Modeling Dual Read/Write Paths for Simultaneous Machine Translation

Shaolei Zhang [1,2], **Yang Feng** [1,2]*
[1]Key Laboratory of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences (ICT/CAS)
[2] University of Chinese Academy of Sciences, Beijing, China
{zhangshaolei20z, fengyang}@ict.ac.cn

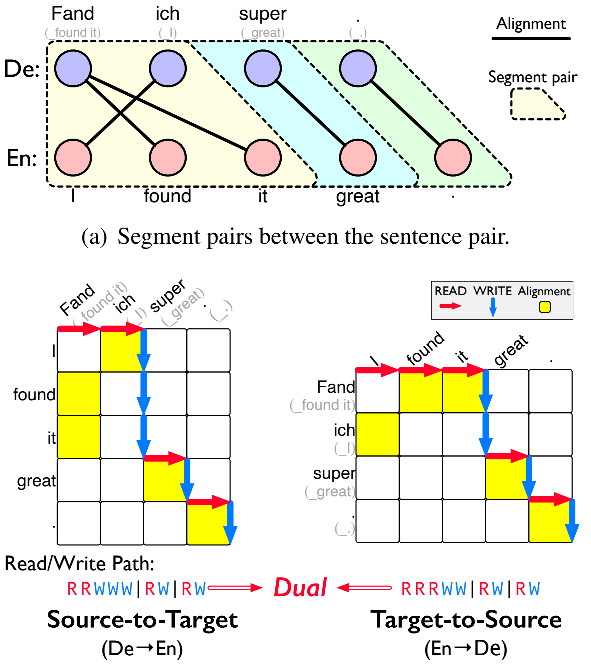
Abstract

Simultaneous machine translation (SiMT) outputs translation while reading source sentence and hence requires a policy to decide whether to wait for the next source word (READ) or generate a target word (WRITE), the actions of which form a *read/write path*. Although the read/write path is essential to SiMT performance, no direct supervision is given to the path in the existing methods. In this paper, we propose a method of dual-path SiMT which introduces duality constraints to direct the read/write path. According to duality constraints, the read/write path in source-to-target and target-to-source SiMT models can be mapped to each other. As a result, the two SiMT models can be optimized jointly by forcing their read/write paths to satisfy the mapping. Experiments on En↔Vi and De↔En tasks show that our method can outperform strong baselines under all latency.

1 Introduction

Simultaneous machine translation (SiMT) (Cho and Esipova, 2016; Gu et al., 2017; Ma et al., 2019; Arivazhagan et al., 2019), which outputs translation while reading source sentence, is important to many live scenarios, such as simultaneous interpretation, live broadcast and synchronized subtitles. Different from full-sentence machine translation which waits for the whole source sentence, SiMT has to decide whether to wait for the next source word (i.e., READ action) or translate a target word (i.e., WRITE action) to complete the translation.

The sequence of READ and WRITE actions in the translation process form a *read/write path*, which is key to SiMT performance. Improper read/write path will bring damage to translation performance as compared to the following WRITE actions too many but not *necessary* READ actions will result in high translation latency while too few but not *sufficient* READ actions will exclude indispensable source information. Therefore, an ideal read/write path is that the READ actions compared to the following WRITE actions are just *sufficient* and *necessary*, which means the source words covered by consecutive READ actions and the target words generated by the following consecutive WRITE actions should be semantically equivalent.



(a) Segment pairs between the sentence pair.

(b) The duality between the read/write paths in two directions.

Figure 1: An example of duality constraints. With duality constraints, the read/write paths of source-to-target and target-to-source translation should project to the same segment pairs between two languages.

Ensuring sufficiency and necessity between READ/WRITE actions will lead to a proper read/write path and thereby good SiMT performance. But unfortunately, the existing SiMT methods, which employ a fixed or adaptive policy, do not consider the sufficiency or necessity in their policy. The fixed policy performs SiMT based on

[*] Corresponding author: Yang Feng.
Code is available at: https://github.com/ictnlp/Dual-Path

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 2461 - 2477
May 22-27, 2022 ©2022 Association for Computational Linguistics

a pre-defined read/write path (Dalvi et al., 2018; Ma et al., 2019), where the number of READ actions before WRITE is fixed. The adaptive policy (Gu et al., 2017; Zheng et al., 2019b; Arivazhagan et al., 2019; Zheng et al., 2019a; Ma et al., 2020; Liu et al., 2021) dynamically decides to READ or WRITE guided by translation quality and total latency, but skips the evaluation of sufficiency and necessity between READ/WRITE actions.

Under these grounds, we aim at introducing the evaluation of sufficiency and necessity between READ/WRITE actions to direct the read/write path without involving external information. As mentioned above, in an ideal read/write path, the source segment (i.e., source words read by the consecutive READ actions) and the corresponding target segment (i.e., target words generated by the following consecutive WRITE actions) are supposed to be semantically equivalent and thus translation to each other, which constitutes a separate *segment pair*. Hence, an ideal read/write path divides the whole sentence pair into a sequence of segment pairs where the source sentence and the target sentence should be translation to each other segment by segment. That means if the translation direction is reversed, an ideal read/write path for target-to-source SiMT can also be deduced from the same sequence of segment pairs. For example, according to the alignment in Figure 1(a), the ideal read/write paths should be 'RRWWW|RW|RW' in De→En SiMT and 'RRRWW|RW|RW' in En→De SiMT, as shown in Figure 1(b), both of which share the same segment pair sequence of $\langle$*Fand ich*, *I fount it*$\rangle$, $\langle$*super*, *great*$\rangle$ and $\langle .,. \rangle$. Therefore, agreement on the segment pairs derived from read/write paths in source-to-target and target-to-source SiMT, called *duality constraints*, can be a good choice to evaluate sufficiency and necessity between READ/WRITE actions.

Based on the above reasoning, we propose a method of *Dual-Path SiMT*, which uses the SiMT model in the reverse direction to guide the SiMT model in the current direction according to duality constraints between their read/write paths. With duality constraints, the read/write paths in source-to-target and target-to-source SiMT should reach an agreement on the corresponding segment pairs. Along this line, our method maintains a source-to-target SiMT model and a target-to-source SiMT model concurrently, which respectively generate their own read/write path using monotonic multi-head attention (Ma et al., 2020). By minimizing the difference between the segment pairs derived from the two read/write paths, the two SiMT models successfully converge on the segment pairs and provide supervision to each other. Experiments on IWSLT15 En↔Vi and WMT15 De↔En SiMT tasks show that our method outperforms strong baselines under all latency, including the state-of-the-art adaptive policy.

2 Background

We first briefly introduce SiMT with a focus on monotonic multi-head attention (Ma et al., 2020).

For a SiMT task, we denote the source sentence as $\mathbf{x} = \{x_1, \cdots, x_J\}$ and the corresponding source hidden states as $\mathbf{m} = \{m_1, \cdots, m_J\}$, where J is the source length. The model generates target sentence $\mathbf{y} = \{y_1, \cdots, y_I\}$ with target hidden states $\mathbf{s} = \{s_1, \cdots, s_I\}$, where I is the target length. During translating, SiMT model decides to read a source word (READ) or write a target word (WRITE) at each step, forming a read/write path.

Read/write path can be represented in multiple forms, such as an action sequence of READ and WRITE (e.g., RRWWWRW$\cdots$), or a path from $(0, 0)$ to (I, J) in the attention matrix from the target to source, where moving right (i.e., $\rightarrow$) means READ action and moving down (i.e., $\downarrow$) means WRITE action, as shown in Figure 1(b).

Mathematically, a read/write path can be represented by a monotonic non-decreasing sequence $\{g_i\}_{i=1}^{I}$ of step i, where the g_i represents the number of source words read in when writing the i^{th} target word y_i. The value of $\{g_i\}_{i=1}^{I}$ depends on the specific SiMT policy, where monotonic multi-head attention (MMA) (Ma et al., 2020) is the current state-of-the-art SiMT performance via modeling READ/WRITE action as a Bernoulli variable.

Monotonic multi-head attention MMA processes the source words one by one, and concurrently predicts a selection probability p_{ij} to indicates the probability of writing y_i when reading x_j, and accordingly a Bernoulli random variable z_{ij} is calculated to determine READ or WRITE action:

$$p_{ij} = \text{Sigmoid}\left(\frac{m_j V^K (s_{i-1} V^Q)^\top}{\sqrt{d_k}}\right), \quad (1)$$

$$z_{ij} \sim \text{Bernoulli}\left(p_{ij}\right), \quad (2)$$

where V^K and V^Q are learnable parameters, d_k is dimension of head. If $z_{ij} = 0$, MMA performs READ action to wait for the next source word x_{j+1}.

