A AT TRANSFORMER IN SYNTHESIS EXPERIMENTS

Table 1: Performance of autoregressive models.

AT Transformer	En-Ro	En-De
Vaswani et al. (2017)	-	27.3
Ghazvininejad et al. (2019)	34.28	27.74
Our implementation	34.25	27.45

In synthesis experiments, we trained all AT models with the standard Transformer-Base configuration: layer=6, dim=512, ffn=2048, head=8. The difference from Ghazvininejad et al. (2019) is that they trained the AT models for 300k steps, but we updated 50k/100k steps on En→Ro and En→De, respectively. Although fewer updates, as shown in Table 1, our AT models have comparable performance with theirs.

B TRAINING ALGORITHM

Algorithm 1 Training Algorithm for Hybrid-Regressive Translation

Input: Training data D including distillation targets, pretrained AT model M_{at}, chunk size k, mixed distillation rate p_{raw}
Output: Hybrid-Regressive Translation model M_{hrt}

1: $\mathrm{M}_{hrt} \leftarrow \mathrm{M}_{at}$ ▷ finetune on pre-trained AT
2: **for** t in $1, 2, \ldots, T$ **do**
3: $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}, \boldsymbol{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}, \boldsymbol{Y}' = \{\boldsymbol{y}'_1, \ldots, \boldsymbol{y}'_n\} \leftarrow$ fetch a batch from D
4: **for** i in $1, 2, \ldots, n$ **do**
5: $\boldsymbol{B}_i = (\boldsymbol{X}_i, \boldsymbol{Y}_i^*) \leftarrow$ sampling $\boldsymbol{Y}_i^* \sim \{\boldsymbol{Y}_i, \boldsymbol{Y}'_i\}$ with $P(\boldsymbol{Y}_i) = p_{raw}$ ▷ mixed distillation
6: **end for**
7: $p_k \leftarrow$ get the chunk-aware proportion by Eq. 1 ▷ curriculum learning
8: $\boldsymbol{B}_{c=k}, \boldsymbol{B}_{c=1} \leftarrow \boldsymbol{B}_{:\lfloor n \times p_k \rfloor}, \boldsymbol{B}_{\lfloor n \times p_k \rfloor:}$ ▷ split batch
9: $\boldsymbol{B}^{at}_{c=k}, \boldsymbol{B}^{mp}_{c=k} \leftarrow$ construct {Skip-AT, Skip-MP} training samples based on $\boldsymbol{B}_{c=k}$
10: $\boldsymbol{B}^{at}_{c=1}, \boldsymbol{B}^{mp}_{c=1} \leftarrow$ construct {AT, MP} training samples based on $\boldsymbol{B}_{c=1}$
11: Optimize M_{hrt} using $\boldsymbol{B}^{at}_{c=k} \cup \boldsymbol{B}^{at}_{c=1} \cup \boldsymbol{B}^{mp}_{c=k} \cup \boldsymbol{B}^{mp}_{c=1}$ ▷ joint training
12: **end for**

Algorithm 1 describes the process of training the HRT model. The HRT model is pre-initialized by a pre-trained AT model (Line 1). During training, the training batch $\boldsymbol{B}_i$ randomly select a raw target sentence $\boldsymbol{Y}_i$ or its distilled version $\boldsymbol{Y}'$ (Line 4-6). Then according to the linear schedule of p_k:

$$p_k = \left(\frac{t}{T}\right)^{\lambda} \tag{1}$$

where λ=1, we can divide $\boldsymbol{B}$ into two parts: $\boldsymbol{B}_{c=1}$ and $\boldsymbol{B}_{c=k}$, where $|\boldsymbol{B}_{c=k}|/|\boldsymbol{B}| = p_k$ (Line 7-8). Next, we construct four kinds of training samples based on corresponding batches: $\boldsymbol{B}^{at}_{c=k}$, $\boldsymbol{B}^{at}_{c=1}$, $\boldsymbol{B}^{mp}_{c=k}$ and $\boldsymbol{B}^{mp}_{c=1}$. Finally, we collect all training samples together and accumulate their gradients to update the model parameters, which results in the batch size being twice that of standard training.

C COMPUTATION COMPLEXITY

In Table 2, we summarized the comparison with Autoregressive Translation (AT), Iterative Refinement based Non-autoregressive Translation (IR-NAT) and Semi-Autoregressive Translation (SAT) Wang et al. (2018).

AT. Although both HRT and AT contain a slow autoregressive generation process, HRT's length is k times shorter than AT. Considering that the computational complexity of self-attention is quadratic with its length, HRT can save more time in autoregressive mode.

Table 2: Compare hybrid-regressive translation (HRT) to autoregressive translation (AT), mask-predicted based non-autoregressive translation (MP), and semi-autoregressive translation (SAT). $Q(i)$ denotes the computation cost in autoregressive mode when producing the i-th token (e.g., the prefix length is $i-1$). $\hat{Q}_b(i)$ denotes the computation cost in non-autoregressive mode when producing i tokens by one shot with a beam size of b. I=4 $\sim$ 10, k is generally 2.

Method	**Steps**	**Computing Cost**
AT	L	$\sum_{i=1}^{L} Q(i)$
IR-NAT	I	I $\times \hat{Q}_{b=5}(L)$
SAT	L/k	L/k $\times (\hat{Q}_{b=5}(k) + \epsilon)$
HRT	L/k + 1	$\sum_{i=1}^{L/k} Q(i \times k)$ + $\hat{Q}_{b=1}(L)$

IR-NAT. Since Skip-AT provides a high-quality target context, HRT does not need to use large beam size and multiple iterations like IR-NAT. The experimental results also show that our light NAT can make up for the increased cost in Skip-AT, and can achieve stable acceleration regardless of the decoding batch size and running device.

SAT. SAT generates segments locally by non-autoregression, but it is still autoregressive between segments. We claim that SAT reduces the decoding steps by k, but each token's calculation remains unchanged. In other words, in the time step i, there are $i-1$ tokens used for self-attention. By contrast, only i/k tokens are involved in our Skip-AT.

REFERENCES

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6111–6120, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1633. URL https://www.aclweb.org/anthology/D19-1633.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 6000–6010, 2017.

Chunqi Wang, Ji Zhang, and Haiqing Chen. Semi-autoregressive neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 479–488, 2018.