REPURPOSING PRETRAINED MODELS FOR ROBUST OUT-OF-DOMAIN FEW-SHOT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Model-agnostic meta-learning (MAML) is a popular method for few-shot learning but assumes that we have access to the meta-training set. In practice, training on the meta-training set may not always be an option due to data privacy concerns, intellectual property issues, or merely lack of computing resources. In this paper, we consider the novel problem of repurposing pretrained MAML checkpoints to solve new few-shot classification tasks. Because of the potential distribution mismatch, the original MAML steps may no longer be optimal. Therefore we propose an alternative meta-testing procedure and combine MAML gradient steps with adversarial training and uncertainty-based stepsize adaptation. Our method outperforms "vanilla" MAML on same-domain and cross-domains benchmarks using both SGD and Adam optimizers and shows improved robustness to the choice of base stepsize.

1 INTRODUCTION

Deep learning approaches have shown improvements based on massive datasets and enormous computing resources. Despite their success, it is still challenging to apply state-of-the-art methods in the real world. For example, in semiconductor manufacturing (Nishi & Doering, 2000), collecting each new data point is costly and time consuming because it requires setting up a new manufacturing process accordingly. Moreover, in the case of a "destructive inspection", the cost is very high because the wafer must be destroyed for measurement. Therefore, learning from small amounts of data is important for practical purposes.

Meta-learning (learning-to-learn) approaches have been proposed for learning under limited data constraints. A meta-learning model optimizes its parameters for the best performance on the distribution of tasks. In particular, few-shot learning (FSL) formulates "learning from limited data" as an n-way k-shot problem, where n is the number of classes and k is the number of labeled samples per class. For each task in FSL, a support set is provided for training, while a query set is provided for evaluation. Ideally, a meta-learning model trained over a set of tasks (meta-training) will generalize well to new tasks (meta-testing).

Model-agnostic meta-learning (MAML) (Finn et al., 2017) is a general end-to-end approach for solving few-shot learning tasks. MAML is trained on the meta-training tasks to learn a model initialization (also known as *checkpoint*) such that a few gradient steps on the support set will yield the best predictions on the query set.

However, in practice it may not always be possible to retrain or finetune on the meta-training set. This situation may arise when the meta-training data is confidential, subject to restrictive licences, contains private user information, or protected intellectual property such as semiconductor manufacturing know-how. Another reason is that one may not have the computing resources necessary for running large-scale meta-training.

In this paper, we consider the novel problem of *repurposing* MAML checkpoints to solve new few-shot classification tasks, *without* the option of (re)training on the meta-training set. Since the meta-testing set (new tasks) may differ in distribution from the meta-training set, the implicit assumption –in end-to-end learning– of identically distributed tasks may not hold, so there is no reason why the meta-testing gradient steps should match the meta-training. Therefore, we investigate various improvements over the default MAML gradient steps for test time adaptation.

Conceptually, our approach consists of collecting information while training the model on a new support set and then proposing ways to use this information to improve the adaptation. In this paper, we consider the variance of model parameters during ensemble training as a source of information to use. We propose algorithms that uses this information both to *adapt* the stepsizes for MAML as well as to generate "task-specific" adversarial examples to help robust adaptation to the new task.

Our **main contributions** are the following:

- We motivate the novel problem of *repurposing* MAML checkpoints to solve cross-domain few-shot classification tasks, in the case where the meta-training set is *inaccessible*, and propose a method based on uncertainty-based stepsize adaptation and adversarial data augmentation, which has the particularity that meta-testing differs from meta-training steps.
- Compared to "vanilla" MAML, our method shows improved accuracy and robustness to the choice of base stepsizes on popular cross-domain and same-domain benchmarks, using both the SGD and Adam (Kingma & Ba, 2014) optimizers. Our results also indicate that adversarial training (AT) is helpful in improving the model performance during meta-testing.

To the best of our knowledge, our work is the first few-shot learning method to combine the use of ensemble methods for stepsize computation and generating adversarial examples from the meta-testing support set for improved robustness. Moreover, our empirical observation of improving over the default meta-testing procedure of MAML motivates further research on alternative ways to leverage published model checkpoints.

2 Related Work

2.1 Meta-learning and Model-agnostic meta-learning

FSL approaches deal with an extremely small amount of training data and can be classified into three categories. First, metric-based approaches solve few-shot tasks by training a feature extractor that maximizes inter-class similarity and intra-class dissimilarity (Vinyals et al., 2016; Snell et al., 2017; Sung et al., 2018). Second, memory-based approaches utilize previous tasks for new tasks with external memories (Santoro et al., 2016; Mishra et al., 2018). Third, optimization-based approaches search for good initial parameters during training and adapt the pretrained model for new tasks (Finn et al., 2017; Lee & Choi, 2018; Grant et al., 2018). We focus on the optimization-based approaches and suggest better training especially for the general family of MAML methods.