If $z_{ij}=1$, MMA sets $g_i=j$ and performs WRITE action to generate y_i based on $x_{\leq g_i}$. Therefore, the decoding probability of $\mathbf{y}$ with parameters $\boldsymbol{\theta}$ is

$$p(\mathbf{y} \mid \mathbf{x};\boldsymbol{\theta})=\prod_{i=1}^{I} p\left(y_i \mid \mathbf{x}_{\leq g_i}, \mathbf{y}_{<i};\boldsymbol{\theta}\right), \tag{3}$$

where $\mathbf{x}_{\leq g_i}$ are first g_i source tokens, and $\mathbf{y}_{<i}$ are previous target tokens.

Note that when integrated into multi-head attention, all attention heads in decoder layers independently determine the READ/WRITE action. If and only when all heads decide to perform WRITE action, the model starts translating, otherwise the model waits for the next source word.

Expectation training Since sampling a discrete random variable z_{ij} precludes back-propagation, MMA applies expectation training Raffel et al. (2017) to replace z_{ij} with a *expected writing probability*, denoted as

$$\boldsymbol{\alpha}=(\alpha_{ij})_{I\times J}, \tag{4}$$

where α_{ij} calculates the expectation probability of writing y_i when reading x_j. Then, the attention distribution and context vectors are accordingly calculated in the expected form.

To trade-off between translation quality and latency, MMA introduces a latency loss $\mathcal{L}_g$ to the training loss:

$$\mathcal{L}(\boldsymbol{\theta})=-\sum_{(\mathbf{x},\mathbf{y})}\log p\left(\mathbf{y} \mid \mathbf{x};\boldsymbol{\theta}\right)+\lambda\mathcal{L}_g, \tag{5}$$

where $\mathcal{L}_g$ measures the total latency, and λ is the weight of latency loss. Please refer to Arivazhagan et al. (2019) and Ma et al. (2020) for more detailed derivation and implementation.

3 The Proposed Method

Our dual-path SiMT model employs a source-to-target (*forward*) model and a target-to-source (*backward*) model, called single-path SiMT, which generate their own read/write path based on MMA. According to duality constraints that the read/write paths of the two single-path SiMT models should share the same segment pair sequence, the two read/write paths should be transposed to each other in principle as shown in Figure 1. But in practice, after transposing one of the read/write paths, there is always a gap between the transposed read/write path and the original one in the reverse translation direction. By closing the gap between the aforementioned transposed and original read/write paths, as shown in Figure 2, duality constraints are introduced into the dual-path SiMT model and thereby the two single-path SiMT models can provide guidance to each other. In what follows, we will introduce how to get the transposed read/write path (Sec.3.1) and how to reduce the gap (Sec.3.2).

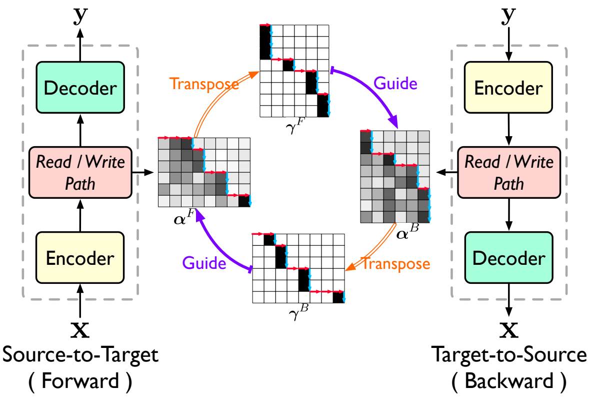
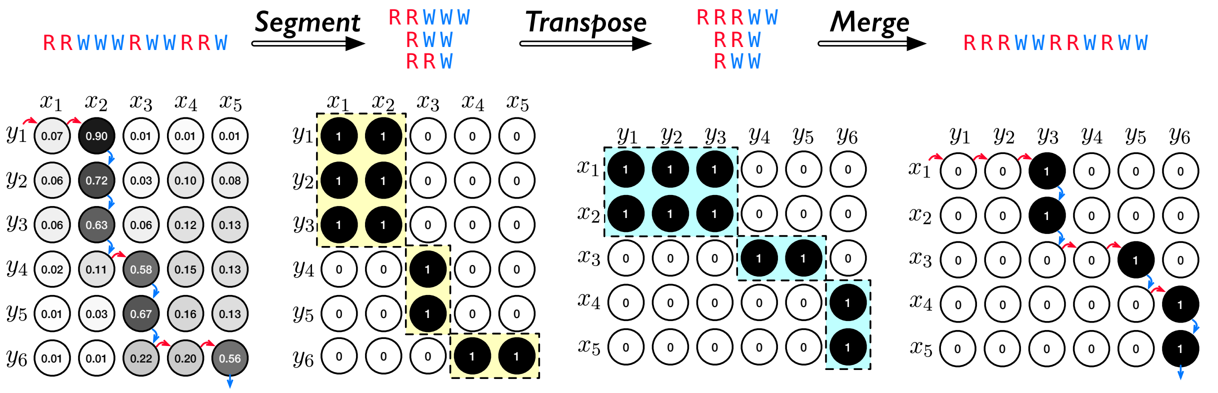


Figure 2: The architecture of dual-path SiMT, consisting of the forward and backward single-path SiMT models. To accomplish the duality constraints, we generate the transposed path of the forward (or backward) read/write path, and use this transposed path to direct the read/write path in another direction; vice versa.

3.1 Transposing the Read/Write Path

The purpose of transposing a read/write path is to get a new read/write path in the reverse direction based on the same segment pairs as the original path. As the transposing process works in the same way for the two directions, we just introduce the process for the forward single-path SiMT. Since there is no explicit read/write path in the training of single-path SiMT model, the transposing process can only use the expected writing probability matrix $\boldsymbol{\alpha}$ as the input, shown in Eq.(4). Similarly, the output of the transposing process is the transposed writing probability matrix $\boldsymbol{\gamma}=(\gamma_{ji})_{J\times I}$ calculated from the transposed read/write path, which will be used to guide the backward single-path SiMT.

The transposing process consists of three steps. First, derive the read/write path from the expected writing probability matrix $\boldsymbol{\alpha}$ and *segment* the sentence pair into a sequence of segment pairs. Second, *transpose* the sequence of segment pairs into the corresponding one for the backward SiMT. Last, *merge* the transposed segment pairs to get the transposed path and then project it to $\boldsymbol{\gamma}$. In the following, we will introduce the steps of segment, transpose and merge in details.



(a) Writing probability $\boldsymbol{\alpha}$

(b) Segment pairs

(c) Transposed segment pairs

(d) Transposed writing probability $\boldsymbol{\gamma}$

Figure 3: Simplified diagrams of generating transposed writing probability $\boldsymbol{\gamma}$ from the writing probability $\boldsymbol{\alpha}$. (a→b) Segment the sentence pair into a sequence of segment pairs. (b→c) Transpose the segment pairs to fit the backward SiMT. (c→d) Merge the transposed segment pairs to get transposed writing probability with the transposed path.

Segment Given the expected writing probability matrix $\boldsymbol{\alpha}$, to get the read/write path, we first find out the source position d_i that the WRITE action for each target position i corresponds to, which is

$$d_i = \underset{j}{\operatorname{argmax}}\ \alpha_{ij}. \tag{6}$$

According to the property of monotonic attention, there are some consecutive WRITE actions that corresponds to the same source position, so the target words generated by the consecutive WRITE actions form a target segment. Formally, we assume there are K target segments in total, denoted as $\mathbf{y} = \{\bar{\mathbf{y}}_1, \cdots, \bar{\mathbf{y}}_k, \cdots, \bar{\mathbf{y}}_K\}$. For each target segment $\bar{\mathbf{y}}_k = (y_{b^y_k}, \cdots, y_{e^y_k})$, where b^y_k and e^y_k are its beginning and end target positions, we can get the corresponding source segment as $\bar{\mathbf{x}}_k = (x_{b^x_k}, \cdots, x_{e^x_k})$ where

$$b^x_k = \begin{cases} 1 & k=1 \\ d_{e^y_{k-1}} + 1 & \text{otherwise} \end{cases} \tag{7}$$

and

$$e^x_k = d_{b^y_k}. \tag{8}$$

Thus the sentence pairs $\langle \mathbf{x}, \mathbf{y} \rangle$ can be segmented into the sequence of segment pairs as $\langle \bar{\mathbf{x}}_1, \bar{\mathbf{y}}_1 \rangle \mid \cdots \mid \langle \bar{\mathbf{x}}_K, \bar{\mathbf{y}}_K \rangle$. By replacing the source words with READ actions and target words with WRITE actions, we can get the action segment pairs. Then, the read/write path is formed by concatenating all the action segment pairs, where the length of the read/write path is equal to the total number of source words and target words.

