used as the validation and test set. We follow the tokenization strategy from the translation example of fairseq[3]. We preprocess our data using byte-pair encoding (Sennrich et al., 2016) for both translation tasks, and learn the shared vocabulary with the joint training corpus in both source and target sides. For evaluation, we use 4-gram BLEU score (Papineni et al., 2002) as the evaluation metric for all language pairs.

Knowledge Distillation As a key ingredient, knowledge distillation (KD) (Zhou et al., 2020) has been proven to reduce the complexity of target data and benefits the training of NAT models efficiently. Following previous works (Gu et al., 2018; Lee et al., 2018), we apply sequence-level knowledge distillation (Kim and Rush, 2016) to generate the training data for NAT models. Specifically, for each sentence pair in a parallel training corpus, we replace the target sentence with the generated translation from a pre-trained autoregressive model. We follow Ghazvininejad et al. (2019) to decode the entire training set for both language pairs.

[1] https://www.statmt.org/wmt14/translation-task
[2] https://www.statmt.org/wmt16/translation-task
[3] https://github.com/pytorch/fairseq/blob/master/examples/translation/prepare-wmt14en2de.sh

#	Model	WMT14 En⇒De	WMT14 De⇒En	WMT16 En⇒Ro	WMT16 Ro⇒En
Autoregressive models					
1	Transformer (Vaswani et al., 2017)	27.74	31.09	34.28	33.99
Fully Non-Autoregressive models					
2	Vanilla-NAT (Gu et al., 2018)	17.69	21.47	27.29	29.06
3	FCL-NAT (Guo et al., 2020a)	25.75	29.50	–	–
4	ReorderNAT (Ran et al., 2021)	22.79	27.28	29.30	29.50
5	Flowseq (Ma et al., 2019)	23.72	28.39	29.73	30.72
6	AXE (Ghazvininejad et al., 2020)	23.53	27.90	30.75	31.54
7	Bag-of-ngrams (Shao et al., 2020)	20.90	24.60	28.30	29.30
8	EM+ODD (Sun and Yang, 2020)	24.54	27.93	–	–
9	Imitate-NAT (Wei et al., 2019)	24.15	27.28	31.45	31.81
10	GLAT (Qian et al., 2021)	25.21	29.84	31.19	32.04
Iterative Non-Autoregressive models					
11	Iter-NAT (Lee et al., 2018)	21.61	25.48	29.32	30.19
12	LaNMT (Shu et al., 2020)	26.30	–	–	29.10
13	CMLM (Ghazvininejad et al., 2019)	27.03	30.53	33.08	33.31
Our work					
14	CMLM (Reimp.)	26.73	30.33	33.02	33.39
15	+ MTC	**27.44**	**31.09**	**34.03**	**34.39**

Table 1: Evaluation of translation performance on the test sets of WMT14 En-De and WMT16 En-Ro. "Reimp." indicates the results of corresponding models obtained by our implementation. "+" denotes appending new features to the above row. "–" means not reported.

Model Configuration For model hyper-parameters, we mainly follow the configurations in (Lee et al., 2018; Gu et al., 2019). Specifically, for both translation tasks, we use the hyper-parameters of *transformer-base* ($d_{\text{model}} = 512$, $d_{\text{hidden}} = 2048$, $n_{\text{layer}} = 6$, $n_{\text{head}} = 8$, $p_{\text{dropout}} = 0.3$). We employ $t_{\text{warmup}} = 10,000$ as the warm-up learning rate schedule. In our implementation, the kernel size of temporal convolution is set to 3. We use weight decay 0.01 as well as label smoothing 0.1. We implement our approach with the open-source toolkit - fairseq (Ott et al., 2019)[4]. All the models are trained for 300K updates on 8 NVIDIA Tesla V100 GPUs with a batch size of 128K tokens using Adam optimizer (Kingma and Ba, 2015).

5.2 Results

Overall Results We evaluate the proposed NAT-MTC approach with the standard practice of knowledge distillation on WMT14 En-De and WMT16 En-Ro datasets. Tabel 1 shows the results of our models and previous non-autoregressive baselines. As seen, our re-implementation (Row 14) of CMLM model achieves comparable performance with the original ones (Ghazvininejad et al., 2019) across all translation tasks, which makes the evaluation convincing in this work. Further, our approach (Row 15) can outperform the strong baseline models (Row 14) by 0.71 and 0.76 BLEU points on En⇒De and De⇒En respectively. Encouragingly, on the En⇒Ro task, our model achieves a significant improvement by up to 1.01 BLEU points. These results clearly demonstrate the effectiveness of explicitly modeling localness for NAT models.

