Language Model Based Target Token Importance Rescaling for Simultaneous Neural Machine Translation

Aditi Jain*
IIT Delhi
mt6190739@iitd.ac.in

Nishant Kambhatla*
School of Computing Science
Simon Fraser University
nkambhat@sfu.ca

Anoop Sarkar
School of Computing Science
Simon Fraser University
anoop@sfu.ca

Abstract

The decoder in simultaneous neural machine translation receives limited information from the source while having to balance the opposing requirements of latency versus translation quality. In this paper, we use an auxiliary target-side language model to augment the training of the decoder model. Under this notion of target adaptive training, generating rare or difficult tokens is rewarded which improves the translation quality while reducing latency. The predictions made by a language model in the decoder are combined with the traditional cross entropy loss which frees up the focus on the source side context. Our experimental results over multiple language pairs show that compared to previous state of the art methods in simultaneous translation, we can use an augmented target side context to improve BLEU scores significantly. We show improvements over the state of the art in the low latency range with lower average lagging values (faster output). [1]

1 Introduction

Simultaneous Machine Translation (SiMT; Grissom II et al. (2014); Cho and Esipova (2016)) is a special case of neural machine translation (NMT; Vaswani et al. (2017)) that aims to produce real time-translations in the target language from a streaming input in the source language. The cornerstone of this task, as well as a key challenge, is the trade-off between the translation quality and the latency in producing the translations. This balance is ensured by a fixed (Ma et al., 2019; Elbayad et al., 2020) or adaptive (Arivazhagan et al., 2019; Ma et al., 2020; Zhang and Feng, 2022b) read/write policy that determines whether to wait for the next source token (a READ action) or to generate a translation (a WRITE action). Adaptive policies dynamically predict the action based on the current source and target contexts (Zheng et al., 2020). Although adaptive policies achieve a better latency/BLEU trade-off, they often fail to account for the varying importance of different tokens when deciding a READ/WRITE action.

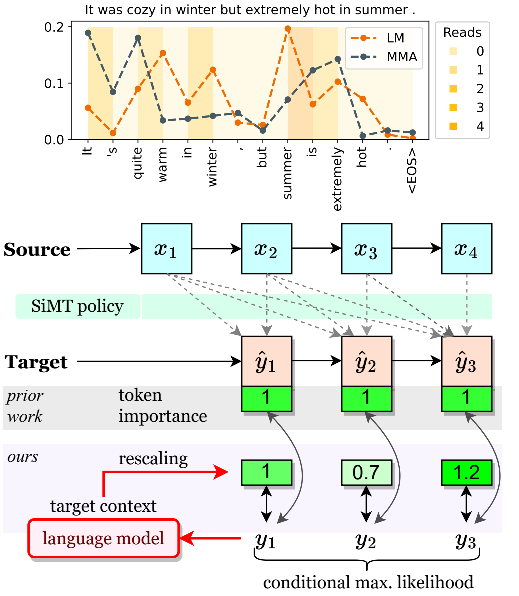


Figure 1: Prior work on simultaneous MT weighs every target token equally. **Top**: Normalized negative log-likelihood (nll) scores of each generated in-context target token as scored by the baseline SiMT along with the number of reads preceding a target token, and a target language model (LM). As the translations are imperfect, the LM shows disagreement by following an opposite nll trend compared to the translation model. **Bottom**: Our method rescales the importance of each target token using the target context during training.

In Figure 1 (top), there is a negative correlation between the normalized negative log likelihoods of output tokens as measured by MMA (a SiMT model with an adaptive policy; Ma et al. 2020) versus a left-to-right language model (LM). This reflects a translation which the SiMT model is confident about, but which the LM regards as poor English (possibly due to the semantic mismatch

[1] github.com/sfu-natlang/target_rescale_siMT

*Equal contribution. Listing order is random. Work performed while AJ was visiting SFU Natlang Lab.

Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023), pages 341–356
July 13-14, 2023 ©2023 Association for Computational Linguistics

evident in "*warm in winter*"). Since a simultaneous policy can only access partial source context, its outputs are likely to reflect imperfect guesses such as these, particularly when translating in real-time between language pairs with different word orderings (Subject-Object-Verb) and very long compounds. As a result, training objectives which treat all translated tokens with equal importance are sub-optimal.

In the context of translation, content words are generally considered more informative than function words (Chen et al., 2020). This is because content words carry the main semantic and lexical meaning of a sentence, while function words provide grammatical context and help to convey the syntactic structure of a sentence. Similarly, high-frequency words that are easier for the translation model to generate may sometimes carry less information than the desirable low-frequency (rare) words that the model struggles to generate (Chen et al., 2017). To this end, Zhang et al. (2022b) proposed to leverage conditional mutual information (MI) to estimate the weight-coefficients between the source and target to reweigh the importance of each target token. However, such an approach hasn't been explored to address simultaneous or streaming MT to the best of our knowledge, and the lack of a complete source context makes the adaptation of this method to SiMT non-trivial. To improve simultaneous MT, Alinejad et al. (2018) proposed a prediction mechanism on the source side to get future information and aid the lack of information on target-side for translation. Instead of directly predicting a source token, Zhang and Feng (2022a) predict its aligned future-position for a given target token to guide its policy. On the other hand, Zhang and Feng (2022b) and Zhang et al. (2022a) explored policies that assign varying importance to source/target tokens based on their level of information, with more informative tokens having a greater influence on the model.

In this paper, we propose a technique to alleviate this problem in SiMT using an information theoretic approach and an adaptive training paradigm. Inspired by the recent work in using pointwise mutual information for guiding the decoder in full-sentence (non-simultaneous) translation (Lee et al., 2022), we differentiate the importance of various target tokens by their dependence on the source sentence. As shown in Figure 1 (bottom), to guide our simultaneous translation model, we incorporate a language model that provides an additional signal indicating the importance of each target token or sentence. This *target-context aware estimation* leverages the relative probabilities of the translation model and language model to guide the generation process by explicitly re-weighting the training loss of each target token in the translation. Experiments show the strength of our simple method, outperforming several strong baselines in terms of both latency and BLEU scores. We perform exhaustive analysis to show that our model performs particularly well on translating low frequency words and longer sentences.