For the example shown in Figure 3(a), the sequence of source positions d_i corresponding to WRITE actions for the whole target sentence is $2, 2, 2, 3, 3, 5$, with the corresponding read/write path RRWWWRWWRRW. Then, we can get the sequence of segment pairs as $\langle x_1\, x_2,\ y_1\, y_2\, y_3 \rangle \mid \langle x_3,\ y_4\, y_5 \rangle \mid \langle x_4\, x_5,\ y_6 \rangle$, and thereby the sequence of action segment pairs as $\langle$RR, WWW$\rangle \mid \langle$R, WW$\rangle \mid \langle$RR, W$\rangle$ shown in Figure 3(b).

Transpose After getting the sequence of segment pairs, the transposed read/write path can be derived from it. As the transposed read/write path is in the form to fit the backward single-path SiMT, the sequence of segment pairs should also be transposed to fit the another direction. According to duality constraints, the sequence of segment pairs is shared by forward and backward SiMT, so we only need to exchange the source segment and target segment in each segment pair, that is from $\langle \bar{\mathbf{x}}_k, \bar{\mathbf{y}}_k \rangle$ to $\langle \bar{\mathbf{y}}_k, \bar{\mathbf{x}}_k \rangle$, where the beginning and end positions of each source/target segment remain the same. Then, we get the corresponding transposed action segment pairs by replacing target words with READ actions and source words with WRITE actions. In this way, we accomplish the transposing of segment pairs. Let's review the example in Figure 3(b), after transposing, the sequence of segment pairs as $\langle y_1\, y_2\, y_3,\ x_1\, x_2 \rangle \mid \langle y_4\, y_5,\ x_3 \rangle \mid \langle y_6,\ x_4\, x_5 \rangle$, and the corresponding sequence of transposed action segment pairs is $\langle$RRR, WW$\rangle \mid \langle$RR, W$\rangle \mid \langle$R, WW$\rangle$ as shown in Figure 3(c).

Merge By merging the transposed action segment pairs, we can get the transposed read/write path. The goal of the transposing process is to get the transposed writing probability matrix $\boldsymbol{\gamma}$ to constrain the excepted writing probability matrix for the backward single-path SiMT. According to the definition of the writing probability matrix, only the last column in the sub-matrix covered by each segment pair corresponds to WRITE actions. Formally, for each transposed segment pair $\langle \bar{\mathbf{y}}_k, \ \bar{\mathbf{x}}_k \rangle$, the following elements in $\boldsymbol{\gamma}$ should have the greatest probability to perform WRITE actions as $\{\gamma_{b_k^x e_k^y}, \cdots, \gamma_{e_k^x e_k^y}\}$. For the three sub-matrices shown in Figure 3(c), only the elements of the last column correspond to WRITE actions as shown in Figure 3(d), which are $\{\gamma_{13}, \gamma_{23}, \gamma_{35}, \gamma_{46}, \gamma_{56}\}$. We employ the $0-1$ distribution to set the value of elements in $\boldsymbol{\gamma}$, where the elements corresponding to WRITE actions are set to 1 and others are set to 0. This is equivalent to the situation that the selection probability for the Bernoulli distribution (in Eq.(2)) is 1.

3.2 Training

Assuming the expected writing probability matrix for the forward single-path SiMT is $\boldsymbol{\alpha}^F$ and its transposed expected writing probability matrix is $\boldsymbol{\gamma}^F$, and similarly in the backward single-path SiMT, the matrices are $\boldsymbol{\alpha}^B$ and $\boldsymbol{\gamma}^B$, respectively. We reduce the gap between the read/write path with the transposed path of read/write path in another direction by minimizing L_2 distance between their corresponding expected writing probability matrix as follows:

$$\Omega^F = \left\| \boldsymbol{\alpha}^F - \boldsymbol{\gamma}^B \right\|_2 \tag{9}$$

$$\Omega^B = \left\| \boldsymbol{\alpha}^B - \boldsymbol{\gamma}^F \right\|_2 . \tag{10}$$

Two L_2 distances are added to the training loss as a regularization term and final training loss is

$$\mathcal{L} = \mathcal{L}\big(\boldsymbol{\theta}^F\big) + \mathcal{L}\big(\boldsymbol{\theta}^B\big) + \lambda_{dual}(\Omega^F + \Omega^B), \tag{11}$$

where $\mathcal{L}\big(\boldsymbol{\theta}^F\big)$ and $\mathcal{L}\big(\boldsymbol{\theta}^B\big)$ are the loss function of the forward and backward single-path SiMT model respectively, calculated as Eq.(5). λ_{dual} is a hyperparameter and we set $\lambda_{dual}=1$ in our experiments.

In the inference time, the forward and backward single-path SiMT models can be used separately, depending on the required translation direction.

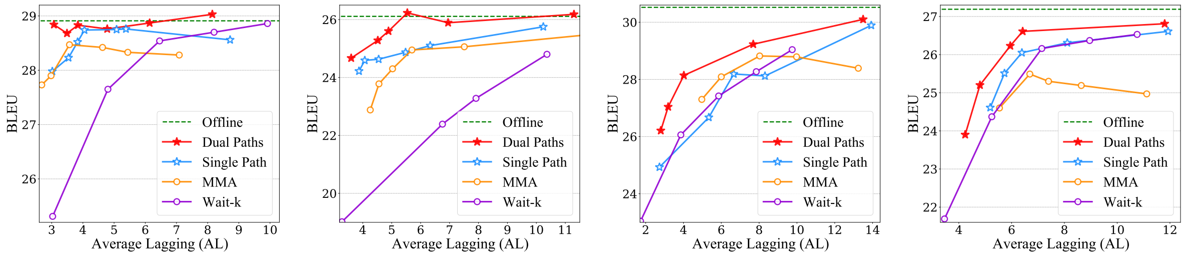
4 Related Work

Dual learning is widely used in dual tasks, especially machine translation. For both unsupervised (He et al., 2016; Artetxe et al., 2019; Sestorain et al., 2019) and supervised NMT (Xia et al., 2017; Wang et al., 2018), dual learning can provide additional constraints by exploiting the dual correlation. Unlike most previous dual learning work on NMT, which use the reconstruction between source and target sequences, we focus on SiMT-specific read/write path and explorer its intrinsic properties.

SiMT policy falls into two categories: fixed and adaptive. For fixed policy, the read/write path is defined by rules and fixed during translating. Dalvi et al. (2018) proposed STATIC-RW, which alternately read and write RW words after reading S words. Ma et al. (2019) proposed wait-k policy, which always generates target k tokens lagging behind the source input. Elbayad et al. (2020) enhanced wait-k policy by sampling different k during training. Han et al. (2020) applied meta-learning in wait-k. Zhang et al. (2021) proposed future-guided training to apply a full-sentence MT model to guide wait-k policy. Zhang and Feng (2021a) proposed a char-level wait-k policy. Zhang and Feng (2021b) proposed a universal SiMT with mixture-of-experts wait-k policy to perform SiMT under arbitrary latency levels.

For adaptive policy, the read/write path is learned and adaptive to the current context. Early adaptive policies used segmented translation (Bangalore et al., 2012; Cho and Esipova, 2016; Siahbani et al., 2018). Gu et al. (2017) trained an agent with reinforcement learning. Alinejad et al. (2018) added a predict operation based on Gu et al. (2017). Zheng et al. (2019a) trained an agent with golden READ/WRITE actions generated by rules. Zheng et al. (2019b) added a "delay" token to read source words. Arivazhagan et al. (2019) proposed MILk, which applied monotonic attention and used a Bernoulli variable to determine writing. Ma et al. (2020) proposed MMA, which is the implementation of MILk on the Transformer and achieved the current state-of-the-art SiMT performance. Zhang et al. (2020) proposed a adaptive segmentation policy. Wilken et al. (2020) used the external ground-truth alignments to train the policy. Liu et al. (2021) proposed cross-attention augmented transducer. Alinejad et al. (2021) introduced a full-sentence model to generate a ground-truth action sequence. Miao et al. (2021) proposed a generative SiMT policy.

