We sincerely thank reviewers for their insightful feedback! We are encouraged that reviewers find our method novel (**R2**,**R3**) and analysis insightful (**R3**). All reviewers (**R1**,**R2**,**R3**,**R4**) agreed that **our method achieved significant improvements in a variety of tasks/settings** (image classification, object detection, instance segmentation, adversarial attack and low data setting) backed with extensive experiments and ablations. We address reviewer comments below.

@**R1**,**R2**,**R3**, Q1: The training cost of GradAug may be several times of typical regularization methods: This is NOT true. As stated in [11], typical regularization methods [11,10,8] require **more training epochs** to converge, while GradAug **converges with less epochs**. Thus the **total training time is comparable**. **The memory cost is also comparable** because we forward and backward sub-networks one by one, only their gradients are accumulated to update the weights. Table 1 shows a comparison on ImageNet. The training cost is measured on an $8\times$ 1080Ti GPU server with a batch size of 512. Mixup and CutMix need 77 and 115 hours to converge, while GradAug converges in 122 hours (120 epochs). So the training cost of our GradAug is comparable with SOTA methods.

Table 1: Training cost comparison on ImageNet. Reference # from paper.

| ResNet-50 | Epochs | Mem (MB) | Mins/epoch | Total hours | Top-1 Acc |
|---|---|---|---|---|---|
| Baseline [10] | 90 | 6973 | 22 | **33** | 76.5 |
| Baseline [10] | 200 | 6973 | 22 | 73 | 76.4 |
| Mixup [10] | 90 | 6973 | 23 | 35 | 76.7 |
| Mixup [10] | 200 | 6973 | 23 | 77 | 77.9 |
| CutMix [11] | 300 | 6973 | 23 | 115 | 78.6 |
| GradAug | 120 | 7145 | 61 | 122 | **78.8** |
| GradAug | 200 | 7145 | 61 | 203 | **78.8** |

@**R4**, Q2: Use stochastic depth [A] to sample depth-shortened sub-nets for GradAug: Great suggestion! To do so, we follow the settings in [A] to randomly drop layers to generate sub-networks. We also utilize random scale transformation and input images are randomly resized to one of $\{32\times 32, 28\times 28, 24\times 24\}$. The results in Table 2 show that **GradAug can be generalized to depth-shortened sub-networks as well**. This also validates the effectiveness of our idea - regularizing sub-networks with differently transformed inputs.

@**R1**,**R4**, Q3: Only a simple sub-network sampling strategy is considered: Our goal is to show the effectiveness of regularizing sub-networks with different transformed inputs. To form sub-networks, we just follow the most common practice in previous literature to scale down the network by network width. As shown in the response to **Q2**, sampling sub-networks by **depth** is also feasible, and the corresponding results (Table 2) also validate its efficacy. Analyzing the effect of different sampling strategies is interesting and we will certainly explore it in future work.

Table 2: Utilizing stochastic depth [A] in GradAug.
[A] "Deep networks with stochastic depth" ECCV 2016

| ResNet-110 | Cifar-10 | | Cifar-100 | |
|---|---|---|---|---|
| | Reported | Reimpl. | Reported | Reimpl. |
| Baseline [A] | 93.59 | 93.49 | 72.24 | 72.21 |
| StochDepth [A] | 94.75 | 94.29 | 75.02 | 75.20 |
| GradAug | - | **94.85** | - | **77.01** |

@**R1**, Q4: Only random scale transformation is considered: This is NOT true. In the paper we conducted *random scale* transformation and *random scale* + *CutMix* (**L185-187**, GradAug+). In the **supplementary material**, we also showed the results of *random rotation* and *random scale* + *random rotation* (confirmed by **R3**). Here, we further present results on ImageNet (Table 3). As suggested by **R3**, we will put these results in the main paper.

Table 3: Different transformations in GradAug on ImageNet.

| ResNet-50 | Top-1 | Top-5 |
|---|---|---|
| Baseline | 76.32 | 92.95 |
| RandScale | 78.79 | 94.43 |
| RandRot | 77.62 | 93.66 |
| RandScale&Rot | 78.66 | 94.40 |

@**R2**, Q5: How GradAug works: We believe there is a misunderstanding about our method. Our idea is leveraging different transformed inputs to regularize sub-networks which are originated from the full-network. We explain our method from two views. First, intuitively, full-network shares the representations learned by sub-networks because they share weights. We illustrate this by showing the CAMs of sub-network and full-network. **Fig. 1** (in paper) shows that full-network shares the attention map of sub-network and it can also use the other network part, which sub-networks don't have, to learn additional features. So full-network can capture more semantic information than sub-networks (**L106-112**). Second, we explain the differences between GradAug and other regularization methods from the perspective of gradient flow. Dropout and its variants randomly drop some connections. This can be viewed as adding **random noises** to the original gradients as explained in Eqs.(1,2,3). GradAug can also be viewed as adding a term to the original gradients (Eq. 4), but this term is the gradients of sub-networks with different transformed inputs. Since **sub-networks are part of the full-network**, we call this term **"self-guided"**. It reinforces good descent directions, leading to improved performance and faster convergence. Indeed, the experimental results show that it significantly improves the performance over Dropout variants (78.8 vs. 77.5 (Shakedrop) [20], 78.1 (Dropblock) [16]) and converges faster in terms of training epochs (120 vs. 180 [20], 270 [16]).

@**R2**, Q6: Comparison to neural network compression: We do NOT agree our approach is analogous to neural network compression. Our goal is to improve the performance of the full network rather than compressing the network.

@**R4**, Q7: Can GradAug be applied to SlimNet, can GradAug-trained network be pruned like SlimNet? GradAug can be applied to SlimNet by feeding different transformed inputs to different widths. We believe the performance can be improved since the full-network is considerably improved. If we do sub-nets sampling by width, GradAug-trained network can be pruned like SlimNet. For example, the performance of sub-net $width = 0.9\times$ is 77.6% on ImageNet.

@**R4**, Q8: Effect of smallest sub-net (SS) and soft label (SL): Ablation is in Table 4. SL is important in GradAug, but the application of SL is not trivial. First, soft labels come for free (from full-net) in GradAug, whether sampling sub-nets by width or depth. Second, we are transferring the knowledge among sub-nets based on **differently transformed inputs**. This is different from traditional KD and label smoothing which usually marginally improve the performance on ImageNet. The effectiveness actually validates our idea of regularizing sub-nets with different inputs. We'll include these results.

Table 4: Effect of SS and SL.

| Model | C-100 | IN-1K |
|---|---|---|
| Baseline | 81.5 | 76.3 |
| GradAug | 84.0 | 78.8 |
| no SS | 83.8 | - |
| no SS&SL | 82.5 | 77.4 |

@**R3**, Q9: Claim on adversarial robustness. Choice of input scales: We will revise the claim to the robustness to FGSM attack. The input scales are determined empirically. We don't want the images to be too small.