Paper #692 Author Feedback

Thank you very much for the thorough reviews. We respond to each comments below.

Responses to Reviewer #1:

> 4.4 "If the solution to the time-centric strategy does not exist, then try the memory-centric strategy next to prioritize memory reduction." I don't understand this. Time-centric refers to the optimal strategy. If that does not exist, that means a solution does not exist. Then why try the other strategy?

It seems that we used a misleading phrasing here and we owe an apology. By the phrase "the solution of the time-centric strategy does not exist," we meant to refer to the case in which the time-centric strategy cannot satisfy the memory budget constraint *even after the application of liveness analysis* — we did not mean the case in which the DP solution does not exist. In such a case of memory shortage, what we suggest is to do the opposite of the objective and prioritize the memory; conduct the memory-centric strategy that maximizes the time, and implement the solution with liveness analysis. This compromise worked well in practice. We will rephrase this part in the revision.

> The runtime of the DP algorithm itself is not mentioned.

Please see Section 5.1; "The exact DP algorithm required more than 80 secs to complete for GoogLeNet and PSPNet, while the approximate DP completed within 1 sec for all networks."

> Is this feasible to be applied on the fly for every minibatch in a dynamic-network setting? For example, a Transformer MT model or lattice-free MMI in speech recognition, where each batch has different input/output lengths.

In this paper, we consider only static graphs. However, as future work, we may extend our algorithm to the dynamic setting by, for example, conducting our algorithm in advance to the set of computation graphs that might become necessary in the course of training. If the variable shape changes over the dataset, we may use maximum shape to develop a computation strategy. We will include this discussion as future work in the revised conclusion.

Responses to Reviewer #2:

> Would like to see some comparisons for sequence models (LSTMs) etc with the relevant work in that category.

If possible, we will try to include the additional experimental results in the revision.

> The directed graph approach works for many models, including "unrolled" sequence models, however for models including loop based sequences it may require some modifications to this approach. I believe it should still work though. The paper would be better if that was covered.

If the number of times the signal goes through each loop is fixed, we can unroll the loop by a simple manipulation on a computational graph. Then, we can apply our algorithm directly. As we mention in our response to Reviewer #1, it may be possible to modify our algorithm to extend the scope of applications. We plan to explore these modifications in future works further.

Responses to Reviewer #3:

> Given that the proposed algorithm generalize beyond what Chen's algorithm can do, I would recommend the authors to include experiments on models that cannot be handled in Chen's algorithm. This will help to strengthen paper.

As we show in the experiment, our algorithm greatly outperforms Chen's algorithm on PSPNet and U-Net in terms of memory consumption, and our method can reduce the computational overhead more greatly than Chen's algorithm when the memory resource is ample. Chen's algorithm is particularly not well-suited to U-Net; because of skip connections, there will always be a massive block in the decomposition of the computation graph. We plan to add a figure to visualize this explanation in the revision.

> I want to point out that there are related treatment of using a tree decomposition https://medium.com/tensorflow/fitting-larger-networks-into-memory-583e3c758ff9 While I know we are not supposed to treat a blog post as existing literature since blogs are not peer-reviewed, the authors should still try to discuss it and give pointers to the related works.

We will make a pointer to the blog-post as an example of the implementation of Chen's algorithm and mention its ideas. We would humbly like to ask the reviewer, however, to recognize that our work is the first work in the community to investigate the algorithm applicable to general graph with appropriate formality and to experimentally verify its efficacy. We also plan to publish the implementation upon the publication.