

Revisiting Non-Autoregressive Translation at Scale

Zhihao Wang^{1,3*}, Longyue Wang^{2*}, Jinsong Su^{1,3†}, Junfeng Yao³, Zhaopeng Tu²

¹School of Informatics, Xiamen University, China

²Tencent AI Lab, China

³Key Laboratory of Digital Protection and Intelligent Processing of Intangible Cultural Heritage of Fujian and Taiwan (Xiamen University), Ministry of Culture and Tourism, China

zhwang@stu.xmu.edu.cn {vinnylywang, zptu}@tencent.com

{jssu, yao0010}@xmu.edu.cn

Abstract

In real-world systems, scaling has been critical for improving the translation quality in autoregressive translation (AT), which however has not been well studied for non-autoregressive translation (NAT). In this work, we bridge the gap by systematically studying the impact of scaling on NAT behaviors. Extensive experiments on six WMT benchmarks over two advanced NAT models show that scaling can alleviate the commonly-cited weaknesses of NAT models, resulting in better translation performance. To reduce the side-effect of scaling on decoding speed, we empirically investigate the impact of NAT encoder and decoder on the translation performance. Experimental results on the large-scale WMT20 En-De show that the asymmetric architecture (e.g. bigger encoder and smaller decoder) can achieve comparable performance with the scaling model, while maintaining the superiority of decoding speed with standard NAT models. To this end, we establish a new benchmark by validating scaled NAT models on the scaled dataset, which can be regarded as a strong baseline for future works. We release code and system outputs at <https://github.com/DeepLearnXMU/Scaling4NAT>.

1 Introduction

Recent years have seen a surge of interest in non-autoregressive translation (NAT) (Gu et al., 2018), which can improve the decoding efficiency by predicting all tokens independently and simultaneously. The majority studies on NAT focus on the base models trained on medium-scale datasets (e.g., Mask-Predict: 69M; WMT14 En-De: 4.5M) (Ghazvininejad et al., 2019), while scaled models and datasets become de facto standard for autoregressive translation (AT) models (e.g., Transformer-Big: 226M; WMT20 En-De:

45.1M) (Ott et al., 2018). The model- and data-level gaps make the progress of NAT lag behind that of AT, which limits the applicability of NAT models to practical scenarios.

This general tendency motivates us to boost NAT models from the scaling perspective, including the amounts of training data and the model size. In this paper, we aim to provide empirical answers to the following research questions:

- **RQ1:** *How does scaling affect NAT behaviours in terms of translation quality and decoding speed?* Scaling neural networks brings dramatic quality gains over translation tasks using AT models (Ariavzhagan et al., 2019), and revisiting existing methods on a large-scale data can obtain more consistent conclusions (Edunov et al., 2018).
- **RQ2:** *Have performance improvements of scaling been accompanied by alleviating commonly-cited weaknesses of NAT?* Several weaknesses exist in NAT, including multimodality problem (Gu et al., 2018), non-fluent outputs (Du et al., 2021) and inadequate translations (Ding et al., 2021c).
- **RQ3:** *Can we establish a new NAT benchmark to reliably translate leaderboard scores to improvements in real-world use of the models?* Although previous studies of NAT have achieved comparable performance with the AT models, they are still validated on small-scale datasets and model sizes using inconsistent evaluation criteria. These gaps make the progress of NAT lag behind that of AT, which limits the applicability of NAT models to practical scenarios.

To answer these research questions, we investigate the effects of different scaling methods on two advanced NAT models. Experimental results show that scaling works well with knowledge distillation to alleviate commonly-cited weaknesses of NAT. The scaled NAT models achieve better translation quality at the expense of decreasing decoding speed. To balance effectiveness and efficiency, we com-

*Equal Contribution.

†Corresponding Author.

pare various component-scaled NAT models and find that scaling architecture in NAT is more asymmetric than that in AT. Accordingly, we introduce a cone architecture for NAT with a deeper and wider encoder and a shallower and narrower decoder, which boosts translation performance and maintain the decoding speed. Specifically, our **main contributions** are as follows:

- We demonstrate the necessity of scaling model and data for NAT models, which narrows the progress gap between NAT and AT models.
- Our study reveals positive effects of scaling on commonly-cited weakness, which makes the standard NAT model sub-optimal.
- We establish a new benchmark, where we evaluate competing scaled NAT models on large-scale datasets in terms of effectiveness and efficiency.
- We provide a better understanding of NAT at scale to help prioritize future exploration towards making NAT a common translation framework.

2 Preliminary

2.1 Non-Autoregressive Translation

Given a source sentence $\mathbf{x} = \{x_1, x_2, \dots, x_{T_X}\}$, an AT model generates each target word \mathbf{y}_t conditioned on previously generated ones $\mathbf{y}_{<t}$, leading to high latency on the decoding stage. In contrast, NAT models break this *autoregressive factorization* by producing all target words independently and simultaneously. Formally, the probability of generating $\mathbf{y} = \{y_1, y_2, \dots, y_{T_Y}\}$ is computed as: $p(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{T_Y} p(\mathbf{y}_t|\mathbf{x}; \theta)$ where T_Y is the length of the target sequence, which is usually predicted by a separate conditional distribution. The parameters θ are trained to maximize the likelihood of a set of training examples according to $\mathcal{L}(\theta) = \arg \max_{\theta} \log p(\mathbf{y}|\mathbf{x}; \theta)$.

Knowledge Distillation Training NAT suffers from the *multimodality problem*, where the conditional independence assumption prevents a model from properly capturing the highly multimodal distribution of target translations (Gu et al., 2018). Accordingly, the sequence-level knowledge distillation (Kim and Rush, 2016) is introduced to reduce the modes of training data by replacing their original target-side samples with sentences generated by an AT teacher (Gu et al., 2018; Zhou et al., 2020). Formally, the original parallel data D_{Raw} and the distilled data D_{KD} can be defined as $D_{\text{Raw}} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ and $D_{\text{KD}} = \{(\mathbf{x}_i, f_{s \rightarrow t}(\mathbf{x}_i)) | \mathbf{x}_i \in$

$D_{\text{Raw}}\}_{i=1}^N$, where $f_{s \rightarrow t}$ represents an AT model trained on D_{Raw} for translating sentences from the source to the target language. N is the total number of sentence pairs in the training data.