Effects of Model Capacity To rule out that the improvement is due to higher modeling capacity, we conduct experiments on matching the number of parameters of our proposed model by adding more self-attention layers to the original CMLM. The results in Table 2 show that although adding self-attention layers has more parameters than our model (88.2M vs. 85.0M), our approach achieves significant improvements (27.44 vs. 26.98). This

[4]https://github.com/pytorch/fairseq

Model	#Enc	#Dec	#Para.	BLEU
Transformer	6	6	64.9M	27.74
CMLM	6	6	66.1M	26.73
	8	8	80.8M	26.87
	9	9	88.2M	26.98
+ MTC	6	6	85.0M	27.44

Table 2: Effects of model capacity on the WMT14 En⇒De task. "#Enc" and "#Dec" indicates the number of self-attention layers respectively. "#Para." denotes the number of trainable parameters.

clearly demonstrates that the improvement of translation quality is due to the inductive bias brought by the architecture modification.

Different Model Architectures To verify the universality of our approach, we also implement our method on top of the non-iterative Vanilla-NAT model (Gu et al., 2018) and the results on WMT14 En-De and WMT16 En-Ro are shown in Table 3. For a fair comparison, we re-implement the Vanilla-NAT model. As seen, our re-implementations greatly outperform original Vanilla-NAT reported by Gu et al. (2018), which makes our evaluation convincing. For instance, compared with the original implementation, our Vanilla-NAT achieves an increase of 2.45 BLEU points on De⇒En task. However, our approach shows a further improvement by 0.91 BLEU points. In particular, on Ro⇒En task, the presented method (+MTC) obtains a significant improvement by 1.43 BLEU scores. This demonstrates the effectiveness and universality of our approach.

Model	En⇒De	De⇒En	En⇒Ro	Ro⇒En
Vanilla-NAT*	17.69	21.47	27.29	29.06
Vanilla-NAT (Reimp.)	19.05	23.92	29.65	28.88
+ MTC	20.02	24.83	30.43	30.31

Table 3: Translation performance on test sets of WMT14 En-De and WMT16 En-Ro. "*" indicates that the results are provided by Gu et al. (2018).

Effects of Decoding Speed To investigate the effects of our approach on decoding speed, we run all models with one sentence at a time on a single GPU and calculate the inference latency on the WMT14 En-De task. The results are shown in Table 4. In contrast to the respective backbone, our model achieves a significant improvement with a very small overhead (2.6× vs. 2.8× for CMLM, 17.4× vs. 17.8× for Vanilla-NAT). This indicates that our approach can efficiently improve the performance of NAT models.

Model	BLEU	Speed
Transformer	27.74	1.0×
CMLM	26.73	2.8×
+ MTC	27.44	2.6×
Vanilla-NAT	19.05	17.8×
+ MTC	20.02	17.4×

Table 4: Decoding speed on the WMT14 En⇒De task.

6 Analysis

In this section, we conduct extensive analyses on the WMT14 En-De translation task to better understand our model in terms of: 1) effects of different strategies, 2) effects of the sentence length, 3) convergence speed, 4) effects of the number of decoding iterations, 5) effects of predicted length candidates, and 6) case study.

#	Encoder	Decoder	BLEU	Δ
1	×	×	26.73	–
2	✓	✓	27.44	+0.71
3	✓	×	27.37	+0.64
4	×	✓	26.87	+0.14

Table 5: Effects of different strategies of incorporating MTC into NAT on the WMT14 En⇒De task.

Effects of Different Strategies We perform experiments on WMT14 En⇒De to investigate the effects of different incorporating strategies, which are shown in Table 5. Specifically, we enumerate the translation results of three strategies, namely introducing MTC into: 1) encoder (Row 3), 2) decoder (Row 4), and 3) both encoder and decoder (Row 2). As seen, augmenting the encoder with MTC improves more BLEU scores than decoder (+0.64 vs. +0.14), and incorporating MTC into both encoder and decoder sides accumulatively achieves the best translation performance (+0.71 BLEU, Row 2). This demonstrates that augmenting encoder with generating localness-aware representations is critical in NAT models and validates the significance of encoder, which was also found by Guo et al. (2020b).

