Self-training Reduces Flicker in Retranslation-based Simultaneous Translation

Sukanta Sen[1*] **Rico Sennrich**[2,1] **Biao Zhang**[1] **Barry Haddow**[1]

[1]School of Informatics, University of Edinburgh
[2]Department of Computational Linguistics, University of Zurich
ssen@ed.ac.uk, sennrich@cl.uzh.ch,
b.zhang@ed.ac.uk, bhaddow@ed.ac.uk

Abstract

In simultaneous translation, the *retranslation* approach has the advantage of requiring no modifications to the inference engine. However, in order to reduce the undesirable flicker in the output, previous work has resorted to increasing the latency through masking, and introducing specialised inference, thus losing the simplicity of the approach. In this work, we show that self-training improves the flicker-latency tradeoff, while maintaining similar translation quality to the original. Our analysis indicates that self-training reduces flicker by controlling monotonicity. Furthermore, self-training can be combined with biased beam search to further improve the flicker-latency tradeoff.

1 Introduction

Simultaneous machine translation systems, which process their input word by word instead of sentence by sentence, must strike a balance between producing output immediately (and so reducing quality because of incomplete input) and waiting for further input (and so increasing latency). Ideally, a good simultaneous translation system will provide a pareto-optimal tradeoff between quality and latency. A straightforward way of doing simultaneous translation is *retranslation* (Niehues et al., 2016), which has the advantage that it can be used with an unmodified machine translation (MT) inference engine, and can perform better than the alternative streaming-based approaches (Arivazhagan et al., 2020b). The disadvantage is that retranslation may change previous output causing *flicker*, leading to a poor user experience, and so flicker needs to be balanced with latency and quality.

We argue that flickering is caused by two different (but related) issues: (i) lexical instability of the translation – the system "changes its mind" as more source is revealed, swapping one word for another[1] and (ii) non-monotonicity of the translation – the system favours a non-monotonic translation, which means it needs high latency in order to avoid flicker. Some of this instability and non-monotonicity is necessary – forced by syntactic differences between source and target, and lack of information in the prefixes – but some is due to arbitrary choices of the model. We aim to reduce these as far as possible.

In non-autoregressive translation (NAT), a related problem, known as the "multimodality" problem (Gu et al., 2018), has been addressed using knowledge distillation (Kim and Rush, 2016, KD). We therefore investigate whether this can also reduce flicker in simultaneous translation. Since the initial model and the distilled model have the same architecture in our work, approximating KD is essentially self-training[2]. We show that a self-trained model is able to achieve the same quality as the initial model, but with improved flicker-latency tradeoff. We also show that self-training (Arivazhagan et al., 2020a) can be combined with biased beam search to further improve the flicker-latency tradeoff. Furthermore, we show experiments that link flicker to monotonicity.

2 Background

2.1 Retranslation

We assume a retranslation approach, where the source is retranslated each time it is updated, and the new output replaces the old. Only the current sentence is retranslated – previous sentences are considered to be fixed. In contrast to streaming approaches (e.g. Ma et al., 2019a; Arivazhagan et al., 2019b), retranslation can use an unmodified inference engine, making it simpler to deploy. The basic retranslation approach can be improved by using *prefix training* (Niehues et al., 2016, 2018), *bi-*

*Work done while at the University of Edinburgh.

[1]An example of this is shown in Appendix C.

[2]Retraining a model on its own output (Clark et al., 2003).

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 3734–3744
May 2-6, 2023

ased beam search and *output masking*[3] (Arivazhagan et al., 2020a).

2.2 Evaluation of Simultaneous Translation

In addition to quality, evaluation of simultaneous translation requires that we consider latency and, if using retranslation, flicker. The quality of the translation can be evaluated by comparing the final output of each sentence with a reference – we will use BLEU (Papineni et al., 2002; Post, 2018), CHRF (Popović, 2015) and COMET (Rei et al., 2020) scores. To measure flicker, we use *normalised erasure* (Arivazhagan et al., 2020a, 2019a), which measures the flicker between consecutive translation outputs by counting the minimum number of tokens that must be deleted from the end of the previous translation in order to produce the next, normalised by output length.

The measurement of latency has been the subject of some debate in the literature, with several different measures proposed (Ma et al., 2019a; Cherry and Foster, 2019; Ansari et al., 2021). In our experiments, we plot the flicker-latency tradeoff by controlling the output mask and recording the effect on flicker. Since mask size correlates with latency, our aim is to improve this mask-flicker tradeoff curve, and so be able to use a shorter mask with the same flicker budget.

