Speed-optimized, Compact Student Models that Distill Knowledge from a Larger Teacher Model: the UEDIN-CUNI Submission to the WMT 2020 News Translation Task

Ulrich Germann[⋆], **Roman Grundkiewicz**[⋆], **Martin Popel**[‡], **Radina Dobreva**[⋆], **Nikolay Bogoychev**[⋆], **Kenneth Heafield**[⋆]
[⋆] University of Edinburgh, UK
[‡] Charles University, Prague, Czech Republic
{ugermann|rgrundki|rdobreva|nbogoych|kheafiel}@inf.ed.ac.uk
popel@ufal.mff.cuni.cz

Abstract

We describe the joint submission of the University of Edinburgh and Charles University, Prague, to the Czech/English track in the WMT 2020 Shared Task on News Translation. Our fast and compact student models distill knowledge from a larger, slower teacher. They are designed to offer a good trade-off between translation quality and inference efficiency. On the WMT 2020 Czech ↔ English test sets, they achieve translation speeds of over 700 whitespace-delimited source words per second[1] on a single CPU thread, thus making neural translation feasible on consumer hardware without a GPU.

1 Introduction

The conventional set-up of the WMT Shared Tasks on News Translation emphasizes translation quality (however measured) above all else. Constraints on the data that may be used for training in the 'constrained' track establish a level playing field in terms of the information available to the translation model and its training process, but there are no constraints on the computational power and effort spent to achieve the results. In contrast, the WNGT Shared Task on Efficient Translation (Heafield et al., 2020) encourages participants to submit systems that are both accurate and efficient during inference (i.e., translation). So far, there has been little interaction between the two tasks.

With our joint submission between the University of Edinburgh (UEDIN) and Charles University, Prague (CUNI), we strive to bridge this gap. We submitted small, efficient systems that distilled knowledge from a more powerful teacher model via sequence-level knowledge distillation (Kim and Rush, 2016). In a nutshell, the procedure can be described as follows:

1. Train a powerful teacher model on the available training data set D.
2. Translate the source side of D plus available monolingual data in the source language and appropriate text domain with the teacher model to generate the training set D'.
3. Train a small student model on D'.

While the computational effort to first train a teacher model and then distill its knowledge into a student model is considerably greater than just training the teacher — in addition to training the teacher, we have to translate the training data and then train a student on that data —, the advantage is at inference time. Even the larger of the two student models we present in this paper can translate on a single CPU core at an acceptable speed (cf. Tab. 4). Translation can further be sped up by quantizing parameters to 8 bits of precision and using Integer General Matrix Multiplication (IntGEMM) for inference. Even though our submissions to the WMT 2020 Shared Task on News Translation were produced by unquantized floating point models, we report performance numbers for quantized models as well to demonstrate their efficacy and show that they can speed up translation by about 10% with negligible trade-offs in terms of BLEU score over unquantized models.

All models used in this work are based on the Transformer architecture (Vaswani et al., 2017). Details are discussed in the sections below; the hyper-parameter settings for each model are listed in Tab. 1. The student models described in this paper can be obtained via https://github.com/browsermt/students.

2 Teacher Models

The teacher models were produced by CUNI, using the Tensor2Tensor deep-learning toolkit (Vaswani et al., 2018)[2]. The teachers were trained on the full

[1] Bogoychev et al. (2020) report translation speeds of up to 3135 source words per second on a single CPU thread; the actual throughput depends not only on the computer hardware used for translation but also on the distribution of translation segment lengths in the test set.

[2] https://github.com/tensorflow/tensor2tensor

Proceedings of the 5th Conference on Machine Translation (WMT), pages 191–196
Online, November 19–20, 2020.

Table 1: Transformer hyper-parameters for T2T teacher and Marian student models.

	teacher		student	
parameter	**cs→en**	**en→cs**	**base**	**tiny**
vocabulary size (spm)	32K	32K	32K	32K
joint vocabulary	yes	yes	yes	yes
encoder layers	6	12	6	6
decoder layers	6	6	2	2
decoder auto-reg.	self-attention	self-attention	SSRU	SSRU
tied embeddings	yes	yes	yes	yes
embedding size	1024	1024	512	256
filter size	4096	4096	2048	1536
number of att. heads	16	16	8	8
att. key size	64	64	64	64
att. value size	64	64	64	64
checkpoints avg.	8	8	exp. smoothing[a]	
back-translation	block-BT	block BT	none	none
beam search alpha	1.0	1.0	1.0	1.0
max training length	150	150	200	200

[a] Exponential smoothing with $\alpha = 0.0001$.

