

ProTran: Profiling the Energy of Transformers on Embedded Platforms

Shikhar Tuli¹, *Student Member, IEEE*, and Niraj K. Jha, *Fellow, IEEE*

Abstract—The abstract goes here.

Index Terms—Embedded platforms, machine learning, transformers.

I. INTRODUCTION

IN recent years, self-attention-based transformer models [1, 2] have achieved state-of-the-art results on tasks that span the natural language processing (NLP), and recently, even the computer vision domain [3]. Increasing computational power and large-scale pre-training datasets has resulted in an explosion in the architecture size of transformer models [4], much beyond the state-of-the-art convolutional neural networks (CNNs). For instance, the Megatron Tuning-NLG [4] has 530B trainable parameters compared to only 928M trainable model parameters in BiT (which uses ResNet-152 with every hidden layer widened by a factor of four, i.e., ResNet-152x4) [5, 6]. However, such massive transformer architectures are not amenable to be run on mobile edge devices, due to a much lower compute budget and memory size.

Even if such a large model is run on these mobile devices, it would incur an extremely high latency. Smaller models may have reasonable latencies, however, they may still not meet the energy or peak power budget for running on edge devices. This could be due to a limited battery size or an intermittent power supply. Thus, there is a need for profiling and benchmarking the latency, energy and peak power consumption of a diverse set of mobile-friendly transformer architectures. This would aid frameworks to leverage hardware-aware neural architecture search (NAS) [7, 8] techniques to find the optimal architecture that maximizes model accuracy while meeting latency, energy and peak power budgets.

Several works have tried to prune transformer models to reduce the number of parameters [9, 10]. Some have also proposed novel attention mechanisms to reduce the number of trainable parameters [11, 12, 13]. Others have run NAS in a design space of transformer architectural hyperparameters to obtain efficient architectures [7, 14, 15]. However, most of these works have only shown gains in the number of model parameters or the number of floating-point operations per second (FLOPs). Such works do not consider latency, energy and power consumption in their optimization loop, while searching for the optimal transformer architecture (Wang

et al. [7] only consider latency for running around 2000 transformer architectures on certain edge devices, and Li et al. [16] consider only a single FPGA). Thus, there is a need to profile not only the accuracy [8], but also the latency, energy and peak power consumption of transformer models on various mobile devices for inclusive design in edge-AI.

Nevertheless, profiling all models in vast design spaces is a challenging endeavor. Hence, in this work, we make the following contributions:

- We implement the FlexiBERT benchmarking framework with its design space of diverse transformer architectures on multiple edge-AI devices, for both training and inference. We measure the latency, energy consumption and peak power draw for the transformer models in the design space. We call this profiling framework that can obtain all hardware measures for a design space of transformer architectures as ProTran.
- We leverage ProTran to train a surrogate model that exploits gradient-based optimization using backpropagation of inputs and heteroscedastic modelling [8, 17] to minimize the overall uncertainty in estimation of each measure in our active learning loop.
- We then leverage the surrogate models along with previously proposed performance predictors [8] to run hardware-aware NAS that gives equivalent performing models in terms of accuracy, but with much lower latency, energy consumption and peak power draw.

The rest of the article is organized as follows. Section II discusses the background and related work in hardware-aware NAS, pruning methods and profiling of transformer models. Section III motivates the need for the ProTran framework using a toy example in our design space.

II. BACKGROUND AND RELATED WORK

This section introduces the relevant background and related works in the fields of hardware-aware NAS, pruning methods and profiling of transformer architectures.

A. Transformer Architectures

Various transformer architectures have been proposed in the past. BERT is one of the most popular transformer architectures that is widely used for language modeling [2]. Its variants leverage mechanisms other than the vanilla self-attention [18] to either optimize performance or reduce model size and complexity. These include – RoBERTa [19] that implements robust pre-training techniques, ConvBERT [20] that uses one-dimensional convolutional operations, MobileBERT [21] that

This work was supported by NSF Grant No. —. S. Tuli and N. K. Jha are with the Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ, 08544, USA (e-mail: {stuli, jha}@princeton.edu).

Manuscript received —; revised —.