2.3 Knowledge Distillation and Self-Training

The idea of sequence-level KD (Kim and Rush, 2016), is to create a smaller *student* model using the predictions of the larger *teacher* model. This has found application in MT efficiency (Junczys-Dowmunt et al., 2018) and in non-autoregressive translation (Zhou et al., 2020). In our work, the student model has the same size as the teacher, and is *self-trained* on teacher output. The output distributions of the student model have lower entropy (Zhou et al., 2020), so the model is less likely to swap between translation hypotheses unnecessarily as the source prefix is extended. Also, since the student model is trained on MT output, where the target order tends to be more similar to the source order (Zhou et al., 2020), it is more likely to avoid unnecessary reorderings, generating a more monotonic translation, which can be built up incrementally. We give experimental evidence for these in the next section.

Chen et al. (2021) also proposed to use pseudo-reference sentences obtained through forward translation of the source sentences to improve simultaneous translation. Unlike our work, they considered a streaming approach (specifically wait-k (Ma et al., 2019b)) where the system can only append to the output; it does not flicker like retranslation. They showed that their approach could improve the quality-latency tradeoff of wait-k using their distillation approach, but to create the training data for the student system they used wait-k and filtering. We avoid these complications by just using the baseline system as the teacher.

3 Experiments

3.1 Data

We test our self-training approach on English↔{German,Czech}. For En↔De we use IWSLT21 (Anastasopoulos et al., 2021) for training, and the concatenation of the 2014 and 2015 test sets for development (early stopping), removing any sentences that overlap with the training set. For En↔Cs, we use the training and validation set from WMT21 (Akhbardeh et al., 2021). Training data sizes are shown in Appendix A. We use *prefix training* (Niehues et al., 2018) to reduce the mismatch between sentence-level training and prefix-based inference at test time. For each parallel sentence pair in the training set, we generate a corresponding prefix pair by truncating a randomly chosen proportion. We treat the validation sets similarly.

We test our systems both on IWSLT test data (derived from TED talks) and on the ESIC test set[4] (Macháek et al., 2021). From IWSLT, we use tst2018 for De↔En, and tst2015/tst2016 combined for Cs↔En. ESIC is derived from the European parliament proceedings, and consists of transcribed speeches in English, together with their simultaneous interpretation into Czech and German (also transcribed). ESIC is aligned at the document level, but not at the sentence level. We use the test portion for evaluation, only for En→X. It has been argued that simultaneous translation is better evaluated (and trained, if possible) on interpreted data (Zhao et al., 2021). However such data is hard to come by, and ESIC is the only such resource for European languages. We remove any segments from the IWSLT test sets that overlap

[3]This means that the last k words are omitted from the output before being passed to the user. This reduces flicker, but increases latency.

[4]https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-3719

with training, and also remove from the training data any Europarl documents with overlap with ESIC.

All data is pre-processed with SentencePiece unigram model (Kudo and Richardson, 2018) with a shared subword (Sennrich et al., 2016b) vocabulary size of 32k.

		En→De		De→En	En→Cs		Cs→En
Metric	Model	ESIC	IWSLT	IWSLT	ESIC	IWSLT	IWSLT
BLEU	T	17.5	27.7	33.4	14.4	24.6	31.3
	S	17.6	27.5	31.7	14.5	25.0	31.3
ChrF	T	58.9	56.9	59.2	51.5	51.5	56.1
	S	58.8	57.2	58.3	51.7	51.7	56.2
COMET	T	.553	.330	.488	.651	.639	.519
	S	.532	.326	.468	.672	.642	.521

Table 1: Comparison between teacher (T) and student (S) models on ESIC and IWSLT test sets. For ESIC, BLEU and CHRF are calculated at document level, i.e. considering each document as a segment. For COMET we use reference-less `wmt20-comet-da` for ESIC and reference-based `wmt20-comet-da` for IWSLT.

3.2 Teacher-Student Training

Our teacher model, which serves as a baseline, is a transformer base (Vaswani et al., 2017) trained[5] with fairseq[6] (Ott et al., 2019).

We use the teacher to translate the training data, using a beam size[7] of 8, then train a student model with the same architecture on this synthetic data.

In Table 1 we show the performance of our baseline system (equivalent to the teacher) and the student system on 6 test sets. Overall, student performance is robust compared to teacher, with same or better scores in Cs↔En and some small losses in De↔En.

To assess whether the student models reduce flicker in retranslation, we use each model in a simulated SLT pipeline and plot flicker-latency tradeoff curves. That is, we use the systems to translate ever-growing prefixes of the source sentences in the testsets, using SLTev (Ansari et al., 2021) to measure the flicker, and varying the output mask to show the tradeoff. A curve for one test set is shown in Figure 1, with full results in Appendix D.

[5]For training hyperparameters, see Appendix B.

