We appreciate the reviewers for the constructive comments on this paper. We will carefully address the grammar mistakes, undetailed plots, unclear statements in the revised manuscript. Major concerns are responded as follows:

One common concern is our baseline for RetinaNet/Mask-RCNN is not strong. It is because our models are trained from scratch (Section 4.5), while most results of other papers or model zoo are fine-tuned from a pre-trained model. The result of Mask-RCNN with 3x scheduler and training from scratch in "Kaiming He, et al. Rethinking ImageNet Pre-Training." is comparable with our baseline (39.5% vs 39.24%).

***R1Q1*: How do the latencies change on GPU?** We show the latencies change in Table 1, which indicates models of similar FLOPs vary greatly in latency, e.g. ResNet18^{AutoBSS} is **18% slower** while Effi-B1^{AutoBSS} is **23% faster**.

Table 1: Average GPU latency per image (Tesla V100-PICe, Pytorch 1.2.0, CUDA 9.0.167, CUDNN 7102).

ResNet18	Latency	ResNet50	Latency	MBV2	Latency	Effi-B0	Latency	Effi-B1	Latency
Original	0.242ms	Original	0.973ms	Original	0.257ms	Original	0.438ms	Original	0.703ms
Rand	0.200ms	Rand	1.22ms	Rand	0.206ms	Rand	0.346ms	Rand	0.544ms
AutoBSS	0.287ms	AutoBSS	1.11ms	AutoBSS	0.252ms	AutoBSS	0.407ms	AutoBSS	0.539ms

***R1Q2*: The improvements are not large.** As pointed out by Reviewer#2, the baseline networks are relatively mature thus the improvements are still significant. We think this is especially the case on common datasets like ImageNet, the improvements will be larger on other datasets.

***R1Q3*: Compare to prior BSS search methods like POP[22] in Table 1.** The network "DF2" in POP has similar FLOPs (1.77G vs 1.8G) and Top1-Acc (73.9% vs 73.91%) compared with ResNet18^{AutoBSS}. But their searching is more sample inefficient (200 vs 64 samplings) and they need some handcrafted modifications like changing the kernel size for the first convolution from 7×7 to 3×3. Methodological comparison can be found in line $90 \sim 93$.

***R2Q1*: Why increasing one dimension of BSSC necessarily make the resulting network perform better than the original network?** Because the resulting network strictly includes the original one, thus has a greater capacity. Take MobileNetV2[7] as an example. Increasing one dimension of BSSC means increasing the number of channels or blocks. For channels, a larger width multiplier always results in better performance (table 4 of [7]). For blocks, residual connection guarantees the newly-introduced inverted residual block no worse than identity mapping.

***R2Q2*: The purpose of BSSC refining.** 1) Bayesian Optimization approach based on a Lipschitz-continuous assumption (page 4 of [33]). The refining makes the model for accuracy, i.e. $f(x)$ satisfy this assumption better. That is, there exists constant C, such that for any two BSSC x_1, x_2: $\|f(x_1) - f(x_2)\| \leq C\|x_1 - x_2\|$. The red dashed line in Figure 2 and Figure 3(b) (mean value plus standard deviation) can be regarded as the upper bound of $\|f(x_1) - f(x_2)\|$. After refined with a linear layer, it becomes much more close to the form of $C\|x_1 - x_2\|$. 2) It helps to aggregate the BSSC with similar accuracies into one cluster, thus they will not be sampled at the same time, otherwise, some evaluations for BSSC will be meaningless.

***R2Q3*: Some unclear details.** **A) How is the one-layer network initialized?** The initial weight is set as an identity matrix (line 158$\sim$159). **B) Is the one-layer network pre-learned on other datasets?** No, the linear layer for Figure 3(b) is trained with 16 BSSC, thus the tens of BSSC evaluated during each searching procedure is enough to train this model. **C) How is the acquisition function optimized and what is the set of candidates for this optimization?** We simply calculate the value of acquisition function at each clustering center and select the largest one (line 164$\sim$165). **D) What is the definition of accuracy discrepancy?** The absolute difference of two accuracies.

***R2Q4*: Add an ablation study on the refining process.** Actually, we have conducted experiments on whether to refine BSSC with a linear model. Without this step, the accuracy for the searched ResNet18^{AutoBSS} is $\sim 0.4\%$ lower. We thought the comparison between Figure 2 and Figure 3(b) could indicate the importance of refining thus didn't report this ablation result, we will add it for the revision.

***R3Q1*: Add more discussions of existing NAS methods like DARTs.** These methods rely on a biased evaluation protocol like early stop or parameter sharing, while we can train each sampled network fully and get an unbiased evaluation. More details about algorithm efficiency and effectiveness will be added in the revision.

***R3Q2*: If the proposed AutoBSS can be used under latency constraints?** Our method supports other constraints like CPU/GPU latency. It can be achieved via using latency instead of FLOPs to construct the candidate set.

***R4Q1*: Why don't you conduct experiments on EfficientNet-B7 or some stronger network?** Actually, EfficientNet-B0/B1 is already a very strong network under the constraint of FLOPs, thus can verify our method. We didn't use EfficientNet-B7 because it is too resource-consuming.

***R4Q2*: Do you try other clustering methods?** Haven't yet. Thanks for your reminder, we will try it for future work.

***R4Q3*: How long is the training time? And what is the number of GPUs?** We take ResNet18 as an example. It needs $\sim$ 20 hours to train a model with 8 GPUs. For each iteration, the 16 sampled models are trained in parallel.

***R4Q4*: Do you have any plan to make your source code and models pubulic?** Yes, we have applied for sharing the code and models for academic society, it is in the approval process.

***R4Q5*: Results on semantic segmentation.** We conduct experiments on PSACAL VOC 2012 for PSPNet and PSANet. The training settings are identical with *Performance.table2* of the github project (hszhao/semseg) created by authors of PSPNet and PSANet. We pre-train our model on ImageNet as well. Because of the task similarity of semantic segmentation and instance segmentation, we directly adopt the BSS searched for the backbone of Mask-RCNN-R50. By adjusting the BSS of backbone, we improve the **mIoU/mAcc/aAcc** (single scale testing) from **77.05/85.13/94.89%** to **78.22/86.50/95.18%** for PSPNet50 and from **77.25/85.69/94.91%** to **78.04/86.79/95.03%** for PSANet50.