Attention Mechanism with Energy-Friendly Operations

Yu Wan[a,b*†] **Baosong Yang**[b*] **Dayiheng Liu**[b] **Rong Xiao**[b†‡] **Derek F. Wong**[a]
Haibo Zhang[b] **Boxing Chen**[b] **Lidia S. Chao**[a]
[a]NLP^{2}CT Lab, University of Macau
nlp2ct.ywan@gmail.com, {derekfw,lidiasc}@umac.mo
[b]Alibaba Group
{yangbaosong.ybs,liudayiheng.ldyh,birong.xr,zhanhui.zhb,
boxing.cbx}@alibaba-inc.com

Abstract

Attention mechanism has become the dominant module in natural language processing models. It is computationally intensive and depends on massive power-hungry multiplications. In this paper, we rethink variants of attention mechanism from the energy consumption aspects. After reaching the conclusion that the energy costs of several energy-friendly operations are far less than their multiplication counterparts, we build a novel attention model by replacing multiplications with either selective operations or additions. Empirical results on three machine translation tasks demonstrate that the proposed model, against the vanilla one, achieves competitable accuracy while saving 99% and 66% energy during alignment calculation and the whole attention procedure. Code is available at: https://github.com/NLP2CT/E-Att.

1 Introduction

Attention mechanism (ATT, Bahdanau et al., 2015; Vaswani et al., 2017; Yang et al., 2018) has demonstrated huge success in a variety of natural language processing tasks (Su et al., 2018; Kitaev and Klein, 2018; Tan et al., 2018; Yang et al., 2019a; Devlin et al., 2019; Zhang et al., 2020). The module learns hidden representations of a sequence by serving each word as a query to attend to all keys in the target sentence, then softly assembling their values. It is a de-facto standard to achieve this via performing linear projections and dot products on representations of queries and keys (Vaswani et al., 2017), resulting in large amount of multiplications. In spite of its promising quality, such kind of paradigm may be not the preferred solution from the energy consumption aspect (Horowitz, 2014; Raffel et al., 2020). How to build a high energy-efficient ATT still remains a great challenge.

Operation (FP32)	**ASIC**	**FPGA**
Addition	0.9	0.4
Multiplication	3.7	18.8

Table 1: Energy cost ($10^{-12} Joule$) of addition and multiplication on ASIC/FPGA chips (You et al., 2020). The representative of ASIC chip is Google's TPU, while Microsoft cloud employs FPGA chips.

Our work starts from in-depth investigations on approaches in ATT context with respect to model compression (Hinton et al., 2015; Jiao et al., 2020) and complexity optimization (Yang et al., 2019b; Raganato et al., 2020; Beltagy et al., 2020; Tay et al., 2021). These approaches can potentially alleviate the problem of high energy consumption in ATT. Nevertheless, intentions of all these methods are not exactly from the energy-friendly perspective, thus overlooking the origin of energy consumed, i.e., basic arithmetic operations in electric equipments. Massive multiplications still remain, consuming far more energy than its additive counterpart on modern devices (Table 1, Li et al., 2020).

To this end, we propose to approach this problem from a new direction – replacing massive multiplications in ATT with cheaper operations. Concretely, we propose a novel energy-efficient attention mechanism (E-ATT). It equips binarized selective operations instead of linear projections over input hidden states, and measures attentive scores using L_1 distance rather than dot-product. Consequently, E-ATT abandons most of multiplications to reach the goal of energy cost reduction.

We examine our method with TRANSFORMER model (Vaswani et al., 2017), and conduct experiments on three machine translation tasks. Compared with conventional ATT, our E-ATT can save more than 99% energy of the vanilla alignment calculation and around 66% energy of the whole

[*] Equal contribution.

[†] Work was done when interning at DAMO Academy, Alibaba Group.

[‡] Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2022, pages 3969 - 3976
May 22-27, 2022 ©2022 Association for Computational Linguistics

attention model. In the meanwhile, our models yield acceptable translation qualities across language pairs. Extensive analyses also demonstrate that E-ATT can functionally model semantic alignments without using multiplications.