[6]To generate training data for the students, we actually used a marian (Junczys-Dowmunt et al., 2018) model, with 60×10^6 parameters, trained on the same data and with the same architecture, which achieves nearly identical BLEU. This was to take advantage of marian's fast inference. All results shown in the paper are with the fairseq models.

[7]We also tested sequence-level interpolation, selecting the highest-scoring translation in an 8-best list according to BLEU and CHRF, but results were very similar.

We can see that in all configurations the student models improve the flicker-latency tradeoff. In section 3.4, we show how the student training data is more monotonic, and the models have lower entropy, echoing Zhou et al. (2020).

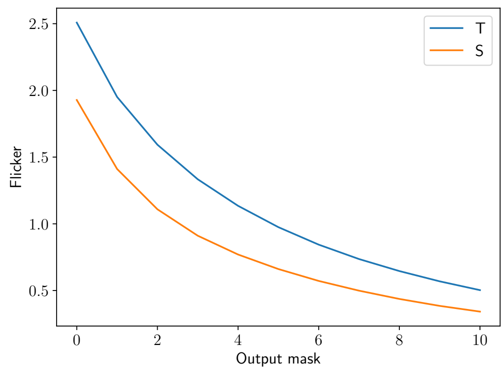


Figure 1: Flicker-latency tradeoff for the teacher (T) and student (S) models, En→De IWSLT. We control latency by varying the output mask.

3.3 Controlling Monotonicity

To show that self-training affects flicker through increased monotonicity, we experiment with controlling the monotonicity of the student training data. We stratify the teacher data into 5 different monotonicity levels using Kendall's Tau on a *fast_align* (Dyer et al., 2013) target–source alignment to measure monotonicity, with an equal stratum size. We add the monotonicity level as pseudo-word, as in Sennrich et al. (2016a), to each source sentence, and train a teacher model on this monotonicity-aware corpus. We then use this teacher to create 5 different student training corpora, using the monotonicity control, and train 5 different students on these corpora.

Table 2 shows the BLEU[8] scores for the monotonicity-controlled models, as well as the teacher and student from the previous section. Using highly monotonic (Mono-1) or non-monotonic (Mono-5) data gives poor quality, but the in-between strata are similar, with Mono-3 slightly better overall. Figure 2 shows a distinctly worse flicker-latency tradeoff for Mono-5, whereas Mono-4 is a bit better than the teacher, and all other students are better. This supports the hypothesized connection between the higher degree of monotonicity in the student training data, and the

[8]Scores for CHRF and COMET are in the Appendix F, but the pattern is similar.

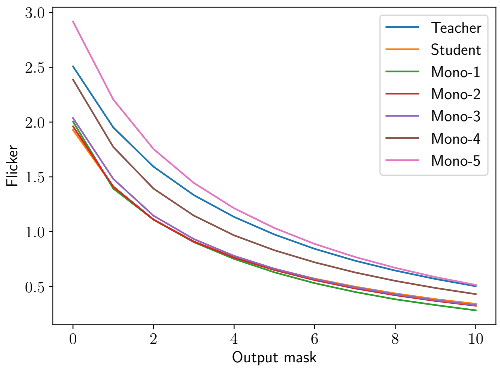


Figure 2: Latency–flicker tradeoff for the En→De IWSLT monotonicity-controlled models. Monotonicity control ranges from 1 (training data created with maximum monotonicity) to 5 (minimum monotonicity).

better flicker-latency tradeoff in the student models. We show the flicker-latency tadeoff curves on more test sets and language pairs in Figure 5 in the Appendix F but the pattern is similar.

		En→De		En→Cs	
Metric	Model	ESIC	IWSLT	ESIC	IWSLT
BLEU	Teacher	17.5	27.7	14.4	24.6
	Student	**17.6**	27.5	14.5	25.0
	Mono-1	8.6	14.4	**14.7**	23.6
	Mono-2	**17.6**	27.4	14.5	25.0
	Mono-3	17.5	**27.9**	14.5	**25.7**
	Mono-4	17.2	26.6	13.8	24.7
	Mono-5	16.0	25.0	12.5	23.0

Table 2: Student models with monotonicity control. Monotonicity ranges from 1 (highest) to 5 (lowest). The best scores are in bold font.

3.4 Monotonicity and Entropy of Student Models

We claimed that student models have lower flicker because they produce more monotonic translations, with less unnecessary variation. Here we provide evidence to support those claims.

