OnlineAugment: Online Data Augmentation with Less Domain Knowledge

Anonymous ECCV submission

Paper ID 206

1 Supplementary Materials

To evaluate OnlineAugment[1] better, we provide three additional experiments in this section. First, we present videos to display the augmentation process during training. The curves, measuring the adaptivity of OnlineAugment, are also given. Finally, we compare the running time of different data augmentation methods.

1.1 Video Demos

We append three video demos in the supplementary materials. They demonstrate the augmentation processes of A-STN, D-VAE, and P-VAE. In each video demo, we randomly select 25 CIFAR-10 training images as anchors. Each frame shows their augmented versions at one training epoch. Specially, we use checkpoints of A-STN, D-VAE, or P-VAE at the beginning of each epoch to produce the demo images. The A-STN video has double images in each frame, where the left and right parts show the inverse affine transformations. For viewing convenience, we tag the epoch numbers at the top of videos.

We can observe diverse and adaptive data augmentations in the demo videos. A-STN can produce nearly 180-degree rotations, large scale variations, and shear effects. In the D-VAE video, smooth local deformations are observable across different areas of images. P-VAE can learn different noise patterns such as horizontal or vertical stripes, small mosaics, and lightness. All three videos share a similar feature: the augmentation strength first increases and then decreases. The videos also correspond to the adaptivity curves in Figure 1.

1.2 Illustration of OnlineAugment Adaptivity

OnlineAugment can adapt to the target learner in training. To demonstrate the adaptivity, we draw curves in Figure 1 to measure the augmentation strength. We can find the augmentations are relatively small at the first few epochs. As the training continues, the augmentations become more substantial as the target learner already learns enough knowledge from the clean data. Finally, the training converges with reduced augmentations. The target learner with a small learning rate may not require significant data augmentations. An exception is P-VAE has a considerable image distance at the beginning. The large noises are due to our initialization of the P-VAE model, which we leave for future studies.

[1] We will release the training code upon acceptance.

Table 1: GPU hours of offline searching time. AutoAugment (AA) [1], Population-based Augmentation (PBA) [3], and Fast AutoAugment (Fast AA) [4] need to search augmentation polices on separate proxy tasks. In contrast, our OnlineAugment has no offline searching cost.

Dataset	AA	PBA	Fast AA	OnlineAugment
CIFAR-10	5000	5.0	3.5	0.0
SVHN	1000	1.0	1.5	0.0
ImageNet	15000	-	450	0.0

Table 2: Per iteration seconds in online training. We measure the time using different input image resolutions and workers in Pytorch data loaders. In the experiments, we train ResNet50 [2] with batch size 128 using 1 RTX 8000 GPU and Intel(R) Xeon(R) Silver 4116 CPUs. Since all the offline methods share the same augmentation policy format, they should have equivalent online training time costs. Thus, we use AutoAugment (AA) [1] to represent all offline methods here. Ours have higher online time costs due to updating augmentation networks.

	32×32 Image (ResNet50, BS: 128)					224×224 Image (ResNet50, BS: 128)				
Workers	AA	A-STN	D-VAE	P-VAE	Comb.	AA	A-STN	D-VAE	P-VAE	Comb.
0	0.17	0.14	0.17	0.17	0.27	0.84	1.21	1.37	1.32	3.29
1	0.17	0.12	0.14	0.15	0.27	0.50	0.94	1.06	1.03	2.97

1.3 Running Time Comparisons

For fair comparisons, we divide the running time into two parts: offline searching time and online training time. The offline data augmentation methods require to learn augmentation policies on separate proxy tasks. Then they apply the policies for online training. Our OnlineAugment, by contrast, performs the online training directly. Tables 1 and 2 give offline and online time comparisons, respectively. Our s is superior in terms of zero offline time cost. For online training, ours is slower as it needs to update both the target learner and augmentation networks in each iteration. According to Table 2, the gap is smaller for small images training. Especially, A-STN is even faster in this case.