2 Preliminary

Conventional Attention Mechanism Given input representations $\mathbf{X} \in \mathbb{R}^{l_1 \times d}$ and $\mathbf{Y} \in \mathbb{R}^{l_2 \times d}$ with l_1, l_2 being sequence length, and d is the input dimensionality. Note l_1 and l_2 may be equal for self-attention pattern, and represent lengths of target and source sequence in cross-attention. ATT first projects the inputs into three representations[1]:

$$\mathbf{Q} = \mathbf{X}\mathbf{W}_Q, \quad [\mathbf{K}; \mathbf{V}] = \mathbf{Y}[\mathbf{W}_K; \mathbf{W}_V], \qquad (1)$$

where $\{\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V\} \in \mathbb{R}^{d \times d}$ are trainable parameters. $\mathbf{Q} \in \mathbb{R}^{l_1 \times d}$, $\{\mathbf{K}, \mathbf{V}\} \in \mathbb{R}^{l_2 \times d}$ are query, key and value representations, respectively. The attention alignment is calculated with dot-product multiplication and softmax activation:

$$\mathbf{M}_{ij} \propto \exp(\frac{\mathbf{Q}_i \mathbf{K}_j^\top}{\sqrt{d}}) \in \mathbb{R}^{l_1 \times l_2}. \qquad (2)$$

Then, the output is derived by multiplying attention weights with value representation $\hat{\mathbf{V}}$:

$$\mathbf{O} = \mathbf{M}\mathbf{V} \in \mathbb{R}^{l_1 \times d}. \qquad (3)$$

As seen, matrix multiplications are massively exploited into conventional ATT.

Related Work Several related approaches potentially alleviate the power-hungry drawback of ATT. One direction relies on model compression by pruning redundant parameters (Denton et al., 2014; Wang et al., 2016; Zhuang et al., 2018) or distilling the learned knowledge from a large model to a smaller one (Hinton et al., 2015; Yim et al., 2017), which still maintains multiplicative operations. Another direction aims at reducing the computational complexity of attention module, e.g. linearly projecting input (Dense, Tay et al., 2021), or randomly initializing and training attention weights (RandInit, Tay et al., 2021). To give a full comparison of energy consumption of these approaches, we conduct the number of multiplicative operations and energy costs across modules in Table 2. As seen, vanilla ATT (Vaswani et al., 2017) involves the most multiplicative operations, and requires the most energy than other methods. Although Dense and RandInit significantly reduce the energy consumption, Tay et al. (2021) point out that these approaches fail to be employed into cross-attention networks, since neither linear transition nor randomly initialized matrix is able to exactly model alignment information across languages.

Model	Alignment		Attention	
	# mul	$\Delta_A(\%)$	# mul	$\Delta_A(\%)$
Vanilla	$2ld^2 + l^2d$	100.00	$3ld^2 + 2l^2d$	100.00
Dense	$ld^2 + l^2d$	51.10	$2ld^2 + 2l^2d$	67.59
RandInit	0	0.00	$ld^2 + l^2d$	33.80
E-ATT	0	0.44	$ld^2 + l^2d$	34.10

Table 2: Calls of multiplication (mul) and energy consumption ratio on ASIC chip (Δ_A) of vanilla ATT, Dense, RandInit, and our model. "Alignment" and "Attention" indicate the statistics are conducted at the level of alignment calculation and the whole attention model, respectively. l and d are sequential length and model size. Please refer to Appendix A for more details.

3 Energy-Efficient Attention Mechanism

In this section, we describe E-ATT by pertinently reducing the multiplicative operations of ATT, including selective operation and L_1 distance.

3.1 Feature Selection with Discreteness

Since the linear transitions of queries and keys (Equation 1) involve massive multiplications within conventional ATT, we propose to modify them with binarized quantization (Liu et al., 2018; Qin et al., 2020). Concretely, the inputs $\mathbf{X}$ and $\mathbf{Y}$ are turned into discrete value with a threshold function $f(\cdot)$:

$$f(x) = \begin{cases} 1 & x > \tau, \\ 0 & \text{otherwise}, \end{cases} \qquad (4)$$

where τ and d are threshold and hidden size, respectively. The derived representations $\tilde{\mathbf{X}} = f(\mathbf{X})$ and $\tilde{\mathbf{Y}} = f(\mathbf{Y})$ contain discrete features composing of zeros and ones. Since this procedure is undifferentiable, we need to predefine a pattern of gradient calculation for $\mathbf{X}$ when receiving back-propagated gradient $\mathbf{Z}$. Wu et al. (2018) pointed out that, when simulating the back-propagated progress across discrete activations, those patterns which peak at the medium of domain reveal better training stabilization and model performance. We thus use a modified Gaussian function during back-propagation

