

# Regularizing Meta-Learning via Gradient Dropout

Hung-Yu Tseng<sup>\*1</sup>, Yi-Wen Chen<sup>\*1</sup>, Yi-Hsuan Tsai<sup>2</sup>,  
Sifei Liu<sup>3</sup>, Yen-Yu Lin<sup>4</sup>, Ming-Hsuan Yang<sup>1,5</sup>

<sup>1</sup>University of California, Merced, <sup>2</sup>NEC Laboratories America,  
<sup>3</sup>Nvidia Research, <sup>4</sup>National Chiao Tung University, <sup>5</sup>Google Research

## 1 Overview

In this document, we first supplement the implementation details. We then present additional experimental results of reinforcement learning and cross-domain few-shot classification. Finally, we compare the proposed DropGrad regularization algorithm with the simulated annealing methods.

## 2 Implementation Details

**Few-Shot Classification.** We use the implementation from [1] to train and evaluate MAML [2] on few-shot classification tasks.<sup>1</sup> In the meanwhile, we modify the same implementation for MetaSGD [3] by ourselves. We verify our implementation by evaluating the MetaSGD model using Conv4, which is the same backbone network adopted in the original paper. The 5-way 5-shot classification results on the mini-ImageNet dataset [4] reported by our implementation and the original paper are  $65.31 \pm 0.66\%$  and  $64.03 \pm 0.94\%$ , respectively.

To train both the MAML and MetaSGD models, we keep the default settings in the original implementation by [1]. We apply the Adam [5] optimizer with the learning rate of 0.001. The mini-batch size is set to be 4. We train the model with 400 epochs and do not apply the learning rate decay strategy.

**Online Object Tracking.** We conduct experiments of online object tracking based on the PyTorch implementation by [6]. For MetaSDNet, the first three convolutional layers of VGG-16 are used as the feature extractor. During meta-training, the last three fully-connected layers are randomly initialized. We only update the last three fully-connected layers in the first 5,000 iterations, and then train the entire network for the remaining iterations. We adopt the Adam optimizer [5] with an initial learning rate of  $10^{-4}$ , and decrease the learning rate to  $10^{-5}$  after 10,000 iterations. In total, we train the network for 15,000 iterations. For MetaCREST, we use the Adam optimizer with a learning rate of  $10^{-6}$ , and train the model for 10,000 iterations.

---

<sup>\*</sup> Equal contribution

<sup>1</sup> <https://github.com/wyharveychen/CloserLookFewShot>

### 3 Reinforcement Learning

We adopt the few-shot reinforcement learning (RL) setting as in [2], which aims to make the system adapt to new experiences and learn the corresponding policy quickly with limited prior experience (*i.e.*, trajectories). In this setting, the support set  $D^s$  contains few trajectories and the corresponding rewards, while the query set  $D^q$  is formed by a set of new trajectories sampled from the running policy. We conduct the experiment with the locomotion tasks simulated by the MuJoCo [7] simulator. Two environments are considered in this experiment: HalfCheetah robot and Ant robot with forward/backward movement, *i.e.*, HalfCheetah-Dir and Ant-Dir.

**Implementation Details.** We adopt the MAML-TPRO [2] framework as the baseline method. Since the rewards are usually not differentiable, policy gradients are calculated for adapting the RL models to new experiences in both inner- and outer-loop optimization. For applying the proposed DropGrad scheme in the RL framework, we augment the policy gradients calculated according to rewards in the support set during the inner-loop optimization. We use a public PyTorch implementation with the default hyper-parameter settings in the experiments.<sup>2</sup>

**Reinforcement Learning Results.** In Figure 1, we present the rewards after the model is optimized with the few trajectories, *i.e.*, in each iteration we perform a one-step policy gradient update for the inner-loop optimization. In both environments, the training process with the proposed DropGrad regularization method converges to favorable rewards compared to the original training without the proposed regularization. This improvement could be attributed to the uncertainty on gradients that provides a better exploration of the policy.