2.2 Advanced Models

The conditional independence assumption results in a performance gap between the NAT model and its AT teacher. A number of recent efforts have explored ways to bridge the performance gap with advanced architectures (Ghazvininejad et al., 2019; Gu et al., 2019; Ding et al., 2020) or training objectives (Shao et al., 2019; Ghazvininejad et al., 2020; Du et al., 2021). Another thread of work focuses on understanding and improving distillation training (Zhou et al., 2020; Ding et al., 2021c; Huang et al., 2022; Ding et al., 2021a,b, 2022). Generally, NAT models can be divided into two categories:

Iterative NAT is proposed to refine previously generated words in each iteration, which allows NAT models to generate target words by capturing partial and noisy dependencies. Mask-Predict (MaskT) (Ghazvininejad et al., 2019) uses the conditional masked language model (Devlin et al., 2019) to iteratively generate the target sequence from the masked input. Levenshtein Transformer (Gu et al., 2019) introduces three steps: deletion, placeholder prediction and token prediction, and the decoding iterations adaptively depend on certain conditions.

Fully NAT is trained to produce one-pass decoding without sacrifice of speed-up. Several studies have been proposed to improve the fully NAT models (Qian et al., 2021; Gu and Kong, 2021). GLAT adopts an adaptive glancing sampling strategy for training, which can be seen as a method of curriculum learning. Furthermore, Gu and Kong (2021) build a new SOTA fully NAT model by combining useful techniques in four perspectives, including training data, model architecture, training objective and learning strategy.

2.3 Experimental Setup

Datasets We not only experiment on the widely-used WMT16 English-Romanian (0.6M) and WMT14 English-German (4.5M) benchmarks, but also broaden the investigation on a large-scale dataset WMT20 English-German (45.1M). We tokenize data using the Moses toolkit, and then split them into subwords using a joint BPE (Sennrich et al., 2016) with 32K merge operations. This

forms a shared vocabulary of 32k, 37k, and 49k for WMT16 En-Ro, WMT14 En-De and WMT20 En-De respectively. Both AT and NAT models are trained on KD data, except as otherwise noted. To generate KD data, we employ Transformer-Big and Transformer-Base as teachers to distill the En-De and En-Ro datasets, respectively.

NAT Models We validate two advanced models, representing iterative and fully NAT respectively:

- *MaskT* (Ghazvininejad et al., 2019) where we follow its optimal settings to keep the iteration number be 10 and length beam be 5.
- *GLAT* (Qian et al., 2021) where we follow their reported configurations to set iteration number and length beam as 1.

Models are re-implemented on top of the Fairseq framework (Ott et al., 2019), which supports training on multiple GPU instances. We employ *large-batch training* (i.e. 480K tokens/batch) to optimize the performance (Ott et al., 2018). We train all NAT models for 300K steps to ensure adequate training, apart from WMT16 En-Ro (30K steps). Following the common practices (Ghazvininejad et al., 2019; Kasai et al., 2020), we evaluate the performance on an ensemble of 5 best checkpoints (ranked by validation BLEU) to avoid stochasticity. More details about NAT training are presented in Appendix A.6.

AT Teachers We closely follow previous works on NAT to apply sequence-level knowledge distillation to reduce the modes of the training data. We trained BASE and BIG Transformer (Vaswani et al., 2017) as the *AT teachers* for En \leftrightarrow Ro and En \leftrightarrow De tasks, respectively. We adopt *large-batch training* (i.e. 458K tokens/batch) to optimize the performance of AT teachers (Ott et al., 2018). Specially, the AT teachers are trained on raw data.

Evaluation For fair comparison, we use case-insensitive tokenBLEU (Papineni et al., 2002) to measure the translation quality on WMT16 En-Ro and WMT14 En-De. We use SacreBLEU (Post, 2018) for the new benchmark WMT20 En-De.

3 Scaling Behaviors of NAT Models

We investigate effects of scaling from three perspectives: translation quality, commonly-cited weaknesses and decoding efficiency. The settings are:

- *Model Scaling*: Based on traditional NAT-Base configurations ((6,6) \times 512), we conduct 1) Width Scaling, where the size of feed-forward

dimensions are enlarged to 1024 (NAT-Big: (6,6) \times 1024); 2) Depth Scaling, where the number of stacked encoder-decoder layers is increased to 24-24 (NAT-Deep: (24,24) \times 512)). We mainly investigate behaviors of NAT-Big as it has similar performance with NAT-Deep, while the training of NAT-Big is more stable.

- *Data Scaling*: The commonly-used datasets for NAT are WMT16 En-Ro and WMT14 En-De, whose sizes are smaller than current AT benchmarks. We mainly experiment NAT models on the WMT20 En-De dataset, which is 10 times larger than previous ones (i.e. WMT16: 0.6M; WMT14: 4.5M; WMT20 45.1M).

3.1 Translation Quality

Results on Benchmarks Table 1 lists the results on the six benchmarks: WMT16 En \leftrightarrow Ro, WMT14 En \leftrightarrow De, and WMT20 En \leftrightarrow De, which are small-, medium- and large-scale datasets, respectively. We experiment MaskT and GLAT models of whose configurations are detailed in Section 2.3. Compared with standard NAT models (“+ Knowledge Distillation”), scaling method (“+ Both”) significantly and consistently improves translation performance (BLEU \uparrow) on medium and large-scale datasets. However, the improvement is not robust on small-scale dataset. An interesting finding is that both model scaling and data scaling are able to narrow the performance gap between fully and iterative NAT models. After model scaling, the average difference between MaskT and GLAT drops from +1.2 (“+ Knowledge Distillation” lines) to +0.5 (“+ Both” lines). Encouragingly, advanced NAT models with model-scaling can perform better than strong AT teachers on larger-scale data. As seen, the performance of “MaskT+Both” is +0.5 higher than the Transformer-Big models on WMT20 En \leftrightarrow De. *This confirms the necessity of scaling model size and data for building practical and robust NAT systems.*

Complementary between Scaling and KD KD is a commonly-used training recipe to boost NAT performance. As shown in Table 1, KD (“+ Knowledge Distillation”) can benefit more for fully NAT than iterative NAT models compared with Raw (+4.1 vs. +2.3 BLEU scores averagely). We also find that KD is more effective on large-scale datasets, where the average improvements are +4.7 and +2.5 on WMT20 and WMT16+14, respectively. This reconfirms the effectiveness of KD training