[1] For simplicity, we omit the bias in linear projections, as well as splits and concatenations in multi-head mechanism.

following Wu et al. (2018):

$$\nabla \mathbf{X} = \sqrt{\frac{2}{\pi}} e^{-2(\mathbf{Z}-\tau)^2}, \tag{5}$$

and the same procedure is applied for $\mathbf{Y}$. Then given parameters $\mathbf{W}_Q, \mathbf{W}_K \in \mathbb{R}^{d\times d}$, we derive query and key representations $\mathbf{Q}, \mathbf{K}$ by applying masked selection function:

$$\tilde{\mathbf{Q}} = g(\tilde{\mathbf{X}}, \tilde{\mathbf{W}_Q}) \in \mathbb{R}^{l_1\times d\times d}, \tag{6}$$

$$\tilde{\mathbf{K}} = g(\tilde{\mathbf{Y}}, \tilde{\mathbf{W}_K}) \in \mathbb{R}^{l_2\times d\times d}, \tag{7}$$

$$\mathbf{Q} = \sum_{i=1}^{d} \tilde{\mathbf{Q}}_{\cdot,i,\cdot}; \quad \mathbf{K} = \sum_{i=1}^{d} \tilde{\mathbf{K}}_{\cdot,i,\cdot}, \tag{8}$$

where $\tilde{\mathbf{W}_Q} \in \mathbb{R}^{l_1\times d\times d}$ and $\tilde{\mathbf{W}_K} \in \mathbb{R}^{l_2\times d\times d}$ are derived by tiling $\mathbf{W}_Q, \mathbf{W}_K$ with l_1 and l_2 times, respectively. $g(\cdot,\cdot)$ represents indexed feature selection defined as follows:

$$g(\mathbf{U}, \mathbf{P}) = \begin{cases} \mathbf{U}_{i,j,\cdot} & \mathbf{P}_{i,j} = 1, \\ \mathbf{0} & \text{otherwise.} \end{cases} \tag{9}$$

3.2 Pairwise L_1 Distance

As the dot-product multiplication can be viewed as similarity calculation between $\mathbf{Q}$ and $\mathbf{K}$, we argue that other similarity estimation methods can play this role as well. Accordingly, we further propose to use pairwise L_1 distance instead, which does not require any multiplication. Attention score calculation in Equation 2 is then modified as:

$$\mathbf{M}_{ij} \propto \exp(-\frac{||\mathbf{Q}_i - \mathbf{K}_j||_1}{\sqrt{d}}), \tag{10}$$

where $||\cdot||_1$ denotes the L_1 norm of inputted vector. Here we use negative L_1 value to ensure that larger distance contributes lower attention score.

4 Experiments

4.1 Dataset Preprocessing

In this paper we evaluate our approach with three widely used machine translation datasets: IWSLT'15 English -Vietnamese (En-Vi), WMT'14 English - German (En-De) and WMT'17 Chinese - English (Zh-En). All datasets are segmented into subwords by byte-pair encoding (BPE, Sennrich et al., 2016) with 32k merge operations. Specially, for the former two tasks, we apply joint BPE for both source and target languages.

Dataset	Train	Dev	Test
En-Vi	13.3K	1,553	1,268
En-De	4.50M	3,000	3,003
Zh-En	20.6M	2,002	2,001

Table 3: Dataset statistics. Each cell represents the number of examples. K: thousand, M: million.

4.2 Experimental Setting

We apply TRANSFORMER-*Base* (Vaswani et al., 2017) setting for all experiments. The model dimensionality is 512, and 6 layers are engaged in both encoder and decoder side. The inner-connection dimensionality for feedforward block is 2,048, and the number of heads in multi-head attention networks is 8. We share the source embedding, target embedding and target softmax projection weight for En-Vi task, and share the latter two matrices for En-De. We modify the learning rate schedule as: $lr = 0.001 \cdot \min\left(\frac{t}{8000}, 1, (\frac{20000}{t})^{0.5}\right)$, where t denotes the current step. Across all tasks, we determine the threshold τ as 1.0.

For both baseline and our model, En-Vi, En-De and Zh-En tasks take 50k, 150k and 200k updates, and each batch contains 4,096, 32,768 and 32,768 source tokens. The dropout ratio is determined as 0.3, 0.1 and 0.1, respectively. All experiments are conducted over 4 NVIDIA V100 GPUs. For each task, we choose the best model over dev set, defining beam size as 4, 4, 10 and decoding alpha as 1.5, 0.6 and 1.5, respectively.