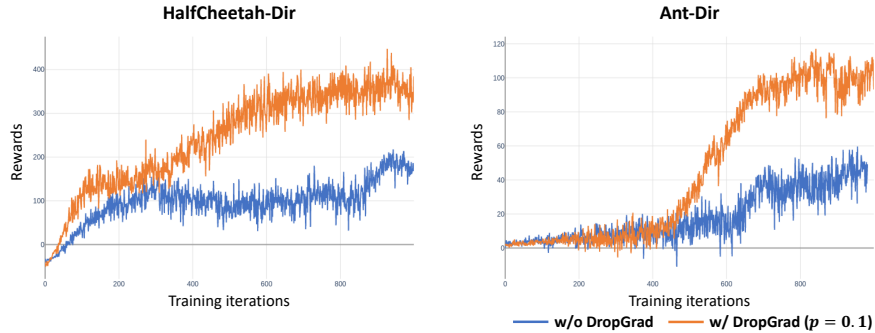
### 4 Cross-Domain Few-Shot Classification

In Figure 2, Figure 3 and Figure 4, we provide more results of the class activation maps (CAMs) [8] for the cross-domain few-shot classification task. The meta-training and meta-testing steps are conducted on the mini-ImageNet and CUB datasets, respectively. We apply the DropGrad scheme on the MAML, MetaSGD, and MetaSGD\* approaches. The results show that models trained with the proposed DropGrad regularization focus on more discriminative regions.

### 5 Comparison to Simulated Annealing

The proposed DropGrad algorithm is also related to simulated annealing (SA) [9]. While conceptually similar to a certain extent, the goals and formulations are significantly different. SA modulates gradients by exploring uncertain solutions

<sup>2</sup> <https://github.com/tristandeleu/pytorch-maml-rl>

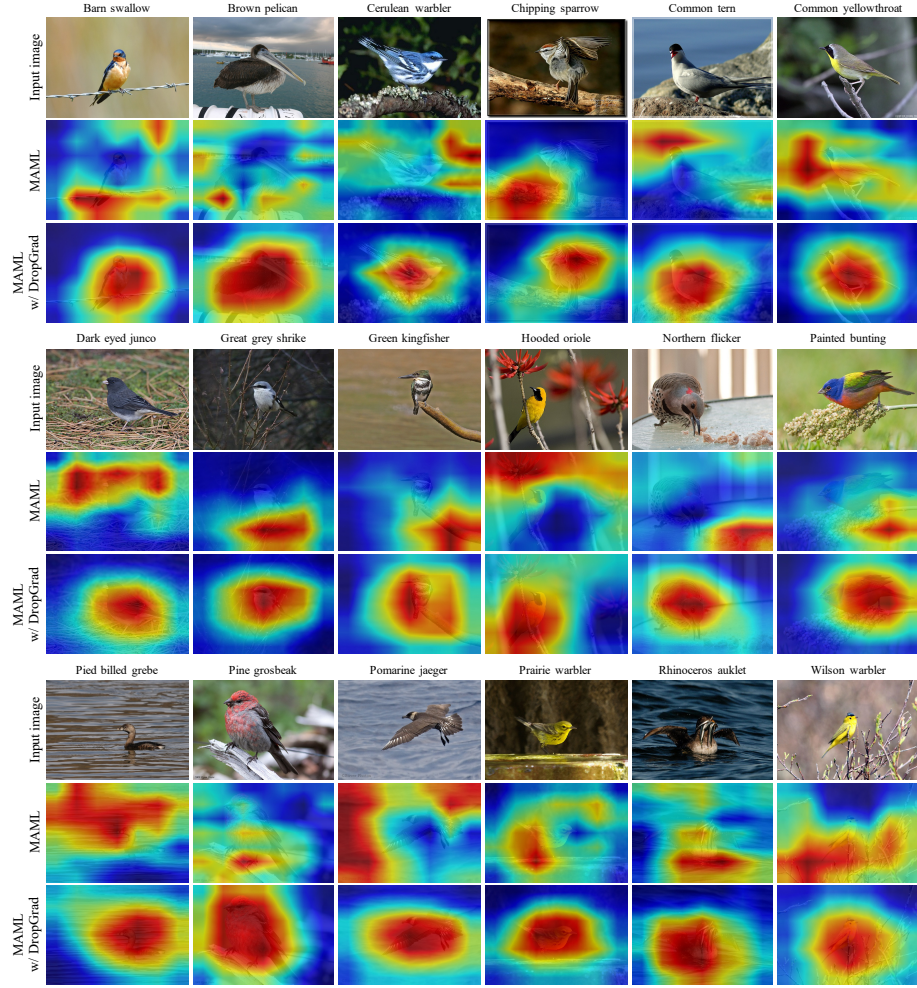


**Fig. 1. Few-shot reinforcement learning results.** Two settings, HalfCheetah-Dir (*left*) and Ant-Dir (*right*), are considered in our experiments using the MAML-TPRO framework. We show the reward curves after the model is updated with the few trajectories and both rewards converge favorably against the original training.