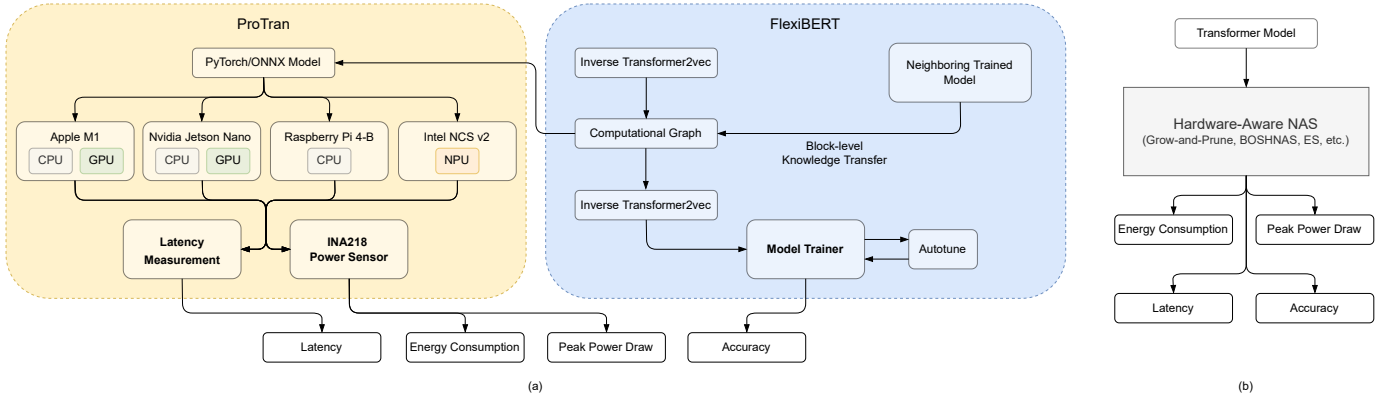


Fig. 1: Overview of the ProTran framework.

uses bottleneck structures and multiple feed-forward stacks, SqueezeBERT [13] that uses grouped convolution operations to approximate the feed-forward stack, etc. Further, architectures like FNet [11] and LinFormer [12] use the Fourier transform or a low-rank approximation of the self-attention operation to aid efficiency.

Many works have tried to device design spaces in order to search for optimal architectural design decisions in a unified manner. For instance, SchuBERT proposes a design space of transformer architectures [22]. However, this work does not consider different types of attention operations and only has *homogenous* models in its design space. In this work, we leverage FlexiBERT, a state-of-the-art benchmarking framework for diverse transformer architectures, which incorporates most popularly used attention operations in a design space of *heterogeneous* and *flexible* transformer architectures [8]. We model the latency, energy and peak power for transformer architectures on a diverse set of embedded platforms, thanks to the FlexiBERT platform.

B. Hardware-Aware NAS

NAS basically searches for the architecture which attains the best accuracy on a specified dataset. However, NAS alone is hardly of much use if the best performing transformer cannot be run on the hardware at hand (or does not meet the hardware performance constraints). Hence, recent works have focused on hardware-aware NAS which directs the architecture search for a target platform. ChamNet proposed accuracy and resource (latency and energy) predictors, and leveraged GP-BO to find the optimal CNN architecture for a given platform [23]. Some works have also proposed *simultaneous* co-design of the hardware and the software design decisions [24, 25, 26]. However, these works are only limited to CNN design spaces.

HAT [7], a recent framework for hardware-aware NAS, trains a large transformer model first and then uses latency feedback to obtain a sub-model for the given hardware platform. However, all sub-models are *homogeneous* and have constant dimensionality on each encoder layer. Further, this work uses a static training recipe, which may not be optimal for every sub-model. Lastly, the design space is highly restricted, which has shown to lead to lower gains [26]. Instead,

other NAS techniques can be leveraged for superior efficient search of the optimal model in a diverse set of transformer architectures [8, 23, 27]. Fig. 1 shows how ProTran leverages the FlexiBERT framework to implement block-level grow-and-prune of transformer architectures to obtain the best-performing model in terms of accuracy, latency, energy and peak power consumption.

C. Energy Profiling of Transformer Models

Profiling the energy consumption of any ML model is a challenging task. This is because, extracting the energy consumed only by the processes of training or running inference for an ML model is non-trivial. Nevertheless, many previous works have profiled the energy consumption of ML architectures. ChamNet trains predictors for the energy consumption of various CNNs in its design space, on different hardware platforms under different latency constraints [23]. FTRANS shows the energy and power consumption for different transformer architectures on an FPGA [16]. However, no NAS approach on transformer architectures has accounted for energy and peak power consumption in the past [7, 8]. Thus, there is a need for light-weight surrogate models for energy and peak power estimation of a diverse set of transformer architectures on various edge-AI platforms. This would not only aid energy-aware NAS for transformer models, but also efficient co-design for optimal edge deployments.