The previous methods often lack the internal supervision on read/write path. Some works use external information such as alignment or generated



(a) En→Vi (b) Vi→En (c) De→En (d) En→De

Figure 4: Translation quality (BLEU) against latency (AL) on the En↔Vi and De↔En. We show the results of Dual Paths, Single Path, MMA (the current SOTA adaptive policy), Wait-k and Offline model.

rule-based sequences to guide the read/write path (Zheng et al., 2019a; Zhang et al., 2020; Wilken et al., 2020; Alinejad et al., 2021). However, these methods rely too much on heuristic rules, and thus their performance is not comparable to jointly optimizing read/write path and translation. Our method internally explorers the duality between the read/write paths in two directions, and accordingly uses the duality to constrain the read/write paths, thereby obtaining better SiMT performance.

5 Experiments

5.1 Datasets

We evaluated our method on four translation directions of the following two public datasets.

IWSLT15[1] **English↔Vietnamese (En↔Vi)** (133K pairs) (Cettolo et al., 2015) We use TED tst2012 (1553 pairs) as validation set and TED tst2013 (1268 pairs) as test set. Following Raffel et al. (2017) and Ma et al. (2020), we replace tokens that the frequency less than 5 by $\langle unk \rangle$. After replacement, the vocabulary sizes are 17K and 7.7K for English and Vietnamese, respectively.

WMT15[2] **German↔English (De↔En)** (4.5M pairs) Following Ma et al. (2020), we use newstest2013 (3000 pairs) as validation set and newstest2015 (2169 pairs) as test set. BPE (Sennrich et al., 2016) is applied with 32K merge operations and the vocabulary is shared across languages.

5.2 System Setting

We conducted experiments on following systems.

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

Wait-k Wait-k policy, the widely used fixed policy Ma et al. (2019), which first reads k source tokens and then writes a target word and reads a word alternately.

MMA[3] Monotonic multi-head attention (MMA) proposed by (Ma et al., 2020), the state-of-the-art adaptive policy for SiMT, which applies monotonic attention on each head in Transformer.

Single Path SiMT model of one translation direction based on monotonic multi-head attention. To avoiding outlier heads[4] that are harmful for the read/write path, we slightly modified MMA for more stable performance. We no longer let the heads in all decoder layers independently determine the READ/WRITE action, but share the READ/WRITE action between the decoder layers.

Dual Paths Dual-path SiMT described in Sec.3.

The implementations of all systems are adapted from Fairseq Library (Ott et al., 2019), based on Transformer (Vaswani et al., 2017), where we apply Transformer-Small (4 heads) for En↔Vi, and Transformer-Base (8 heads) for De↔En. For 'Dual Paths', the forward and backward models are used to complete the SiMT on two translation directions at the same time. To perform SiMT under different latency, we set various lagging numbers[5] k for 'Wait-k', and set various latency weights[67] λ for 'MMA', 'Single Path' and 'Dual Paths'.

We evaluate these systems with BLEU (Papineni

[1] nlp.stanford.edu/projects/nmt/

[2] www.statmt.org/wmt15/

[3] github.com/pytorch/fairseq/tree/master/examples/simultaneous_translation

[4] Since MMA requires all heads in decoder layers to independently decide READ/WRITE action and starts translating only when all heads select WRITE action, some outlier heads that perform too many READ actions will result in higher latency. Ma et al. (2020) try to control this phenomenon by adding some loss functions, but it still cannot avoid some outlier heads waiting for too many words, which seriously affects the impair the necessity between the READ/WRITE actions in read/write path (Ma et al., 2020; Zaidi et al., 2021).

[5] For both En↔Vi and De↔En: $k = 1, 3, 5, 7, 9$

[6] For En↔Vi: $\lambda = 0.01, 0.05, 0.1, 0.2, 0.3, 0.4$

[7] For De↔En: $\lambda = 0.1, 0.2, 0.25, 0.3, 0.4$

	AL	BLEU
Dual Paths	7.69	29.23
-w/o Segment	7.61	27.24
-w/o Ω^B	8.57	28.66
-w/o Ω^F, Ω^B	8.31	28.12

Table 1: Ablation study with $\lambda = 0.2$. 'w/o Segment': remove the segment operation in transposing process of read/write path, and directly perform transposition. 'w/o Ω^B': remove Ω^B in Eq.(11), only constrain forward model. 'w/o Ω^F, Ω^B': remove the duality constraints between read/write paths.

et al., 2002) for translation quality and Average Lagging (AL) (Ma et al., 2019) for latency. Average lagging evaluates the number of words lagging behind the ideal policy. Given read/write path g_i, AL is calculated as

$$\mathrm{AL} = \frac{1}{\tau} \sum_{i=1}^{\tau} g_i - \frac{i-1}{|\mathbf{y}| \, / \, |\mathbf{x}|}, \tag{12}$$

where $\tau = \underset{i}{\operatorname{argmax}} \left(g_i = |\mathbf{x}| \right)$, and $|\mathbf{x}|$ and $|\mathbf{y}|$ are the length of the source sentence and target sentence respectively. The results with more latency metrics are shown in Appendix D.

5.3 Main Results

Figure 4 shows the comparison between our method and the previous methods on 4 translation directions. 'Dual Paths' outperforms the previous methods under all latency, and more importantly, the proposed duality constraints can improve the SiMT performance on both source-to-target and target-to-source directions concurrently.

Compared to 'Wait-k', our method has significant improvement, especially under low latency, since the read/write path in 'Wait-k' is fixed and cannot be adjusted. Compared to 'MMA', the state-of-the-art adaptive policy, our 'Single Path' achieves comparable performance and is more stable under high latency. 'MMA' allows each head of each layer to independently predict a read/write path, where some outlier heads will affect the overall performance, resulting in a decline in translation quality under high latency (Ma et al., 2020). Our method applies a common read/write path instead of the heads in each layer to predict READ/WRITE, thereby reducing the possibility of outlier heads. Based on 'Single Path', 'Dual Paths' further improves the SiMT performance by modeling the duality constraints between read/write paths, especially under low latency. Besides, our method improves the SiMT performance even close to the full-sentence MT on En↔Vi, which shows that the more precise read/write path is the key to SiMT performance. Additionally, under the same latency weight λ, our method tends to have lower latency than 'MMA' on De↔En. The 'Single Path' reduces the unnecessary latency caused by outlier heads, and the duality constraints further improve the necessity of reading source content, thereby achieving lower latency.

6 Analysis

We conducted extensive analyses to understand the specific improvements of our method. Unless otherwise specified, all results are reported on De→En.

6.1 Ablation Study

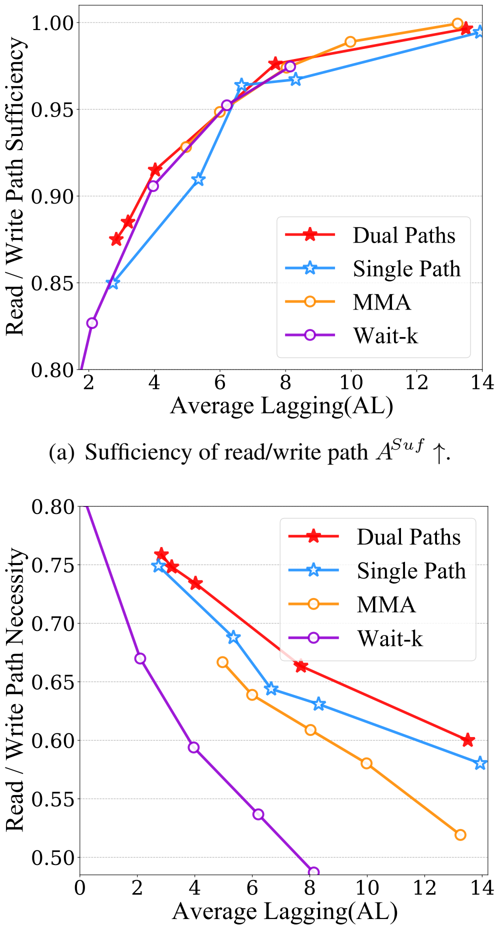
We conducted ablation studies on the duality constraints, where we use direct transposition to replace transposing process of read/write path, only constrain the forward single-path model or remove the duality constraints. As shown in Table 1, the proposed method of transposing the read/write path is critical to translation quality, showing the importance of the segment operation. Besides, mutual constraining between forward and backward single-path model is more conducive to SiMT performance than only constraining one of them or removing constraints.