to escape from the local minimum during the training stage. On the other hand, our DropGrad method drops the *inner* gradient to introduce uncertainty in the forward pass of the gradient-based meta-learning framework.

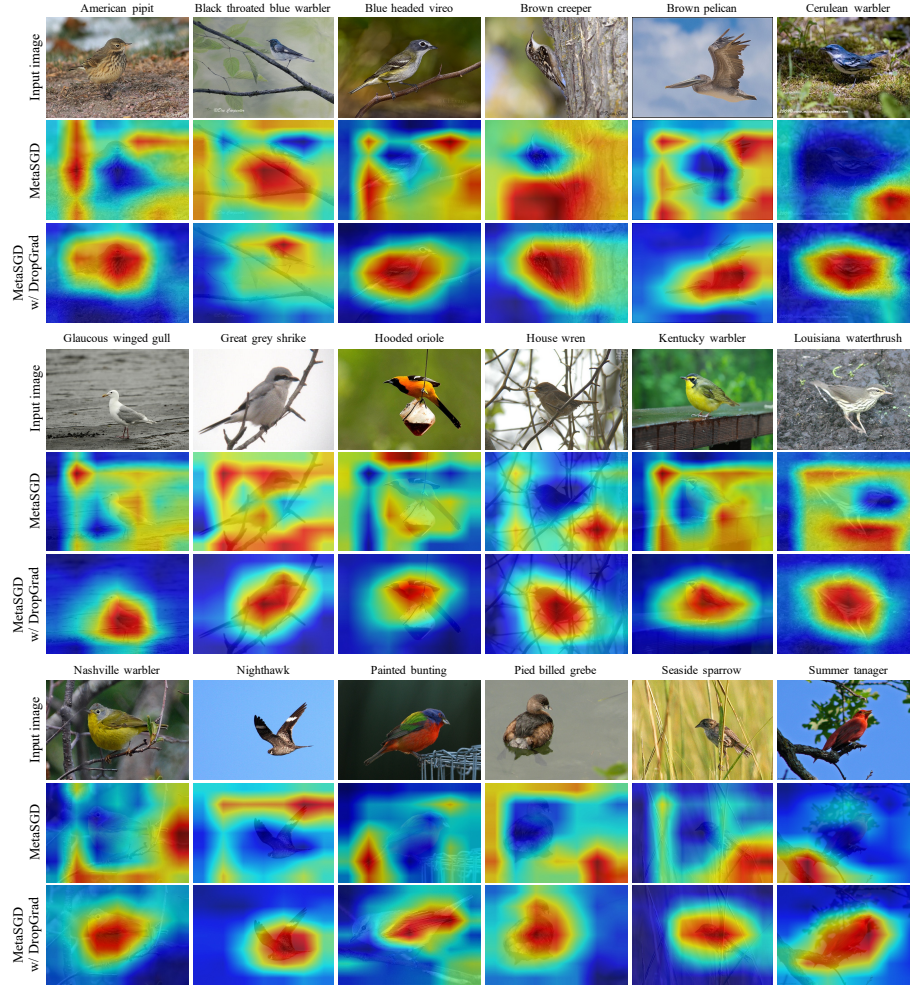
## References

1. Chen, W.Y., Liu, Y.C., Kira, Z., Wang, Y.C., Huang, J.B.: A closer look at few-shot classification. In: ICLR. (2019) 1
2. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML. (2017) 1, 2
3. Li, Z., Zhou, F., Chen, F., Li, H.: Meta-sgd: Learning to learn quickly for few shot learning. arXiv preprint arXiv:1707.09835 (2017) 1
4. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning. In: ICLR. (2017) 1
5. Kinga, D., Adam, J.B.: Adam: A method for stochastic optimization. In: ICLR. (2015) 1
6. Park, E., Berg, A.C.: Meta-tracker: Fast and robust online adaptation for visual object trackers. In: ECCV. (2018) 1
7. Todorov, E., Erez, T., Tassa, Y.: Mujoco: A physics engine for model-based control. In: IEEE International Conference on Intelligent Robots and Systems. (2012) 2
8. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: CVPR. (2016) 2
9. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. *Science* **220** (1983) 671–680 2

