We thank all the reviewers for their constructive comments. Please see our responses below.

R1-3: novelty and relevant work. The key contribution of our work is the development of an efficient and memory-friendly architecture for video understanding. Our approach is purely based on 2D convolutions. Nevertheless, it outperforms or performs comparably to many more costly 3D models. Especially, our proposed TAM layers have been shown more effective than 3D temporal convolutions and some recently proposed spatiotemporal approaches that are structurally more sophisticated (Table 1 and 3). We hope that our findings and results in the paper are helpful and will make the community to rethink the efficacy of 2D and 3D architectures in learning spatiotemporal feature representations.

We thank the reviewers for pointing out some related (or missing) references. We note that some of them such as Timeception, SlowFast and TSM are concurrent with our work. Here we briefly describe the main differences between these approaches and ours, and more discussions will be added to the final manuscript. Timeception basically applies the concept of “Inception" to the temporal domain for capturing long-range temporal dependencies in a video. The Timeception layers involve group convolutions at different time scales while our TAM layers only use depthwise convolution. As a result, the Timeception has significantly more parameters than the TAM (10% vs. 0.1% of the total model parameters). As for SlowFast, it differs from our approach in that a) it uses 3D convolutions for temporal modeling; and b) it achieves efficiency by balancing the number of input frames and channels at different network branches. Compared to TSM, our approach is more generalized, more extensible, and in particular more effective, as shown in the paper.

R1-3: new results. After the submission, we further trained optical-flow models on the Something-Something V2 dataset and applied model ensemble with the corresponding RGB models. Our 2-stream models improve top-1 accuracy over the RGB models by **2.2%-2.8%** on the validation set. On the leaderboard, we are currently the 2^{nd} best on top-1 accuracy and the 1^{st} on top-5 accuracy.

R2 and R3: code release. We will release our code and models for this work as well as the scripts for data preparation, model training and evaluation. In the meanwhile, we would be delighted to share as much material as possible to help independent replication and validation of our work. In addition, the Big-Little Net code is publicly available at `https://github.com/IBM/BigLittleNet`, which should be helpful for the adoption of our work.

R1: performance of 8×2 models on ImageNet. We realized that Line 255 in the paper might have confused R1. To clarify that, all our models using 8×2 frames in Table 1-4 (*bLVNet-TAM*-8×2) were fine tuned from 2D models pretrained on ImageNet. Then, the models *bLVNet-TAM*-16×2 were learnt from *bLVNet-TAM*-8×2 and *bLVNet-TAM*-24×2 from *bLVNet-TAM*-16×2 and *bLVNet-TAM*-32×2 from *bLVNet-TAM*-24×2, respectively. We found that learning in such a progressive way is not only effective, but also faster than fine tuning from ImageNet.

R3: odd or even frames as input. We do not enforce that the big branch must operate on odd frames and the little branch on even frames. Instead the big branch can take either of a pair of frames as input and the other frame goes to the little branch. We will clarify this in the final manuscript.

R3: complexity of TAM. Like TSN, our approach has a training-time complexity proportional to the number of input frames because the TAM layers are highly light-weighted compared to the backbone network. As shown in Fig. 3 in the paper, when using the same number of input frames, our models allow for a batch size of about 2 times larger than TSN in training. Note that The TAM operates on ‘r’ *frames* rather than ‘r’ *clips*.

R3: evaluation setup (question c-e)). R3 is right about the “single-crop single-clip” setup, which means a single clip is formed for each video in test by picking a pair of frames from a set of uniformly split segments of the video. The results in Table 1, 2 and 4 are reported based on such a setup. The ‘Frames’ column refers to the total number of frames used in inference, but with a single crop per frame only. Differently, the “multi-crop multi-clip” setup can be considered as repeating “single-crop single-clip” multiple times at different time instances and at different cropping locations in a test video. In such a case, the TOTAL number of frames used in inference is thus the product of the number of frames used in “single-crop single-clip”, the number of crops and the number of clips. For example, in Table 3, our *bLVNet-TAM*-8×2 uses 16×3 (crops)×3 (clips) frames for evaluation while TSM-8 uses 8×3 (crops)×10 (clips) frames.

R3: performance improvement over SOTA. While being efficient, our approach achieves the state of the art accuracy on the Something-Something and Moments datasets (see Table 1, 2 and 4). It’s worthy to note that our approach only uses RGB information, but still outperforming the previously best 2-stream models based on both RGB and optical flow information. In addition, our recently trained two-stream model (*bLVNet-TAM*-32×2) is 2.8% better than TSM-16 at top-1 accuracy (66.8% vs. 64.0%) on the SS-V2 validation set, and our approach are ranked the 2^{nd} on the Something-Something leaderboard (2% better than the TSM-16 on the leaderboard, 66.34% vs. 64.33%).