Effects of Sentence Length We investigate the translation results of CMLM and our approach on

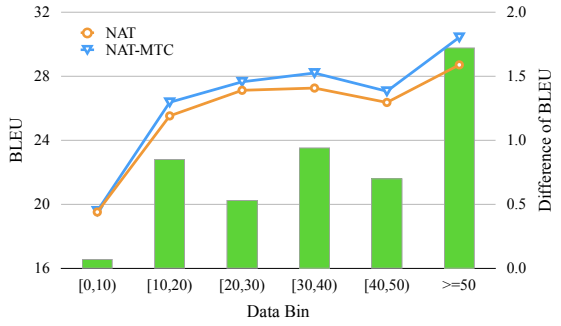
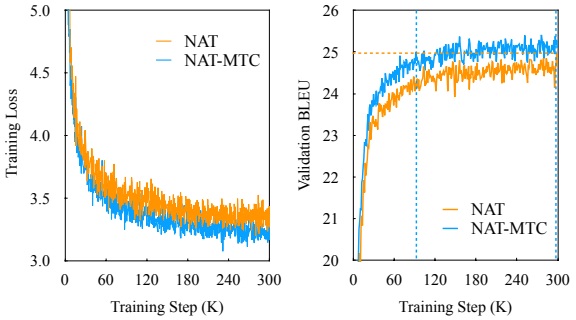


Figure 4: Translation performance on the WMT14 En⇒De test set with respect to different lengths of target sentences. The left axis denotes the BLEU scores of NAT and NAT-MTC models, while the right axis indicates the difference of BLEU scores for both models.

the WMT14 En⇒De task with respect to different lengths of target sentences, which is shown in Figure 4. Specifically, the translations are allocated into distinct buckets based on the respective lengths of corresponding reference sentences and then we evaluate the BLEU scores for each bucket. For comparison, we also show the relative change of BLEU score between NAT-MTC and NAT models. As expected, the presented approach achieves improvements over the baseline system across all buckets. In particular, for longer sentences (≥ 50), our model improves by 1.72 BLEU points. This clearly reveals that explicitly modeling localness effectively benefits long-distance dependencies in NAT models.



(a) Training Loss (b) Validation BLEU

Figure 5: Learning curves on the WMT14 En⇒De translation task.

Convergence Speed We present the training process to observe the effects of our approach on optimization, which is shown in Figure 5. As seen, our approach converges faster concerning the training loss and validation BLEU score. In particular, the translation performance is significantly boosted during training. Specifically, the NAT model achieves the best BLEU score at 298K training step (indicated by a vertical line with the color yellow in Figure 5b). However, our model achieves an equivalent BLEU point at 92K training steps (indicated by a vertical line with the color blue in Figure 5b). Therefore, our approach largely benefits convergence speed in training and can achieve comparable translation accuracy with a reduction of 70% training steps.

Iter.	BLEU			Rep.		
	NAT	NAT-MTC	Δ	NAT	NAT-MTC	Δ
2	23.12	24.39	+1.27	2.43%	2.08%	-0.35%
4	25.62	26.69	+1.07	0.58%	0.45%	-0.13%
6	26.43	27.16	+0.73	0.31%	0.24%	-0.07%
8	26.45	27.21	+0.76	0.22%	0.16%	-0.06%
10	26.73	27.44	+0.71	0.16%	0.13%	-0.03%

Table 6: The translation performance and percentage of repeating words on the WMT14 En⇒De task, varied by the number of decoding iterations.

Effects of Iteration We study the translation performance and repetitive words as the number of decoding iterations changes, which is shown in Table 6. Specifically, the percentage of repetitive words is defined as $\beta = C_{rep}/C$, where C_{rep} enumerates the number of repetitive words (refer to words, which are equivalent to adjacent words) and C indicates the total number of words within decoding sentences. As seen, with the increase of the number of iterations, our model improves the translation performance, while reducing repetitive words. Besides, our approach reduces more repeating words (-0.35%) while obtaining more improvements in translation performance (+1.27 BLEU). This clearly confirms that our approach benefits the reduction of repetitive translations and alleviates the multi-modality problem in NAT models.

#Cands.	NAT	NAT-MTC	Δ
1	26.08	27.26	+1.18
2	26.04	27.32	+1.28
3	26.73	27.51	+0.78
4	26.50	27.40	+0.90
5	26.73	27.44	+0.71

Table 7: Translation performance on the WMT14 En⇒De task. "#Cands." indicates the number of candidates with different predicted length.