Training data for student models is more monotonic In order to calculate the monotonicity of the training data, we use Kendall's tau score. We first extract word alignments from the training data using *fast_align* (Dyer et al., 2013) to forward-align source and target. For each sentence pair we express the alignment as a function $a : i \to j$, and construct the two lists $1, \ldots, T$ and $a(1), \ldots, a(T)$ where T is the target length. We then calculate the Kendall's tau between the two lists, repeat for each sentence pair in the corpus, and average. We repeat the calculation for the original training data and for the student training set. The results are shown in Table 3. We can see that in all cases, the student training data is more monotonic than the original teacher training data.

Model	En→De	De→En	En→Cs	Cs→En
Teacher	0.793	0.788	0.849	0.836
Student	0.857	0.801	0.906	0.880

Table 3: Kendall's tau scores. Higher scores indicate more monotonicity.

Student models have lower entropy distributions For each of our models, we calculate the mean per-token entropy, by considering the probability distribution over the vocabulary at each time step. The entropies are shown in Table 4.

		Entropy	
Pair	Test set	Teacher	Student
En→De	ESIC	0.371	0.220
	IWSLT	0.295	0.228
De→En	IWSLT	0.273	0.160
En→Cs	ESIC	0.443	0.251
	IWSLT	0.417	0.238
Cs→En	IWSLT	0.335	0.213

Table 4: Mean per-token entropies for each language pair test set combination.

We can see from Table 4 that the token entropies are consistently lower for student models, suggesting that the distributions are more "peaky", and so less likely to flicker between multiple output tokens with similar probabilities.

3.5 Self-training and Biased Beam Search

We investigate the combination of our self-training approach with biased beam search (Arivazhagan et al., 2020a). The idea of biased beam search (or "prefix biasing") is to reduce flicker in retranslation by modifying inference so that the translation of the current prefix is "biased" towards the translation of the last prefix. The model for inference has an extra term which penalises it for departing from the previous translation. As the current translation is being generated, once the hypothesis departs from the previous translation, we stop applying the bias penalty, reverting to the unmodified MT model.

Before the previous translation is used for biasing, it is normally *masked*; i.e., the right-most k tokens are removed. Without applying this mask, biased beam search seriously reduces quality by forcing inference to follow poor-quality early decisions. This *bias mask* is different from the output mask used in earlier experiments (which controls latency) although in previous work the bias and output mask are typically set to the same value.

We implemented biased beam search in fairseq and, based on previous work, we set the bias strength $\beta = 0.25$. After comparing different bias masks (Appendix E) we set the mask to 6 for ESIC and 10 for IWSLT.

We sweep across output masks to generate latency–flicker tradeoff curves in Figure 3 (with full results in Appendix E). We compare teacher and student models, with and without biased beam search. We can see from the graphs that biased beam search is effective in improving the latency–flicker tradeoff, but that the student models still improve over the teacher with biased beam search. The disadvantages of biased beam search are that it requires careful tuning of the prefix mask in order to avoid damaging quality, and that it requires a modified inference engine. The inference engine requires access to the previous translation, creating challenges for scalability. In contrast, our self-training approach requires no modifications to inference. Furthermore, since biased beam search relies on aligning the current translation with the previous one, it is hard to apply when the translation cannot be aligned – for example in a cascaded system where the ASR can rewrite its output.

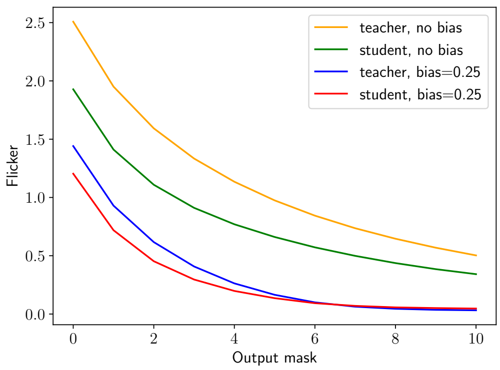


Figure 3: Latency-flicker tradeoff for teacher-student models with and without biased beam search for the En→De IWSLT.

4 Conclusion

We show that self-training reduces the flicker in retranslation-based simultaneous translation, whilst retaining quality. Our experiments link this flicker reduction to increased monotonicity and reduced entropy of the self-trained model. Although biased beam search can obtain larger reductions in flicker, it requires more careful parameter tuning, and a modified inference engine.

5 Limitations

Language Pairs We conducted our experiments using two European language pairs where source and target are linguistically similar. We show that we are able to reduce word order divergence between source and target text through forward translation which helps in reducing the flicker. However, a more challenging case will be using languages from different linguistic families with radically different word orders (such as English–Japanese) which may limit to which extent we are able to reduce the word order divergence between source and target through synthetic data creation.

Evaluation Whilst we show quality evaluation across three different metrics, we were not able to add human evaluation due to resource and space constraints. An additional consideration for simultaneous ST is that it is not clear what the combined effect of flicker, latency and quality is on human perception, and there has been limited work on this (Javorský et al., 2022).

