First of all, we would like to thank all reviewers for the feedback and questions.

=== Reviewer #1 ===

Q: “The idea of shrinking the network is somewhat not novel ...”
A: Indeed, shrinking spatial dimension is not an entirely new idea. However, despite its prevalence in ConvNets, this idea hasn’t not been successfully applied in NLP, especially under the context of pretraining where (1) the model capacity and scalability are critical, but (2) reducing the length could harm the capacity. Thus, this work can be seen a proof of concept such a trade-off can be beneficial even under the context of pretraining.

Q: “The computational efficiency was evaluated by FLOPs.”
A: We agree FLOPs don’t tell the whole story. Therefore, in addition to FLOPs, we indeed compare the exact running time in the Appendix C.3, which also shows the efficiency gain of the proposed model.

=== Reviewer #2 ====

Q: “The idea of this method is interesting, but seems a little complex... it is difficult to decide the optimal block layout”
A: This is indeed a difficulty future work needs to consider. Also, the optimal layout could be problem dependent. Hence, a more elegant solution in our opinion should enable some simple layout finetuning mechanism “after pretraining”.

Q: “have you tried to fine-tune the encoder plus decoder for classification task?”
A: Yes. We tried to finetune the encoder + decoder version on the GLUE benchmark with B6-6-6H768 pretrained in large-scale setting. Out of 8 tasks, using the decoder only improves the performances on 4 tasks, leading to about 0.3 points gains on average compared to the version without decoder, which is not that significant considering the cost.

==== Reviewer #3 ====

Q: “I would like to see how the model performs on natural language generation tasks”:
A: The reason why we focus mostly on language understanding tasks is that they are arguably more influenced by the success of language pretraining, which requires significant computation. But we do agree that how to apply this idea (enabling operations on the sequence length) to generation tasks is definitely a very interesting future direction.

Q: “How does the technique scale to longer sequences?”
A: We tried to pretrain a proposed model with sequence length $T = 1024$ and $D = 1024$. For relative pretraining loss and speed, we observe very similar patterns with $T = 512$ and $D = 1024$. But due to the much longer total time, we didn’t finish the pretraining nor finetuning. In our opinion, “sequence compression” is a necessary component for handling super-long sequences, just like how humans handle such cases.

==== Reviewer #4 ====

Q: “Why are the training speeds of Funnel-Transformer faster than standard Transformer”
A: Firstly, the complexity of a standard Transformer layer to process of sequence of length T and hidden dimension D is $O(T^2D + TD^2)$. Hence, by reducing the length from T to a shorter one $T' = T/2^m (m \geq 1)$, the FLOPs needed are also reduced accordingly. Secondly, although computation can be done in parallel *in theory*, current computational device still cannot finish all parallelizable operations *in a single clock*. In practice, these operations still need to be done sequentially. Therefore, reducing FLOPs requirement of the model improves the running time.

Q: “only measure the actual running time by performing 1000 steps gradient descent with the fixed length”
A: As described from Line 223 through 225, the *number of training steps* and *batch size* of the base setting are exactly the same as those in the original BERT model. Also, during pretraining, the input sequence always has a fixed length. Hence, the measured running times simply proportional to the total pretraining times for both models.

Q: “trading sequential resolution for more layer”
A: As we discussed in Section 2.3, reducing the sequence length will inevitably lead to capacity drop but the capacity drop can be compensated by re-investing the *saved FLOPs* in stacking more cheaper layers. Fundamentally, the key question is: Given the **same amount of FLOPs**, should we invest the FLOPs in (a) fewer full-length layers, or (b) more reduced-length layers? This work explores this fundamental question and shows that option (b) can empirically improve the performance. In comparison, simply increasing the layers of standard transformer will require much more FLOPs.

Q: “What is the significance of length reduction”
A: By comparing the performance between L12H768 and B4-4-4H768 and between L24H1024 and B8-8-8H1024, the current length reduction mechanism by itself can mildly harm the performance in certain tasks but achieve the same performance in others, with the benefit of over 40% fewer FLOPs. Hence, the ultimate effects are largely task-specific.

Q: “there should be more theoretical analysis and experiments on length reduction ...”
A: We agree that developing theoretical understanding of length reduction / compression is an important direction for future work. Meanwhile, we believe identifying the scientific possibility of the idea, designing a practically scalable instantiation, and empirically showing the potential of this direction are also valuable and necessary steps.