4.3 Experimental Results

As shown in Table 4, vanilla model achieves best performance over all translation tasks. However, replacing conventional attention networks with E-ATT does not lead to significant performance drop, with small decrease of 0.15∼0.78 BLEU score. Besides, after referring the statistics from Table 1 and 2, our E-ATT module takes 34.10%/33.83% energy of conventional ATT. These results reveal that, E-ATT can achieve competitive translation quality, and more importantly, significantly reduce the energy consumption of attention.

4.4 Ablation Study

We further conduct ablation experiments on En-Vi task. As seen in Table 5, using discrete feature selection instead of linear transition slightly harms performance, with 0.61 BLEU score de-

Attention mechanism	**En-Vi**	**En-De**	**Zh-En**	**ASIC** (%)	**FPGA** (%)
Vanilla	30.26 ± 0.07	27.60 ± 0.04	24.28 ± 0.08	100.00	100.00
E-ATT	29.48 ± 0.08	27.45 ± 0.04	24.23 ± 0.06	**34.10**	**33.83**

Table 4: Averaged BLEU scores (%) upon test set on IWSLT'15 En-Vi, WMT'14 En-De and WMT'17 Zh-En tasks over 5 independent runs. E-ATT gives comparable results against conventional ATT, reducing the energy cost at 65.90%/66.17% on ASIC/FPGA chip in attention procedure. Since the energy cost is difficult to be empirically evaluated, we report the theoretical values following the common practice (Chen et al., 2020; You et al., 2020).

Model	**BLEU** (%)
Vanilla	**28.12**
Replace with discrete selection	27.51
Replace with L_1 distance	28.05
E-ATT	27.45

Table 5: Model performance with component replacements over En-Vi dev set. Using L_1 distance as similarity measurement dose not harm model performance.

-	Vanilla	Dense	RandInit	E-ATT
Vanilla	**28.12**	19.92	19.31	27.72
Dense	27.48	19.43	19.21	27.60
RandInit	27.36	18.98	18.83	27.48
E-ATT	28.08	19.85	19.67	27.45

Table 6: BLEU score (%) of different model hybrids with modifying self-attention (horizontal) and cross-attention (vertical) network upon En-Vi dev set. E-ATT can achieve good performance when applied as cross-attention modules, whereas Dense or RandInit can not.

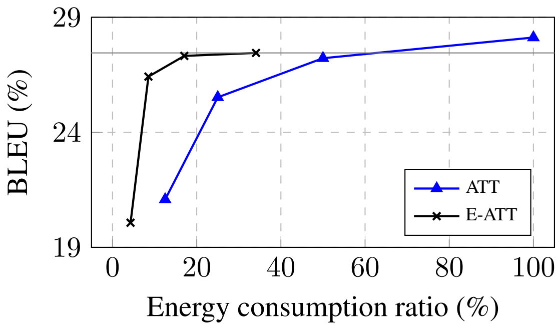

Figure 1: Performance and energy consumption of different models with knowledge distillation on En-Vi dev set. We regard the energy consumption of ATT baseline as 1, and accumulatively halve the dimensionality of model till untrainable (from 512 to 64). Energy consumption is estimated over ASIC. E-ATT requires far less energy to meet up the baseline performance.

crease. Besides, replacing dot-product attention with L_1 distance does not significantly affect model performance. We can conclude that: 1) the performance gap between E-ATT and vanilla model mainly stems from the usage of discrete feature selection; and 2) L_1 distance can measure the similarity of vectorized representations and give modest performance compared to baseline.

5 Analyses

5.1 Hybrid Attention Networks

We conduct a series of experiments involving hybrids of attention networks among vanilla ATT, Dense, RandInit, and E-ATT module in Table 6. As shown, the conventional attention network performs the best among all models. Our module performs well when served as either self-attention or cross-attention modules. Besides, for all cases applying Dense/RandInit as cross-attention modules, models perform significantly worse, identical with the findings in Tay et al. (2021). On the contrary, E-ATT module can give better performance with marginal performance drop comparing with baseline, indicating that E-ATT module is capable of providing adequate semantic alignments across languages for translation. Besides, it is encouraging to see that our method works compatibly with other modules with marginal performance drop.