CzEng 2.0 dataset (Kocmi et al., 2020)[3], consisting of genuine (authentic) parallel data as well as monolingual news data translated by CUNI's transformer systems from WMT 2018 (Popel, 2018) to generate back-translated synthetic training data (Sennrich et al., 2016). Rather than shuffling and mixing authentic and synthetic training data, the teacher models were trained on alternating blocks of authentic and synthetic data ("block-regime back-translation" (block-BT); Popel et al., 2020), spending about 10 hours of training time on each block.

The model parameters for the final teacher models were obtained by checkpoint averaging over the last 8 checkpoints of the training process, saved in hourly intervals.

The en→cs teacher model used in this work also produced CUNI's primary submission to the WMT 2020 Shared Task on News Translation ("CUNI-Transformer"; Popel, 2020). However, the CUNI submission used a beam size of 4 instead of 8 as used in this work, resulting in a BLEU score on the WMT 2020 en→ test set that is 0.2 lower than the BLEU score reported in Tab. 4.

The cs→en teacher model used in this work has only 6 encoder layers as opposed to the 12 encoder layers used in CUNI's primary submission to the Shared Task, resulting in a BLEU score on the WMT 2020 test set that is 1.0 BLEU points lower than the score achieved by the model used for CUNI's primary submission.

3 Student Models

The smaller, more efficient student models were trained by UEDIN with the Marian NMT toolkit (Junczys-Dowmunt et al., 2018a)[4]. The students were trained on artificial training data produced by knowledge distillation (Kim and Rush, 2016), where the target side of the parallel data is the teacher model's translation of the source side. The basic idea is that the teacher guides the student towards translations that can be achieved with the teacher's knowledge.

3.1 Student Model Architectures

The student models use the architecture proposed by Kim et al. (2019) with improvements by Bogoychev et al. (2020). Apart from using fewer layers and fewer dimensions in each layer, the main difference of the students from the conventional transformer architecture is the use of Simpler Simple Recurrent Units (SSRU; Kim et al., 2019) instead of the self-attention mechanism in decoder part of the transformer. For the sake of simplicity, our student models use exponential smoothing of model parameters with a smoothing parameter of 0.0001 instead of the checkpoint averaging used to produce the final teacher models.

For each translation direction, we trained two models: a base transformer and a 'tiny' transformer with fewer decoder layers and a smaller number of embedding and filter dimensions; specifications are shown in Tab. 1.

3.2 Data Preparation

To create artificial training data for the students, we used the original parallel section of the CzEng 2.0 dataset but no back-translations. Instead, we translated ca. 40 million sentences from the mono-

[3] http://ufal.mff.cuni.cz/czeng

[4] https://github.com/marian-nmt/marian

Table 2: Data used for training the models (in millions of sentence pairs).

data set	teacher		student	
	cs→en	en→cs	cs→en	en→cs
CzEng 2.0 parallel (original)	61.1m	61.1m		
CzEng 2.0 parallel (pre-filtered)			42.3m	42.3m
Back-translated news (CzEng 2.0 'mono')	50.6m	76.2m		
Teacher-translated news			50.1m	43.0m
Top 90% according to alignment score			83.2m	76.8m
Total used	111.7m	137.3m	83.2m	76.8m

lingual English NewsCrawl corpus[5] (2018 and 2019) for en→cs and 50 million sentences from the monolingual Czech NewsCrawl corpus (2013–2019) for cs→en.

Prior to translation with the respective teacher model, we filtered and de-duplicated the data. Filtering consisted of the following steps:

- Sentence-level deduplication.
- Removal of excessively long sentences (longer than 120 space-separated tokens; note that the sentence length limit for training in terms of subword units was 200; cf Tab. 1).
- Removal of sentence pairs that were not identified as the correct language by the fastText language identifier (Joulin et al., 2017, 2016) Python module.[6]
- For parallel data, removal of sentence pairs with length ratio larger than 2.5 (in terms of words of untokenized text), i.e. the longer sentence could not be more than 2.5 times as long as the shorter one.
- Removal of sentences in which less than half the words contain an alphabetical character or less than half the characters belong to the alphabet of the specific language.