6.2 Evaluation of Read/Write Path

The read/write path needs to ensure sufficient content for translation and meanwhile avoid unnecessary latency, where the aligned source position[8] is always considered as the oracle position to perform WRITE in previous work (Wilken et al., 2020; Arthur et al., 2021). Therefore, we propose two metrics A^{Suf} and A^{Nec} to measure the *sufficiency* and *necessity* between the READ/WRITE actions in a path via alignments. We denote the ground-truth aligned source position of the i^{th} target word as a_i, and the read/write path is represented by g_i, which is the number of source words read in when writing the i^{th} target word. For sufficiency, A^{Suf} is used to evaluate whether the aligned source word is read before writing the target word, calculated as

$$A^{Suf} = \frac{1}{I} \sum_{i=1}^{I} \mathbb{1}_{a_i \leq g_i}, \tag{13}$$

[8] For many-to-one alignment from source to target, we choose the furthest source word. For the target words with no alignment, we ignore them.



(a) Sufficiency of read/write path A^{Suf} ↑.

(b) Necessity of read/write path A^{Nec} ↑.

Figure 5: Sufficiency evaluation and necessity evaluation of the read/write path.

where $\mathbb{1}_{a_i \leq g_i}$ counts the number of $a_i \leq g_i$, and I is the target length. For necessity, A^{Nec} is used to measure the distance between the output position g_i and the aligned source position a_i, calculated as

$$A^{Nec} = \frac{1}{|a_i \leq g_i|} \sum_{i, a_i \leq g_i} \frac{a_i}{g_i}, \qquad (14)$$

where the best case is $A^{Nec} = 1$ for $g_i = a_i$, performing WRITE just at the aligned position and there is no unnecessary waiting. The more detailed description please refers to Appendix A.

As shown in Figure 5, we evaluate the A^{Suf} and A^{Nec} of read/write path on RWTH De→En alignment dataset [9], whose reference alignments are manually annotated by experts. The read/write paths of all methods perform similarly in sufficiency evaluation and our method performs slightly better at low latency. Except that the fixed policy 'Wait-k' may be forced to start translating before reading the aligned source word under the lower latency, 'MMA' and our method can almost cover more than 85% of the aligned source word when starting translating. In the necessity evaluation, our method surpasses 'Wait-k' and 'MMA', and starts translation much closer to the aligned source word, which shows that duality constraints make read/write path more precise, avoiding some unnecessary waiting. Note that while avoiding unnecessary waiting, our method also improves the translation quality (see Figure 4) under the same latency, which further shows the importance of a proper read/write path for SiMT performance.

Latency	**Duality of Read/Write Path (IoU) between De→En and En→De**		
	MMA	Single Path	Dual Path
High	0.4755	0.5328	0.6346
Middle	0.5132	0.5898	0.6962
Low	0.6046	0.7169	0.7466

Table 2: Duality of read/write path (IoU score) between De→En and En→De.

6.3 Effect of Duality Constraints

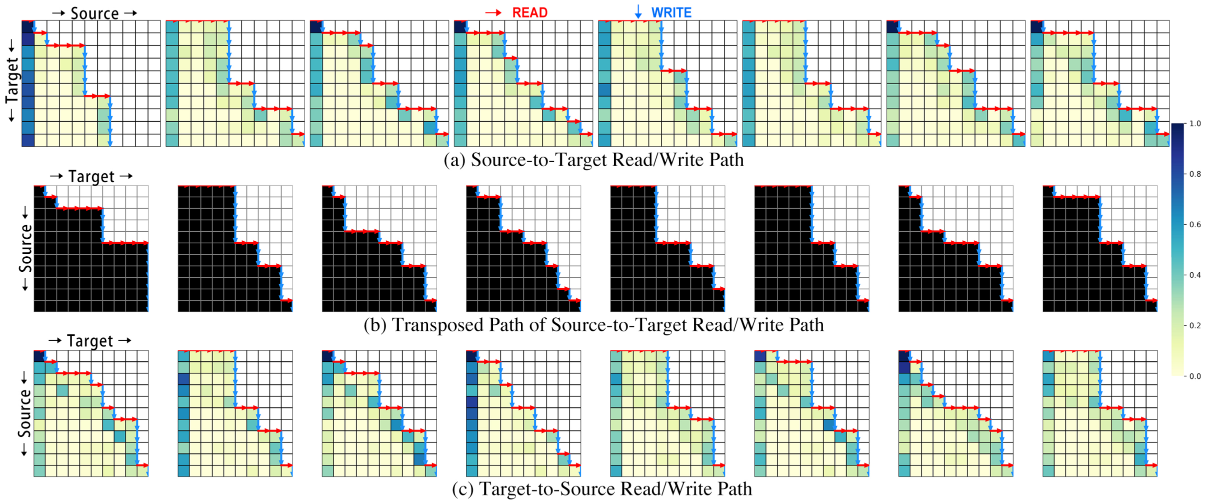
To verify that our method improves the duality of two read/write paths, we conduct duality evaluation between source-to-target and target-to-source read/write paths. Specifically, we first express both the original read/write path on target-to-source and the transposed path of source-to-target read/write path in the form of matrices, and then calculate the *Intersection over Union score* (IoU) between the area below them (see Figure 6), which is regarded as the duality between the read/write path in the two directions. The higher IoU score indicates that the two paths are more consistent on common segment pairs, i.e., stronger duality. Appendix B gives the detailed calculation of IoU score.

The results of duality evaluation are reported in Table 2, where our method effectively enhances the duality of source-to-target and target-to-source read/write paths under all latency levels. This shows that with dual-path SiMT, the read/write paths in source-to-target and target-to-source are more in agreement on the sequence of segment pairs between the sentence pair.

6.4 Dual Read/Write Paths Visualization

Figure 6 shows the read/write path visualization of a De↔En example. In 'Dual Paths', there is a

[9] https://www-i6.informatik.rwth-aachen.de/goldAlignment/



(a) Source-to-Target Read/Write Path

(b) Transposed Path of Source-to-Target Read/Write Path

(c) Target-to-Source Read/Write Path

Figure 6: Read/write path visualization of a De↔En example (De: '*die Lehr@@ er@@ bildung fand in Bam@@ berg statt .*' ↔ En: '*the teacher training course was in Bam@@ berg .*'). (a) and (c) show the read/write path and attention distribution in two single-path SiMT model, where the shade of the color indicates the attention weight. (b) shows the transposed path of the source-to-target read/write path. '→': READ action to wait for a source word, '↓': WRITE action to generate a target word. Note that 8 sub-figures respectively represent 8 read/write paths, assigned to 8 heads and shared between decoder layers, and the attention is averaged on all decoder layers.

Systems	λ^F	λ^B	**De→En**		**En→De**	
			AL	**BLEU**	**AL**	**BLEU**
MMA	0.3	-	6.00	27.29	-	-
Single Path	0.3	-	5.34	26.67	-	-
Dual Paths	0.3	0.2	4.71	27.39	6.43	25.53
	0.3	0.3	3.19	27.04	4.80	25.20
	0.3	0.4	3.00	27.01	3.77	23.62

Table 3: Performance under different settings of latency weight, where λ^F and λ^B are the latency weight of the forward and backward single-path SiMT model respectively.

strong duality between the read/write paths in two translation directions, where the target-to-source read/write path (Figure 6(c)) and the transposed path of the source-to-target read/write path (Figure 6(b)) have a high degree of overlap. In particular, the read/write paths in our method exhibit a clear division on segment pairs.

6.5 Analysis on Forward/Backward Latency

To analyze the relationship between the forward and backward single-path SiMT model in terms of the latency setting, we set the latency weight (λ in Eq.(5)) of the forward and backward single-path SiMT model to different values, denoted as λ^F and λ^B respectively (the greater the latency weight, the lower the model latency). Table 3 reports the effect of different settings of λ^B on the performance of the forward single-path model.

After applying backward model and the duality constraints, our method has a much lower latency and similar translation quality compared with 'MMA' and 'Single Path'. As the latency of the backward model decreases (λ^B becomes larger), the latency of the forward model also gradually decreases, which shows that the latency of the forward and backward models are strongly correlated. Overall, regardless of the setting of λ^F and λ^B, 'Dual Paths' obtains a better trade-off between latency and translation quality. Furthermore, we can get a slightly larger or smaller latency by adjusting the combination of λ^F and λ^B.

7 Conclusion

In this paper, we develop the dual-path SiMT to supervise the read/write path by modeling the duality constraints between SiMT in two directions. Experiments and analyses we conducted show that our method outperforms strong baselines under all latency and achieves a high-quality read/write path.