III. MOTIVATION

This section motivates the need for rigorous profiling of transformer architectures on diverse edge-AI platforms and the development of light-weight surrogate models for various hardware performance measures.

A. Energy Reduction on Mobile Platforms

Fig. ?? plots the power consumption for running model inference for BERT-Tiny [28] on the SST-2 task [29], on different hardware platforms. Fig. ??(a) and (b) show the central processing unit (CPU), graphics processing unit (GPU) and dynamic random access memory (DRAM) power consumption on an iPad (with the Apple M1 SoC) [30]. These energy

profiles can be compared against that of a regular GPU in Fig. ??(c) (we use Nvidia A100 for the baseline GPU). These figures show that the Apple M1 SoC has much lower power consumption throughout its operation when compared to the A100.

As can be seen from Fig. ??, the Apple M1 SoC on its integrated GPU outperforms the traditional A100 GPU in terms of energy and peak power consumption, and is also close in terms of latency. This motivates the need for profiling the latency, energy and peak power consumption of various transformer architectures on a diverse set of embedded platforms. This would aid efficient design of transformer architectural design decisions for different edge deployments.

B. Challenges and Proposed Solutions

The example shown in Fig. ?? only runs inference of a small transformer architecture. For larger architectures, and even for training, much more memory is required. Such large models (e.g. BERT-Large) may not fit into the memory of the integrated CPU or GPU. To counter this, gradient accumulation over multiple subsets of the mini-batch could be exploited, however, at the cost of increased latency. A challenge that remains is to efficiently search for optimal transformer architectures that can fit into the memory while also not compromising on accuracy. For this, different approaches can be experimented, including NAS, grow-and-prune, etc. Further, quantization and reduction of the activations inside the model could be explored for alleviate memory bottlenecks [7].