2.2 Uncertainty for model training

Uncertainty is an important criterion for measuring the robustness of a neural network (NN). Bayesian neural networks (BNN) (Blundell et al., 2015) obtain model uncertainty by placing prior distributions over the weights $p(\omega)$. This uncertainty has been used to adapt the stepsizes during continual learning in Uncertainty-guided Continual BNNs (UCB) (Ebrahimi et al., 2020). For each parameter, UCB scales its stepsize inversely proportional to the uncertainty of the parameter in the BNN to reduce changes in important parameters while allowing less important parameters to be modified faster in favor of learning new tasks. Our approach also decreases the stepsizes for "uncertain" parameters, but using a different notion of uncertainty, and instead in the context of FSL with a pretrained MAML checkpoint.

Unlike BNNs, deep ensembles (Lakshminarayanan et al., 2017) estimate uncertainty of NNs using ensembles over randomly initialized parameters combined with adversarial training (AT). Deep ensembles have been successfully used to boost predictive performance and AT has been used to improve robustness to adversarial examples. Lakshminarayanan et al. (2017) showed that they produce predictive uncertainty estimates comparable in quality to BNNs. Deep ensembles, however, are not directly applicable for FSL tasks, as training randomly initialized parameters from scratch with a limited amount of training data yields poor performance. Our approach is partly inspired from deep ensembles, adapting it to the FSL setting by using instead a parameter perturbations of the MAML checkpoint model rather than a random initialization. We use in particular a multiplicative Gaussian perturbation that rescales the parameters, as the information content of the weights is said to be invariant to their scale (Wen et al., 2018).

2.3 ADVERSARIAL TRAINING FOR ATTACK AND DEFENSE

Deep NNs are sensitive to adversarial attacks (Goodfellow et al., 2015; Madry et al., 2018; Moosavi-Dezfooli et al., 2016). These methods generate an adversarial sample to fool a trained model, where the generated image looks identical to the original one for humans. Goodfellow et al. (2015) proposed the fast gradient sign method (FGSM) which generates adversarial example using sign of input gradient. While stronger attacks have been proposed (such as using projected gradient descent (Madry et al., 2018)), we will focus on the FGSM approach in this paper for simplicity. In FSL, adversarial attack and defense have also been studied. ADversarialMeta-Learner (Yin et al., 2018) utilized one-step adversarial attack for generating adversarial samples during meta-training. There is little consideration, however, for the degradation of accuracy in the original sample in adversarial defense approaches. Xie et al. (2020) reported improvement using generated examples for adversarial attack on the large scale training. Motiian et al. (2017) propose adversarial domain adaptation for FSL. Despite similar keywords, their work is distinct from ours and relies on using generative adversarial networks *discriminators* to confuse domains, whereas we rely on adversarial *examples* for improved robustness. To our best knowledge, there is no prior FSL work which uses adversarial examples to enhance the model at test-time.

2.4 DOMAIN ADAPTATION

Domain adaptation methods (DA) (Ben-David et al., 2010) attempt to alleviate the distribution mismatch between source and target domains. Most of the recently proposed DA approaches are based on generative adversarial networks (GANs) (Goodfellow et al., 2014). GAN-based DA approaches require having access to the large amount of unlabeled data from both the source and target datasets (Tzeng et al., 2017; Zhang et al., 2019; Wilson & Cook, 2020). DA methods cannot be directly applied in the FSL scenario due to limited number of target domain samples (*shots*). Some domain-adaptive FSL (DA-FSL) methods have been proposed in the case where only a very few samples from the target domain are available (Motiian et al., 2017; Zhao et al., 2020).

DA and DA-FSL methods cannot be directly applied to our setting because we assume that the meta-training dataset is *inaccessible* — mirroring real world situations in which access to the meta-training set is restricted by privacy and confidentiality concerns. Our approach differs from DA and DA-FSL by not requiring access to the meta-training dataset (source domain).

3 PROPOSED METHOD

At meta-testing time, MAML normally uses the support set to compute *fixed* gradient steps, which were "calibrated" using end-to-end learning during meta-training. That is, the learned initialization is such that a fixed combination of *stepsize* and *loss* result in the desired result. However, those stepsizes and losses may be suboptimal on the new task, especially if the new task is out-of-domain with respect to the meta-training tasks.