Acknowledgements

We thank all the anonymous reviewers for their insightful and valuable comments. This work was supported by National Key R&D Program of China (NO. 2017YFE0192900).

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.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2019. An effective approach to unsupervised machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 194–203, 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.

Srinivas Bangalore, Vivek Kumar Rangarajan Sridhar, Prakash Kolan, Ladan Golipour, and Aura Jimenez. 2012. Real-time incremental speech-to-speech translation of dialogs. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 437–445, Montréal, Canada. Association for Computational Linguistics.

Mauro Cettolo, Niehues Jan, Stüker Sebastian, Luisa Bentivogli, R. Cattoni, and Marcello Federico. 2015. The iwslt 2015 evaluation campaign.

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

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, and Stephan Vogel. 2018. Incremental decoding and training methods for simultaneous translation in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 493–499, New Orleans, Louisiana. Association for Computational Linguistics.

Maha Elbayad, Laurent Besacier, and Jakob Verbeek. 2020. Efficient Wait-k Models for Simultaneous Machine Translation.

Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor O.K. Li. 2017. Learning to translate in real-time with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1053–1062, Valencia, Spain. 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.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Dan Liu, Mengge Du, Xiaoxi Li, Ya Li, and Enhong Chen. 2021. Cross attention augmented transducer networks for simultaneous translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 39–55, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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*.

Yishu Miao, Phil Blunsom, and Lucia Specia. 2021. A generative framework for simultaneous machine translation. In *Proceedings of the 2021 Conference* *on Empirical Methods in Natural Language Processing*, pages 6697–6706, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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 the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

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 of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Colin Raffel, Minh-Thang Luong, Peter J. Liu, Ron J. Weiss, and Douglas Eck. 2017. Online and linear-time attention by enforcing monotonic alignments. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2837–2846. PMLR.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Lierni Sestorain, Massimiliano Ciaramita, Christian Buck, and Thomas Hofmann. 2019. Zero-shot dual machine translation.

Maryam Siahbani, Hassan Shavarani, Ashkan Alinejad, and Anoop Sarkar. 2018. Simultaneous translation using optimized segmentation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 154–167, Boston, MA. Association for Machine Translation in the Americas.

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 I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Yijun Wang, Yingce Xia, Li Zhao, Jiang Bian, Tao Qin, Guiquan Liu, and Tie-Yan Liu. 2018. Dual transfer learning for neural machine translation with marginal distribution regularization. In *AAAI*, pages 5553–5560.

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.

Yingce Xia, Tao Qin, Wei Chen, Jiang Bian, Nenghai Yu, and Tie-Yan Liu. 2017. Dual supervised learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3789–3798. PMLR.

Mohd Abbas Zaidi, Sathish Indurthi, Beomseok Lee, Nikhil Kumar Lakumarapu, and Sangha Kim. 2021. Infusing future information into monotonic attention through language models. *arXiv preprint arXiv:2109.03121*.

Ruiqing Zhang, Chuanqiang Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. 2020. Learning adaptive segmentation policy for simultaneous translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2280–2289, 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, Yang Feng, and Liangyou Li. 2021. Future-guided incremental transformer for simultaneous translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14428–14436, Online.

Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. 2019a. 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.

Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. 2019b. Simultaneous translation with flexible policy via restricted imitation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5816–5822, Florence, Italy. Association for Computational Linguistics.

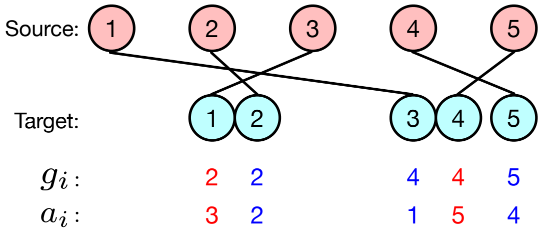


Figure 7: Schematic diagram of evaluating the read/write path in terms of sufficiency and necessity. The black line indicates the ground-truth alignments between the target and source. g_i is the number of source words read in when generating the i^{th} target word. a_i is the ground-truth aligned source position of the i^{th} target word. $a_i > g_i$ (numbers colored in red) means that the i^{th} target word is forced to be translated in advance before reading its aligned source word.

A Evaluation Metrics of Read/Write Path

In Sec.6.2, we propose two metrics A^{Suf} and A^{Nec} to measure the *sufficiency* and *necessity* of the read/write path using alignments. Here, we give a more detailed calculation of them.

Given the ground-truth alignments, we denote the aligned source position of the i^{th} target word as a_i. Specifically, for one-to-many alignment from target to source, we choose the furthest source word as it aligned source position. For a read/write path, we denote the number of source words read in when generating the i^{th} target word as g_i. Figure 7 gives an example of the calculation of a_i and g_i.

Sufficiency A^{Suf} measures how many aligned source words are read before translating the target word (i.e., $a_i \leq g_i$), which ensures the faithfulness of the translation, calculated as

$$A^{Suf} = \frac{1}{I}\sum_{i=1}^{I} \mathbb{1}_{a_i \leq g_i}, \tag{15}$$

where $\mathbb{1}_{a_i \leq g_i}$ counts the number that $a_i \leq g_i$. Taking the case in Figure 7 as an example, the sufficiency is calculated as $A^{Suf} = \frac{1}{5} \times (0+1+1+0+1) = \frac{3}{5}$, where the 1^{st} and 4^{th} target word are translated before read their aligned source word ($a_i > g_i$).

Necessity A^{Nec} measures how far the output position g_i is from the aligned position a_i, where the closer output position to the alignment position indicates that the read/write path outputs earlier, and there is less unnecessary latency. A^{Nec} is calculated as

$$A^{Nec} = \frac{1}{|a_i \leq g_i|}\sum_{i, a_i \leq g_i} \frac{a_i}{g_i}, \tag{16}$$

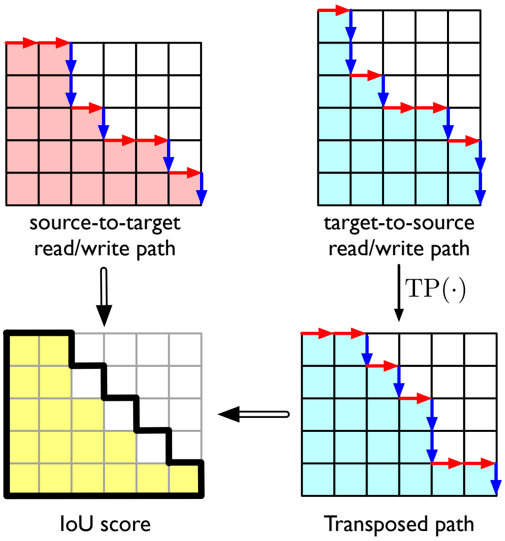


Figure 8: Schematic diagram of calculating the Intersection over Union score (IoU) to evaluate the dual degree of source-to-target and target-to-source read/write path. The yellow area represents the union of the areas below two paths, and the area enclosed by the black line represents the intersection.

Note that A^{Nec} only focuses on aligned positions that are read before output position (i.e., $a_i \leq g_i$). In the case shown in Figure 7, the necessity is calculated as $A^{Nec} = \frac{1}{3} \times \left(\frac{2}{2}+\frac{1}{4}+\frac{4}{5}\right) = \frac{41}{60}$, where we only consider the the 2^{th}, 3^{rd} and 5^{th} target word.

B IoU Score for Duality Evaluation

To verify that our proposed method does make the read/write path of source-to-target and target-to-source more dual, we calculate the *Intersection over Union score* (IoU) to evaluate the duality in Sec.6.3. Following, we describe the detailed calculation of IoU score.

Figure 8 gives an example of calculating the IoU score. Given the source-to-target and target-to-source read/write path $\mathbf{P}^{s2t}$ and $\mathbf{P}^{t2s}$ in the binary matrix form, we first generate the transposed path $\mathbf{TP}^{s2t}$ of $\mathbf{P}^{t2s}$ with proposed method of transposing the read/write path in Sec.3.1. Then, we calculate the intersection over union score between binary matrices $\mathbf{P}^{s2t}$ and $\mathbf{TP}^{s2t}$:

$$\mathrm{IoU} = \frac{\mathrm{Sum}\left(\mathbf{P}^{s2t} \cap \mathbf{TP}^{s2t}\right)}{\mathrm{Sum}\left(\mathbf{P}^{s2t} \cup \mathbf{TP}^{s2t}\right)}, \tag{17}$$

where the larger IoU score means that the source-to-target and target-to-source read/write path are much more dual. Ideally, the best case is $\mathrm{IoU} = 1$,

Hyperparameter	IWSLT15 En↔Vi	WMT15 De↔En
encoder layers	6	6
encoder attention heads	4	8
encoder embed dim	512	512
encoder ffn embed dim	1024	1024
decoder layers	6	6
decoder attention heads	4	8
decoder embed dim	512	512
decoder ffn embed dim	1024	1024
dropout	0.3	0.3
optimizer	adam	adam
adam-β	(0.9, 0.98)	(0.9, 0.98)
clip-norm	0	0
lr	5e-4	5e-4
lr scheduler	inverse sqrt	inverse sqrt
warmup-updates	4000	4000
warmup-init-lr	1e-7	1e-7
weight decay	0.0001	0.0001
label-smoothing	0.1	0.1
max tokens	16000	2400×4×8

Table 4: Hyperparameters of our experiments.

which means the source-to-target and target-to-source read/write path are exactly in the dual form and reach the agreement on the sequence of segment pairs.