2 Background

Target adaptive training (Lin et al., 2017) in NMT addresses the token imbalance problem (Gu et al., 2020). While a translation model is conventionally trained with conditional maximum likelihood estimation or cross-entropy:

$$\mathcal{L}_{\text{CE}}(\mathbf{x}, \mathbf{y}) = -\sum_{j=1}^{N} \log p\left(y_j \mid \mathbf{y}_{<j}, \mathbf{x}\right) \quad (1)$$

adaptive training rescales this objective by assigning static or dynamic weights to further guide the translation model:

$$\mathcal{L}_{\text{adapt}}(\mathbf{x}, \mathbf{y}) = -\sum_{j=1}^{N} w_j \log p\left(y_j \mid \mathbf{y}_{<j}, \mathbf{x}\right) \quad (2)$$

Frequency based approaches (Gu et al., 2020; Xu et al., 2021) to assign these weights are promising but maintaining a frequency count can be an expensive overhead and would not be directly transferable to a simultaneous setting. More recently, Zhang et al. (2022b) proposed to leverage pointwise mutual information (MI) to estimate the weight-coefficients between the source $\mathbf{x}$ and target $\mathbf{y}$ as :

$$\text{MI}(\mathbf{x}, \mathbf{y}) = \log\left(\frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x}).p(\mathbf{y})}\right) \quad (3)$$

which can reflect the importance of target tokens for translation models.

Monotonic Infinite-Lookback Attention Arivazhagan et al. (2019) models a Bernoulli variable to make the READ or WRITE decision at every time step, while processing the input sequence incrementally. Ma et al. (2020) present monotonic multihead attention to extend this policy to the multihead attention of transformers. For each encoder state in MMA, every head in the cross-attention of each decoder layer produces a probability $p_{i,j}$ that dictates if it should write target token y_j while having read till the source token x_i, or wait for more inputs. This is computed using the softmax energy:

$$\begin{aligned} \text{energy}_{i,j} &= \left(\frac{m_j W^K \left(s_{i-1} W^Q \right)^T}{\sqrt{d_k}} \right)_{i,j} \\ p_{i,j} &= \text{Sigmoid}\left(\text{energy}_{i,j}\right) \end{aligned} \tag{4}$$

where m signifies the encoder states, W the input projection matrix for query Q and key K, and d_k is the dimension of the attention head. The probability $p_{i,j}$ is then used to parameterize the Bernoulli random variable:

$$b_{i,j} \sim \text{Bernoulli}\left(p_{i,j}\right) \tag{5}$$

If $b_{i,j} = 1$ then the model performs a WRITE action on y_j based on previous source tokens, otherwise it performs a READ.

Our method is based on MMA and we use it as our main simultaneous policy. To mitigate the negative impact of outlier heads[2] on the read/write path, we have made slight modifications to MMA to ensure more stable performance. Instead of allowing the heads in each decoder layer to independently determine the READ/WRITE action, we now share the READ/WRITE action between the decoder layers. This adjustment helps to avoid outlier heads that could potentially disrupt the system performance and stability (Indurthi et al., 2022).

3 Approach

3.1 Target-context Aware Information Quotient

Inspired by Lee et al. (2022), we leverage the pointwise mutual information (MI) between each target token and its source context under the condition of previous target context. For a target token y_j and the streaming source context $\mathbf{x} \leq i$, factoring in the partially constructed target prefix $\mathbf{y} < j$ gives the *target information quotient* (TIQ) is calculated as:

$$\begin{aligned} \text{TIQ}\left(y_j\right) &= \log\left(\frac{p\left(y_j, \mathbf{x}_{\leq i} \mid \mathbf{y}_{<j}\right)}{p\left(y_j \mid \mathbf{y}_{<j}\right) \cdot p\left(\mathbf{x}_{\leq i} \mid \mathbf{y}_{<j}\right)}\right) \\ &= \log\left(\frac{p\left(y_j \mid \mathbf{x}_{\leq i}, \mathbf{y}_{<j}\right) \cdot p\left(\mathbf{x}_{\leq i} \mid \mathbf{y}_{<j}\right)}{p\left(y_j \mid \mathbf{y}_{<j}\right) \cdot p\left(\mathbf{x}_{\leq i} \mid \mathbf{y}_{<j}\right)}\right) \\ &= \log\left(\frac{p\left(y_j \mid \mathbf{x}_{\leq i}, \mathbf{y}_{<j}\right)}{p\left(y_j \mid \mathbf{y}_{<j}\right)}\right) \\ &= \log\left(\frac{p_{\text{SiMT}}\left(y_j\right)}{p_{\text{LM}}\left(y_j\right)}\right) \end{aligned} \tag{6}$$

where $p_{\text{SiMT}}(.)$ is the simultaneous translation model probability and $p_{\text{LM}}(.)$ is the auxiliary target-side language model of the same size as the translation decoder. By decomposing the conditional joint distribution, this can be formalized as the log quotient of the streaming translation model probability and target language model probability. This captures the information of a target token conditioned on the target context and uses it to rescale the loss, thereby making the model pay more attention to more "informative" words.

To incorporate weights into the adaptive training objective (equation 2), two separate weights are used:

Token-level weight is used to determine weights of loss from each target token y_j and streaming source context, conditioned on the obtained partial translation at the current timestep. We use a token TIQ measure and normalise it to reduce variance:

$$\text{TIQ}_{\text{tok}} = \left(\text{TIQ}(y_j) - \mu^{tok}\right) / \sigma^{tok} \tag{7}$$

where μ^{tok}, and σ^{tok} are the mean and standard deviation of $\text{TIQ}(y_j)$ respectively, for every sentence.

