Tempo: Accelerating Transformer-Based Model Training through Memory Footprint Reduction

Muralidhar Andoorveedu[1], Zhanda Zhu[2,3], Bojian Zheng[1,3], Gennady Pekhimenko[1,3]

1 UNIVERSITY OF TORONTO

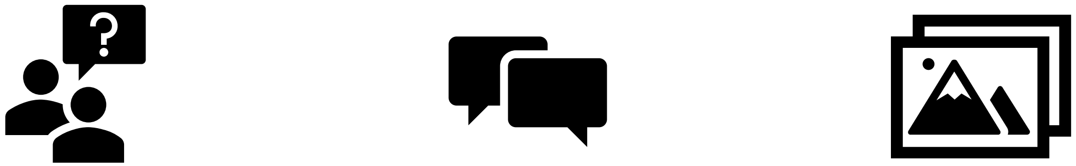
2 上海交通大學 SHANGHAI JIAO TONG UNIVERSITY

3 VECTOR INSTITUTE

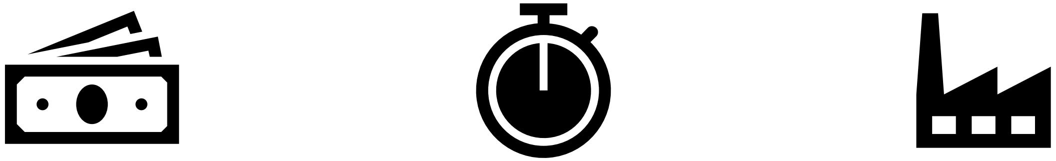
https://github.com/UofT-EcoSystem/Tempo

Problem Overview

- Transformer-based models [1] are increasingly relevant to tasks such as question answering, paraphrasing, and even image processing [2, 3, 4].



- However, training Transformer-based models is also expensive [5, 6]!



- There is a strong incentive to reduce the training time of these models.

Potential Angle: Look at the Memory Footprint!

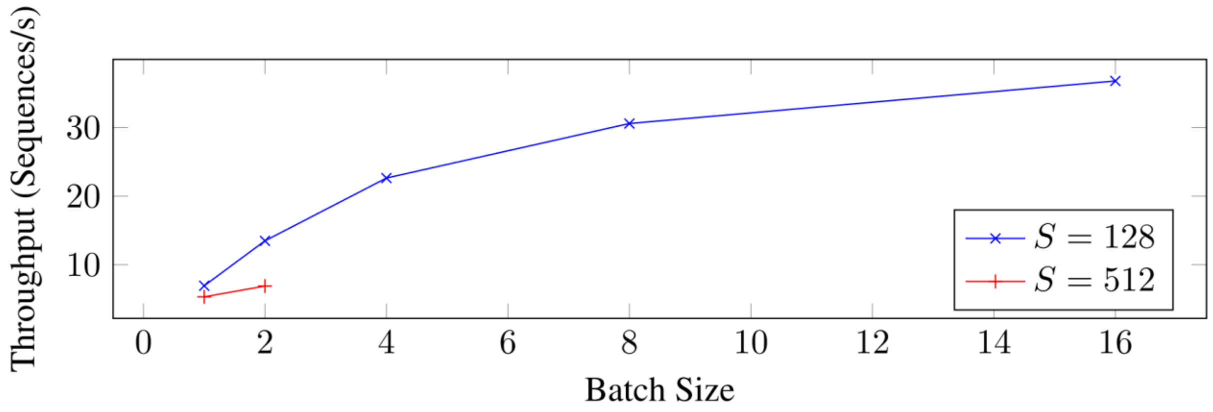
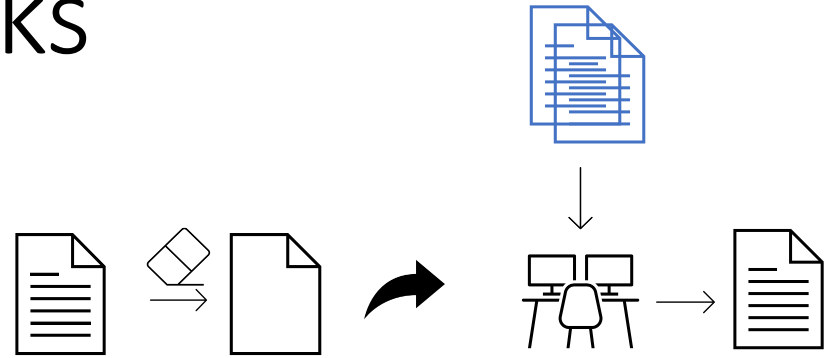