In the calculation of IoU score, for 'MMA' and 'Single Path', the source-to-target and target-to-source read/write paths come from independent models in the two directions respectively. For 'Dual Paths', the source-to-target and target-to-source read/write paths come from the forward and backward single-path SiMT model concurrently.

C Hyperparameters

All systems in our experiments use the same hyperparameters, as shown in Table 4.

D Numerical Results with More Metrics

We also compare 'Dual Paths' and 'Single Path' with previous methods on the latency metrics Average Proportion (AP) (Cho and Esipova, 2016) and Differentiable Average Lagging (DAL) (Arivazhagan et al., 2019). In this section, we first give the definition of AP and DAL, and then report the expanded results and numerical results of the main experiment (Sec.5.3), using AP, AL, DAL as latency metrics.

D.1 Latency Metrics

Average Proportion (AP) (Cho and Esipova, 2016) measures the proportion of the area above a read/write path. Given the read/write path g_i, AP is calculated as

$$\mathrm{AP} = \frac{1}{|\mathbf{x}|\,|\mathbf{y}|}\sum_{i=1}^{|\mathbf{y}|} g_i. \tag{18}$$

Differentiable Average Lagging (DAL) (Arivazhagan et al., 2019) is a differentiable version of average lagging, which can be integrated into training. Given the read/write path g_i, DAL is calculated as

$$g_i^{'} = \begin{cases} g_i & i = 1 \\ \max\left(g_i, g_{i-1}^{'} + \frac{|\mathbf{x}|}{|\mathbf{y}|}\right) & i > 1 \end{cases}, \tag{19}$$

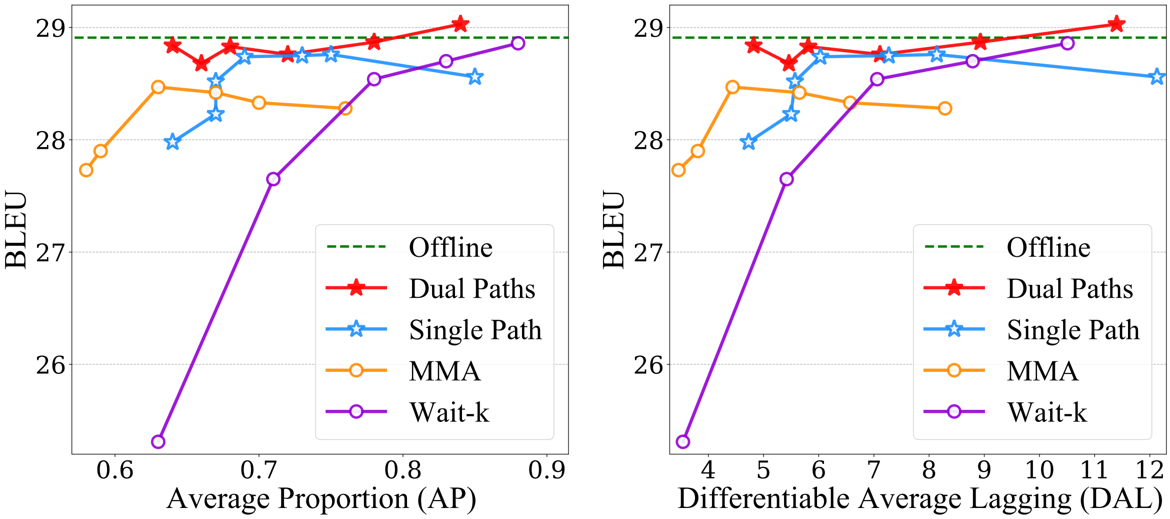
$$\mathrm{DAL} = \frac{1}{|\mathbf{y}|}\sum_{i=1}^{|\mathbf{y}|} g_i^{'} - \frac{i-1}{|\mathbf{x}|\,/\,|\mathbf{y}|}. \tag{20}$$

D.2 Expand Results

Figure 9, 10, 11, 12 respectively show the expanded results on IWSLT15 En↔Vi and WMT15 De→En, measured by AP and DAL.

D.3 Numerical Results

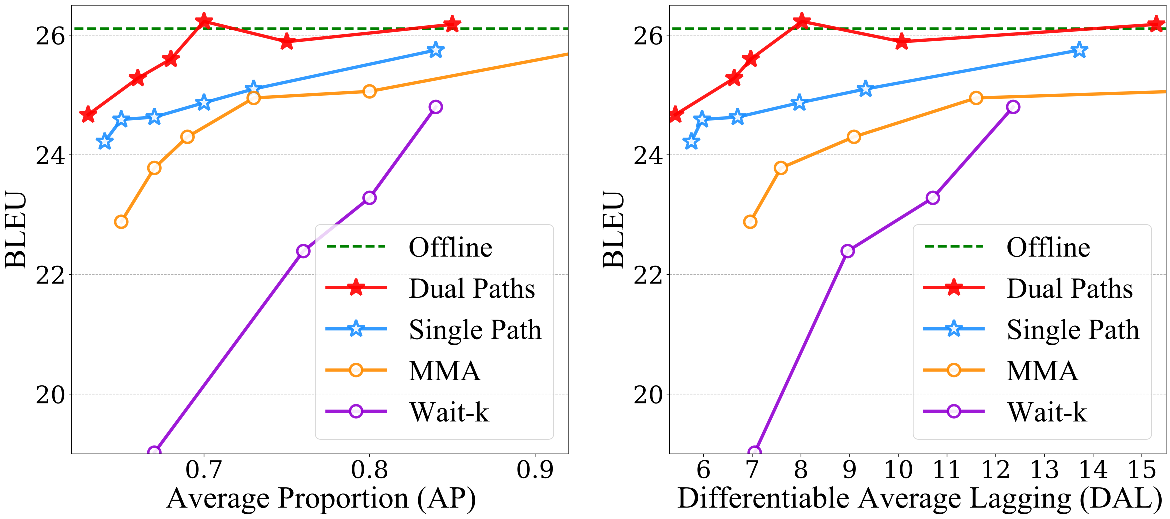
Table 5, 6, 7, 8 respectively report the numerical results on IWSLT15 En↔Vi and WMT15 De→En, measured by AP, AL, DAL, and BLEU.



(a) IWSLT15 En→Vi, AP

(b) IWSLT15 En→Vi, DAL

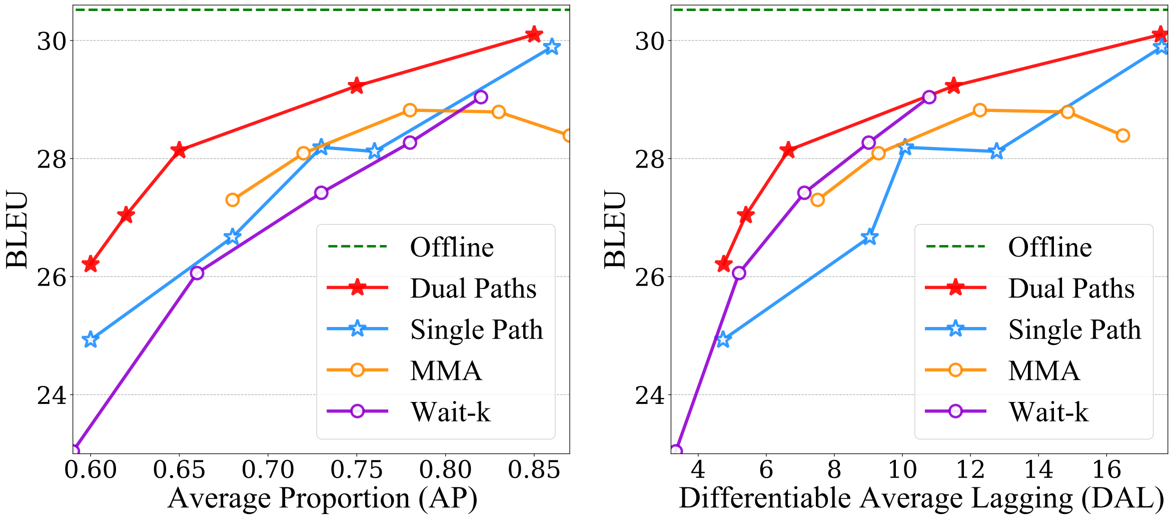
Figure 9: Results on IWSLT15 En→Vi, measured with AP and DAL.