Sentence-level weight on the other hand, is token-level TIQ is aggregated and averaged over the target sentence length $|\mathbf{y}|$:

$$\text{TIQ}_{\text{sen}} = \left(\frac{1}{|y|} \sum_{j=1}^{|y|} \text{TIQ}(y_j) - \mu^{sen} \right) / \sigma^{sen} \tag{8}$$

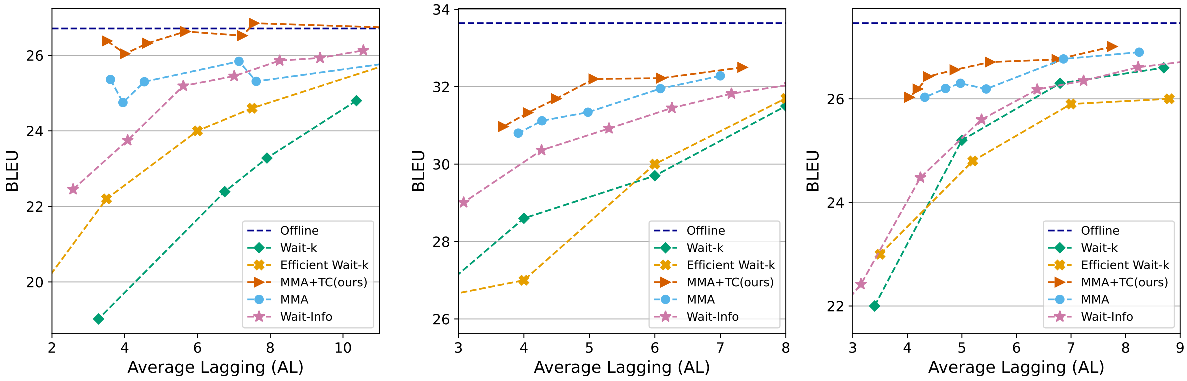
where μ^{sen}, and σ^{sen} are the mean and standard deviation of $\text{TIQ}(y_j)$ respectively, over a batch.

The final rescaling factor to assign weights in equation 2 is calculated as:

$$w_j = \left(\lambda_{\text{tok}} \text{TIQ}_{\text{tok}} + 1\right) \cdot \left(\lambda_{\text{sen}} \text{TIQ}_{\text{sen}} + 1\right) \tag{9}$$

The rescaling allows the model to learn the source side information for a particular target token y_j, while also factoring in the target context so far.

[2] In MMA, every head in the transformer multihead attention independently decides its read/write action and has access to all previous encoder states. The write action only takes place when the slowest head has arrived to a write decision.



(a) Vi→En

(b) De→En

(c) En→De

Figure 2: Results on IWSLT15 Vi → En (a), and IWSLT14 En⇔De (b,c)

Given that information (from source) is constrained in the nature of this task, this additional signal of the target context acts as reinforcement for translation. The likelihood score from the LM should serve to strengthen the predictive capability of the decoder. Frequent words would have a higher LM score and therefore a smaller weight w_j. On the other hand, rare words would be scored lower by the LM, and thus have a higher rescaling weight w_j, allowing the model to focus on them more.

3.2 Final Training Objective with Adaptive Weights and Latency Constraints

In MMA models, following Ma et al. (2020), we use the weighted average of differentiable lagging metric $\mathcal{C}$ (Arivazhagan et al., 2019) over all the attentions heads as the weighted average latency L_{avg} constraint[3].

The MMA model uses both these loss terms in its final loss, with the hyperparameters λ_{avg} and λ_{var} respectively. Combining the latency average loss and the target-context aware information quotient, the final training objective for our model is:

$$\mathcal{L}_{\text{MMA+TC}} = L_{\text{adapt}}(\text{TIQ}) + \lambda_{\text{avg}} L_{\text{avg}} \quad (10)$$

where L_{adapt} is adaptive cross-entropy loss from equation 2 with TIQ (equation 9) as its rescaling-weight, and λ_{avg} is a hyperparameter to control the latency constraint.

4 Experiments

4.1 Data

IWSLT'15 English ↔ Vietnamese (133K pairs) with TED tst2012 (1553 pairs) as validation set and TED tst2013 (1268 pairs) as test set. The vocabulary sizes of English and Vietnamese are 17K and 7.7K respectively.

IWSLT'14 English ↔ German (160K pairs) with validation set and test set of 7283 and 6750 pairs respectively. The vocabulary size of German is 13.5K and 9.98K for English.

4.2 Baselines and Model Settings

The following are the **main baselines** we compare our method against:

Offline Transformer (Vaswani et al., 2017) model for full-sentence translation.

Wait-k policy (Ma et al., 2019) which is a fixed-policy that reads k source tokens initially, and then alternates between reading and writing.

Efficient Wait-k (Elbayad et al., 2020) uses multiple k's to train a Wait-k model and relieves the constraint of test k being equal to train k.

Monotonic Multihead Attention (MMA; Ma et al. (2020)) extends infinite lookback attention (Arivazhagan et al., 2019) to all the Transformer heads.

Wait-Info (Zhang et al., 2022a) quantifies source and target token info to decide R/W action.

We also juxtapose our method against several **other baselines** on the En → Vi direction:

[3] Early experiments with other policies such as GMA and Wait-info showed the approach to be ineffective. The explicit latency loss in MMA is crucial for the working of target adaptive training for simultaneous MT.

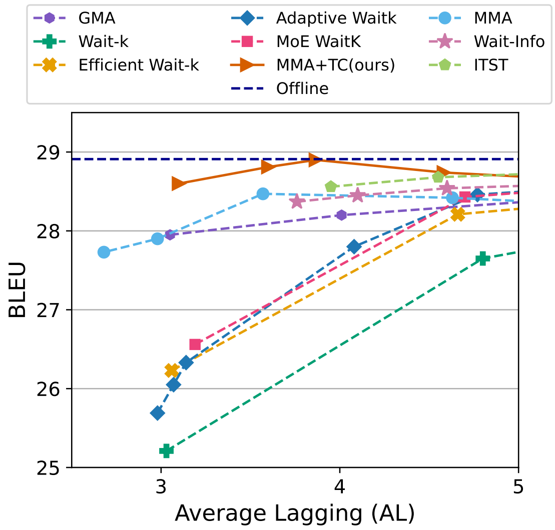


Figure 3: Performance of several methods on the En→Vi dataset in the *low latency* (AL<5) window.

Gaussian Multihead Attention (GMA; Zhang and Feng (2022a)) that predicts the aligned source position for a target token and rescales attention with a gaussian distribution centred at this position.

ITST (Zhang and Feng, 2022b) finds the optimal information transport between source and target.

Adaptive Wait-k(Zheng et al., 2020) dynamically chooses an optimal k in the wait-k policy at every step.

MoE Wait-k (Zhang and Feng, 2021b) uses attention heads as experts trained with different k with the wait-k policy.

MMA+TC (ours) is the proposed MMA model with target context aware adaptive training objective. We use an auxiliary target-side LM decoder of the same configuration as the MT decoder. Note that the LM is only used during training and discarded at test time. We do not use extra data.

The implementation of our method is based on fairseq (Ott et al., 2019). Following MMA, we use transformer (Vaswani et al., 2017) with 6 encoder and decoder layers and 4 monotonic attention heads for the IWSLT datasets En↔Vi, De↔En. All baselines are trained with same configurations and are trained with 16k tokens. Our auxiliary language model follows the decoder settings in the model.

