**Rebuttal for PID 8695, "Simultaneous Network Channel and Spatial Pruning via Discrete Variable QCQP"**

We thank the reviewers for the time, and the effort spent providing feedback. We appreciate the encouraging comments (R1: "I liked the proposed method", R2: "provide a new way to jointly consider the importance and resource contribution of DNN weights", R3: "interesting to model the pruning constraints", R4: "might be more important in the long run"). First, we would like to address common questions and answer the specific questions from each reviewer.

| Network | Method | Top5 Acc↑ | FLOPs(%)↓ |
|---|---|---|---|
| VGG16 | base | 90.4 | 100(%) |
| | [25] | 84.5 | **51.7**(%) |
| | ours(c) | 87.2 | **51.7**(%) |
| | ours(c+s) | **87.6** | **51.7**(%) |
| Network | Method | Top1 Acc↑ | FLOPs(%)↓ |
| MobileNetV2 | NetAdapt | **70.9** | 75% |
| | AMC | 70.8 | **70**% |
| | ours(c) | 69.7 | **70**% |

Table 1: Test accuracy and FLOPs on VGG16 and MobileNetV2

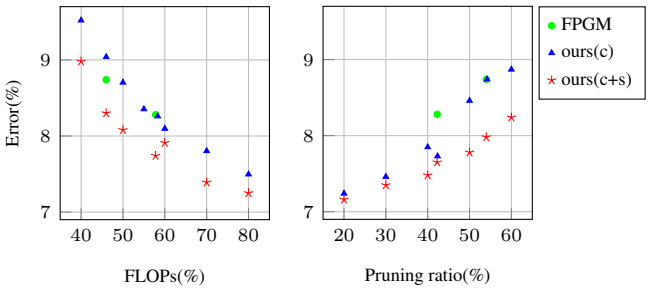


Figure 1: Pruned test error at different FLOPs(left), and Pruning ratio(right) on CIFAR-10

**[R1,2] Experiments on different network architectures** Table 1 compares our methods with [25] on VGG16 and compares our method with NetAdapt and AMC on MobileNetV2. In VGG16, our methods outperform [25] at the given FLOPs, $51.7\%$. Since spatial pruning in depthwise convolution cannot reduce time complexity, we performed only 'ours(c)' on MobileNetV2. In MobileNetV2, 'ours(c)' shows slightly lower performance than AMC and NetAdapt.

**[R1,4] Make it clear why you think they have such problems, discuss the relationship of avoiding overpruning** According to [10], the post-hoc pruning process produces dead neurons and subsequently *inactive* weights. Predicting the exact number of inactive weights after post-hoc channel pruning is a difficult problem, as discussed in Section 3.1 of [19]. Therefore, existing post-hoc channel pruning approaches resort to heuristics to remove some of these inactive weights remaining in the pruned network. For example, in FPGM, inactive weights, which are not removed after following heuristics after pruning process, account for $3\%$ of total FLOPs both in ResNet 20, 56 on CIFAR-10. However, our method prevents the existence of any inactive weights in the pruned network, as proven in Sec. C in Supp.

**[R1] Ablation for different nonsequential operations** Skip addition is the most widely used nonsequential operation as in ResNet and MobileNetV2. Also, skip concatenation in DenseNet could be easily extended from 'Eq.2'.

**[R1] Ablation for the comparison to the related works on this contribution** We performed an ablation experiment where the nonsequential constraints in 'Eq.4' are replaced with $u = v$ ignoring the skip connection as in existing pruning methods such as [19] and FPGM. We observed this results in $1.2\%$ drop in test accuracy at given FLOPs, $40\%$.

**[R1] Ablations for different FLOPs, memory constraints** We performed an ablation study on different FLOPs and Pruning ratio in Figure 1. For all Pruning ratio and FLOPs, our method strictly outperforms FPGM in a large margin.

**[R2] It is not clear how the spatial pruning can reduce time complexity.** As stated in L89-90, spatial pruning can reduce time complexity by assembling the group-sparse filter matrix into the compact matrix. Also, the time complexity reduction is experimentally shown in [17].

**[R2] How is the accuracy of the solution from Algorithm 1?, How is the time\memory requirement for solving the QCQP derived from the pruning problem?** We denote directly solving 'Eq.4' with CPLEX as 'Algorithm 0'. Algorithm 0 is not scalable even in ResNet20 on CIFAR-10. Therefore, we compare Algorithm 0 and 1 for the first 8 layers of ResNet56. We set $B = 2$ and adjust $\gamma$ in $0.1$ steps. Algorithm 1 yields the objective '100.16' in '12-min', and algorithm 0 yields the objective '106.27' in '1-hr'. Also, Algorithm 1 requires '3-hr' \'4GB' in ResNet50 on CIFAR-10.

**[R3] What is the difference between directly ranking parameters and pruning the ones with less magnitude?** In 'Eq.1', objective function without the constraints on $A$ is the unstructured pruning as in Han et al.[10]. However, constraints on $A$ enforce the structure on $A$, which is specified by $r$. We will make this clearer in the final version.

**[R3] Structure pruning in the first layer leads to corresponding pruning of the second layer. Is this the quadratic couple?** Yes, in 2 fully-connected layers without any nonsequential operations, it is the quadratic coupling of two subsequent layers. The quadratic coupling becomes more complex with the nonsequential operations like skip addition.

**[R3] Is it necessary to include A for it can be replaced by r** In 'Eq.1', we introduce $A$ for simplicity in objective function and constraints. Also, 'Eq.1' is written as a QCQP form in terms of $r$ in Sec. A in Supp.

**[R4] Can the update of u, v, q be performed alternately?** Since $u$ is dependent on $q$, 'Eq.4' is the optimization problem of $v, q$. Update of $v, q$ can be performed alternately. However, initially optimizing $u, v$ in algorithm 1 can significantly reduce the number of variables by about 10 times compared to optimizing $u, v, q$ directly in ResNet.

**[R4] Necessary to clearly explain the reason for setting the parameters ($\gamma, B$)** $\gamma$ temporarily increases the resource budget to ensure channel pruning in $u, v$ leaves a room for spatial pruning in $q$. $B$ is the size of the optimization subproblem to handle the time complexity.

**References**


[10] S. Han, J. Pool, J. Tran, and W. Dally. Learning both weights and connections for efficient neural network. In *NIPS*, 2015

[17] V. Lebedev and V. Lempitsky. Fast convnets using group-wise brain damage. In *CVPR*, 2016

[19] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf. Pruning filters for efficient convnets. *ICLR*, 2017

[25] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz. Pruning convolutional neural networks for resource efficient inference. *ICLR*, 2017