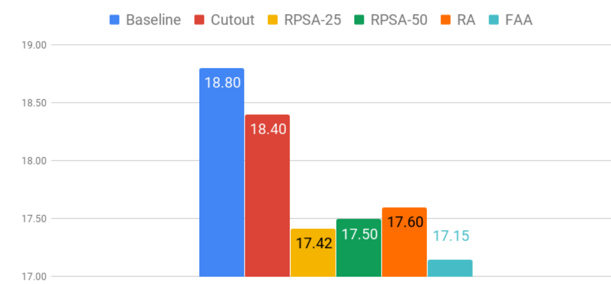
We thank all the reviewers for their efforts in reviewing our paper. We first address the common concern of all reviewers.

Common

Comparison with Random Search Experiment. The goal of Fast AutoAugment (FAA) is to propose an algorithm that can find a set of augmentation policies *faster* than AutoAugment (AA) given the *same search space*. Therefore, we addressed that the proposed FAA performs better than the random search, since AA outperforms the random search in [3] while FAA achieves similar performances to AA. However, in order to empirically clarify it, we performed additional experiments with two random search strategies, suggested by Reviewer 1 and 3, on the given search space: (1) **Randomly pre-selected augmentations (RPSA)** (suggested by Reviewer 1), which first selects a certain number (25/50) of augmentation policies randomly from the search space, and then trains a network (WResNet28x10) using the selected augmentations over 200 epochs; (2) **Random augmentations (RA)** (suggested by Reviewer 3), that independently samples an augmentation policy for each train input from the whole search space during training with 400 epochs (two times more epochs than AA and FAA).

Both the RPSA and RA are performed on CIFAR-100 and repeated 20 times. As shown in the right figure, the performances of the RPSA is better than Cutout but not improved as the number of selected (sub-)policies increases. And the best performance obtained by RPSA is still worse than FAA[1]. In addition, the RA achieves a little bit worse result than those obtained by RPSA, and the improvement by RA is also less than that by FAA. It is noted that even though we take into account the search time of FAA on CIFAR-10/100 (3.5 hours), the training time for FAA with 200 epochs including the search time is shorter than the training time for the RA with 400 epochs. We will include these experimental results in the revised paper.



Reviewer 1

Details of search strategy and Cutout. We use the classification loss (categorical cross entropy) as an evaluation measure ($\mathcal{L}$ in Equation 3) for each candidate policy. The FAA is able to select "Cutout", since "Cutout" can (probabilistically) eliminate irrelevant backgrounds and improve the classification accuracy when the inference is performed on a (well-) trained network. We will include these statements in the revised paper. **Reproducibility**. We observed the similar performance variances from the FAA when compared with AA. In addition, we omit the statement about our public source codes due to the anonymization policy. We will comment on our public source codes in the final paper.

Reviewer 2

Justification of the search objective of FAA. The proposed search objective pursues to find label-preserving transformations that generates unseen but plausible missing data samples. It is noted that the non-augmented original data samples are also taken into account by probabilistically augmenting the data space when evaluating a candidate policy. Namely, it does not transform but augment the data space which has to be correctly predicted by a classification network for better generalization. This perspective is also inline with the motivation of Bayesian DA [34]. We empirically verify this by comparisons with random searches. We will include these statements in the revised paper.

Reviewer 3

As a reviewer mentioned, the main contribution of this paper is to remove the requirement of a separate retraining from scratch for evaluating each policy, which allows to efficiently use Bayesian optimization. We will emphasize this point in the revised paper.

Number of sub-policies found by FAA. Due to the efficiency in the proposed search process, contrary to AA, the FAA can fastly find more numbers of optimized augmentation policies, almost regardless of its number. Therefore, we can consider the number of sub-policies as a hyperparameter to tune, since the training time overhead by increased number of sub-policies is also limited as shown in the below explanation. Having this in mind, we performed FAA with different numbers of sub-policies and determined the number of sub-policies that produces the best average performances across different datasets and networks. However, as shown in Figure 3 in the submitted paper, the performances obtained by 25 numbers of sub-policies are also comparable to those by more numbers of sub-policies. We will include this statement in the revised paper. **Practicality of FAA from the training time perspective**. When we use a multi-threading functionality for data augmentation as like a "DataLoader" in PyTorch, we observe that there is no actual extension of training time by augmentation from FAA in comparison to the baseline without augmentation. Moreover, even when we perform both the data augmentation and weight updating by SGD in a single thread as a sequential processing, the increased training time that we observe is only 10-20% over 200 epochs; in total, less than 5 hours on CIFAR-10/100 with WResNet28x10 and a single V100 GPU. We will include this in the revised paper.

[1] We fix the cosine scheduling for SGD and re-run the training with policies found by FAA.