4.3 Evaluation

We evaluate using BLEU (Papineni et al., 2002) for translation quality and Average Lagging (AL) (Ma et al., 2019) for latency. AL denotes the lagging behind the ideal policy (Wait-0). Other metrics used are Average Proportion (AP) (Cho and Esipova, 2016) and Differentiable Average Lagging (DAL) (Arivazhagan et al., 2019). Given a read/write policy g_i, AL is :

$$\mathrm{AL} = \frac{1}{\tau}\sum_{i=1}^{\tau} \mathrm{g}_i - \frac{i-1}{|y|/|x|} \tag{11}$$

where $\tau = \operatorname{argmax}_i (g_i = |x|)$, $|x|$ and $|y|$ are source sentence and target sentence lengths respectively.

5 Results

Figure 2 shows the comparison of BLEU vs. Latency (in terms of Average Lagging) of our method against previous methods on the IWSLT'15 Vi → En and IWSLT'14 En ↔ De directions. For Vi → En, we observe a significant improvement in the BLEU scores at the same latencies, compared to the baselines. We also reach the offline translation quality in low AL on this dataset. In the En → De, De → En directions too, there is a boost in the translation quality, more noticeably for lower latencies. The plots show that our method boosts translation quality in the earlier latencies and the effect of reweighing is more pronounced in these regions, where the source context is more limited. In higher latency regions, when the source information window increases, the other baselines start to reach our BLEU score in the English-German directions.

In Figure 3, we compare against several state-of-the-art methods on the En → Vi. Our method gets better translation quality compared all others, in the *low-latency* zone, matching the offline score at 3.86 AL. We show the BLEU vs. AL plot in a low latency range to compare performance in the more challenging area of this task, the low latency points.

6 Analysis

6.1 Token-level vs. Sentence-Level Weight

Ablation Study The two hyperparameters in our method are Sentence-Level Weight and Token-Level Weight, which determine the sentence and token-level effect of rescaling with LM. In Fig. 5

Token Order (Descending)	Avg. Freq.	Ref (%)	MMA (%)	MMA+ TC (%)
[0, 10%)	1385	85.56	87.63	87.21
[10, 30%)	56	6.89	6.48	6.34
[30, 50%)	20	2.19	1.75	**1.95**
[50, 70%)	11	1.30	0.70	**0.86**
[70, 100%]	6	0.95	0.26	**0.31**

Table 1: Avg. frequency on the training set and the proportion of tokens of different frequencies in the test set and the translations generated by the baseline and our model.

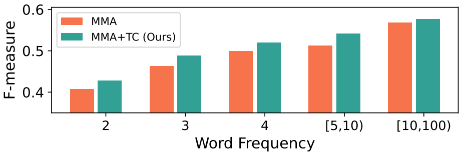


Figure 4: F-measure between model outputs and reference tokens for the low-frequency words, bucketed by frequency of the reference token.

we report the BLEU scores with different hyperparameter settings on Vi-En. (AL across the table are similar as experiments are done with the same λ). We set the values of these hyperparameters to 0.2 in all our experiments.

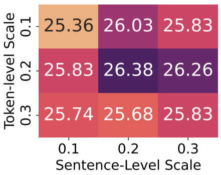


Figure 5: MMA+TC with different combinations for tok-level scale (λ_{tok}) and sent-level scale (λ_{sen}) values.

6.2 Effect on Low-frequency Words

With reweighing loss using the Language Model likelihood, we aim to reduce the effect of frequency imbalance in the corpus on training. We compare our translations against MMA on rare and frequent words. In addition to an overall BLEU improvement, we also see an improvement in the F-measure of rare words. As shown in Figure 4, our method does better on extremely rare words (freq $\leq$ 10). Table 1 shows that while the baseline overfits to the most frequent words, our method captures rare words, from the bottom two frequency bins (50-70% and 70-100%), better. The results show that our method makes the model train better on rare words and remedy the effect of token imbalance.

POS	**Ref**	**MMA (%)**	**+TC (%)**	**MSE (↓)**
ADJ	1497	82.1	**83.5**	0.18 | **0.16**
ADV	1323	83.5	**87.6**	0.20 | **0.12**
INTJ	74	**98.6**	94.6	**0.01** | 0.04
NOUN	4187	90.5	**93.4**	0.09 | **0.06**
PROPN	1315	99.4	99.4	- | -
VERB	3226	94.0	**95.7**	0.06 | **0.04**

Table 2: Our method generates more content words than the baseline MMA. Columns 2 and 3 show the percentage of the reference content words recovered in MMA and MMA+TC (in blue) respectively. The last column shows normalized mean squared error (MSE) of the recovered content words wrt reference. Lower MSE values are better.

Content word occurrences. Zhang et al. (2022a) show that focusing on the right content words in the target is crucial to getting the necessary target information in a subcutaneous translation setting. Following Moradi et al. (2019) we inspect the content words generated by our model using spacy to get POS tags over the translations. As evident from Table 2, our model recovers more content words in the translations wrt the reference.