We translated the cleaned data with the Tensor2Tensor teacher model with a beam size of 8. A small proportion of 'odd' sentences that had escaped our cleaning process, for example sentences with several long URLs that resulted in very long token sequences after segmentation into subword units, forced us to use a relatively small batch size of 8–24 sentences to avoid out-of-memory errors. For load balancing, we split the translation load into blocks of 10,000 sentences each and parallelized the translation process over dozens of machines. Using comparatively many translation blocks gave us flexibility in scaling the translation operation in response to resource availability.

[5] http://data.statmt.org/news-crawl/
[6] https://pypi.org/project/fasttext/

Despite the small batch size, 32 of our 10,0000-sentence input chunks still failed to translate due to memory limitations. A cursory investigation revealed that these often contained undesirable material (such as Javascript and HTML code that had somehow survived the filtering process), so that we decided to simply discard those blocks of data.

We made no effort to optimize translation speed and throughput for the teacher models in Tensor2Tensor; translation time for a single 10,000-sentence block was ca. 30–45 minutes, depending on sentence lengths and hardware used.

Table 3: Distribution of teacher-produced translations chosen by sentence-level BLEU score over their respective ranks in the decoder beam.

rank	en→cs	cs→en
1	32.22%	31.24%
2	15.20%	15.63%
3	12.25%	12.21%
4	9.79%	9.87%
5	8.89%	8.88%
6	7.73%	7.85%
7	7.26%	7.40%
8	6.67%	6.93%

For the authentic parallel data, we selected from the 8 top-scoring final translation hypotheses for each source sentence the one with the highest sentence-level BLEU score with respect to the original target side of the data. Table 3 shows the distribution of the hypotheses selected over the respective beam ranks. For the monolingual data, for which we obviously have no human reference translations, we simply chose the highest-scoring translation. Sentence pairs where the translation contained the same whitespace-separated sequence of words three or more times in a row, or the same sequence of one or more characters in five or more subsequent repetitions (which can happen when the recursive decoder goes into a loop) were discarded.

We subsequently tokenized the synthetic teaching data (source and translations by the teacher model) with SentencePiece (Kudo and Richardson, 2018),

using a joint vocabulary for both languages with a size of 32,000 tokens. This vocabulary is also used by the final systems. The tokenized training data was word-aligned in both translation directions with FastAlign (Dyer et al., 2013). Directional word alignments were then symmetrized with the grow-diag-final-and symmetrization algorithm (Koehn et al., 2003).

These word alignments serve mainly three purposes: (a) to guide the attention mechanism during training of the student models (Liu et al., 2016) with guided alignment (Chen et al., 2016); (b) to produce shortlists of translation candidates to limit the choice of target words that need to be considered during inference (Junczys-Dowmunt et al., 2018b); and (c) to give us a rough estimate of translation quality via average per-token alignment scores for each sentence pair. We used these scores to discard the bottom 10% of our artificial training data.

Based on our experiments, the guided alignment training is neither required for student model training, nor does it improve BLEU scores on the development set. However, it encourages the guided decoder layer to mimic word alignments, which can be useful in post-processing.[7] We used default settings from Marian for the guided alignment training.

3.3 Quantized Models

Floating point operations are computationally more expensive than integer operations. However, as Han et al. (2016) have shown, neural network inference does not require the high precision of representation and computation that 32-bit floating point numbers offer. Devlin (2017) suggests a simple quantization mechanism for quantizing parameters to 16-bit integer precision and notes that support for off-the-shelf 8-bit integer matrix multiplication is lacking. Bogoychev et al. (2020) fill that gap and provide an 8-bit quantization and fine tuning scheme for Marian based on the intgemm library;[8] we used that scheme for our models. The model parameters are quantized offline from *float32* to *int8*, and during translation, the activations are quantized just prior to each GEMM operation. The GEMM operation is performed in 8-bit integers, and then the result is de-quantized back to *float32*. Despite the extra quantization and de-quantization involved, the increased speed at which 8-bit integer multiplication is performed more than compensates for it. Bogoychev et al. (2020) observe that smaller student presets lose BLEU when quantized. In order to counteract that, we perform model fine tuning following the work of Aji and Heafield (2020): We replace the GEMM routine implementation with a custom one that is damaged, according to the quantization scheme and perform several thousand mini-batch updates of the model. The damaged GEMM implementation can only produce 255 unique float values (corresponding to the 8-bit integer dequantization range) and the model quickly learns to work with those values and recovers some of the BLEU lost compared to untuned quantized model.