Model	Iter.	Size	W16 (0.6M)		W14 (4.5M)		W20 (45.1M)	
			En-Ro	Ro-En	En-De	De-En	En-De	De-En
AT Models								
Transformer-Base (<i>En↔Ro teacher</i>)	n/a	69M	33.9	34.1	-	-	-	-
Transformer-Big (<i>En↔De teacher</i>)	n/a	226M	-	-	29.2	32.0	32.4	41.7
Existing NAT Models								
AXE (Ghazvininejad et al., 2020)	1	69M	30.8	31.5	23.5	27.9	n/a	n/a
Fully-NAT (Gu and Kong, 2021)	1	70M	33.8	33.9	27.5	31.1	n/a	n/a
DisCo (Kasai et al., 2020)	5	69M	33.2	33.3	27.3	31.3	n/a	n/a
Imputer (Saharia et al., 2020)	5	69M	34.4	34.1	28.2	31.8	n/a	n/a
CMLMC (Huang et al., 2022)	10	73M	34.6	34.1	28.4	31.4	n/a	n/a
Our NAT Models								
MaskT with <i>iterative decoding</i>		69M	33.9	33.6	24.7	29.1	27.2	36.6
+ Knowledge Distillation	10	69M	34.8	33.8	27.5	31.1	31.3	40.6
+ Width Scaling (i.e., NAT-Big)		226M	34.6	33.2	24.9	29.6	30.2	38.7
+ Both		225M	34.7	34.0	28.2	31.2	32.9	42.1
GLAT with <i>fully decoding</i>			71M	30.0	31.2	19.3	26.7	24.0
+ Knowledge Distillation	1	70M	32.3	32.6	26.2	30.3	30.6	40.0
+ Width Scaling (i.e., NAT-Big)		230M	32.0	32.3	21.8	27.6	28.8	38.6
+ Both		229M	34.5	34.2	27.4	30.9	32.3	41.1

Table 1: Translation performance (BLEU \uparrow) of NAT models on translation tasks with different data sizes. “Iter.” indicates the number of iterations while “Size” shows the number of model parameters. “+ Width Scaling” denotes scaled NAT architecture similar to Transformer-Big (i.e., (6,6) \times 1024). “+ Knowledge Distillation” represents training models on KD data instead of the original ones.

especially on large data. The model scaling (“+ Width Scaling”) can also improve NAT models by enhancing the model ability on learning difficult data. The conclusions of model scaling are similar to KD: 1) it benefits more for fully NAT (+1.0 vs. +2.2 BLEU); 2) it is more effective on large-scale datasets (+3.0 vs. +0.9 BLEU). Combining scaling with KD (“+ Both”) can further improve standard MaskT and GLAT (“+ Knowledge Distillation”) by +0.7 and +1.3, which illustrates that they exhibit complementary properties for NAT models. We extensively analyze the reasons behind this in Section 3.3. *Scaling and KD are related to and complement one another for NAT models.* The conclusion on complementary between scaling and KD also holds for depth scaling (detailed in Appendix §A.1). The deep models also have similar performance with big ones, but depth scaling is difficult to train with side effect on inference speed. Therefore, we employ NAT-Big as our testbed in following experiments, unless otherwise specified.

3.2 Difference between NAT and AT Scaling

The scaling behavior of AT models has been studied (Wang et al., 2019a), which seems similar to

NAT in terms of BLEU score. Different from autoregressive Transformer, NAT predicts target tokens independently and simultaneously, which may lead to different scaling behaviors of NAT models. Starting from this intuition, we further compare NAT and AT scaling from the perspective of linguistic properties. Probing tasks (Conneau et al., 2018) can quantitatively measure the linguistic knowledge embedded in encoder representations. We follow Hao et al. (2021) to analyze Base and Big models trained on WMT20 En \rightarrow De KD data. The experimental results on WMT20 En \rightarrow De raw data are also provided in Appendix §A.3.

As depicted in Table 2, scaling improves NAT and AT models on syntactic (+1.4% vs. +1.7%, averagely) and semantic (+0.7% vs. +1.0%, averagely) abilities. However, their behaviors are quite different on surface tasks (-0.1% vs. +12.9%, averagely), which test the ability of preserving global information contained in sentence embeddings. Specifically, scaling improves ability of AT models on “WC” subtask (+18.4%), while this weakens NAT ability (-3.5%). Besides, NAT-Base model preserves more surface information than AT-Base (SeLen: 80.6% vs. 78.1%; WC: 81.3% vs. 55.6%).

Task	MaskT			AT			
	Base	Big	Δ	Base	Big	Δ	
Surface	SeLen	80.6	83.9	+3.3	78.1	85.4	+7.3
	WC	81.3	77.8	-3.5	55.6	74.0	+18.4
Syntactic	TrDep	35.2	36.9	+1.7	35.8	36.9	+1.1
	ToCo	70.8	73.0	+2.2	69.0	72.9	+3.9
	Bshif	49.6	49.9	+0.3	50.1	50.1	0
Semantic	Tense	83.8	85.1	+1.3	84.4	85.5	+1.1
	SubN	79.7	80.9	+1.2	79.7	80.0	+0.3
	ObjN	81.5	82.0	+0.5	80.6	82.1	+1.5
	SoMo	49.9	49.9	0	49.7	49.9	+0.2
	CoIn	53.4	53.9	+0.5	53.0	55.0	+2.0

Table 2: Performance (accuracy \uparrow) of probing tasks for evaluating linguistic properties embedded in the learned representations of AT and NAT models (Width Scaling).

3.3 Analysis on NAT Weaknesses

We analyze effects of scaling on commonly-cited weaknesses: 1) *multimodality* indicated by token repetition ratio (Gu et al., 2018); 2) *generation fluency* calculated by language model (LM) perplexity (Du et al., 2021); 3) *translation adequacy* measured by word translation accuracy (Ding et al., 2021c). Table 3 shows the results. Examples about NAT weaknesses are listed in Appendix A.5.

Scaling Alleviates Multimodality Problem Repeated token percentage is a commonly-used metric of measuring multimodality in a NAT model (Saharia et al., 2020). A NAT model may consider many possible translations at the same time due to the independent predictions of target tokens. Accordingly, the NAT output typically contains some repetitive tokens, especially for fully NAT (1.1% vs. 2.7%). Similar to KD, scaling is an alternative method to significantly reduce the repetition percentage for NAT models (-0.5% and -1.0%). In addition, combining KD and scaling can further alleviate the repetition problem, which is consistent with the translation quality in Table 1.

Scaling Improves Generation Fluency NAT models typically suffer from fluency problems because they only have limited capabilities to model dependencies between the target tokens (Kasner et al., 2020; Gu and Kong, 2021). We measure the fluency of output with a public released LM,¹ which is trained on the News Crawl corpus. The

¹<https://github.com/pytorch/fairseq/tree/main/examples/wmt19>.