5.2 Knowledge Distillation

Knowledge distillation is a representative of model compression approach (Hinton et al., 2015; Tang et al., 2019). We further conduct experiments on ATT models with various dimensionalities compressed by knowledge distillation. Figure 1 shows the energy consumption and performance of different models with modified dimensionality d. As seen, by accumulatively halving d from 512, both ATT and E-ATT progressively loses the quality. However, E-ATT can give a better trade-off between model performance and energy consumption than knowledge distillation methods.

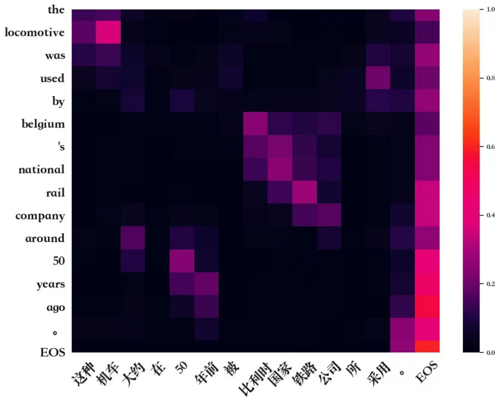

Figure 2: Case study from WMT'17 Zh-En dev set. E-ATT performs well on cross-lingual alignments.

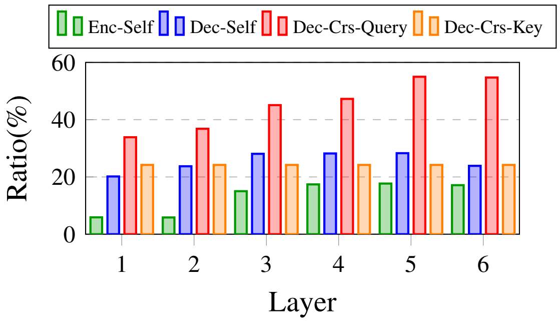

Figure 3: Ratio of nonzero values in the representations of E-ATT. Enc-Self: encoder self-attention ; Dec-Self: decoder self-attention; Dec-Crs-Query/Key: query/key representation for decoder cross-attention. Query representations in cross-attention are the most active.

5.3 Case Study

We visualize the averaged attention values over one case from WMT'17 Zh-En dev set. As seen, our model can give good aligned information, where preposition phrase "around 50 years ago" is arranged at the end of sentence in English, while its aligned phrase is at the front in Chinese. This reveals that, our E-ATT can perform well on modeling the cross-lingual alignments.

5.4 Binarization Statistics

We further collect the ratio of nonzero values ρ for each attention module in Figure 3, we can see that it increases with the number of encoder layers, denoting that more information is arranged into attentive calculation at higher layer of source side. However, for decoder E-ATT, the ratio meets its peak at middle layers, revealing that decoder E-ATT are most active at the middle term of semantic processing. Interestingly, ratio in the query of cross-attention modules, which align source and target semantics, is higher for the layer closer to output. As the binarized key representation of each cross-attention module is equivalent, higher ratio of nonzero values in query representation means that, E-ATT at higher decoder layer provides more information for cross-lingual alignments, thus enrich the information for translation.

6 Discussion and Conclusion

In this paper, we empirically investigate the high energy-consumption problem in ATT. We argue that the alignment modeling procedure can be achieved with additions other than multiplications, thus reducing the energy costs. Extensive analyses suggest that: 1) Binarized representations marginally harm the feature extraction procedure; and 2) L_1 distance can be efficiently exploited to measure alignment among queries and keys. Compared to baseline, our approach can yield considerable quality of translations, and significantly save energy in attention mechanism. Although we have shown the superiority of E-ATT, considering the whole TRANSFORMER block[2], the use of E-ATT brings 17% energy reduction. We hope this work can attract more researches on energy-efficient models. It is worth to further design techniques that reduce the energy cost of other modules in TRANSFORMER.

Acknowledgements

The authors would like to send great thanks to all reviewers and meta-reviewer for their insightful comments. This work was supported in part by the Science and Technology Development Fund, Macau SAR (Grant No. 0101/2019/A2), the Multi-year Research Grant from the University of Macau (Grant No. MYRG2020-00054-FST), and Alibaba Group through Alibaba Research Intern Program.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *International Conference on Learning Representations (ICLR)*.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. *arXiv preprint arXiv:2004.05150*.

[2] A TRANSFORMER block consists of a multi-head attention layer and a feed-forward layer. Please refer to Appendix A for more calculation details.

