#R1# **Using Bounding Boxes.** In fact, we considered a more general assumption where bounding boxes are absent, in terms that annotating datasets with localization boxes is usually expensive and time-consuming in realistic applications. We agree that it is a nice idea to exploit the bounding boxes of ImageNet, and are happy to explore it in GFNet.

#R1# **Upper Bound of Performance.** Thanks for the suggestion. The accuracy of GFNets will approach the models receiving the whole images (e.g., 224x224) when using larger patches (e.g., with DenseNet-121, the gaps are 0.6%/0.1% for 96x96/128x128). We will add more comparisons on this point in our revision.

#R1# **Complex Training Process.** Given that no ground-truth bounding box is provided, we believe that RL is crucial to search for a superior patch selection strategy. The RL part is not complex since we simply use an off-the-shelf PPO algorithm. The iterative process is indeed not indispensable, but in experiments, it improves the accuracy (e.g., $\sim 0.5\%$ with MobileNet-V3) compared with training all components simultaneously. We will make these clear in revision.

#R1# **Code.** We will release all the code and pre-trained models upon the acceptance of this paper.

Table 1: Results using 32x32 (left) and 64x64 (right) patches. Due to time limits, more results will be added in revision.

32x32 Patches	$t=1$	$t=2$	$t=3$	$t=4$
Random Policy	41.65%	45.31%	47.66%	49.10%
GF-ResNet-50	**41.93%**	**52.04%**	**55.32%**	**57.49%**

64x64 Patches	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$	$t=8$	$t=9$	$t=10$
Random Policy	60.88%	64.37%	66.53%	67.87%	68.90%	69.52%	70.05%	70.41%	70.73%	70.94%
GF-ResNet-50	**61.03%**	**68.24%**	**70.18%**	**71.61%**	**72.36%**	**73.10%**	**73.64%**	**73.84%**	**74.07%**	**74.27%**

#R2# **Selection of RL Algorithm.** We use PPO since it is shown to outperform REINFORCE in terms of effectiveness, efficiency and stability in its original paper. We will study the impacts of its components in our revision.

#R2# **Performance of Random Policy.** Thanks for pointing out this issue. There are three reasons why random policy performs well. First, the sequential classification task on ImageNet is not that difficult for random policy. It has been proven in [1] that ImageNet-trained CNNs are strongly biased towards recognizing image textures rather than shapes. Even random patches can capture textures (local patterns) well, leading to acceptable performance. Second, in ablation study, we use relatively large patches of 96x96 (approaching 1/4 of 224x224 images), which are very likely to contain some class-discriminative regions even with random sampling. As shown in Table 1, GFNet outperforms random policy by 8% and 4% when using 32x32 and 64x64 patches. Third, for fair comparison, we also augment the random policy with the *Glance Step*, which contributes to an excellent preliminary prediction. In addition, it is actually challenging for GFNet to learn to identify class-discriminative regions since we considered a very general setting where no localization annotation (e.g., bounding boxes) is available. We admit that there may still exist space to design better RL algorithms, and will focus on this point in the future. ([1] Geirhos R, et al. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In ICLR, 2019.)

#R2# **Comparisons with MSDNet.** As suggested, we compare GFNet with MSDNet in Figure 1.

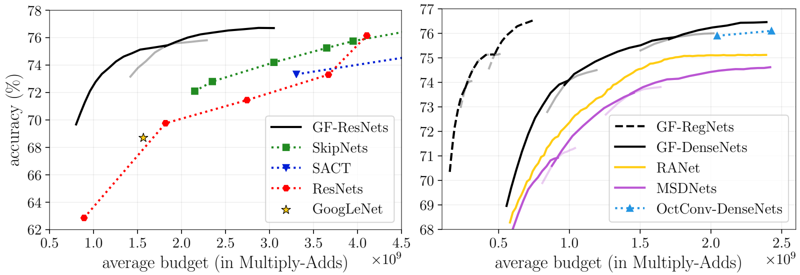

Figure 1: Performance of GFNets and baselines. OctConv-ResNets are not presented as they use a different network architecture from us (a pre-act version). We will implement pre-act GF-ResNets and present comparisons in revision.

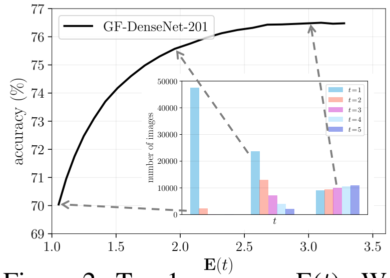

Figure 2: Top-1 acc. v.s. E(t). We show the numbers of images using different values of t in several points.

#R3# **More Baselines.** We compare SACT with GFNet in Figure 1. Since SBNet is proposed for 3D object detection, we do not directly compare it with ours. We will thoroughly discuss the relations of GFNet to the two works in revision.

#R3# **Clarity.** Thanks for the suggestion. The averaged budget refers to the mean computational cost of each test sample. Each curve in Figure 4&5 of the paper corresponds to an individual model. The curves are across since each model has the highest efficiency only within a certain range of computational budgets. We will make these points clear.

#R4# **More Baselines.** We compare OctConv, SkipNets and RANet with GFNet in Figure 1. In fact, We believe that OctConv and SkipNet are orthogonal techniques with our method since they can be used as CNN backbones in GFNet.

#R4# **Value of** E(t)**.** As suggested, we show the graph of E(t) v.s. accuracy in Figure 2. We also present the plots of t v.s. number of images. One can observe that the performance of GFNet is significantly improved by letting images exit later in the *Focus Stage*, which is realized via adjusting the confidence thresholds online (without additional training).

#R4# **Implications to Other Vision Tasks.** Although we only focus on the most general classification task in this paper, we note that it is causable to extend GFNet to other vision tasks. For example, in objective detection, we can obtain the preliminary prediction using low-resolution inputs (*Glance*), and then *focus* the computation on "important" high-resolution local regions to sequentially find all the objects. We will add these discussions in revision.

#R4# **Relation to SkipNet.** Indeed, GFNet is similar to SkipNet in terms of the dynamic architecture. However, we focus on reducing spatial redundancy, while they adaptively skip unnecessary layers. We will include more discussions.