Model	Repetition \downarrow		PPL \downarrow		WA \uparrow	
	#	Δ	#	Δ	#	Δ
MaskT	1.1%	-	66	-	71.3%	-
+ KD	0.2%	-0.9%	55	-11	73.0%	+1.7%
+ Scale	0.6%	-0.5%	59	-7	72.2%	+0.9%
+ Both	0.1%	-1.0%	52	-14	73.4%	+2.1%
GLAT	2.7%	-	98	-	70.7%	-
+ KD	1.2%	-1.5%	70	-28	72.6%	+1.9%
+ Scale	1.7%	-1.0%	79	-19	72.1%	+1.4%
+ Both	0.8%	-1.9%	64	-34	73.1%	+2.4%
Golden	0.02%	-	54	-	-	-

Table 3: Analyses of translation outputs generated by NAT models on the WMT20 De \rightarrow En test set. Lower repeated token percentages (“Repetition”) represent lower multimodality in a model. Lower perplexities (“PPL”) denote better fluency while higher word translation accuracy (“WA”) denotes better adequacy. # is the absolute value and Δ is the difference over NAT-Base models. We also list the results of reference (“Golden”).

results show that either KD or scaling can consistently decrease the PPL in all cases (-7 \sim -34). We attribute the improvement of fluency to that KD reduces the learning difficulty by simplifying training data while scaling enhances the model ability by introducing larger parameters. Besides, the complementarity between KD and scaling still holds in terms of fluency measurement. Encouragingly, scaled model without KD performs closely to the standard NAT models, showing that scaling has the potential to directly learn from the raw data of complex modes.

Scaling Enhances Translation Adequacy NAT often suffers from two kinds of adequacy errors which were empirically observed by previous studies: 1) incomplete translation, due to incomplete transfer of source side information (Wang et al., 2019b); 2) lexical choice, due to choosing a target lexeme which inadequately expresses the source meaning (Ding et al., 2021c). Following Neubig et al. (2019), we measure the word accuracy which defined as F-measure of system outputs with respect to the reference. It can demonstrate how much a system over- or under-produces words of a specific type as well. As expected, NAT models with KD or scaling have higher word accuracy (+0.9% \sim +1.9%), resulting in better translation performance (BLEU \uparrow in Table 1). Combining KD and scaling can further improve translation quality by

Model	Size	Speed ₁		Speed _{max}		BLEU
		#	Δ	#	Δ	
MaskT	69M	8.9	-	166	-	31.3
+ Scale	225M	8.4	0.94×	92	0.55×	32.9
GLAT	70M	58.8	-	2160	-	30.6
+ Scale	229M	54.4	0.93×	1772	0.82×	32.3

Table 4: Decoding speed (sentences/s \uparrow) of scaled NAT models (i.e., NAT-Big (6, 6) \times 1024) on the WMT20 En \rightarrow De task. We test the decoding speed when translating one sentence (Speed₁) or hardware-maximum mini-batches (Speed_{max}). # is the absolute value and Δ is the speedup ratio over NAT-Base.

increasing word accuracy (+2.1% \sim +2.4%).

3.4 Discussion on Decoding Efficiency

Although scaling produces significant performance gains, someone may argue that model scaling introduces more parameters, which will increase latency at decoding stage. Following previous studies, we carefully investigate effects of scaling on decoding efficiency for NAT. We employ two metrics:

- *Speed₁*, which measures speed when translating one sentence at a time (Gu et al., 2018). This is used in standard practice and aligns with applications like instantaneous MT that translates text input from users immediately.
- *Speed_{max}*, which measures speed when translating in mini-batches as large as the hardware allows (Kasai et al., 2021). This corresponds to scenarios where one wants to translate a large amount of text given in advance.

As illustrated in Table 4, adding $3\times$ parameters definitely decreases the decoding speed (Speed₁: $0.93\times \sim 0.94\times$ and Speed_{max}: $0.55\times \sim 0.82\times$). In terms of Speed_{max}, scaling harms the iterative NAT more than fully NAT models ($0.55\times$ vs. $0.82\times$). Besides, we test the decoding speed of the MaskT-Deep model ((24, 24) \times 512) and find that Speed₁ rapidly declines to $0.28\times$. These results suggest that scaling method increases translation quality (BLEU \uparrow) at the expense of decoding speed (Speed \downarrow), especially on Speed_{max}.

This findings motivates us to design a better scaling architecture for NAT, taking both performance and time cost into consideration. Kasai et al. (2021) pointed out that some NAT models have little advantage when translating a large amount of text given in advance. Accordingly, we use Speed_{max} as default when discussing translation speed.

Model	Speed _{max}			BLEU		
	AT	MaskT	GLAT	AT	MaskT	GLAT
No Scaling						
Base	1.00×	1.00×	1.00×	33.0	31.3	30.6
Component Scaling						
Enc.	0.99×	0.96×	0.89×	33.5	33.1	32.0
Dec.	0.74×	0.56×	0.85×	33.0	32.5	31.0
Both	0.72×	0.55×	0.82×	34.0	32.9	32.3

Table 5: Translation performance (BLEU \uparrow) and relative decoding speed of component-level width-scaled NAT models on the WMT20 En \rightarrow De. Speed_{max} means speedup ratio of scaled NAT models over Base ones.

4 New NAT Benchmark

Most NAT models are implemented upon the encoder-decoder framework, where the encoder summarizes the source sentence and the decoder learns to generate target words. We ask: *how to scale this framework?* In this section, we empirically search for a better NAT architecture by considering both effectiveness and efficiency.

4.1 Discussion on Architecture Symmetry

Previous studies usually propose asymmetric architectures for AT such as the one with deep encoder and shallow decoder (Kasai et al., 2021). The main reason is that increasing the number of layers, especially in the decoder, deteriorates the latency of translation and memory costs. We verify the architecture symmetry of NAT models by investigating impacts of component-level scaling on translation quality and decoding speed. More specifically, we enlarge the size of layer dimensions in either encoder or decoder, or both components. Table 5 shows results of component-level width-scaling on the WMT20 En-De dataset. Results of component-level depth-scaling are shown in Appendix §A.1.