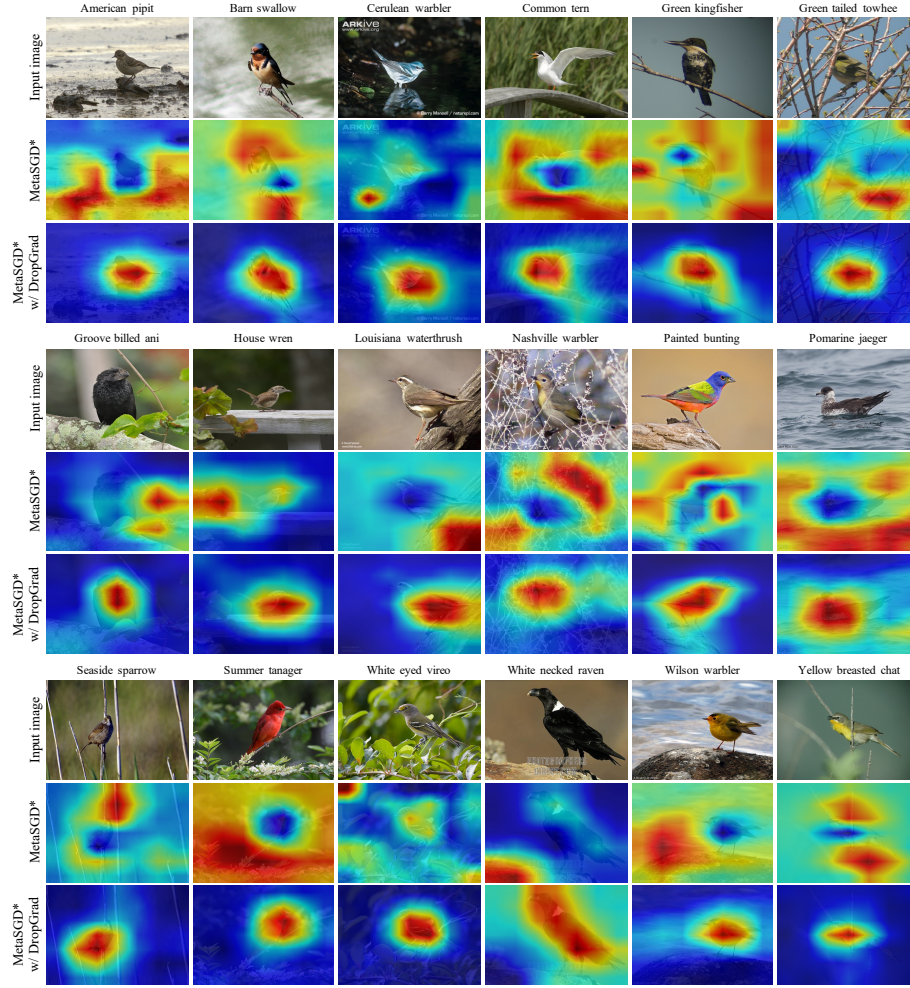


**Fig. 2. Class activation maps (CAMs) for cross-domain 5-shot classification.** The mini-ImageNet and CUB datasets are used for the meta-training and meta-testing steps, respectively. Models trained with the proposed DropGrad (the third row for each example) focus more on the objects than the original models (the second row for each example).





**Fig. 3. Class activation maps (CAMs) for cross-domain 5-shot classification.** The mini-ImageNet and CUB datasets are used for the meta-training and meta-testing steps, respectively. Models trained with the proposed DropGrad (the third row for each example) focus more on the objects than the original models (the second row for each example).



**Fig. 4. Class activation maps (CAMs) for cross-domain 5-shot classification.** The mini-ImageNet and CUB datasets are used for the meta-training and meta-testing steps, respectively. Models trained with the proposed DropGrad (the third row for each example) focus more on the objects than the original models (the second row for each example).