Hanting Chen, Yunhe Wang, Chunjing Xu, Boxin Shi, Chao Xu, Qi Tian, and Chang Xu. 2020. AdderNet: Do We Really Need Multiplications in Deep Learning? In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1468–1477.

Emily Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. 2014. Exploiting Linear Structure within Convolutional Networks for Efficient Evaluation. In *Neural Information Processing Systems (NIPS)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling The Knowledge in A Neural Network. *arXiv preprint arXiv:1503.02531*.

Mark Horowitz. 2014. 1.1 Computing's Energy Problem (and What We Can Do About It). In *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for Natural Language Understanding. In *Empirical Methods in Natural Language Processing (EMNLP): Findings*.

Nikita Kitaev and Dan Klein. 2018. Constituency Parsing with A Self-Attentive Encoder. In *Association for Computational Linguistics (ACL)*.

Zhengjie Li, Yufan Zhang, Jian Wang, and Jinmei Lai. 2020. A Survey of FPGA Design for AI Era. *Journal of Semiconductors*, 41(2):021402.

Xuan Liu, Di Cao, and Kai Yu. 2018. Binarized LSTM Language Model. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Haotong Qin, Ruihao Gong, Xianglong Liu, Xiao Bai, Jingkuan Song, and Nicu Sebe. 2020. Binary Neural Networks: A Survey. *Pattern Recognition*, 105:107281.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21:140:1–140:67.

Alessandro Raganato, Yves Scherrer, and Jörg Tiedemann. 2020. Fixed Encoder Self-Attention Patterns in Transformer-Based Machine Translation. In *Empirical Methods in Natural Language Processing (EMNLP): Findings*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Association for Computational Linguistics (ACL)*.

Dehua Song, Yunhe Wang, Hanting Chen, Chang Xu, Chunjing Xu, and DaCheng Tao. 2021. AdderSR: Towards Energy Efficient Image Super-Resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jinsong Su, Jiali Zeng, Deyi Xiong, Yang Liu, Mingxuan Wang, and Jun Xie. 2018. A Hierarchy-to-Sequence Attentional Neural Machine Translation Model. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(3):623–632.

Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi. 2018. Deep Semantic Role Labeling with Self-attention. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*.

Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019. Distilling Task-Specific Knowledge from BERT into Simple Neural Networks. *arXiv preprint arXiv:1903.12136*.

Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. 2021. Synthesizer: Rethinking Self-Attention for Transformer Models. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*.

Ashish Vaswani, 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 (NIPS)*.

Yunhe Wang, Chang Xu, Shan You, Dacheng Tao, and Chao Xu. 2016. CNNpack: Packing Convolutional Neural Networks in the Frequency Domain. In *Advances in Neural Information Processing Systems (NIPS)*.

Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. 2018. Spatio-Temporal Backpropagation for Training High-performance Spiking Neural Networks. *Frontiers in Neuroscience*, 12:331.

Baosong Yang, Jian Li, Derek F Wong, Lidia S Chao, Xing Wang, and Zhaopeng Tu. 2019a. Context-aware self-attention networks. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, pages 387–394.

Baosong Yang, Zhaopeng Tu, Derek F Wong, Fandong Meng, Lidia S Chao, and Tong Zhang. 2018. Modeling localness for self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4449–4458.

Baosong Yang, Longyue Wang, Derek F Wong, Lidia S Chao, and Zhaopeng Tu. 2019b. Convolutional self-attention networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4040–4045.

Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. 2017. A Gift from Knowledge Distillation: Fast Optimization, Network Minimization and Transfer Learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Haoran You, Xiaohan Chen, Yongan Zhang, Chaojian Li, Sicheng Li, Zihao Liu, Zhangyang Wang, and Yingyan Lin. 2020. ShiftAddNet: A Hardware-Inspired Deep Network. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Biao Zhang, Deyi Xiong, Jun Xie, and Jinsong Su. 2020. Neural Machine Translation With GRU-Gated Attention Model. *IEEE Transactions on Neural Networks and Learning Systems*, 31(11):4688–4698.

Zhuangwei Zhuang, Mingkui Tan, Bohan Zhuang, Jing Liu, Yong Guo, Qingyao Wu, Junzhou Huang, and Jinhui Zhu. 2018. Discrimination-Aware Channel Pruning for Deep Neural Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

A Energy Ratio Calculation

We calculate the energy cost following the common practice (Chen et al., 2020; You et al., 2020). Note that, we follow suggestions in Song et al. (2021) to omit the energy calculation of activate functions, such as relu and softmax, because they are specially designed over some modern AI chips, which requires far less energy than additive operation.