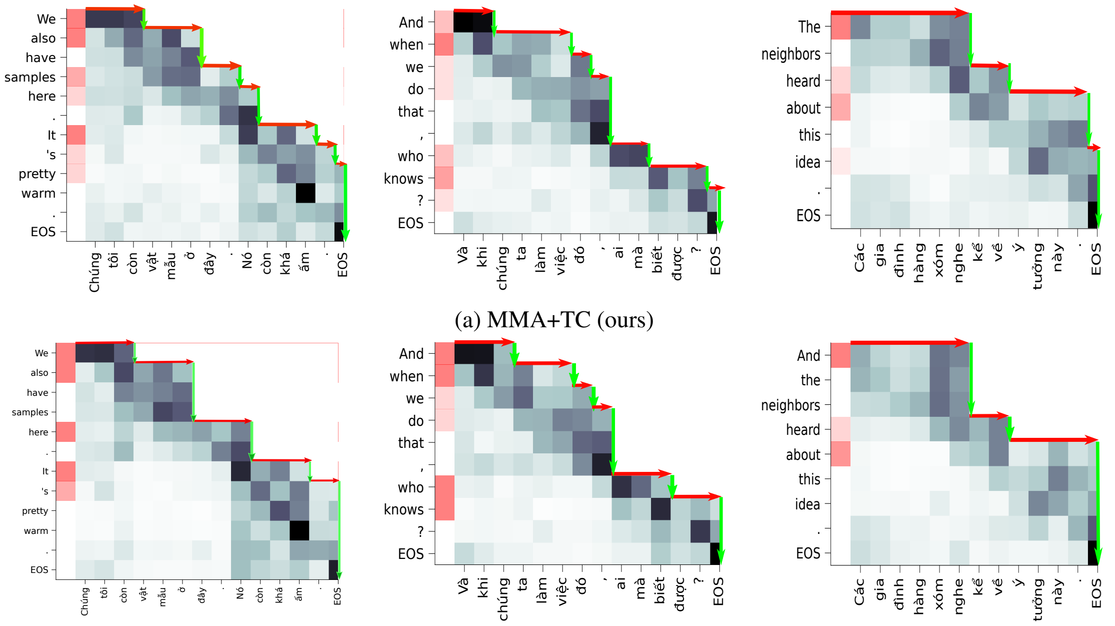
6.3 Effect on Translation Length

Following the rationale of Lakew et al. (2019) in NMT and Zhang and Feng (2022d) in simultaneous translation, we inspect the translation quality of our model on varying target sentence lengths in Figure 8 and observe that our method shows a big improvement in BLEU on the longer sentences. Our method prevents the model from over-producing words (as seen in the top figure in Figure 8). We hypothesize that this is because the model does not generate as many words (and overuse them) from the most frequent word bin (see Table 1, top 10% bin) as MMA. Our target sentence lengths are consistently less than MMA's and are closer to the ground truth sentence lengths (as shown in the bin 0, Fig. 8 (top)).

6.4 Effect on Translation Paths

Attention Heatmaps and READ/WRITE Sequences. Figure 6 compares attention heatmaps from MMA and MMA+TC (our method) on the Vi → En direction. As evident, our method performs READ actions in smaller intervals between predicting consecutive WRITE actions.

Consider the READ/WRITE actions generated by MMA and MMA+TC for the given source sen-



(a) MMA+TC (ours)

(b) MMA (baseline)

Figure 6: Attention heatmap comparison on the Vi → En direction. The Read-Write policy is drawn with red and green arrows respectively. The pink column at the start denotes the source tokens read to produce the target token on the left (darker implies more source words read, and white denotes 0 reads between consecutive target tokens)

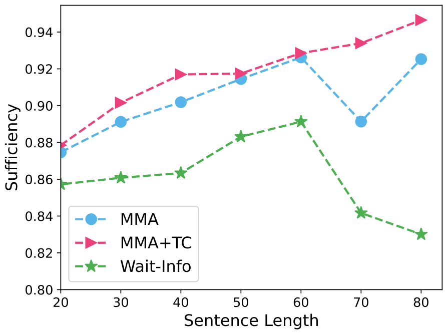


Figure 7: Sufficiency as a function of target length. All models produce translation with an AL of 4.

tence are:

Src: Chúng tôi còn vt mu đây . Nó còn khá m .

MMA: RRR W RRR WWW RRR WW RRR W RR WWWWW

Ours: RRR W RRR WW RR W R WW RRR W R W R WWWW

In this example, MMA reads more than required for a write in certain places. It shows that at a similar lag, our model gets a higher probability of a WRITE action, compared to MMA, after having read the same number of source words.

***Sufficiency* of the READ actions.** Zhang and Feng (2022c) introduce a metric of sufficiency A^{Suf} in Read/Write paths with the notion that too many but not all necessary READs would result in high latency while few but not sufficient READ actions would exclude needed information and could cause poor translation quality. When the ground truth aligned source position of the j^{th} target word is denoted by a_j, and the number of source words read when writing target j^{th} word is denoted by r_j:

$$A^{Suf} = \frac{1}{|y|}\sum_{j=1}^{|y|} \mathbb{1}_{a_j \le r_j} \qquad (12)$$

We compare our method against MMA and Wait-Info on AL=4 with the sufficiency metric. Using equation(12) across sentences of varying lengths, we evaluate the read-write paths of each model, against reference alignments from Eflomal (Östling and Tiedemann, 2016)[4]. In Figure 7, we can see a clearly increasing and higher score on sufficiency as compared to the baselines - Wait-Info and MMA. This signifies that our target-context augmented training helps the model read sufficient source tokens required for producing a translation, while maintaining the same latency as others, showing that the model learns and correctly gauges the information it requires to translate a target token, and

[4] We use the Eflomal library to get alignment priors from IWSLT'15 Vi-En train set, and use them to generate alignments for the test set. https://github.com/robertostling/eflomal

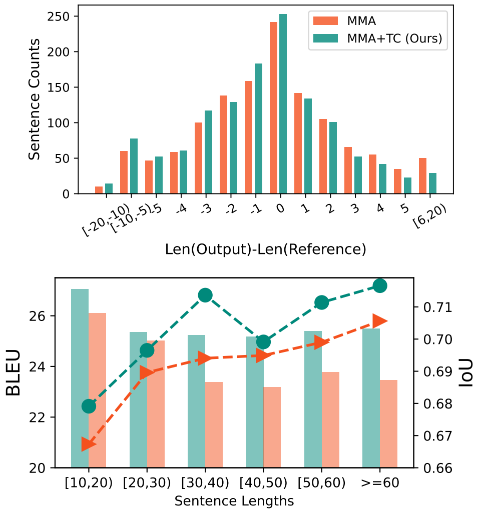


Figure 8: **Top**: Length difference compared to ref. **Bottom**: Sentence BLEU bucketed by target length (shown in bars), and the ratio of aligned READ actions for each bucket (IoU scores, Eqn. 13) shown with lines.

makes READ actions accordingly.

Ratio of Aligned READ actions. We compare MMA and our Read-Write policy against the reference source-target alignments by computing the overlap between the hard alignments and the translation path for all output translations :

$$IoU_{a,r} = \sum_{i=1} \left(\frac{intersection(a_i, r_i)}{union(a_i, r_i)} \right) \tag{13}$$

where a_i is the reference alignment matrix for the i^{th} sentence, made by setting all aligned source positions to 1 and r_i is the upper triangular matrix set to 1 using reads from the policy.[5] The IoU scores for our policy and for MMA are shown in Figure 8 (bottom) with varying sentence lengths. Our policy shows a stronger adherence to the source-target monotonic alignment path.