Translation Performance Clearly the scaling approach improves the translation quality in all cases, although there are still considerable differences among the variants (“Component Scaling” vs. “No Scaling”). Introducing encoder- and decoder-scaling individually improves translation performance over the standard MaskT by +1.8 and +1.2 BLEU points respectively. As seen, scaling encoder and decoder are not equivalent in terms of translation performance. This asymmetric phenomenon is more severe than that in AT models. The possible reason is that NAT model need to

Model	Base	Enc.	Dec.	Both
MaskT	81.3	92.4	85.2	77.8
AT	55.6	93.0	87.3	74.0

Table 6: Performance (accuracy \uparrow) of “WC (word-content)” probing of component-level width-scaled NAT and AT models.

spend a substantial amount of its capacity in disambiguating source and target words under the conditional independence assumption. However scaling both encoder and decoder cannot always achieve better performance compared with individual scaling. This is opposite to AT models, which can further increase by +0.5 BLEU point. *To sum up, 1) scaling NAT is more asymmetric than AT; 2) complementary between encoder and decoder in NAT is weaker than that in AT.*

Decoding Efficiency Compared with Base models, scaling encoder has minimal side-effect on the decoding speed (MaskT: $0.96\times$; GLAT: $0.89\times$). The conclusion still holds on AT models ($0.99\times$). However, scaling decoder has a large impact on decoding speed (MaskT: $1.00\times \rightarrow 0.56\times$; GLAT: $1.00\times \rightarrow 0.85\times$). It is worth noting that iterative NAT is more sensitive to decoder-scaling than fully NAT. The main reason is that iterative mechanism should occupy many times of GPU memory, resulting in smaller mini-batches when calculating Speed_{\max} . Furthermore, there is no further speed decrease when scaling both encoder and decoder components (MaskT: $0.56\times \rightarrow 0.55\times$; GLAT: $0.85\times \rightarrow 0.82\times$). *To sum up, 1) The decoding latency is mainly attributed to scaling decoder; 2) Scaling decoder of iterative NAT comes at the cost of a much larger time cost than fully NAT.*

Linguistic Probing As discussed in Section 3.3, NAT and AT models have different scaling behaviors on learning word-content linguistics. We further investigate the effects at component level in Table 6. *To sum up, asymmetric scaling can enhance the capability of NAT on learning word-content knowledge. The conclusion still holds on AT.*

4.2 Asymmetric Scaling Method

To find a better scaling architecture, we conduct ablation study on a variety of scaled NAT models. Based on the findings, we propose a new NAT architecture to boost translation quality without increasing latency during inference.

#	Encoder		Decoder		Size	Performance	
	#L	Dim.	#L	Dim.		Speed	BLEU
1	6	512	6	512	69M	$1.00\times$	42.0
2	6	1024	6	512	170M	$0.96\times$	42.6
3	12	512	6	512	105M	$0.99\times$	42.4
4	12	1024	6	512	246M	$0.95\times$	43.1
5	12	1024	6	256	217M	$1.32\times$	43.1
6	12	1024	3	512	231M	$1.29\times$	42.7
7	12	1024	3	256	213M	$1.58\times$	43.0

Table 7: Ablation Study of different NAT architectures varying from scaling methods (i.e. number of stacked layer and dimension of feed-forward) and scaling components (i.e. encoder and decoder). “Speed” shows the speedup ratio over NAT-Base, where we measure the decoding speed in terms of S_{\max} . We evaluate MaskT on WMT20 En \rightarrow De validation set.

Ablation Study Seven MaskT models with different architectures are investigated on WMT20 En \rightarrow De dataset. These models are varying from scaling methods (i.e. depth and width) and scaling components (i.e. encoder and decoder). Table 7 shows the variant configurations and the corresponding performances in terms of decoding speedup and translation quality. The #1 is an NAT-Base model, which contains 6 encoder layers and 6 decoder layers with feed-forward dimensions being 512 (i.e. $(6, 6)\times 512$). As shown in #2~4, widening or deepening the encoder component can boost translation quality (BLEU \uparrow) with decreasing the decoding efficiency lightly (Speed \downarrow). Compared with the best encoder-scaling architecture (#4), further widening the decoder counterpart (#5) fails to increase the BLEU scores (43.1 vs. 43.1) but decrease the decoding speed ($0.95\times$ vs. $1.32\times$). To better trade off efficiency and effectiveness, we make the decoder shallower and smaller based on the #4 model. Encouragingly, the #6 and #7 models still achieve comparable translation quality while increasing the speed of decoding to some extent (42.7 vs. 43.0 BLEU and $1.29\times$ vs. $1.58\times$ Speed). This confirms our hypothesis that NAT models need an asymmetric framework when considering both translation quality and decoding speed.

Cone Scaling Motivated by the ablation study, we propose a “Cone” architecture for NAT, of which encoder is deep and big while the decoder is shallow and small (i.e. $(12\times 1024, 3\times 256)$). As shown in Table 8, we adapt the cone-scaling to

Model	Speed	Size	W16 (0.6M)		W14 (4.5M)		W20 (45.1M)	
			En-Ro	Ro-En	En-De	De-En	En-De	De-En
AT Models								
AT-Base	n/a	69M	33.9	34.1	-	-	-	-
AT-Big	n/a	226M	-	-	29.2	32.0	32.4	41.7
NAT Models								
MaskT-Base	1.00×	69M	34.8	33.8	27.5	31.1	31.3	40.6
MaskT-Big	0.55×	225M	34.7	34.0	28.2	31.2	32.9	42.1
MaskT-Cone	1.58×	213M	35.0	34.5	28.4	31.1	33.2	42.1
GLAT-Base	1.00×	70M	32.3	32.6	26.2	30.3	30.6	40.0
GLAT-Big	0.82×	229M	34.5	34.2	27.4	30.9	32.3	41.1
GLAT-Cone	0.90×	215M	34.5	34.2	27.7	31.1	32.5	41.2

Table 8: Translation performance (BLEU \uparrow) of the proposed NAT models on translation tasks with different data sizes. ‘‘Cone’’ denotes scaling NAT architecture to $(12 \times 1024, 3 \times 256)$. ‘‘Speed’’ shows the speedup ratio over NAT-Base, where we measure the decoding speed in terms of S_{\max} .