Acknowledgements

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement 825460 (ELITR), and from UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee [grant number 10039436 UTTER]. RS acknowledges funding from the Swiss National Science Foundation (project MUTAMUR; no. 176727).

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–93, Online. Association for Computational Linguistics.

Antonios Anastasopoulos, Ondřej Bojar, Jacob Bremerman, Roldano Cattoni, Maha Elbayad, Marcello Federico, Xutai Ma, Satoshi Nakamura, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Sebastian Stüker, Katsuhito Sudoh, Marco Turchi, Alexander Waibel, Changhan Wang, and Matthew Wiesner. 2021. Findings of the IWSLT 2021 Evaluation Campaign. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 1–29, Bangkok, Thailand (online). Association for Computational Linguistics.

Ebrahim Ansari, Ondřej Bojar, Barry Haddow, and Mohammad Mahmoudi. 2021. SLTEV: Comprehensive evaluation of spoken language translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 71–79, Online. Association for Computational Linguistics.

Naveen Arivazhagan, Colin Cherry, Te I, Wolfgang Macherey, Pallavi Baljekar, and George Foster. 2019a. Re-Translation Strategies For Long Form, Simultaneous, Spoken Language Translation. *arXiv e-prints*, page arXiv:1912.03393.

Naveen Arivazhagan, Colin Cherry, Te I, Wolfgang Macherey, Pallavi Baljekar, and George Foster. 2020a. Re-Translation Strategies For Long Form, Simultaneous, Spoken Language Translation. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, Chung-Cheng Chiu, Semih Yavuz, Ruoming Pang, Wei Li, and Colin Raffel. 2019b. 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.

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, and George Foster. 2020b. Re-translation versus streaming for simultaneous translation. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 220–227, Online. Association for Computational Linguistics.

Junkun Chen, Renjie Zheng, Atsuhito Kita, Mingbo Ma, and Liang Huang. 2021. Improving simultaneous translation by incorporating pseudo-references with fewer reorderings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5857–5864, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Colin Cherry and George Foster. 2019. Thinking Slow about Latency Evaluation for Simultaneous Machine Translation. *arXiv e-prints*, page arXiv:1906.00048.

Stephen Clark, James Curran, and Miles Osborne. 2003. Bootstrapping POS-taggers using unlabelled data. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 49–55.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2018. Non-Autoregressive Neural Machine Translation. In *Proceedings of ICLR*.

Dávid Javorský, Dominik Macháek, and Ondej Bojar. 2022. Comprehension of subtitles from re-translating simultaneous speech translation.

Marcin Junczys-Dowmunt, Kenneth Heafield, Hieu Hoang, Roman Grundkiewicz, and Anthony Aue. 2018. Marian: Cost-effective High-Quality Neural Machine Translation in C++. In *Proceedings of WNMT*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-Level Knowledge Distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327. Association for Computational Linguistics. Event-place: Austin, Texas.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. 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. 2019a. 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.

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. 2019b. 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.

Dominik Macháek, Matú ilinec, and Ondej Bojar. 2021. Lost in Interpreting: Speech Translation from Source or Interpreter? In *Proceedings of Interspeech*.

Jan Niehues, Thai Son Nguyen, Eunah Cho, Thanh-Le Ha, Kevin Kilgour, Markus Müller, Matthias Sperber, Sebastian Stüker, and Alex Waibel. 2016. Dynamic Transcription for Low-latency Speech Translation. In *Proceedings of Interspeech*.

Jan Niehues, Ngoc-Quan Pham, Thanh-Le Ha, Matthias Sperber, and Alex Waibel. 2018. Low-latency neural speech translation. In *Proceedings of Interspeech*.

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*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics. Type: Conference proceedings (article).

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40, San Diego, California. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. 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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *CoRR*, abs/1706.03762.

Jinming Zhao, Philip Arthur, Gholamreza Haffari, Trevor Cohn, and Ehsan Shareghi. 2021. It is not as good as you think! evaluating simultaneous machine translation on interpretation data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6707–6715, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chunting Zhou, Jiatao Gu, and Graham Neubig. 2020. Understanding knowledge distillation in non-autoregressive machine translation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

A Training Data

Corpus	Sentence pairs
English-German	
Europarl	1.79 M
Rapid	1.45 M
News Commentary	0.35 M
OpenSubtitle	22.51 M
TED corpus	206 K
MuST-C.v2	248 K
English-Czech	
Europarl	645 K
ParaCrawl	14 M
CommonCrawl	161 K
News Commentary	260 K
CzEng2.0	36 M[9]
Wikititles	410 K
Rapid	452 K

B Training Parameters

The non-default hyperparameters for Fairseq are shown in Table 5.