Figure showing the maximum batch size is 2 for BERT Large on a 2080Ti at S=512

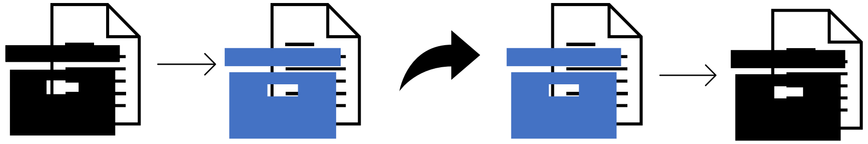
- Increasing the batch size can improve GPU compute utilization [7].
- Activation memory is the main contributor to the memory footprint compared to parameters, gradients, and optimizer states [8].

Overview of Prior Works

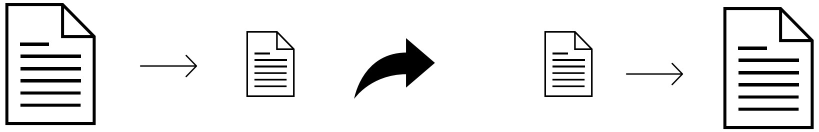
- Checkpointing
  - Checkmate [9]
  - Sublinear Memory Cost [10]



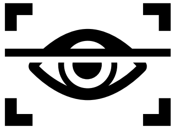
- Offloading
  - vDNN [11]
  - Capuchin [12]



- Compression/Quantization
  - ActNN [13]

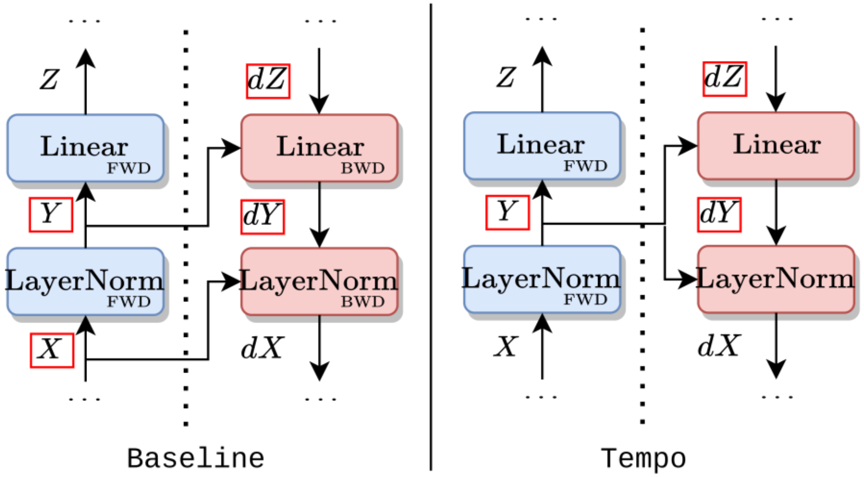


- CNN Specific
  - Gist [14]
  - In-place ABN [15]



Tempo Techniques

- Tempo applies Transformer-specific optimizations that are missed by general techniques
- In-place GELU (3)
- In-place LayerNorm (2)
  - Alternative derivation for the backward pass
- Sub-Layer Dropout Recomputation (1)



Retained activations for the LayerNorm backward pass on the Baseline and Tempo.