MaskT and GLAT models, and evaluate them on six benchmarks. In general, our method achieves comparable performance with big models while retaining low latency during inference. As seen, the cone-scaling improve standard MaskT model by +0.9 BLEU averagely) and $1.58 \times$ decoding speedup (over MaskT-Big by +0.2 BLEU and $2.87 \times$ Speed). Besides, the cone-scaling improves standard GLAT model by +1.5 BLEU but decreases decoding speed by $-0.90 \times$ (over GLAT-Big by +0.1 BLEU and $1.10 \times$ Speed). Surprisingly, our method can further benefit the translation quality, leading to much better performance than AT teachers (MaskT: +0.2 BLEU averagely). This emphasizes the need for scaling NAT as a standard procedure. This can be used as a new benchmark over NAT models to convey the extent of the challenges they pose. We also measure translation quality with METEOR (Banerjee and Lavie, 2005), which incorporates semantic information by calculating either exact match, stem match, or synonymy match. As shown in Table 9, the cone scaling consistently achieves the best performance. Results on more datasets are listed in Appendix §A.4.

5 Conclusion and Future Work

In this study we target bridging the gap of model and data scale between NAT and AT models by investigating the scaling behaviors of NAT models. We find that simply scaling NAT models (NAT-Big) can significantly improve translation performance, especially on large-scale training data. To better balance effectiveness and efficiency, we empiri-

Model	MaskT		GLAT	
	En-De	De-En	En-De	De-En
Base	45.1	34.5	44.4	34.0
Big	46.3	34.9	45.9	34.5
Cone	46.7	35.0	46.2	34.5

Table 9: Translation quality of proposed NAT models in terms of METEOR (\uparrow) on WMT20 En \leftrightarrow De tasks.

cally study the contributions of scaling encoder and scaling decoder, and find that scaling NAT is more asymmetric than AT. Based on the observations, we design a new scaling architecture with deeper and wider encoder and shallower and narrower decoder (NAT-Cone), which achieves comparable performance with NAT-Big without scarifying decoding speed. Our study empirically indicates the potential to make NAT a practical translation system as its AT counterpart.

However, the SOTA NAT models (including Scaling NAT) still rely on the distillation by an AT teacher. Future work will investigate better techniques to train scaled NAT models from scratch (i.e. without distillation). We additionally experiment larger NAT models in Appendix §A.2, which can be regarded as a preliminary experiments for this. We will also explore scaling NAG models in other NLP tasks, such as keyphrase generation (Xie et al., 2022) and text-to-table generation (Li et al., 2023). The advent of large language models (LLMs) like GPT-4 has ushered in a new era in MT (Lyu et al., 2023; Jiao et al., 2023a,b; Wang et al., 2023; He

et al., 2023). This innovation is causing us to reconsider conventional paradigms, especially with regards to NAT models.

Limitations

We list the main limitations of this work as follows:

- **Limited NAT Models.** The conclusions in this paper are drawn from two representative NAT models, which may be not necessarily well suited for other NAT models. The main reason is that experiments on six WMT benchmarks have cost a large number of GPU resources. We therefore appeal to future works compare more NAT models using the new benchmarks.
- **Carbon Emissions.** This work totally costed 40,000 GPU hours (around 8,160 kg of CO₂), because 1) large numbers of experiments; and 2) scaled neural networks and training data require more GPU resources. However, we hope our empirical results can help other researchers to reduce the expense of redundant model training.

Ethics Statement

We take ethical considerations very seriously, and strictly adhere to the ACL Ethics Policy. This paper focuses on empirical evaluations on large-scale datasets and scaled NAT models, which can be seen as a reality check. Both the datasets and models used in this paper publicly available and have been widely adopted by studies of machine translation. We ensure that the findings and conclusions of this paper are reported accurately and objectively.

Acknowledgements

The project was supported by National Key Research and Development Program of China(No. 2020AAA0108004), National Natural Science Foundation of China (No. 62276219), and Natural Science Foundation of Fujian Province of China (No. 2020J06001). We also thank the reviewers for their insightful comments.

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. [Massively multilingual neural machine translation in the wild: Findings and challenges](#). arXiv.

Satanjeev Banerjee and Alon Lavie. 2005. [Meteor: An automatic metric for mt evaluation with improved correlation with human judgments](#). In *ACL*.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. [What you can cram into a single vector: Probing sentence embeddings for linguistic properties](#). In *ACL*.

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. 2021a. [Progressive multi-granularity training for non-autoregressive translation](#). In *ACL-IJCNLP Findings*.

Liang Ding, Longyue Wang, Xuebo Liu, Derek F Wong, Dacheng Tao, and Zhaopeng Tu. 2021b. [Rejuvenating low-frequency words: Making the most of parallel data in non-autoregressive translation](#). In *ACL-IJCNLP*.

Liang Ding, Longyue Wang, Xuebo Liu, Derek F. Wong, Dacheng Tao, and Zhaopeng Tu. 2021c. [Understanding and improving lexical choice in non-autoregressive translation](#). In *ICLR*.

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*.

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

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. [Understanding back-translation at scale](#). In *EMNLP*.

Marjan Ghazvininejad, V. 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 Zettloyer. 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, Changan Wang, and Junbo Zhao. 2019. [Levenshtein transformer](#). In *NeurIPS*.