C Example of Flicker

An example of a translation which flickers between two similar possibilities is shown in Table 6.

Param	Value
label-smoothing	0.1
criterion	label_smoothed_cross_entropy
patience	10
arch	transformer
optimizer	adam
adam-betas	0.9, 0.98
lr	5e-4
lr-scheduler	inverse_sqrt
warmup-updates	4000
clip-norm	0.0
weight-decay	0.0001
dropout	0.3
update-freq	2
max-tokens	3000
best-checkpoint-metric	bleu
maximize-best-checkpoint-metric	True

Table 5: Fairseq training hyperparameters (non-default) for 4 GPU training.

Source	I hope you will have a little time and energy to focus on another report which is, despite its technicality, quite important for all of us.
Target:	Ich Ich hoffe, Ich hoffe, Sie Ich hoffe, Sie Ich hoffe, Sie haben Ich hoffe, Sie haben ein Ich hoffe, Sie werden ein wenig Zeit Ich hoffe, Sie haben etwas Zeit Ich hoffe, Sie haben etwas Zeit und Ich hoffe, Sie werden etwas Zeit und Energie haben, Ich hoffe, Sie haben etwas Zeit und Energie, um sich Ich hoffe, Sie haben etwas Zeit und Energie, um sich auf Ich hoffe, Sie werden ein wenig Zeit und Energie haben, um sich auf ein anderes Thema Ich hoffe, Sie haben etwas Zeit und Energie, um sich auf einen weiteren Bericht zu konzentrieren, Ich hoffe, Sie haben etwas Zeit und Energie, um sich auf einen anderen Bericht zu konzentrieren, ⋮ Ich hoffe, Sie werden ein wenig Zeit und Energie haben, um sich auf einen anderen Bericht zu konzentrieren, der trotz seiner Formalität für uns alle sehr wichtig ist.

Table 6: Examples of flicker caused by the teacher model. *Source* is the original full sentence which is input as a growing input prefix. *Target* is the output prefix in successive retranslations.

D Flicker-Latency Tradeoff

In Figure 4, we show the flicker-latency tradeoff for all language-pair and testset combinations.

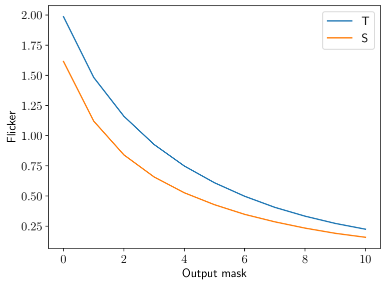
E Biased Beam Search

We consider the effect of the bias mask on full sentence translation quality, as measured by BLEU. The bias mask is measured in sentencepiece tokens. Based on Figure 6, we set the bias mask to 6 for ESIC and 10 for IWSLT, in order to avoid a loss of BLEU.

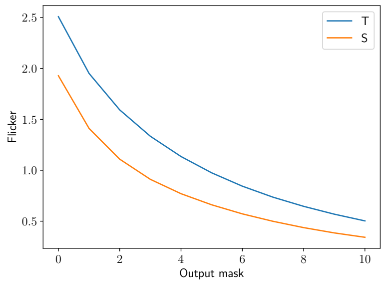
In Figure 7, we show flicker-latency tradeoffs for all language pair and testset combinations.

F Controlling Monotonicity

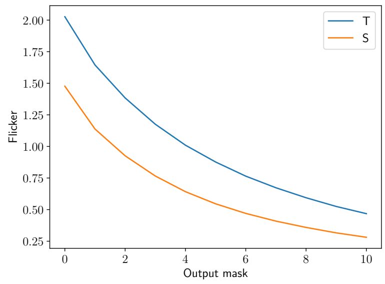
We have shown the scores for CHRF and COMET for monotonicity control experiments in addition to BLEU in Table 7. In Figure 5, we show the flicker-latency tradeoff for monotonicity-controlled experiments for all language-pair and testset combinations.