Effects of Length Candidates We present the translation performance with respect to the change

Source	Er selbst war im Jahr 2004 das erste Mal in Mauretanien im Cheijk-Zajed-Krankenhaus in Nouakchott.
Target	In 2004, he visited the **Cheijk-Zajed Hospital** in Nouakchott in Mauritania *for the first time*.
NAT	He himself was in Mauritania at the **Cheijk Zajed Cheijk Hospital** in Nouakchott in 2004.
NAT-MTC	He himself was in Mauritania *for the first time* at the **Cheijk-Zajed Hospital** in Nouakchott in 2004.

Table 8: A case study on the WMT14 De⇒En translation task. Phrases formatted as bold or underline indicate the problem of repetitive and incomplete translations in the baseline but fixed by our model.

of predicted length candidates. As shown in Table 7, when the number of candidates is reduced to one, our model drops by fewer BLEU scores compared with the NAT model (-0.18 vs. -0.65). In addition, our model performs extremely stable with the change of length candidates. These observations verify the robustness of our approach.

Case Study We further carry out a case study to intuitively illustrate the performance of our approach and baseline. Table 8 shows an example randomly selected from the test set on the WMT14 De⇒En translation task. As seen, introducing the mechanism of generating localness-aware representations into NAT can produce more fluent and adequate translations. For instance, the German words "*das erste Mal*" are ignored by baseline, while the NAT-MTC model accurately translates it into "*for the first time*". Besides, NAT tends to generate repetitive words (e.g., "*Cheijk Zajed Cheijk Hospital*"), while our model corrects this issue. This demonstrates that our model can comprehensively generate localness-aware representations in terms of words, phrases and patterns.

7 Related Work

Fully Non-Autoregressive Models Gu et al. (2018) first introduced non-autoregressive machine translation, which enables the sequence generation in parallel and reduces the inference latency significantly. Specifically, through sequence-level knowledge distillation (Kim and Rush, 2016) and modeling fertility as a latent variable, it maintains a relatively competitive translation quality as opposed to the autoregressive Transformer. The idea of modeling dependency as latent variables has been investigated extensively (Kaiser et al., 2018; Sun and Yang, 2020; Gu and Kong, 2021; Qian et al., 2021; Du et al., 2021). Kaiser et al. (2018) proposed to model a shorter sequence as discrete latent variables, which are generated autoregressively. Subsequently, this short latent sequence is utilized to decode the output sequence in parallel. In addition, a glancing mechanism with adaptively sampling words from the reference (Qian et al., 2021) was exploited to improve the translation performance of non-iterative NAT.

Non-Autoregressive Models with Iterative Refinement To alleviate the multi-modality problem, a line of researches (Lee et al., 2018; Stern et al., 2018; Ghazvininejad et al., 2019; Gu et al., 2019; Saharia et al., 2020; Ding et al., 2021, 2022; Huang et al., 2022) introduce an iterative refinement process to maintain the translation accuracy. Lee et al. (2018) first presented an iterative approach, which interprets the entire model as a latent variable and each refinement step as a denoising process. In addition, Stern et al. (2018) proposed to make predictions for multiple time steps by introducing a blockwise parallel decoding scheme. Inspired by the pretraining approach (Devlin et al., 2019; Lample and Conneau, 2019), Ghazvininejad et al. (2019) utilized a masked language modeling to predict any subset of the target sentence, which is based on both the source sentence and a generated translation with partially masking.

8 Conclusion

In this paper, we propose to improve NAT models via explicitly modeling localness. First of all, we conduct a thorough empirical study on the attention mechanism and reveal that compared with autoregressive models, existing NAT models lack the ability to effectively modeling local information. Furthermore, we incorporate temporal con-

volutions into both encoder and decoder sides to enhance localness-aware representations in NAT models. Empirical results on a variety of language pairs and two advanced NAT models demonstrate the effectiveness and universality of our approach. Further analyses confirm that the proposed method benefits translations for long sentences and accelerates convergence during training.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *EMNLP*.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*.

Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *ICML*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Liang Ding, Longyue Wang, Xuebo Liu, Derek F Wong, Dacheng Tao, and Zhaopeng Tu. 2021. Progressive multi-granularity training for non-autoregressive translation. In *ACL*.