- Yongchang Hao, Shilin He, Wenxiang Jiao, Zhaopeng Tu, Michael Lyu, and Xing Wang. 2021. [Multi-task learning with shared encoder for non-autoregressive machine translation](#). In *NAACL*.
- Zhiwei He, Tian Liang, Wenxiang Jiao, Zhuosheng Zhang, Yujiu Yang, Rui Wang, Zhaopeng Tu, Shuming Shi, and Xing Wang. 2023. [Exploring human-like translation strategy with large language models](#). arXiv.
- Xiao Shi Huang, Felipe Perez, and Maksims Volkovs. 2022. [Improving non-autoregressive translation models without distillation](#). In *ICLR*.
- Wenxiang Jiao, Jen tse Huang, Wenxuan Wang, Xing Wang, Shuming Shi, and Zhaopeng Tu. 2023a. [Parrot: Translating during chat using large language models](#). arXiv.
- Wenxiang Jiao, Wenxuan Wang, Jen tse Huang, Xing Wang, and Zhaopeng Tu. 2023b. [Is chatgpt a good translator? a preliminary study](#). arXiv.
- Jungo Kasai, James Cross, Marjan Ghazvininejad, and Jiatao Gu. 2020. [Non-autoregressive machine translation with disentangled context transformer](#). In *ICML*.
- Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah Smith. 2021. [Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation](#). In *ICLR*.
- Zdeněk Kasner, Jindřich Libovický, and Jindřich Helcl. 2020. [Improving fluency of non-autoregressive machine translation](#). arXiv.
- Yoon Kim and Alexander M Rush. 2016. [Sequence-level knowledge distillation](#). In *EMNLP*.
- Tong Li, Zhihao Wang, Liangying Shao, Xuling Zheng, Xiaoli Wang, and Jinsong Su. 2023. [A sequence-to-sequence&set model for text-to-table generation](#). In *ACL Findings*.
- Chenyang Lyu, Jitao Xu, and Longyue Wang. 2023. [New trends in machine translation using large language models: Case examples with chatgpt](#). arXiv.
- Graham Neubig, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, Xinyi Wang, and John Wieting. 2019. [compare-mt: A tool for holistic comparison of language generation systems](#). arXiv.
- 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*.
- Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. [Scaling neural machine translation](#). In *WMT*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *ACL*.
- Matt Post. 2018. [A call for clarity in reporting bleu scores](#). In *WMT*.
- 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*.
- 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. 2019. [Minimizing the bag-of-ngrams difference for non-autoregressive neural machine translation](#). In *AAAI*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *NeurIPS*.
- Longyue Wang, Chenyang Lyu, Tianbo Ji, Zhirui Zhang, Dian Yu, Shuming Shi, and Zhaopeng Tu. 2023. [Document-level machine translation with large language models](#). arXiv.
- Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. 2019a. [Learning deep transformer models for machine translation](#). In *ACL*.
- Yiren Wang, Fei Tian, D. He, T. Qin, ChengXiang Zhai, and T. Liu. 2019b. [Non-autoregressive machine translation with auxiliary regularization](#). In *AAAI*.
- Binbin Xie, Xiangpeng Wei, Baosong Yang, Huan Lin, Jun Xie, Xiaoli Wang, Min Zhang, and Jinsong Su. 2022. [Wr-one2set: Towards well-calibrated keyphrase generation](#). In *EMNLP*.
- Chunting Zhou, Graham Neubig, and Jiatao Gu. 2020. [Understanding knowledge distillation in non-autoregressive machine translation](#). In *ICLR*.

A Appendix

A.1 Results of Depth Scaling

Main Results We also exploit impacts of the depth scaling on NAT performance. Table 10 shows the results of MaskT and GLAT models on WMT20 En-De. In general, most of conclusions in width scaling still hold for depth scaling. Furthermore, deep and big models achieve comparable performances on KD data (MaskT: 33.1 vs. 32.9 and GLAT: 33.0 vs. 32.3 on En→De; MaskT: 41.9 vs. 42.1 and GLAT: 41.2 vs. 41.1 on De→En) using a comparable number of parameters (MaskT: 201M vs. 225M and GLAT: 203M vs. 229M). On raw data, the performance gap between fully NAT (GLAT) and iterative NAT (MaskT) can be completely overcome by depth scaling (NAT-Base: 24.0 vs. 27.0 and NAT-Deep: 31.3 vs. 30.2 on En→De; NAT-Base: 36.1 vs. 36.6 and NAT-Deep: 39.8 vs. 39.3 on De→En), while width scaling can only bridge the gap between fully NAT and iterative NAT (NAT-Base: 24.0 vs. 27.0 and NAT-Big: 28.8 vs. 30.2 on En→De; NAT-Base: 36.1 vs. 36.6 and NAT-Big: 38.6 vs. 38.7 on De→En). It indicates that depth scaling is a better way to improve the performance of fully NAT than width scaling.

Model	Size	WMT20	
		En-De	De-En
MaskT	69M	27.2	36.6
+ Distillation	69M	31.3	40.6
+ Depth Scaling	202M	30.2	39.3
+ Both	201M	33.1	41.9
GLAT	71M	24.0	36.1
+ Distillation	70M	30.6	40.0
+ Depth Scaling	203M	31.3	39.8
+ Both	203M	33.0	41.2

Table 10: Translation performance (BLEU \uparrow) of NAT-Deep (“Depth Scaling”) models on WMT20 En↔De tasks. The size of NAT-Deep model is (24, 24)×512, which has comparable parameters (202M vs. 226M) with the NAT-Big model in Table 1.

Deeper Scaling To further explore the characteristics of depth scaling for NAT models, we deepen the encoder and decoder to 54-54 layers. Results on WMT20 En→De are shown in Table 11. Compared with the experimental results in Table 10, deeper NAT models (from (24,24) to (54,54)) improve higher performance on raw data than KD

Model	Size	BLEU
MaskT-Deep (54, 54)	422M	31.4
+ Distillation	422M	33.1
GLAT-Deep (54, 54)	424M	32.2
+ Distillation	423M	33.3

Table 11: Translation performance of NAT-Deep (“Depth Scaling”) models on WMT20 En→De task. The size of NAT-Deep model is (54, 54)×512. Relevant experimental results of NAT-Base are shown in Table 10.

Model	Size	BLEU	
		Raw	KD
MaskT	69M	27.2	31.3
+ Deep Enc.	221M	30.8	33.9
+ Deep Dec.	271M	29.2	32.7
+ Deep Both	422M	31.4	33.1
GLAT	71M	24.0	30.6
+ Deep Enc.	222M	29.1	32.8
+ Deep Dec.	273M	27.4	31.2
+ Deep Both	424M	32.2	33.3

Table 12: Translation performance of component-depth-scaled NAT models on WMT20 En→De task. The size of NAT model is (54, 6)×512 or (6, 54)×512.

data (MaskT: +1.2 vs. +0.0; GLAT: +0.9 vs. +0.3). Encouragingly, depth scaling on raw data outperforms standard models trained on distillation data.

Component-Level Deeper Scaling To further verify the symmetry of NAT architecture, we conduct experiments on component-level depth scaling. Experimental results on WMT20 En→De are shown in Table 12. Regarding model type (fully or iterative NAT) and data type (raw or KD), performance gap between depth scaling encoder and depth scaling decoder is significant and stable. It indicates that scaling encoder is more important for NAT model than scaling decoder, which still holds for width scaling in Table 5. Besides, comparing the depth scaling in Table 10 and the deep encoder in Table 12, NAT models with deep encoder show better performance than symmetric deep NAT ones.

A.2 Results of Larger NAT Models

In order to explore the upper-bound of translation performance for NAT, we enlarge models with both depth and width scaling. The model sizes are increased to 831M (MaskT) and 835M (GLAT). Re-

Model	Size	BLEU	
		Raw	KD
MaskT	69M	27.2	31.3
+ Scaling	831M	31.7	34.2
GLAT	71M	24.0	30.6
+ Scaling	835M	31.4	33.4

Table 13: Translation performance of large NAT models on WMT20 En→De task. The size of NAT-Large model is (54, 6)×1024.