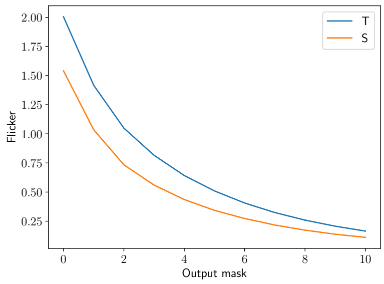
(a) En→De ESIC



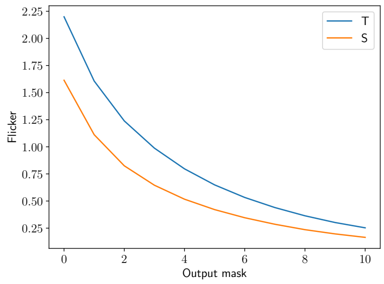
(b) En→De IWSLT



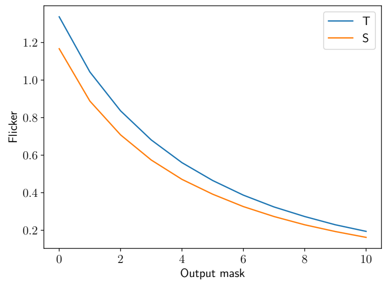
(c) De→En IWSLT



(d) En→Cs ESIC

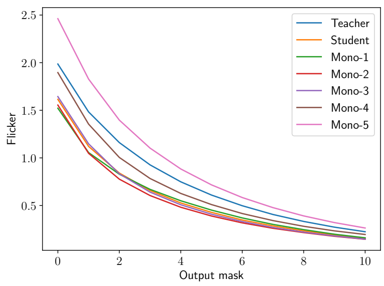


(e) En→Cs IWSLT

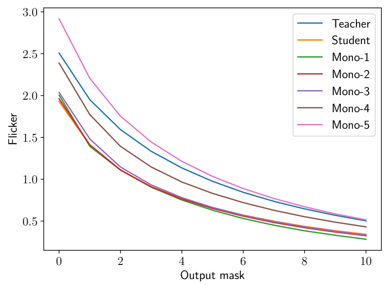


(f) Cs→En IWSLT

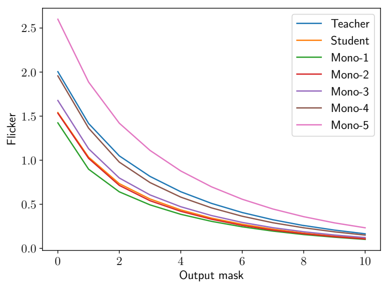
Figure 4: Flicker-latency tradeoff for the teacher-student models. We control latency by varying the output mask.



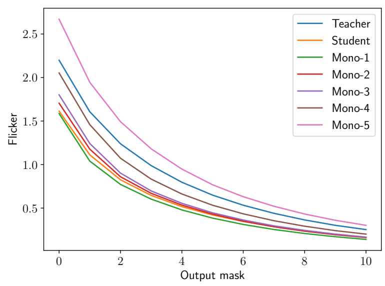
(a) En→De ESIC



(b) En→De IWSLT

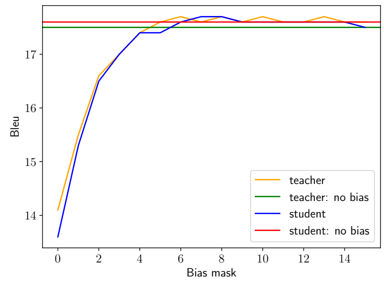


(c) En→Cs ESIC

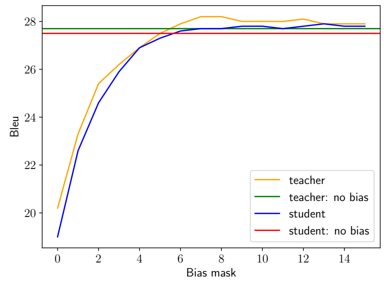


(d) En→Cs IWSLT

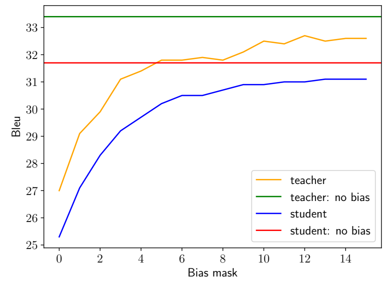
Figure 5: Latency–flicker tradeoff for the monotonicity-controlled models. Monotonicity control ranges from 1 (training data created with maximim monotonicity) to 5 (minimum monotonicity).