For the common case where input $\mathbf{X} \in \mathbb{R}^{l_1 \times d}$ is projected into $\mathbf{Q} \in \mathbb{R}^{l_1 \times d}$ with $\mathbf{W} \in \mathbb{R}^{d \times d}$:

$$\mathbf{Q} = \mathbf{X}\mathbf{W}^\top + b, \tag{11}$$

the number of multiplicative operations is $l_1 \times d \times d + l_1 \times d = l_1 d^2 + l_1 d$. For the number of additive operations, it is also $l_1 d^2 + l_1 d$. Specially, if $d \gg 1$, we can omit the term $l_1 d$ for simplicity.

We calculate the energy cost from three levels: the alignment calculation which is used to measure the attention score, the whole attention model which is the core of our work, and the widely used TRANSFORMER (Vaswani et al., 2017) block, which contains a multi-head attention layer and a feedforward layer. We simply set the sequence length of inputs $\mathbf{X}$ and $\mathbf{Y}$ to l. Then:

Calculation Level	Δ_A	Δ_F
Alignment Calculation	0.45	0.05
Attention Model	34.09	33.83
TRANSFORMER Block	83.17	83.10

Table 7: Energy consumption ratio (%) at each calculation level on En-Vi translation task compared to conventional TRANSFORMER design.

Alignment Calculation Two linear prjections are arranged to obtain query and key representations, yielding $2ld^2$ additive/multiplicative operations. Both query and key representations are used to derive attention logits. In dot-product, the number of required multiplicative operations is $l \times d \times l = l^2 d$. The total numbers of additive/multiplicative operations are both $2ld^2 + l^2 d$.

Attention Model In order to obtain value representations, attention model requires ld^2 additive/multiplicative operations. Besides, applying weighted sum over value representations with attention weights requires $l^2 d$ multiplicative/additive operations. Overall, the numbers of multiplicative/additive operations in the whole attention model are $3ld^2 + 2l^2 d$.

TRANSFORMER Block Although representations in multi-head attention are splitted into h heads, in which dimension is d_h ($d = h \times d_h$), the number of multiplicative/additive operations is also $h \times l \times d_h \times l = l^2 h d_h = l^2 d$. There are one output linear transition for the output of multiple heads and two additional linear transitions in feedforward layer, resulting in $ld^2 + 4ld^2 + 4ld^2 = 9ld^2$ additional additive/multiplicative operations. The overall multiplication operations in a TRANSFORMER Block is $9ld^2 + 3ld^2 + 2l^2 d = 12ld^2 + 2l^2 d$.

Same as the steps above, we can calculate the required energy consumption of other modules. For example, considering our proposed E-ATT, the numbers of additive and multiplicative operations are $2ld + l^d$ and 0 in alignment calculation, $ld^2 + 2ld + 2l^2 d$ and $ld^2 + l^2 d$ in attention model, $10ld^2 + 2ld + 2l^2 d$ and $10ld^2 + l^2 d$ in TRANSFORMER block.

In this way, we can calculate the ratio of energy cost referring to the statistics in Table 1. For example, the ratio between E-ATT and conventional attention model on ASIC chip is:

$$\Delta_A = \frac{0.9(ld^2 + 2ld + 2l^2d) + 3.7(ld^2 + l^2d)}{0.9(3ld^2 + 2l^2d) + 3.7(3ld^2 + 2l^2d)} \tag{12}$$

Similarly, that on FPGA chip is:

$$\Delta_F = \frac{0.4(ld^2 + 2ld + 2l^2d) + 18.8(ld^2 + l^2d)}{0.4(3ld^2 + 2l^2d) + 18.8(3ld^2 + 2l^2d)} \tag{13}$$

For the IWSLT'15 En-Vi task, with d being 512, the averaged length of dataset is $\bar{l} = 22$. We can get the result Δ_A = 34.09%, Δ_F = 33.83%, thus the energy reduction ratio is $1 - \Delta_A$ = 65.91%, $1 - \Delta_F$ = 66.17%. Similarly, energy consumption ratios at the level of alignment calculation are 99.55% and 99.95%. Those of TRANSFORMER block are 83.17% and 83.10%, respectively.