7 Related Work

Simultaneous Translation. Fixed Policy methods (Ma et al., 2019; Elbayad et al., 2020) follow the fixed rule of waiting for the first k source tokens before generating a target token, and alternate thereafter. Adaptive Wait-k (Zheng et al., 2020) dynamically chooses the best k at every step. Han et al. (2020) applied meta learning in wait-k. Zhang and Feng (2021b) use each attention head as an expert of wait-k policy whereas Zhang and Feng (2021a) introduce a character level wait-k policy. But fixed policy methods aren't feasible for complex inputs and cannot adapt to them. Full-sentence MT has also been leveraged to augment the policy with future information (Zhang et al., 2020; Alinejad et al., 2021). But using such oracle or gold (Zheng et al., 2019; Arthur et al., 2021) READ/WRITE actions does not optimize policy with translation quality. Alinejad et al. (2018) proposes providing future-information on the source side using prediction. Grissom II et al. (2014) predict unseen verbs and uses reinforcement learning to learn when to trust these predictions and when to wait for more input. In contrast, we leverage target side context to strengthen the simultaneous translations.

Zhang and Feng (2022c) train two models on either language directions and make their policies converge. Wilken et al. (2020) propose external ground-truth alignments to train the policy. Papi et al. (2023) use cross attention scores to guide policy. Infinite-lookback (Arivazhagan et al., 2019) and chunkwise (Chiu* and Raffel*, 2018) attention propose to use a soft monotonic attention over previous encoder states. We use a variant of the policy proposed by Ma et al. (2020) that adapts monotonic attention to the multihead architecture of the Transformer. GMA (Zhang and Feng, 2022a) predicts the aligned source position of the current target token and rescales attention based on it. But these methods treat all words equally during training whereas our method improves upon MMA via adaptive training.

Some recent work explores capturing and quantifying *information* from the source tokens and use it to model READ/WRITE actions (Zhang et al., 2022a; Zhang and Feng, 2022b). But these works do not use the target context in their information. Unlike their quantization method, we present a simple scoring by using an auxiliary target-side LM.

Adaptive Training for MT. Target adaptive objectives have been explored by (Lin et al., 2017) which uses probability of a class to scale, but actually only scale down high frequency classes; (Jiang et al., 2019) which directly uses normalized frequency count but have high variance. (Gu et al., 2020) use a chi-square and an exponential distribution function with frequency. However these use only static word frequency. BMI (Xu et al., 2021) attempt to capture mutual information between each source and target token. CBMI (Zhang et al., 2022b) incorporate target context as well, in

[5] We choose this metric to show the extent to which the policy follows the source-target alignments. In an ideal setting, $IoU = 1$.

mutual information. However, these adaptive methods are not directly transferable to the streaming nature of our task.

8 Conclusion

We have presented a simple technique for rescaling target-token importance in simultaneous translation using an information theoretic approach and an adaptive training paradigm. We differentiate the importance of various target tokens by their dependence on the source sentence. To guide our simultaneous translation model, we incorporate a target-side language model that provides an additional signal indicating the importance of each target token or sentence under the condition of the previous target context. Our model shows strong performance on several datasets and outperforms several state-of-the-art techniques in the low latency range (AL<5). Further analysis shows that our technique is better able to translate long sentences and those with rare words. We also showed that the translation path (read/write action sequence) has a stronger correlation to the source-target alignment.

Limitations and Future Work

Since our auxiliary target-side LM decoder is spawned with the same configuration as the MT decoder, this significantly adds to the model size at training time. This makes it difficult to scale/slower to train with translation models of large size. While this problem can be easily mitigated by using a GPU of larger memory, we would like to explore more efficient ways of incorporating the target context which we leave for future work. Secondly, even though our method gives a significant boost to translation quality in the early latencies, it relies on the MMA (Ma et al., 2020) policy that has some limitations in terms of latency because of a suboptimal decision making using multiple heads (Indurthi et al., 2022). While our policy shows improvement, it could be further optimized, for instance, in following reference alignments more closely which would have a positive effect on latency. Finally, using additional monolingual data is also a viable direction for future work to strengthen the language model used in the approach.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. We would also like to thank Logan Born for the numerous discussions that helped shape this paper and Dr. Angel Chang for lending us the GPU resources. The research was partially supported by the Natural Sciences and Engineering Research Council of Canada grants NSERC RGPIN-2018-06437 and RGPAS-2018-522574 and a Department of National Defence (DND) and NSERC grant DGDND-2018-00025 to the third author.

References

Ashkan Alinejad, Hassan S. Shavarani, and Anoop Sarkar. 2021. Translation-based supervision for policy generation in simultaneous neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1734–1744, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ashkan Alinejad, Maryam Siahbani, and Anoop Sarkar. 2018. Prediction improves simultaneous neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3022–3027, Brussels, Belgium. Association for Computational Linguistics.

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, Chung-Cheng Chiu, Semih Yavuz, Ruoming Pang, Wei Li, and Colin Raffel. 2019. Monotonic infinite lookback attention for simultaneous machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1313–1323, Florence, Italy. Association for Computational Linguistics.

Philip Arthur, Trevor Cohn, and Gholamreza Haffari. 2021. Learning coupled policies for simultaneous machine translation using imitation learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2709–2719, Online. Association for Computational Linguistics.

Kehai Chen, Rui Wang, Masao Utiyama, and Eiichiro Sumita. 2020. Content word aware neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 358–364, Online. Association for Computational Linguistics.

Kehai Chen, Rui Wang, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. 2017. Context-aware smoothing for neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 11–20, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Chung-Cheng Chiu* and Colin Raffel*. 2018. Monotonic chunkwise attention. In *International Conference on Learning Representations*.

Kyunghyun Cho and Masha Esipova. 2016. Can neural machine translation do simultaneous translation?

Maha Elbayad, Laurent Besacier, and Jakob Verbeek. 2020. Efficient Wait-k Models for Simultaneous Machine Translation. In *Proc. Interspeech 2020*, pages 1461–1465.

Alvin Grissom II, He He, Jordan Boyd-Graber, John Morgan, and Hal Daumé III. 2014. Don't until the final verb wait: Reinforcement learning for simultaneous machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1342–1352, Doha, Qatar. Association for Computational Linguistics.

