R2, R4: The improvements are marginal/slightly better. Our gains are indeed large. Here're some highlights:

- EvoNorm-S0 is the state-of-the-art in the small batch size regime (Table 4), outperforming BN-ReLU by 7.8% on ResNet-50 and by 7.3% on MobileNetV2.
- We achieve clear gains over other influential works such as GroupNorm (GN). On ResNet-50, EvoNorm-S0 outperforms GN-ReLU by 1.2% (large batch) and 1.1% (small batch) (Table 4).
- On Mask R-CNN, GN-ReLU improves BN-ReLU by 0.6mAP whereas EvoNorm-B0 improves BN-ReLU by 1.9mAP (Table 5). 1.9mAP gain on detection/instance segmentation is considered to be very significant in the vision community (e.g., it is comparable with the gains of RetinaNet [1] over its previous models). [1]

We'd also like to emphasize that EvoNorms beat BN-ReLU on 12 (out of 14) different classification models/training settings (Table 3 & 4) and all instance segmentation models (Table 5). B0 also improves over BN-ReLU on BigGAN in terms of FID (Table 6). These are significant considering the predominance of BN-ReLU in ML models.

R3: "the overall search algorithm lacks some novelty." We argue that our work should not simply be evaluated as "yet another AutoML paper" (with the expectation that some fancy search algorithms must be proposed), but rather under the scope of designing new normalization-activation techniques which are central to deep learning (BatchNorm paper is cited 20000 times, LayerNorm is cited 2000 times and GroupNorm is cited 700 times). While most existing norm-act layers are in the form of $\frac{x-\text{mean}(x)}{\text{std}(x)}$ followed by a scalar-to-scalar transformation, we design highly unconventional layers that perform equally well/better on a variety of important vision tasks (see EvoNorm-B0, which does not center the feature maps or require explicit activation functions). The novelty of these discovered layers, as well as the new insights, should not be diminished because they were discovered automatically instead of manually.

R2, R4: Can EvoNorms generalize to deeper variants (e.g., ResNet-101) and architecture families not included during search (e.g., DenseNet)? The answer is yes. In Secs 5.1-5.3 we conducted extensive experiments to verify that EvoNorms can perform well on previously unseen architectures (as well as tasks beyond classification), including MnasNet, EfficientNet-B5, Mask R-CNN + FPN/SpineNet and BigGAN–none of them was used during search. Below we also provide additional results on ResNet-101 and DenseNets (as requested by R4):

Model	DenseNet-BC-121	DenseNet-BC-169	ResNet-101	ResNet-101 + Mask R-CNN	Batch Independent?
BN-ReLU	74.9	76.0	79.3	43.8	No
EvoNorm-B0	**75.8**	**76.4**	**79.5**	**45.1**	No
GN-ReLU	74.5	75.8	79.0	44.0	Yes
EvoNorm-S0	**75.7**	**76.5**	**79.4**	**44.5**	Yes

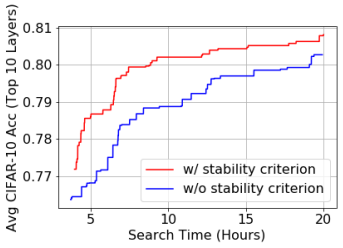
Table 1: ImageNet Accuracies & COCO box mAPs. ImageNet models were trained using the +aug+2×+cos setup with batch size 8×128. Like all the other experiments in the paper, we do not tune hyperparameters wrt EvoNorms.

R3: Why including 0, v_0, v_1 in the computation graph? They are basic elements in popular activation functions. E.g., nodes 0 and v_1 are required to form graphlets like $\max(x, 0)$ (ReLU) and $x\sigma(v_1 x)$ (SiLU/Swish), respectively. Moreover, they can also be used by evolution in creative ways–see the expression of EvoNorm-B0.

R1, R4: The search space design could be improved. Our current search space contains 12 element-wise ops and 12 aggregation ops (3 types of statistical moments, each with 4 possible index sets). While expanding the search space further is an interesting future direction, we believe the current search space is sufficient to demonstrate the concept for the following reasons: (i) it is already challenging enough (Figure 2: none of 5000 random layers can outperform BN-ReLU), and (ii) it is also interesting enough to yield highly novel design patterns, such as EvoNorm-B0.

R1: Is there a hold-out set for layer search? Yes. The layers were evolved on CIFAR-10 and then reranked using 10% of ImageNet's training set (L204-205 & L547-549 in Appendix D). The test set is never used during search.

R2: Stability criterion might reject promising layers early on. This is a valid concern. If so, one would expect the stability criterion to eventually hurt the accuracies of the best layers during evolution and/or slow down the search progress. We provide an ablation study (as suggested by R2) in the figure on the right hand side, showing that this is not the case. In fact, the stability criterion speeds up the overall search progress by up to 2×. One possible explanation is that layers that can achieve high accuracies are also likely numerically stable.



[1] [1] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In CVPR 2017.