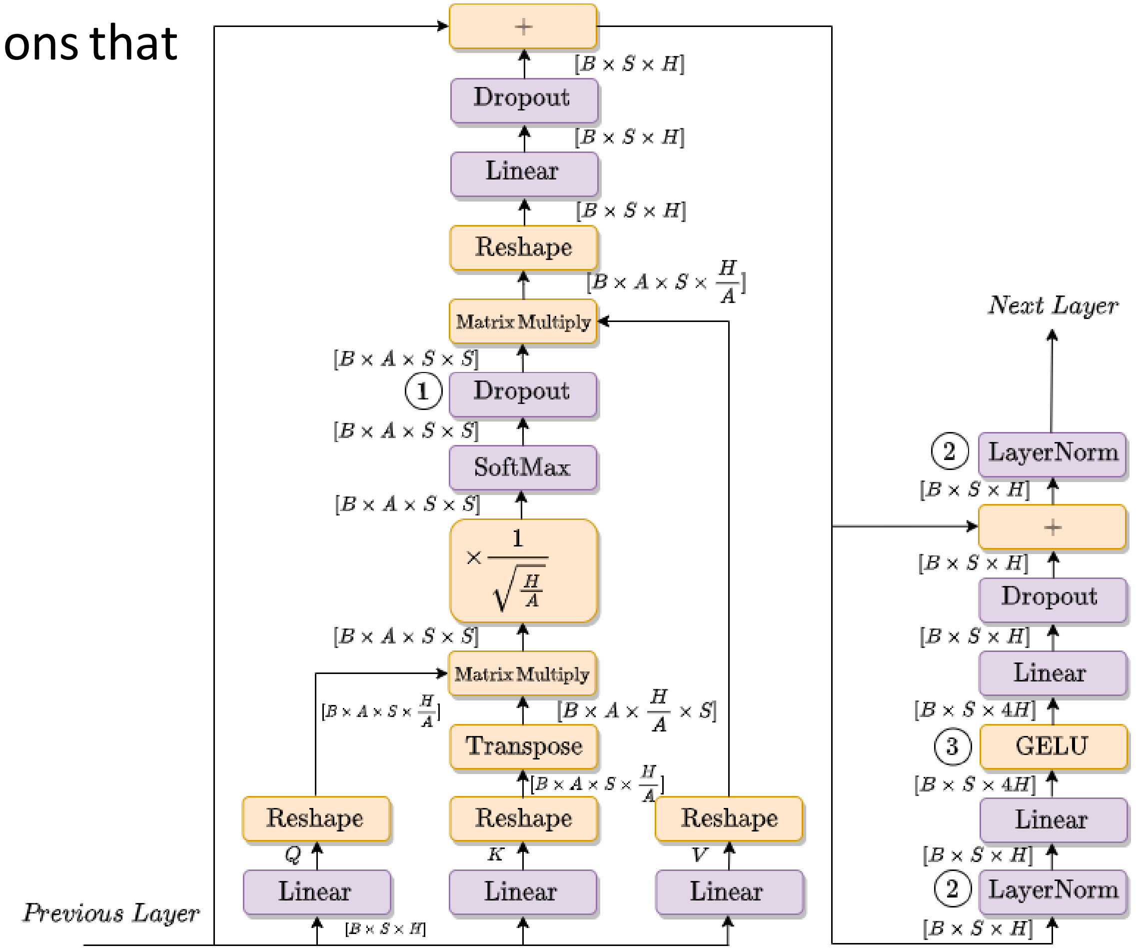
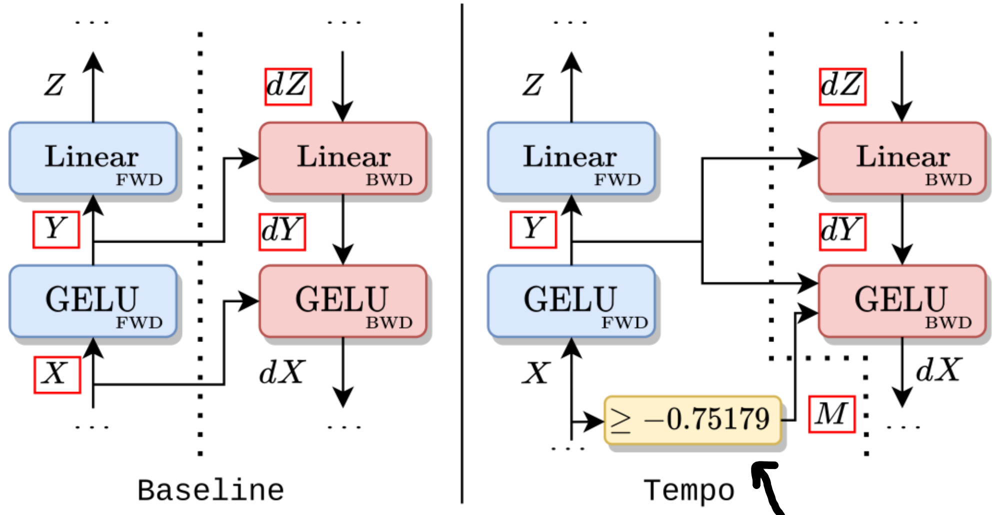


Diagram of a BERT [16] encoder layer with sizes of intermediates. The points at which our method is applied is annotated.