Task	MaskT			AT			
	Base	Big	Δ	Base	Big	Δ	
Surface	SeLen	81.4	87.3	+5.9	81.2	87.7	+6.5
	WC	76.6	70.5	-6.1	55.1	70.3	+15.2
Syntactic	TrDep	34.6	35.9	+1.3	35.6	36.5	+0.9
	ToCo	70.7	73.0	+2.3	69.5	73.7	+4.2
	Bshif	49.2	49.6	+0.4	49.5	49.7	+0.2
Semantic	Tense	82.6	84.1	+1.5	83.5	85.0	+1.5
	SubN	79.3	80.6	+1.3	79.3	81.8	+2.5
	ObjN	81.2	81.1	-0.1	80.5	82.3	+1.8
	SoMo	49.8	49.9	+0.1	49.9	49.9	0
	CoIn	53.6	54.1	+0.5	52.8	53.5	+0.7

Table 14: Performance (accuracy ↑) of probing tasks for evaluating linguistic properties embedded in the learned representations of AT and NAT models (with width scaling) on **raw data**.

sults are listed in Table 13. To the best of our knowledge, 34.2 BLEU could be a SOTA performance among existing NAT models. Comparing the performance of base NAT model on KD data and that of large NAT model on raw data, scaling may be an alternative way to replace knowledge distillation (MaskT: 31.7 vs. 31.3 and GLAT: 31.4 vs. 30.6).

A.3 Results of Probing Tasks

To compare the scaling behaviors of AT and NAT models further, more experiments of probing task are conducted. The representations come from the NMT models trained on WMT20 En→De raw data and the experimental results are depicted in Table 14. The differences of scaling behaviors between AT and NAT models on raw data are similar to that on KD data in Table 2.

Model	W16 (0.6M)	W14 (4.5M)	W20 (45.1M)		
	Ro-En	En-De	De-En	En-De	De-En
MaskT					
Base	31.5	40.4	29.7	45.1	34.5
Big	31.5	40.9	29.8	46.3	34.9
Cone	31.7	41.0	29.8	46.7	35.0
GLAT					
Base	31.0	39.3	29.2	44.4	34.0
Big	31.6	40.2	29.6	45.9	34.5
Cone	31.7	40.7	29.7	46.2	34.5

Table 15: Translation performance (METEOR ↑) of the proposed NAT models on translation tasks with different data sizes. “Cone” denotes scaling NAT architecture to (12×1024, 3×256). The results of WMT16 En→Ro are unavailable due to the lack of alignment information on Romanian for METEOR.

Source	Sechs Maschinengewehre des Typs MG3 sind nach wie vor verschwunden.
Refer.	Six MG3 machine guns are still missing.
GLAT-Base	Six MG3- machine machine guns have still disappeared.
GLAT-Big	Six MG3 machine guns have still been missing.

Table 16: Examples about repetition for NAT models. Repetitive tokens are highlighted in **red color**.

A.4 Evaluation with METEOR

To make the results convincing, another evaluation metric, METEOR, is used to measure the scaling behavior of NAT models. Different from BLEU, METEOR incorporates semantic information by calculating either exact match, stem match, or synonymy match. The results of METEOR are calculated with Multeval.² More experimental results of METEOR are provided in Table 15, which are similar to the results of BLEU in Table 8.

A.5 Commonly-Cited Weaknesses of NAT

In this paper, we study the commonly-cited weaknesses of NAT from the following three perspectives: 1) *multimodality* indicated by token repetition ratio; 2) *generation fluency* calculated by language model perplexity; 3) *translation adequacy* measured by word translation accuracy. To illus-

²<https://github.com/jhclark/multeval>.

Source	国庆长假临近，人们的假期计划也逐渐敲定。
Refer.	As the National Day holiday approaches, people’s holiday plans are gradually being finalized.
GLAT-Base	The National Day long holiday near, people people’s plans plans gradually gradually gradually.
GLAT-Big	The National Day holiday is approaching, people’s holiday plans are gradually worked out.

Table 17: Examples about fluency for NAT models. The key spans are highlighted in red color.

Source	曼西内利 当时 虽然有 上小学，但 后来 没有 毕业。
Refer.	Although Mancinelli entered elementary school, he did not graduate.
GLAT-Base	Manthinelli attended primary school at the time but but did not graduate.
GLAT-Big	Mancinelli went attended primary school at the time but did not not graduate.

Table 18: Examples about word accuracy for NAT models. The key tokens are highlighted in red color.

trate the effect of scaling on commonly-cited weaknesses of NAT, examples are listed in Table 16, Table 17 and Table 18 respectively.

A.6 Training of NAT models

We adopt Transformer-Base/Big configurations for all NAT models: both encoder and decoder contain 6 layers with 8/16 attention heads, the hidden dimension is 512/1024, and the feedforward layer dimension is 2048/4096. We train all NAT models with a big batch size of 480K. We train MaskT, GLAT models for 300K steps.

We list the training budget in Table 19. More details about training hyper-parameters can be found in the training scripts of different NAT models.

Model	Size	GPU Hours
AT-Base	69M	352h
AT-Big	226M	616h
MaskT-Base	69M	320h
MaskT-Big	226M	584h
GLAT-Base	71M	816h
GLAT-Big	230M	1120h

Table 19: The number of parameters and training budget (in GPU hours with 8 A100 for 300K steps). More detailed training hyper-parameters can be found in the training scripts of the different NAT models.

ACL 2023 Responsible NLP Checklist

A For every submission:

- A1. Did you describe the limitations of your work?
Section of Limitations
- A2. Did you discuss any potential risks of your work?
Section of Ethics Statement
- A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1.
- A4. Have you used AI writing assistants when working on this paper?
Left blank.

B Did you use or create scientific artifacts?

Left blank.

- B1. Did you cite the creators of artifacts you used?
Not applicable. Left blank.
- B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.
- B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.
- B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.
- B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Not applicable. Left blank.

C Did you run computational experiments?

Section 3 and 4 and Appendix

- C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 2.3, Section of Limitations and Appendix A.6

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

- C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?

Section 2.3, and Appendix A.6

- C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?

Section 4

- C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?

Section 3 and 4.

D Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

- D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?

Not applicable. Left blank.

- D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?

Not applicable. Left blank.

- D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?

Not applicable. Left blank.

- D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?

Not applicable. Left blank.

- D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?

Not applicable. Left blank.