Liang Ding, Longyue Wang, Shuming Shi, Dacheng Tao, and Zhaopeng Tu. 2022. Redistributing low-frequency words: Making the most of monolingual data in non-autoregressive translation. In *ACL*.

Liang Ding, Longyue Wang, Di Wu, Dacheng Tao, and Zhaopeng Tu. 2020. Context-aware cross-attention for non-autoregressive translation. In *COLING*.

Zi-Yi Dou, Zhaopeng Tu, Xing Wang, Shuming Shi, and Tong Zhang. 2018. Exploiting deep representations for neural machine translation. In *EMNLP*.

Cunxiao Du, Zhaopeng Tu, and Jing Jiang. 2021. Order-agnostic cross entropy for non-autoregressive machine translation. In *ICML*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *ICML*.

Marjan Ghazvininejad, Vladimir Karpukhin, Luke Zettlemoyer, and Omer Levy. 2020. Aligned cross entropy for non-autoregressive machine translation. In *ICML*.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *EMNLP*.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *ICLR*.

Jiatao Gu and Xiang Kong. 2021. Fully non-autoregressive neural machine translation: Tricks of the trade. In *ACL-Findings*.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In *NeurIPS*.

Junliang Guo, Xu Tan, Linli Xu, Tao Qin, Enhong Chen, and Tie-Yan Liu. 2020a. Fine-tuning by curriculum learning for non-autoregressive neural machine translation. In *AAAI*.

Junliang Guo, Linli Xu, and Enhong Chen. 2020b. Jointly masked sequence-to-sequence model for non-autoregressive neural machine translation. In *ACL*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.

Xiao Shi Huang, Felipe Pérez, and Maksims Volkovs. 2022. Improving non-autoregressive translation models without distillation. In *ICLR*.

Lukasz Kaiser, Samy Bengio, Aurko Roy, Ashish Vaswani, Niki Parmar, Jakob Uszkoreit, and Noam Shazeer. 2018. Fast decoding in sequence models using discrete latent variables. In *ICML*.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. In *EMNLP*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *NeurIPS*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. In *NeurIPS*.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *EMNLP*.

Zhuohan Li, Di He, Fei Tian, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Hint-based training for non-autoregressive translation. In *EMNLP*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP*.

Xuezhe Ma, Chunting Zhou, Xian Li, Graham Neubig, and Eduard Hovy. 2019. Flowseq: Non-autoregressive conditional sequence generation with generative flow. In *EMNLP*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *NAACL-HLT: Demonstrations*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu, and Lei Li. 2021. Glancing transformer for non-autoregressive neural machine translation. In *ACL*.

Qiu Ran, Yankai Lin, Peng Li, and Jie Zhou. 2021. Guiding non-autoregressive neural machine translation decoding with reordering information. In *AAAI*.

Chitwan Saharia, William Chan, Saurabh Saxena, and Mohammad Norouzi. 2020. Non-autoregressive machine translation with latent alignments. In *EMNLP*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *ACL*.

Chenze Shao, Jinchao Zhang, Yang Feng, Fandong Meng, and Jie Zhou. 2020. Minimizing the bag-of-ngrams difference for non-autoregressive neural machine translation. In *AAAI*.

Raphael Shu, Jason Lee, Hideki Nakayama, and Kyunghyun Cho. 2020. Latent-variable non-autoregressive neural machine translation with deterministic inference using a delta posterior. In *AAAI*.

Mitchell Stern, Noam Shazeer, and Jakob Uszkoreit. 2018. Blockwise parallel decoding for deep autoregressive models. In *NeurIPS*.

Zhiqing Sun and Yiming Yang. 2020. An em approach to non-autoregressive conditional sequence generation. In *ICML*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NeurIPS*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

Yiren Wang, Fei Tian, Di He, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2019. Non-autoregressive machine translation with auxiliary regularization. In *AAAI*.

Bingzhen Wei, Mingxuan Wang, Hao Zhou, Junyang Lin, and Xu Sun. 2019. Imitation learning for non-autoregressive neural machine translation. In *ACL*.

Baosong Yang, Zhaopeng Tu, Derek F. Wong, Fandong Meng, Lidia S. Chao, and Tong Zhang. 2018. Modeling localness for self-attention networks. In *EMNLP*.

Chunting Zhou, Jiatao Gu, and Graham Neubig. 2020. Understanding knowledge distillation in non-autoregressive machine translation. In *ICLR*.