In-place GELU

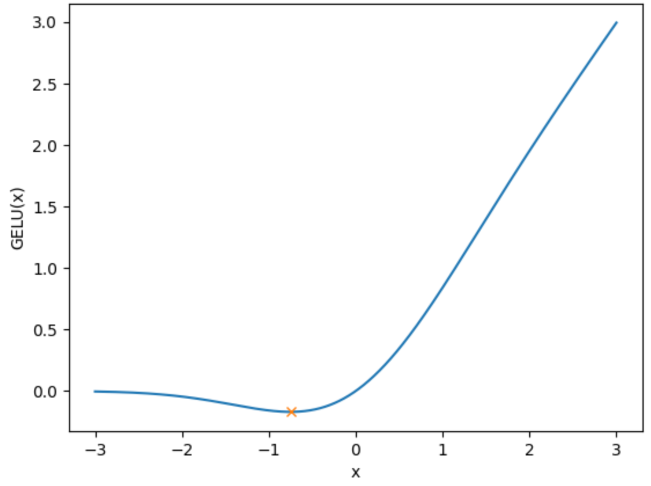
- 4 Key Ideas
  - Invert GELU operator to avoid storing X
  - Compose inverse with regular backward gradient calculation to "fuse kernels"
  - Use polynomial approximation since there is no nice form for this composite function
  - Store a mask bit since it is not bijective



Retained activations for the GELU backward pass on the Baseline and Tempo.

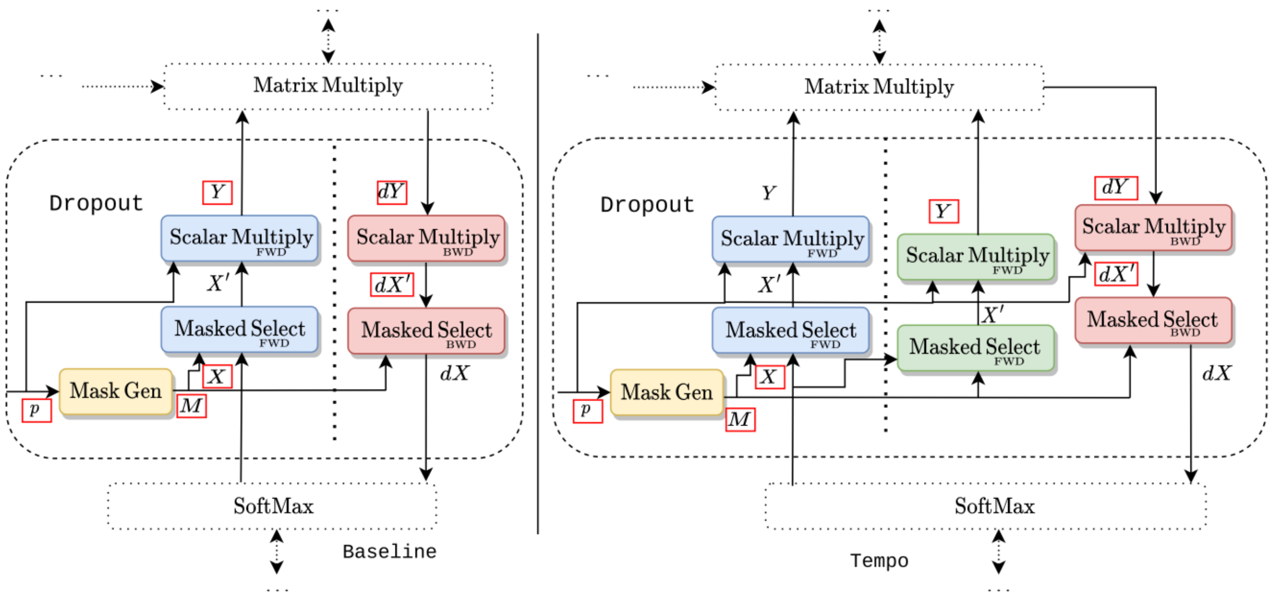
Generates boolean for invertibility

Inverse + Derivative



Graph of the GELU [17] function with minimum point indicated.