Our method is based on the assumption that the support set can be used to improve the meta-testing procedure itself, beyond merely serving as training examples. We start by leveraging the support set to estimate task-specific uncertainties over the model parameters. Then, we propose two improvements over the "vanilla" MAML gradient steps : we *scale* the gradient steps using layer-wise stepsizes computed from the support set and we train using *task-specific adversarial examples*.

3.1 MOTIVATING HYPOTHESES

At meta-testing time, we start by assuming that we can estimate *task-specific uncertainties* over the model parameters. One possibility, which we adopt in Section 3.2, is to train deep ensembles (Lakshminarayanan et al., 2017) on the support set and use the ensemble to estimate variances over model parameters. Each model learns slightly different parameters, and yields slightly different gradients. We regard the variance over the parameters and input gradients as task-specific uncertainties. Given the uncertainty estimates, we propose two modifications to the original MAML gradient steps.

Proposal 1 (Task-specific stepsizes) ***Use lower stepsizes for model parameters with higher variance.***

In our case, the variance could be further amplified if high-variance components were to be moved with large stepsizes. Therefore, we carefully move high-variance components with a lower stepsize with the hope is that we can limit the variance over the model parameters. This approach can be related with the fact that lower stepsizes should be taken for SGD when the gradients are very noisy (Dieuleveut et al., 2020).

Proposal 2 (Task-specific adversarial examples) ***Use adversarial examples with higher adversarial perturbation on input components with higher variance over the input gradient.***

The intuition is that if only slightly perturbed models (from the ensemble) disagree on parts of the input gradient, then it means that they disagree on what to learn and therefore those parts of the input are more vulnerable to attack. Therefore, we propose to use AT with stronger adversarial perturbation in the weak parts of the input, with the hope to incur improved robustness on those parts of the input.

We regard adversarial training as a form of *data augmentation* or *regularization* at meta-testing time, which we hope allows the model to overcome the limited size of the support set.

3.2 Uncertainty-based gradient steps at test-time

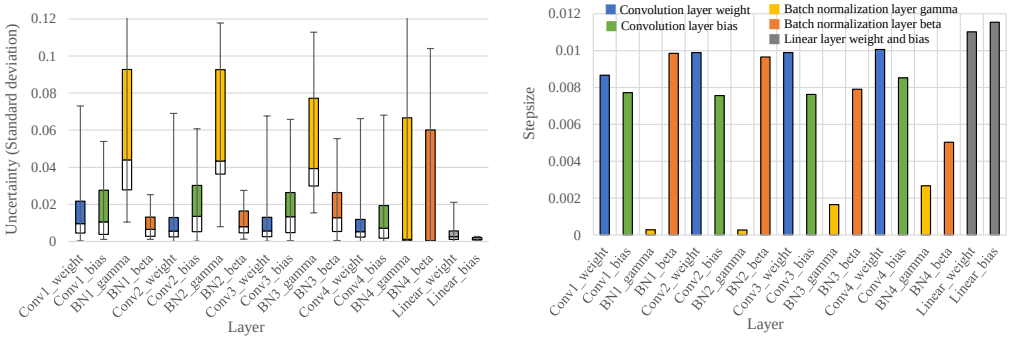
We improve over the default MAML gradient steps by implementing the ideas presented in the previous section. The resulting approach is detailed in **Algorithm 1** and is a combination of uncertainty-based stepsize adaption (USA, based on Proposal 1) and uncertainty-based fast gradient sign method (UFGSM), adversarial training (AT), and generating additional adversarial examples from deep ensembles (EnAug) which are based on Proposal 2.

Algorithm 1: Uncertainty-based Gradient Steps at Test-time

Data: New task support set $D^{\text{Spt}} = \{(x_1, y_1), \ldots, (x_{n\times k}, y_{n\times k})\}$ with n ways and k shots.
Require: Base stepsize α, pretrained weights θ_0, Gaussian variance σ, AT coefficient ϵ, size of ensemble M, number of gradient steps T, selection coefficients in $\{0, 1\}$ for: base stepsize λ_α, adversarial loss λ_{AT}, and augmented cross-entropy λ_{Aug} .