4 Results

In Table 4, we show the performance of the three models in terms of BLEU scores for the WMT 2020 cs↔en test sets and translation speed. Teacher models ran on an Nvidia GeForce GTX 1080 with a batch size of 16. Student models were run on a single CPU core on an Intel Intel(R) Xeon(R) CPU E5-2680 0 @ 2.70GHz with a batch size of 64. It should be noted that we made no effort to optimize the teachers for translation speed.

Text segments in the WMT 2020 cs↔en test sets are aligned at the paragraph level; we therefore split the provided segments into individual sentences prior to translation with Moses-style sentence splitting[9] and restored paragraphs afterwards.

All BLEU scores were computed with SacreBLEU.[10] For the en→cs teacher model, removing repetitions and adapting quotation marks to Czech spelling conventions boosted the BLEU score by 1.6 BLEU points; for student models, this post-processing is not necessary. Having been trained on post-processed teacher output, the student models have learned this correctly. Except where indicated, translation was with a greedy search (beam size 1) and a shortlist of 50 translation candidates per source word.

Due to resource congestion, we were not able to fully train the models by the submission deadline; our submissions are based on the systems with the best validation BLEU score available at the time. For validation, we used the concatenation of all parallel data for the respective translation direction from the WMT test sets from the years 2008 through 2019 where the original language of the data is the source language for the translation direction in question.

In terms of BLEU score on the WMT 2020 test set, the submitted primary system for cs→en is ca. 0.5 BLEU points below the final system; the en→cs system incidentally outperforms the final system by 0.5 BLEU points, as shown in Tab. 4.

[7] For example, for handling HTML tags in translated texts.

[8] https://github.com/kpu/intgemm

[9] https://github.com/ugermann/ssplit-cpp, which is a C++ reimplentation of the Moses sentence splitter, currently covering only a subset of the languages supported by the Moses Sentence splitter (no non-roman scripts).

[10] Post (2018); BLEU+case.mixed+lang.${src}-${trg}+numrefs.1 +smooth.exp+test.wmt20+tok.13a +version.1.4.13

Table 4: BLEU results for teacher and student models (base, tiny) on the WMT20 test set.

	cs→en			en→cs		
system	BLEU	time[a]	words/sec.	BLEU	time	words/sec.
teacher (no postprocessing)[b]	27.6	782 sec.	33	34.0	1131 sec.	39
teacher (with postprocessing)[b]				35.6	1131 sec.	39
base (float32, primary sub.)[b]	27.7[c]	424 sec.	61	36.3[d]	637 sec.	69
base (float32)[b]	28.2	294 sec.	88	35.8	465 sec.	95
base (float32)	27.9	101 sec.	256	35.7	151 sec.	292
base (8-bit quant., untuned)	27.5	90 sec.	287	34.4	136 sec.	324
base (8-bit quant., tuned)	27.8	88 sec.	294	35.3	136 sec.	324
base (8-bit quant., tuned, precomp[e])	27.9	89 sec.	291	35.7	135 sec.	326
tiny (float32)	27.0	38 sec.	681	34.7	59 sec.	746
tiny (8-bit quant., untuned)	25.6	34 sec.	761	31.9	55 sec.	815
tiny (8-bit quant., tuned)	26.9	35 sec.	739	32.9	55 sec.	815
tiny (8-bit quant., tuned, precomp[e])	26.9	35 sec.	739	32.8	53 sec.	830

[a] Inference time for core test set without additional test sets.
[b] Beam size 8; postprocessing: remove repetitions, adapt quotation marks to Czech conventions.
[c] After 65K updates, shortlist size 100.
[d] After 190K updates, shortlist size 100.
[e] Pre-computed scaling factor for quantization, see Sec. 5.1 in Bogoychev et al. (2020) for details.