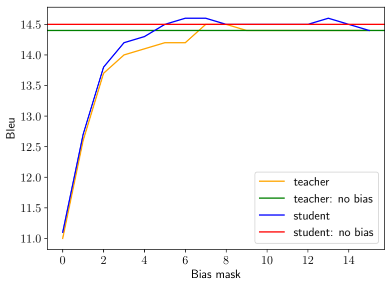
(a) En →De ESIC



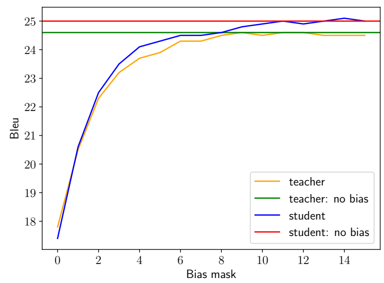
(b) En →De IWSLT



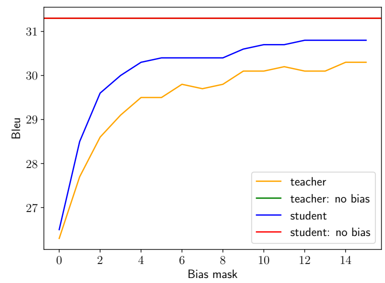
(c) De →En IWSLT



(d) En→Cs ESIC

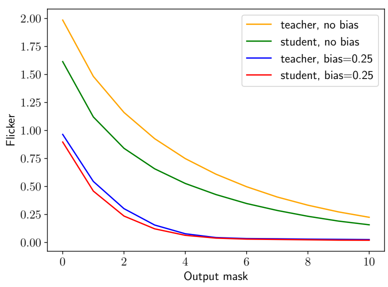


(e) En→Cs IWSLT

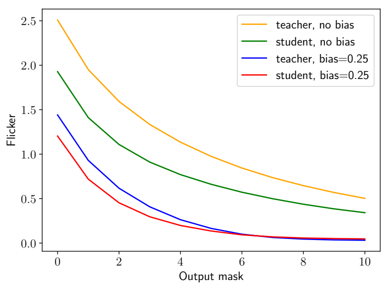


(f) Cs→En IWSLT

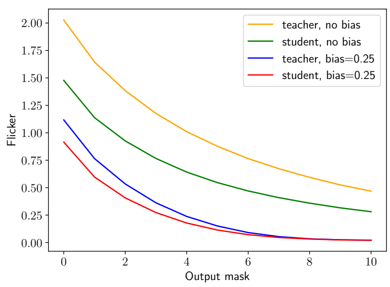
Figure 6: Dependence of BLEU on bias mask when applying biased beam search.



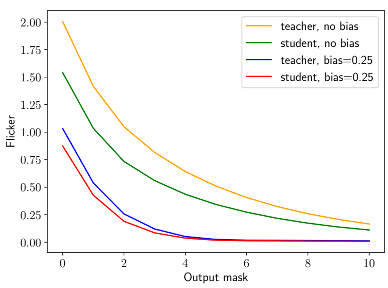
(a) English → German ESIC



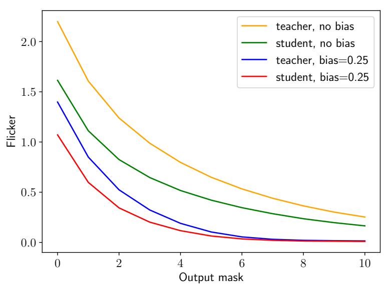
(b) English → German IWSLT



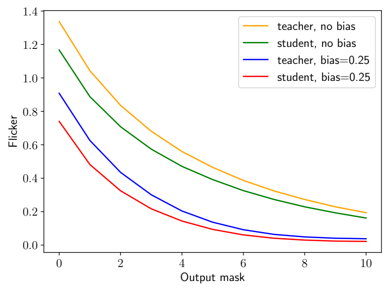
(c) German → English IWSLT



(d) English → Czech ESIC



(e) Czech → English IWSLT



(f) Czech → English IWSLT

Figure 7: Flicker vs mask on biased beam search.

		En→De		En→Cs	
Metric	Model	ESIC	IWSLT	ESIC	IWSLT
BLEU	Teacher	17.5	27.7	14.4	24.6
	$\text{Student}_{\text{model}}$	17.6	27.5	14.5	25.0
	Mono-1	8.6	14.4	14.7	23.6
	Mono-2	17.6	27.4	14.5	25.0
	Mono-3	17.5	27.9	14.5	25.7
	Mono-4	17.2	26.6	13.8	24.7
	Mono-5	16.0	25.0	12.5	23.0
ChrF	Teacher	58.9	56.9	51.5	51.5
	$\text{Student}_{\text{model}}$	58.8	57.2	51.7	51.7
	Mono-1	42.4	39.6	51.3	50.7
	Mono-2	58.7	57.3	51.8	52.0
	Mono-3	59.0	57.8	51.7	52.2
	Mono-4	59.0	56.8	51.4	51.4
	Mono-5	58.5	55.0	50.7	50.2
COMET	Teacher	.553	.330	.651	.639
	$\text{Student}_{\text{model}}$	.532	.326	.672	.642
	Mono-1	.510	-0.028	.639	.597
	Mono-2	.526	.295	.650	.636
	Mono-3	.530	.326	.678	.641
	Mono-4	.535	.313	.677	.639
	Mono-5	.518	.247	.633	.577

Table 7: Full results of student models with monotonicity control.