REFERENCES

- [1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 5998–6008. [Online]. Available: <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>
- [2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, 2019, pp. 4171–4186.
- [3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proceedings of the International Conference on Learning Representations*, 2021. [Online]. Available: <https://openreview.net/forum?id=YicbFdNTTy>
- [4] S. Smith, M. Patwary, B. Norick, P. LeGresley, S. Rajbhandari, J. Casper, Z. Liu, S. Prabhunoye, G. Zerveas, V. Korthikanti, E. Zheng, R. Child, R. Y. Aminabadi, J. Bernauer, X. Song, M. Shoybi, Y. He, M. Houston, S. Tiwary, and J. Sun, "Using deepspeed and megatron to train megatron-turing NLG 530b, A large-scale generative language model," *CoRR*, vol. abs/2201.11990, 2022. [Online]. Available: <https://arxiv.org/abs/2201.11990>
- [5] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby, "Big Transfer (BiT): General visual representation learning," in *European Conference on Computer Vision*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., 2020, pp. 491–507.
- [6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [7] H. Wang, Z. Wu, Z. Liu, H. Cai, L. Zhu, C. Gan, and S. Han, "HAT: Hardware-aware transformers for efficient natural language processing," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Jul. 2020, pp. 7675–7688. [Online]. Available: <https://aclanthology.org/2020.acl-main.686>
- [8] Anonymous, "FlexiBERT: Expanding the flexibility of transformer architectures and exploring a heterogeneous design space," *International Conference on Machine Learning*, 2022, in review.
- [9] M. A. Gordon, K. Duh, and N. Andrews, "Compressing BERT: studying the effects of weight pruning on transfer learning," *CoRR*, vol. abs/2002.08307, 2020. [Online]. Available: <https://arxiv.org/abs/2002.08307>
- [10] Z. Yan, H. Wang, D. Guo, and S. Han, "Micronet for efficient language modeling," in *Proceedings of the NeurIPS 2019 Competition and Demonstration Track*, ser. *Proceedings of Machine Learning Research*, H. J. Escalante and R. Hadsell, Eds., vol. 123. PMLR, 08–14 Dec 2020, pp. 215–231. [Online]. Available: <https://proceedings.mlr.press/v123/yan20a.html>
- [11] J. Lee-Thorp, J. Ainslie, I. Eckstein, and S. Ontañón, "FNet: Mixing tokens with Fourier transforms," *CoRR*, vol. abs/2105.03824, 2021. [Online]. Available: <https://arxiv.org/abs/2105.03824>
- [12] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, "Linformer: Self-attention with linear complexity," *CoRR*, vol. abs/2006.04768, 2020. [Online]. Available: <https://arxiv.org/abs/2006.04768>
- [13] F. Iandola, A. Shaw, R. Krishna, and K. Keutzer, "SqueezeBERT: What can computer vision teach NLP about efficient neural networks?" in *Proceedings of SustainNLP: Workshop on Simple and Efficient Natural Language Processing*. Online: Association for Computational Linguistics, Nov. 2020, pp. 124–135. [Online]. Available: <https://aclanthology.org/2020.sustainlp-1.17>
- [14] J. Xu, X. Tan, R. Luo, K. Song, J. Li, T. Qin, and T.-Y. Liu, "NAS-BERT: Task-agnostic and adaptive-size BERT compression with neural architecture search," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery Data Mining*, 2021, p. 1933–1943.
- [15] Y. Yin, C. Chen, L. Shang, X. Jiang, X. Chen, and Q. Liu, "AutoTinyBERT: Automatic hyper-parameter optimization for efficient pre-trained language models," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 5146–5157. [Online]. Available: <https://aclanthology.org/2021.acl-long.400>
- [16] B. Li, S. Pandey, H. Fang, Y. Lyv, J. Li, J. Chen, M. Xie, L. Wan, H. Liu, and C. Ding, "FTRANS: Energy-efficient acceleration of transformers using fpga," in *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design*, ser. ISLPED '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 175–180. [Online]. Available: <https://doi.org/10.1145/3370748.3406567>
- [17] S. Tuli, S. R. Poojara, S. N. Srirama, G. Casale, and N. R. Jennings, "COSCO: Container orchestration using co-simulation and gradient based optimization for fog computing environments," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 1, pp. 101–116, 2021.
- [18] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 2, 2018, pp. 464–468.
- [19] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019. [Online]. Available: <http://arxiv.org/abs/1907.11692>
- [20] Z.-H. Jiang, W. Yu, D. Zhou, Y. Chen, J. Feng, and S. Yan, "ConvBERT: Improving BERT with span-based dynamic convolution," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 12 837–12 848. [Online]. Available: <https://proceedings.neurips.cc/paper/2020/file/96da2f590cd7246bbde0051047b0d6f7-Paper.pdf>
- [21] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou, "MobileBERT: A compact task-agnostic BERT for resource-limited devices," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Jul. 2020, pp. 2158–2170. [Online]. Available: <https://aclanthology.org/2020.acl-main.195>
- [22] A. Khetan and Z. Karnin, "schuBERT: Optimizing elements of BERT," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Jul. 2020, pp. 2807–2818. [Online]. Available: <https://www.aclweb.org/anthology/2020.acl-main.250>
- [23] X. Dai, P. Zhang, B. Wu, H. Yin, F. Sun, Y. Wang, M. Dukhan, Y. Hu, Y. Wu, Y. Jia, P. Vajda, M. Uyttendaele, and N. K. Jha, "Chamnet: Towards efficient network design through platform-aware model adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [24] Y. Lin, M. Yang, and S. Han, "NAAS: Neural accelerator architecture search," *CoRR*, vol. abs/2105.13258, 2021.
- [25] J. Lin, W.-M. Chen, Y. Lin, J. Cohn, C. Gan, and S. Han, "MCUNet:

- Tiny deep learning on IoT devices,” in *Proc. Int. Conf. Neural Information Processing Systems*, vol. 33, 2020, pp. 11 711–11 722.
- [26] S. Tuli, C. H. Li, R. Sharma, and N. K. Jha, “CODEBench: A neural architecture and hardware accelerator co-design framework,” in *Proc. Trans. Computer-Aided Design of Integrated Circuits and Systems*, 2022, in review.
 - [27] X. Dai, H. Yin, and N. K. Jha, “NeST: A neural network synthesis tool based on a grow-and-prune paradigm,” *IEEE Trans. Computers*, vol. 68, no. 10, pp. 1487–1497, 2019.
 - [28] I. Turc, M. Chang, K. Lee, and K. Toutanova, “Well-read students learn better: The impact of student initialization on knowledge distillation,” *CoRR*, vol. abs/1908.08962, 2019. [Online]. Available: <http://arxiv.org/abs/1908.08962>
 - [29] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, “GLUE: A multi-task benchmark and analysis platform for natural language understanding,” in *Proceedings of the EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Nov. 2018, pp. 353–355. [Online]. Available: <https://aclanthology.org/W18-5446>
 - [30] Apple. (2020) Apple unleashes M1. [Online]. Available: <https://www.apple.com/newsroom/2020/11/apple-unleashes-m1/>