We thank the reviewers for their thoughtful and constructive feedback.

Reviewer 1: The reviewer notes that the proposed conditional convolution method is novel and shows promising results. We agree further results could help strengthen the generality and applicability of our method and will add the following:

- The reviewer suggests that results would be more convincing if tested on more network structures. We additionally apply our CondConv approach to MnasNet-A1, one of the state-of-the-art mobile architectures designed with architecture search. **Adding CondConv with 8 experts to the last two block groups improves top-1 accuracy of MnasNet-A1 on Imagenet from 74.9% to 76.2%, while only increasing MADDs from 312M to 325M. This significantly outperforms MnasNet-A2 at 75.6% accuracy and 340M MADDs.** This complements our original results, which cover three different commonly used architectures (MobileNetV1, MobileNetV2, and ResNet-50) and two different tasks (classification and detection).
- The reviewer suggests we compare our approach with squeeze-and-excite (SE). CondConv improves MnasNet-A1 performance, which includes SE layers. In addition, we ran experiments on the MobileNet-V1 baseline. On MobileNetV1 (0.25x), adding SE to every layer improves performance by 2%. Our CondConv approach with 32 experts improves performance by 10%. **CondConv and SE together achieves 10.2% improvement**.
- We agree that additional experiments are needed to support the claim that "CondConv blocks at later layers improve final accuracy more than those at earlier layers". We directly compare a CondConv-MobileNetV1 (0.25x) model with 32 experts at layer 7 only against a model with 32 experts at layer 13 only. **32 experts at layer 7 only improves accuracy by 2%, while 32 experts at layer 13 only improves accuracy by 5%**.

The reviewer further mentions relevant work around deformable convolutions and MSDNet which we will add to our discussions. Finally, the reviewer mentions the explanation for Figure 4 is not clear. We intended to show many routing weights had values close to 0.0, which suggests experts can be sparsely activated. We believe the new results suggested by the reviewer better demonstrate the applicability of our approach, and will consider moving Figure 4 to the supplementary material in the final manuscript.

Reviewer 2: We apologize for the confusion with regards to our approach, and we will work hard to clarify. The reviewer mentions a parallel between our CondConv approach and Inception modules, but the methods are quite different. The basic idea behind Inception modules is to apply multiple convolutional kernels of different sizes to the same input in a layer and combine their outputs. This requires computing an expensive convolution for each kernel. The novel insight of our work is that we can achieve the capacity of combining multiple convolutions of the same size with a *single* convolution. We achieve this by first computing a single *example dependent kernel* for the convolution. On CondConv-MobileNetV1 (0.25x) with 32 kernels per layer, our approach achieves the same capacity and performance as combining 32 separate convolutions, with only 7% of the multiply-adds. We will use this to improve our discussion.

The reviewer then mentions specific questions with our discussion of W_i. In our CondConv approach, the W_i are tensors of the same shape as the original kernels for the convolutional layer being replaced, with the same number of channels. Every kernel W_i in a layer is used for every example throughout training to compute a single example dependent kernel for the convolution. The routing weights and kernels are trained together with gradient descent.

The reviewer then suggests we analyze other non-linear activation functions to compute routing weights r(x). We initially compare the Softmax and Sigmoid functions, because they are commonly used in the mixture of experts and conditional computation literature. On CondConv-MobileNetV1 (0.25x) with 32 kernels per layer, **using ReLU for r(x) achieves 56% Imagenet accuracy while tanh achieves 60%, compared to Sigmoid which achieves 62%**.

The reviewer finally notes our proposed method may be impractical for mobile devices due to the number of parameters. While we agree that our method is impractical in memory-constrained settings, our CondConv approach is applicable to many scenarios constrained by inference time and quality but not memory. This includes systems to process video and speech in real-time on servers or desktop computing machines. We plan to expand on this motivation in our discussion.

Reviewer 3: The reviewer found the work interesting, and noted that the writing and illustrations are clear.

The reviewer is concerned that increasing the number of experts for CondConv-ResNet-50 does not improve performance. We find that CondConv-ResNet-50 with 8 experts per layer achieves lower training loss than 2, but suffers from over-fitting due to larger capacity. We hypothesize that with enough data and/or regularization, more experts leads to better accuracy, which we see with smaller models on Imagenet. To test this hypothesis, we additionally experiment with more aggressive regularization techniques based on Shake-Shake. In new results, we are able to **improve Imagenet accuracy of CondConv-ResNet-50 with 8 experts from 77.7 to 78.6, higher than accuracy with 2 experts**. In the case that dataset size and regularization are constant, a good approach to identify the right number of experts is a grid search, as we did originally in the case of CondConv-ResNet-50.

Finally, the reviewer mentions relevant work for dynamic filter networks (DFN), which we will add to our discussion. One key aspect that differentiates our work is our approach to generate large kernels for modern CNN architectures (the largest kernel in DFN is 81 parameters) for classification and detection (rather than next frame and stereo prediction).