5 Conclusion

We presented student models that distill knowledge from a larger teacher model without loss in BLEU performance. (In fact, for the WMT 2020 test set, our larger student models technically outperform the teacher in terms of BLEU, but we consider that difference accidental.) At the same time, they are significantly faster and do not require a GPU for inference, making it possible to perform neural machine translation on consumer-grade hardware without the use of a GPU.

Acknowledgements

This work was supported by funding from the European Union's Horizon 2020 research and innovation programme under grant agreements No 825303 (Bergamot) and 825627 (European Language Grid).

This work was performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service (http://www.csd3.cam.ac.uk/), provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/P020259/1), and DiRAC funding from the Science and Technology Facilities Council (www.dirac.ac.uk). The work has been using language resources developed and distributed by the LINDAT/CLARIAH-CZ project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2018101).

References

Aji, Alham Fikri and Kenneth Heafield. 2020. Compressing neural machine translation models with 4-bit precision. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 35–42, Online. Association for Computational Linguistics.

Bogoychev, Nikolay, Roman Grundkiewicz, Alham Fikri Aji, Maximiliana Behnke, Kenneth Heafield, Sidharth Kashyap, Emmanouil-Ioannis Farsarakis, and Mateusz Chudyk. 2020. Edinburgh's submissions to the 2020 Machine Translation Efficiency Task. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 218–224, Online. Association for Computational Linguistics.

Chen, Wenhu, Evgeny Matusov, Shahram Khadivi, and Jan-Thorsten Peter. 2016. Guided alignment training for topic-aware neural machine translation. *Proceedings of AMTA 2016*.

Devlin, Jacob. 2017. Sharp models on dull hardware: Fast and accurate neural machine translation decoding on the CPU. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2820–2825, Copenhagen, Denmark. Association for Computational Linguistics.

Dyer, Chris, 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. Code at `https://github.com/clab/fast_align`.

Han, Song, Huizi Mao, and William J. Dally. 2016. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Heafield, Kenneth, Hiroaki Hayashi, Yusuke Oda, Ioannis Konstas, Andrew Finch, Graham Neubig, Xian Li,

and Alexandra Birch. 2020. Findings of the fourth workshop on neural generation and translation. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 1–9, Online. Association for Computational Linguistics.

Joulin, Armand, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Joulin, Armand, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Junczys-Dowmunt, Marcin, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018a. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics. Code at `https://github.com/marian-nmt/marian-dev`.

Junczys-Dowmunt, Marcin, Kenneth Heafield, Hieu Hoang, Roman Grundkiewicz, and Anthony Aue. 2018b. Marian: Cost-effective high-quality neural machine translation in C++. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 129–135, Melbourne, Australia. Association for Computational Linguistics.

Kim, Yoon 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, Austin, Texas. Association for Computational Linguistics.

Kim, Young Jin, Marcin Junczys-Dowmunt, Hany Hassan, Alham Fikri Aji, Kenneth Heafield, Roman Grundkiewicz, and Nikolay Bogoychev. 2019. From research to production and back: Ludicrously fast neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 280–288, Hong Kong. Association for Computational Linguistics.

Kocmi, Tom, Martin Popel, and Ondrej Bojar. 2020. Announcing CzEng 2.0 parallel corpus with over 2 gigawords. *arXiv preprint arXiv:2007.03006*.

Koehn, Philipp, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133.

Kudo, Taku 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. Code at `https://github.com/google/sentencepiece`.

Liu, Lemao, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Neural machine translation with supervised attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3093–3102, Osaka, Japan. The COLING 2016 Organizing Committee.

Popel, Martin. 2018. CUNI transformer neural MT system for WMT18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 482–487, Belgium, Brussels. Association for Computational Linguistics.

Popel, Martin. 2020. CUNI English-Czech and English-Polish Systems in WMT20: Robust document-level training. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*, Online. Association for Computational Linguistics.

Popel, Martin, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtskỳ. 2020. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11(1):1–15.

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

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Vaswani, Ashish, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2tensor for neural machine translation. *CoRR*, abs/1803.07416.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.