$D^{\text{Aug}} \leftarrow \emptyset$, $\alpha^{\text{adap}} \leftarrow \alpha$
for $m = 1$ **to** M **do**
  $\theta_0^m \leftarrow \theta_0(1 + \mathcal{N}(0, \sigma))$ ▷`Initialize deep ensemble`
end
for $t = 1$ **to** T **do**
  for $m = 1$ **to** M **do**
    $\theta_t^m \leftarrow \theta_{t-1}^m - \alpha^{\text{adap}} \odot \nabla_{\theta_{t-1}^m}[\mathcal{L}(D^{\text{Spt}}, \theta_{t-1}^m) + \mathcal{L}_{\text{AT}}(D^{\text{Spt}}, \theta_{t-1}^m)]$
    $D^{\text{Aug}} \leftarrow D^{\text{Aug}} \cup \{(\mathcal{A}_{\theta_{t-1}^m}(x, y), y) \,|\, (x, y) \in D^{\text{Spt}}\}$ ▷`EnAug`
  end
  $\alpha^{\text{adap}} \leftarrow \texttt{USA}(\alpha, \theta_t^{1:M})$ ▷`Algorithm 2`
  $D^{\text{Aug}} \leftarrow D^{\text{Aug}} \cup \{\texttt{UFGSM}(\theta_0, \theta_t^{1:M}, x, y) \,|\, (x, y) \in D^{\text{Spt}}\}$ ▷`Algorithm 3`
end
$\alpha \leftarrow \lambda_\alpha \alpha + (1 - \lambda_\alpha)\alpha^{\text{adap}}$
for $t = 1$ **to** T **do**
  $\theta_t \leftarrow \theta_{t-1} - \alpha \odot \nabla_{\theta_{t-1}}[\mathcal{L}(D^{\text{Spt}}, \theta_{t-1}) + \lambda_{\text{AT}}\mathcal{L}_{\text{AT}}(D^{\text{Spt}}, \theta_{t-1}) + \lambda_{\text{Aug}}\mathcal{L}(D^{\text{Aug}}, \theta_{t-1})]$
end

Denote $\mathcal{L}(D, \theta) = 1/|D| \sum_{(x,y)\in D} l_\theta(x, y)$ the cross-entropy loss for model θ over the labeled dataset $D = \{(x, y) \ldots\}$, $\mathcal{A}_\theta(x, y) = x + \epsilon \texttt{sign}(\nabla_x l_\theta(x, y))$ the FGSM adversarial example computed from (x, y) and $\mathcal{L}_{\text{AT}}(D, \theta) = 1/|D| \sum_{(x,y)\in D} l_\theta(\mathcal{A}_\theta(x, y), y)$ the resulting adversarial



(a) Uncertainty of layers

(b) Uncertainty-based stepsize adaption

Figure 1: USA converts uncertainty into layer-wise adapted stepsize. Here $\alpha = 0.01$. (a) is standard deviation of trained ensemble models' weights. Each layer has a difference uncertainty. (b) is adapted stepsize by USA. Each layer has a different stepsize based on the uncertainty.

cross-entropy, which we refer to as **AT**. For easy reference, all notations are summarized in **Appendix A.1**.

Starting from the pretrained MAML checkpoint θ_0, we perturb the model parameters with multiplicative Gaussian noise to create a deep ensemble $(\theta_0^m)_{1 \leq m \leq M}$ (**lines 2-4**). Then, we repeat the following for T steps. At time t, run gradient descent on each model θ_t^m of the ensemble (**line 7**), where the *loss* is a combination of the usual cross-entropy $\mathcal{L}$ and AT loss $\mathcal{L}_{\text{AT}}$ as in (Lakshminarayanan et al., 2017) but on the support set, and the stepsizes α^{adap} are updated using **USA** (details below). Also, we generate adversarial examples using FGSM and **UFGSM** (details below) and store them into D^{Aug} (**lines 8 and 11**). Finally, we run T gradient steps on the original checkpoint, where the loss is a combination of cross-entropy on the support set D^{Spt}, AT loss on the support set, and cross-entropy on the ensemble-augmented support set D^{Aug} (**lines 13-16**). Note that we recover the original MAML steps for ($\lambda_{\text{AT}} = \lambda_{\text{Aug}} = 0, \lambda_\alpha = 1$), while we refer to the case ($\lambda_{\text{AT}} = \lambda_{\text{Aug}} = 1, \lambda_\alpha = 0$) as our full method.

USA. We propose *uncertainty-based stepsize adaptation* (USA), which assigns lower stepsizes to layers[1] with higher uncertainty (**Algorithm 2**) – we call this loosely an "inverse-relationship" below. Let α denote the default (scalar) stepsize and $\theta^{1:M}$ the parameters of the ensemble models, where M is the size of the ensemble and the number of layers in model is L. To compute the adapted layer-wise stepsizes α^{adap}, we compute u, the parameter-wise standard deviation of the model parameters over the ensemble (**line 2**), apply an inverse-relationship transformation which flips the max with the min (**line 3**), average over each layer (**line 4**) and L_1-normalize the result (**line 5**). The design choices for USA is explained in **Appendix A.2**. The resulting stepsizes α^{adap} have an inverse-relationship with the variance of each layer. We give an example of applying USA to the 4-ConvNet architecture on *mini*ImageNet (**Figure 1**). On the left, we plot the layer-wise standard deviations of the parameters and on the right the corresponding USA stepsizes, which follow an inverse relationship with the standard deviations.