There are several possible ways to improve the training efficiency of OnlineAugment. One is to reduce the frequency of updating the augmentation networks A-STN, D-VAE, and P-VAE. Currently, we update them in each iteration of updating the target learner. The online training time will significantly decrease if updating them less frequently. Another direction is to optimize the architectures of A-STN, D-VAE, and P-VAE. We can also reduce the training costs by using more compact models.

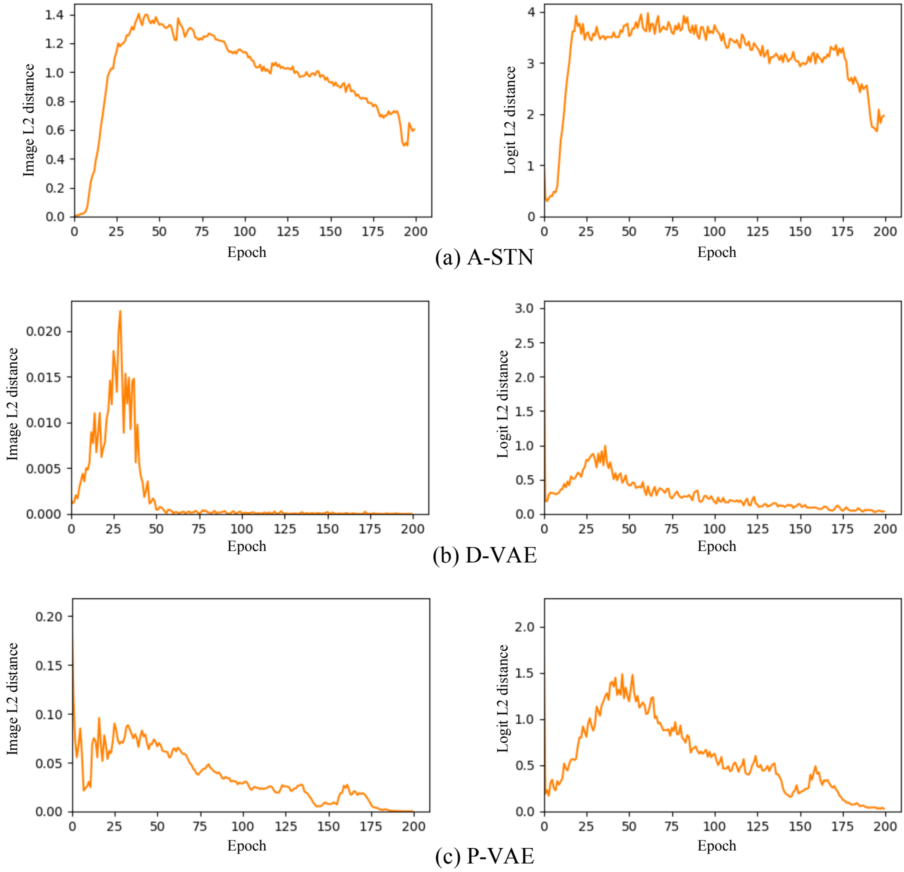

Fig. 1: Illustration of augmentation strengths for A-STN, D-VAE, and P-VAE along epochs. We measure the strengths using the L2 distances between the clean and augmented data. The **left** and **right** columns show the L2 distances in the image and logit spaces. The trend is that the augmentation strengths increase in the early stages of training, while during the target network converges, the augmentation magnitude gradually decreases.

References

1. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation strategies from data. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 113–123 (2019)
2. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
3. Ho, D., Liang, E., Stoica, I., Abbeel, P., Chen, X.: Population based augmentation: Efficient learning of augmentation policy schedules. arXiv preprint arXiv:1905.05393 (2019)
4. Lim, S., Kim, I., Kim, T., Kim, C., Kim, S.: Fast autoaugment. In: Advances in Neural Information Processing Systems. pp. 6662–6672 (2019)