(a) IWSLT15 Vi→En, AP

(b) IWSLT15 Vi→En, DAL

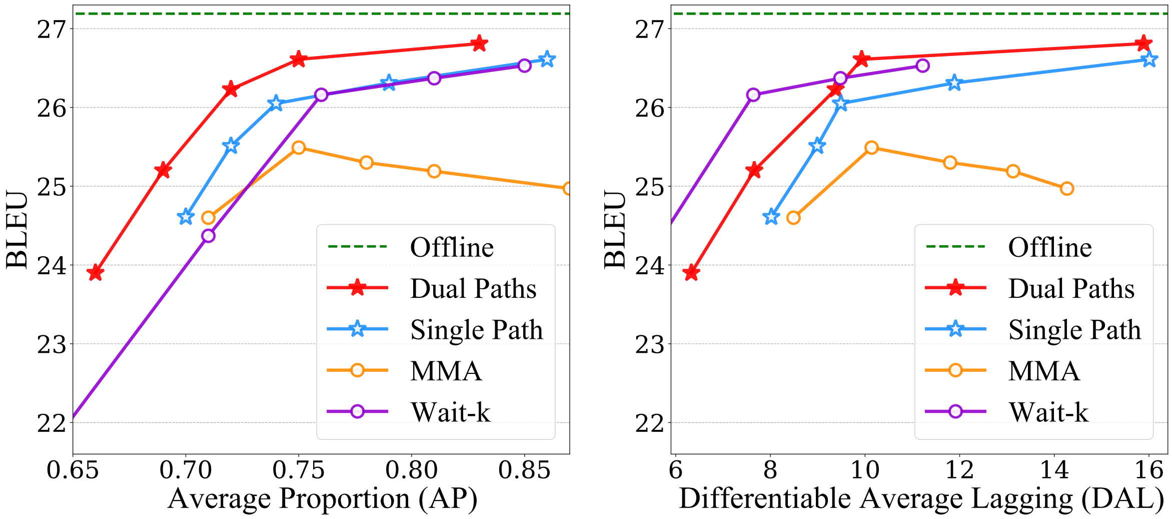
Figure 10: Results on IWSLT15 Vi→En, measured with AP and DAL.



(a) WMT15 De→En, AP

(b) WMT15 De→En, DAL

Figure 11: Results on WMT15 De→En, measured with AP and DAL.



(a) WMT15 En→De, AP

(b) WMT15 En→De, DAL

Figure 12: Results on WMT15 En→De, measured with AP and DAL.

IWSLT15 En→Vi				
Offline				
	AP	AL	DAL	BLEU
	1.00	22.08	22.08	28.91
Wait-k				
k	AP	AL	DAL	BLEU
1	0.63	3.03	3.54	25.31
3	0.71	4.80	5.42	27.65
5	0.78	6.46	7.06	28.54
7	0.83	8.21	8.79	28.70
9	0.88	9.92	10.51	28.86
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
Single Path				
λ	AP	AL	DAL	BLEU
0.5	0.64	3.02	4.73	27.98
0.4	0.67	3.54	5.50	28.23
0.3	0.67	3.83	5.57	28.52
0.2	0.69	4.05	6.03	28.74
0.1	0.73	5.08	7.27	28.75
0.05	0.75	5.38	8.14	28.76
0.01	0.85	8.72	12.13	28.56
Dual Paths				
λ	AP	AL	DAL	BLEU
0.4	0.64	3.07	4.82	28.84
0.3	0.66	3.49	5.46	28.68
0.2	0.68	3.84	5.81	28.83
0.1	0.72	4.78	7.11	28.76
0.05	0.78	6.14	8.93	28.87
0.01	0.84	8.15	11.40	29.03

Table 5: Numerical results of IWSLT15 En→Vi.

IWSLT15 Vi→En				
Offline				
	AP	AL	DAL	BLEU
	1.00	27.56	27.56	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
MMA				
λ	AP	AL	DAL	BLEU
0.4	0.65	4.26	6.96	22.08
0.3	0.67	4.56	7.59	22.98
0.2	0.69	5.03	9.09	23.50
0.1	0.73	5.70	11.60	24.15
0.05	0.80	7.51	15.70	24.26
0.01	0.95	15.55	23.95	25.04
Single Path				
λ	AP	AL	DAL	BLEU
0.4	0.64	3.87	5.75	24.22
0.3	0.65	4.07	5.97	24.59
0.2	0.67	4.55	6.70	24.63
0.1	0.70	5.48	7.97	24.87
0.05	0.73	6.33	9.33	25.10
0.01	0.84	10.24	13.72	25.75
Dual Paths				
λ	AP	AL	DAL	BLEU
0.4	0.63	3.60	5.42	24.67
0.3	0.66	4.52	6.63	25.28
0.2	0.68	4.89	6.97	25.60
0.1	0.70	5.54	8.02	26.23
0.05	0.75	6.95	10.07	25.89
0.01	0.85	11.30	15.30	26.18

Table 6: Numerical results of IWSLT15 Vi→En.

WMT15 De→En				
Offline				
	AP	AL	DAL	BLEU
	1.00	27.77	27.77	30.52
Wait-k				
k	AP	AL	DAL	BLEU
1	0.52	0.02	1.84	16.95
3	0.59	1.73	3.34	23.04
5	0.66	3.86	5.20	26.06
7	0.73	5.86	7.12	27.42
9	0.78	7.85	9.01	28.27
11	0.82	9.75	10.79	29.04
MMA				
λ	AP	AL	DAL	BLEU
0.4	0.68	4.97	7.51	27.30
0.3	0.72	6.00	9.30	28.09
0.25	0.78	8.03	12.28	28.82
0.2	0.83	9.98	14.86	28.79
0.1	0.87	13.25	16.48	28.39
Single Path				
λ	AP	AL	DAL	BLEU
0.4	0.60	2.73	4.73	24.93
0.3	0.68	5.34	9.04	26.67
0.25	0.73	6.66	10.08	28.19
0.2	0.76	8.31	12.77	28.12
0.1	0.86	13.93	17.62	29.89
Dual Paths				
λ	AP	AL	DAL	BLEU
0.4	0.60	2.80	4.75	26.21
0.3	0.62	3.19	5.40	27.04
0.25	0.65	4.02	6.65	28.14
0.2	0.75	7.69	11.51	29.23
0.1	0.85	13.50	17.59	30.10

Table 7: Numerical results of WMT15 De→En.

WMT15 En→De				
Offline				
	AP	AL	DAL	BLEU
	1.00	26.56	26.56	27.19
Wait-k				
k	AP	AL	DAL	BLEU
1	0.56	1.52	2.38	16.72
3	0.64	3.46	3.97	21.69
5	0.71	5.25	5.72	24.37
7	0.76	7.14	7.64	26.16
9	0.81	8.96	9.48	26.37
11	0.85	10.76	11.22	26.53
MMA				
λ	AP	AL	DAL	BLEU
0.4	0.71	5.54	8.49	24.60
0.3	0.75	6.69	10.14	25.49
0.25	0.78	7.40	11.80	25.30
0.2	0.81	8.64	13.13	25.19
0.1	0.87	11.12	14.27	24.97
Single Path				
λ	AP	AL	DAL	BLEU
0.4	0.70	5.19	8.02	24.61
0.3	0.72	5.73	8.99	25.51
0.25	0.74	6.39	9.49	26.05
0.2	0.79	8.11	11.89	26.31
0.1	0.86	11.93	16.01	26.61
Dual Paths				
λ	AP	AL	DAL	BLEU
0.4	0.66	4.24	6.33	23.90
0.3	0.69	4.80	7.66	25.20
0.25	0.72	5.95	9.38	26.23
0.2	0.75	6.42	9.93	26.61
0.1	0.83	11.80	15.89	26.81

Table 8: Numerical results of WMT15 En→De.