Shuhao Gu, Jinchao Zhang, Fandong Meng, Yang Feng, Wanying Xie, Jie Zhou, and Dong Yu. 2020. Token-level adaptive training for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1035–1046, Online. Association for Computational Linguistics.

Hou Jeung Han, Mohd Abbas Zaidi, Sathish Reddy Indurthi, Nikhil Kumar Lakumarapu, Beomseok Lee, and Sangha Kim. 2020. End-to-end simultaneous translation system for IWSLT2020 using modality agnostic meta-learning. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 62–68, Online. Association for Computational Linguistics.

Sathish Reddy Indurthi, Mohd Abbas Zaidi, Beomseok Lee, Nikhil Kumar Lakumarapu, and Sangha Kim. 2022. Infusing future information into monotonic attention through language models.

Shaojie Jiang, Pengjie Ren, Christof Monz, and Maarten de Rijke. 2019. Improving neural response diversity with frequency-aware cross-entropy loss. New York, NY, USA. Association for Computing Machinery.

Surafel Melaku Lakew, Mattia Di Gangi, and Marcello Federico. 2019. Controlling the output length of neural machine translation. In *Proceedings of the 16th International Conference on Spoken Language Translation*, Hong Kong. Association for Computational Linguistics.

Youngwon Lee, Changmin Lee, Hojin Lee, and Seungwon Hwang. 2022. Normalizing mutual information for robust adaptive training for translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8008–8015, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, Hua Wu, and Haifeng Wang. 2019. STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3025–3036, Florence, Italy. Association for Computational Linguistics.

Xutai Ma, Juan Miguel Pino, James Cross, Liezl Puzon, and Jiatao Gu. 2020. Monotonic multihead attention. In *International Conference on Learning Representations*.

Pooya Moradi, Nishant Kambhatla, and Anoop Sarkar. 2019. Interrogating the explanatory power of attention in neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 221–230, Hong Kong. Association for Computational Linguistics.

Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with Markov Chain Monte Carlo. *Prague Bulletin of Mathematical Linguistics*, 106:125–146.

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 *Proceedings of NAACL-HLT 2019: Demonstrations*.

Sara Papi, Marco Turchi, and Matteo Negri. 2023. Alignatt: Using attention-based audio-translation alignments as a guide for simultaneous speech translation.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Patrick Wilken, Tamer Alkhouli, Evgeny Matusov, and Pavel Golik. 2020. Neural simultaneous speech translation using alignment-based chunking. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 237–246, Online. Association for Computational Linguistics.

Yangyifan Xu, Yijin Liu, Fandong Meng, Jiajun Zhang, Jinan Xu, and Jie Zhou. 2021. Bilingual mutual information based adaptive training for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 511–516, Online. Association for Computational Linguistics.

Shaolei Zhang and Yang Feng. 2021a. ICT's system for AutoSimTrans 2021: Robust char-level simultaneous translation. In *Proceedings of the Second Workshop on Automatic Simultaneous Translation*, pages 1–11, Online. Association for Computational Linguistics.

Shaolei Zhang and Yang Feng. 2021b. Universal simultaneous machine translation with mixture-of-experts wait-k policy. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7306–7317, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shaolei Zhang and Yang Feng. 2022a. Gaussian multi-head attention for simultaneous machine translation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3019–3030, Dublin, Ireland. Association for Computational Linguistics.

Shaolei Zhang and Yang Feng. 2022b. Information-transport-based policy for simultaneous translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 992–1013, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Shaolei Zhang and Yang Feng. 2022c. Modeling dual read/write paths for simultaneous machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2461–2477, Dublin, Ireland. Association for Computational Linguistics.

Shaolei Zhang and Yang Feng. 2022d. Reducing position bias in simultaneous machine translation with length-aware framework. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6775–6788, Dublin, Ireland. Association for Computational Linguistics.

Shaolei Zhang, Yang Feng, and Liangyou Li. 2020. Future-guided incremental transformer for simultaneous translation. In *AAAI Conference on Artificial Intelligence*.

Shaolei Zhang, Shoutao Guo, and Yang Feng. 2022a. Wait-info policy: Balancing source and target at information level for simultaneous machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2249–2263, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Songming Zhang, Yijin Liu, Fandong Meng, Yufeng Chen, Jinan Xu, Jian Liu, and Jie Zhou. 2022b. Conditional bilingual mutual information based adaptive training for neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2377–2389, Dublin, Ireland. Association for Computational Linguistics.

Baigong Zheng, Kaibo Liu, Renjie Zheng, Mingbo Ma, Hairong Liu, and Liang Huang. 2020. Simultaneous translation policies: From fixed to adaptive. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2847–2853, Online. Association for Computational Linguistics.

Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. 2019. Simpler and faster learning of adaptive policies for simultaneous translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1349–1354, Hong Kong, China. Association for Computational Linguistics.

A Hyperparameters

Hyperparameter	IWSLT'15 En ↔ Vi IWSLT'14 De ↔ En
encoder layers	6
encoder attention heads	4
encoder embed dim	512
encoder ffn embed dim	1024
decoder layers	6
decoder attention heads	4
decoder embed dim	512
decoder ffn embed dim	1024
dropout	0.3
optimizer	adam
adam-β	(0.9,0.98)
clip-norm	0
lr	5e-4
lr scheduler	inverse sqrt
warmup-updates	4000
warmup-init-lr	1e-7
weight decay	0.0001
label-smoothing	0.1
max tokens	16000

Table 3: Hyperparameters used in our experiments

All models were trained on 2 x Titan RTX with 24 GB memory each. An entire training run finishes within 2.5 hours with fp32 completing about 40 epochs.