Sub-Layer Dropout Recomputation



Recomputation of Y inside the Dropout layer.

- Attention memory is quadratic in the sequence length
- Can quickly recompute Y through cheap operations
- Saves a large amount of memory with minimum overhead

Results

- **2x and 1.5x** batch size increase vs. Baseline on BERT Large at a Sequence Length of 512 on 2080Ti and V100 GPUs respectively.
- **16% and 5%** improvement in throughput for these configurations.
- **39%** improvement on BERT Base modified to use a Hidden Layer Size of 3072 at a Sequence Length of 512 on an A100
- **27%** improvement on BERT Large modified to use a Sequence Length of 1024 and 12 Layers on an A100
- Up to **19% and 26%** improvement on 2080Ti for GPT2 [17] and RoBERTa [18] respectively.

Hardware

Models

Sequence Lengths

Hidden Layer Sizes

Conclusion

- Transformer training requires more efficient training
- Activation memory footprint reduction can improve training performance
- Tempo is a method that takes advantage of Transformer-based model specifics, improving performance for a low-cost compared to existing works
- Results show improvement across a variety of different parameters.

References

[1]	**Ashish Vaswani et al., "Attention is All you Need", In Advances in Neural Information Processing Systems (NeurIPS), 2017.**
[2]	Pranav Rajpurkar et al., "SQuAD: 100,000+ questions for machine comprehension of text", In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2016.
[3]	William B. Dolan and Chris Brockett, "Automatically constructing a corpus of sentential paraphrases", In Proceedings of the Third International Workshop on Paraphrasing (IWP), 2005.
[4]	Alexey Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", In 9th International Conference on Learning Representations, (ICLR), 2021.
[5]	Andrei Ivanov et al., "Data Movement Is All You Need: A Case Study on Optimizing Transformers", In Proceedings of Machine Learning and Systems (MLSys), 2021.
[6]	Emma Strubell, Ananya Ganesh, and Andrew McCallum, "Energy and Policy Considerations for Deep Learning in NLP", In Proceedings of the 57th Conference of the Association for Computational Linguistics, (ACL), 2019.
[7]	P. Goyal et al., “Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour”, CoRR, 2017.
[8]	H. Zhu et al., “Benchmarking and Analyzing Deep Neural Network Training”, in 2018 IEEE International Symposium on Workload Characterization (IISWC), 2018.
[9]	P. Jain et al., “Checkmate: Breaking the Memory Wall with Optimal Tensor Rematerialization”, in Proceedings of Machine Learning and Systems (MLSys), 2020.
[10]	T. Chen, B. Xu, C. Zhang, and C. Guestrin, “Training Deep Nets with Sublinear Memory Cost”, CoRR, 2016.
[11]	M. Rhu, N. Gimelshein, J. Clemons, A. Zulfiqar, and S. W. Keckler, “vDNN: Virtualized Deep Neural Networks for Scalable, Memory-Efficient Neural Network Design”, in The 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2016.
[12]	X. Peng et al., “Capuchin: Tensor-Based GPU Memory Management for Deep Learning”, in Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), 2020.
[13]	J. Chen et al., “ActNN: Reducing Training Memory Footprint via 2-Bit Activation Compressed Training”, in International Conference on Machine Learning (ICML), 2021.
[14]	A. Jain, A. Phanishayee, J. Mars, L. Tang, and G. Pekhimenko, “Gist: Efficient Data Encoding for Deep Neural Network Training”, in International Symposium on Computer Architecture (ISCA), 2018.
[15]	S. Rota Bulò, L. Porzi, and P. Kontschieder, “In-Place Activated BatchNorm for Memory-Optimized Training of DNNs”, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
[16]	Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding", In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019.
[17]	D. Hendrycks and K. Gimpel, “Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units”, CoRR, vol abs/1606.08415, 2016.
[18]	Yinhan Liu, et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach", CoRR, 2019
[19]	Alec Radford et al., "Language models are unsupervised multitask learners", OpenAI blog, 2019.