B Detailed Results

IWSLT15 En-Vi Transformer-Small					
Full-sentence MT		AP	AL	DAL	BLEU
		1.00	22.08	22.08	28.91
MMA	λ	AP	AL	DAL	BLEU
	0.4	0.58	2.68	3.46	27.73
	0.3	0.59	2.98	3.81	27.90
	0.2	0.63	3.57	4.44	28.47
	0.1	0.67	4.63	5.65	28.42
	0.04	0.70	5.44	6.57	28.33
	0.02	0.76	7.09	8.29	28.28
Wait-K	k	AP	AL	DAL	BLEU
	1	0.63	3.03	3.54	25.21
	3	0.71	4.80	5.42	27.65
	5	0.78	6.46	7.06	28.34
	7	0.83	8.21	8.79	28.60
	9	0.88	9.92	10.51	28.69
Efficient Wait-K	k	AP	AL	DAL	BLEU
	1	0.63	3.06	3.61	26.23
	3	0.71	4.66	5.20	28.21
	5	0.78	6.38	6.94	28.56
	7	1.96	8.13	8.69	28.62
	9	0.87	9.80	10.34	28.52
Wait-Info	$\mathcal{K}$	AP	AL	DAL	BLEU
	1	0.67	3.76	4.33	28.37
	2	0.69	4.10	4.71	28.45
	3	0.71	4.60	5.28	28.54
	4	0.74	5.28	5.97	28.59
	5	0.77	6.01	6.71	28.70
	6	0.80	6.80	7.51	28.78
	7	0.82	7.61	8.33	28.80
	8	0.84	8.39	9.11	28.82
MMA+TC	λ	AP	AL	DAL	BLEU
	0.55	0.66	3.1	5.12	28.6
	0.5	0.67	3.60	5.78	28.81
	0.3	0.68	3.86	6.12	28.9
	0.2	0.71	4.58	7.22	28.74
	0.1	0.74	5.34	8.18	28.65
	0.01	0.89	9.89	14.37	28.67

Table 4: Experiments on IWSLT15 English → Vietnamese

IWSLT15 Vi - En Transformer-Small					
Full-sentence MT (Offline)		AP	AL	DAL	BLEU
		1.00	27.56	27.56	26.11
MMA	λ	AP	AL	DAL	BLEU
	0.4	0.63	3.60	6.96	25.36
	0.3	0.64	3.95	7.59	24.75
	0.2	0.67	4.54	9.09	25.33
	0.1	0.75	7.14	11.60	25.84
	0.05	0.77	7.61	15.70	25.31
	0.01	0.88	13.63	23.95	26.11
Wait-K	k	AP	AL	DAL	BLEU
	1	0.42	-2.89	1.62	7.57
	3	0.53	-0.18	3.24	14.66
	5	0.61	1.49	5.08	17.44
	7	0.67	3.28	7.05	19.02
	9	0.76	6.75	8.96	22.39
	11	0.80	7.91	10.71	23.28
	13	0.84	10.37	12.36	24.80
Wait-Info	$\mathcal{K}$	AP	AL	DAL	BLEU
	4	0.62	2.58	5.06	22.45
	5	0.67	4.08	6.27	23.75
	6	0.72	5.61	7.72	25.19
	7	0.76	7.01	9.19	25.45
	8	0.79	8.26	10.66	25.86
	9	0.82	9.37	11.98	25.93
	10	0.84	10.56	13.30	26.13
MMA+TC	λ	AP	AL	DAL	BLEU
	0.4	0.63	3.51	5.902	26.38
	0.3	0.65	4.01	6.558	26.04
	0.2	0.67	4.62	7.527	26.32
	0.1	0.71	5.67	9.212	26.63
	0.05	0.76	7.23	10.579	26.52
	0.04	0.77	7.55	11.76	26.85
	0.01	0.89	13.31	18.627	26.67

Table 5: Experiments on IWSLT15 Vietnamese → English

IWSLT15 De-En Transformer-Small					
Full-sentence MT		AP	AL	DAL	BLEU
(Offline)		1.00	22.97	22.97	33.64
MMA	λ	AP	AL	DAL	BLEU
	0.4	0.67	3.91	6.36	30.8
	0.3	0.69	4.27	6.84	31.12
	0.2	0.72	4.97	7.82	31.34
	0.1	0.77	6.08	9.47	31.95
Wait-Info	$\mathcal{K}$	AP	AL	DAL	BLEU
	1	0.57	1.32	2.53	26.26
	2	0.59	1.97	3.17	27.39
	3	0.64	3.08	4.35	29.01
	4	0.69	4.27	5.61	30.36
	5	0.739	5.30	6.84	30.92
	6	0.77	6.26	8.03	31.45
	7	0.80	7.17	9.09	31.82
	8	0.82	8.06	9.94	32.05
Wait-K	k		AL		BLEU
	3		1.8		26
	5		4		28.6
	7		6		29.7
	9		8		31.5
Efficient Wait-K	k		AL		BLEU
	3		2		26.4
	5		4		27
	7		6		30
	9		8		31.7
MMA+TC	λ	AP	AL	DAL	BLEU
	0.5	0.66	3.68	5.92	30.97
	0.4	0.68	4.06	6.51	31.33
	0.3	0.70	4.49	7.12	31.69
	0.2	0.73	5.06	7.93	32.2
	0.1	0.77	6.10	9.54	32.22

Table 6: Experiments on IWSLT14 German → English

IWSLT15 En-De Transformer-Small					
Full-sentence MT		AP	AL	DAL	BLEU
(Offline)		1.00	22.21	22.21	27.46
MMA	λ	AP	AL	DAL	BLEU
	0.5	0.69	4.32	6.42	26.03
	0.4	0.71	4.70	6.95	26.20
	0.3	0.72	4.97	7.28	26.30
	0.2	0.74	5.44	7.96	26.19
	0.1	0.79	6.86	9.72	26.77
	0.05	0.84	8.25	11.42	26.91
Wait-Info	$\mathcal{K}$	AP	AL	DAL	BLEU
	1	0.61	2.62	3.09	21.75
	2	0.63	3.15	3.89	22.42
	3	0.68	4.24	5.30	24.48
	4	0.73	5.36	6.77	25.60
	5	0.77	6.38	8.09	26.18
	6	0.80	7.23	9.18	26.35
	7	0.83	8.23	10.35	26.61
	8	0.86	9.25	11.46	26.74
Wait-K	k		AL		BLEU
	3		3.41		22.00
	5		5.00		25.21
	7		6.83		26.32
	9		8.72		26.61
Efficient Wait-K	k		AL		BLEU
	3		3.51		23.01
	5		5.27		24.80
	7		7.03		25.93
	9		8.81		26.11
MMA+TC	λ	AP	AL	DAL	BLEU
	0.6	0.68	4.04	6.07	26.03
	0.5	0.69	4.19	6.25	26.19
	0.4	0.69	4.38	6.52	26.43
	0.3	0.71	4.87	7.14	26.56
	0.2	0.74	5.51	8.09	26.71
	0.1	0.79	6.74	9.80	26.76
	0.06	0.82	7.75	10.94	27.01

Table 7: Experiments on IWSLT14 English $\rightarrow$ German