[1] In this study, we choose to use layer-wise stepsizes, but it is also possible to use kernel-wise or parameter-wise stepsizes. The motivation for using layer-wise stepsizes is because features from the same layer tend to have the same level of abstraction (Zeiler & Fergus, 2014).

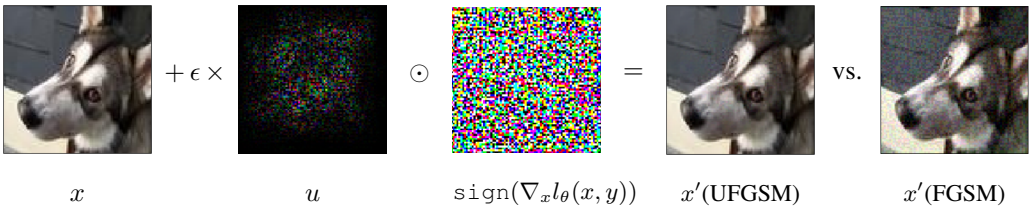


Figure 2: Applying UFGSM to 4-ConvNet on *mini*ImageNet with $\epsilon = 0.05$. Starting from the clean image x, we add the signed gradient $\texttt{sign}(\nabla_x l_\theta(x,y))$ after rescaling it by the uncertainty over the input gradient u, to generate the adversarial example x'(UFGSM). Note how UFGSM generated a more natural image than FGSM (rightmost, $u = 1$).

Algorithm 2: USA

Function $\texttt{USA}(\alpha,\ \theta_t^{1:M})$
  $u = \texttt{Std}(\theta_t^{1:M})$
  $c = \texttt{Max}(u) - u + \texttt{Min}(u)$
  $\mu^l \leftarrow$ average of c over each layer l
  $\alpha^{\text{adap}} = \alpha\mu^l/(1/L\sum_{l=1}^{L}\mu^l)$
  return α^{adap}
end

Algorithm 3: UFGSM

Define: $\texttt{MinMaxNorm}(x) = \frac{x-\texttt{Min}(x)}{\texttt{Max}(x)-\texttt{Min}(x)}$

1 **Function** $\texttt{UFGSM}(\theta,\ \theta_t^{1:M},\ x,\ y)$
2 $u = \texttt{Std}(\{\nabla_x l_{\theta'}(x,y) \mid \theta' \in \theta_t^{1:M}\})$
3 $u = \texttt{MinMaxNorm}(u)$
4 $x' = x + \epsilon u \odot \texttt{sign}(\nabla_x l_\theta(x,y))$
5 **return** x'
6 **end**

UFGSM. We also propose *uncertainty-based FGSM* (UFGSM) to generate adversarial examples, with higher adversarial perturbation on input components with higher variance over the input gradient (**Algorithm 3**). Starting from an image x and label y, we compute the input gradient for each model from the ensemble and calculate the standard deviation u over the ensemble (**line 2**). Then, we linearly map u between 0 and 1 (**line 3**), compute the FGSM adversarial example for the pre-trained model θ_0 and rescale it using u, so that areas of higher variance get more perturbation. We give an example of applying UFGSM to *mini*ImageNet in **Figure 2**.

4 Experimental Evaluation

We train MAML on *mini*ImageNet (Vinyals et al., 2016) training split; we then apply our method on the resulting checkpoint. We evaluate our model on the test split of *mini*ImageNet – for the same-domain setting – as well as CUB-200-2011 (Welinder et al., 2010), Traffic Sign (Houben et al., 2013) and VGG Flower (Nilsback & Zisserman, 2008) – for the cross-domain setting. These datasets are denoted as Mini, Birds, Signs and Flowers respectively. A desirable feature for an optimizer is to maintain good performance in a broad range of stepsizes (Asi & Duchi, 2019). Therefore, we evaluate our approach not only at the optimal stepsize, but also over a broad range of base stepsizes[2] from 10^{-4} to 1. We evaluate the performance with three metrics: All, Top-1 and Top-40%.[3] If two methods have comparable Top-1 performance, but one has better Top-40% performance, then that method is more robust to the choice of base stepsize. Detailed experimental setup and pretrained model selection are included in **Appendix A.3**. Our code is available at `https://anonymous.4open.science/r/08ef52cf-456a-4e36-a408-04e1ad0bc5a9/` (anonymous project).

4.1 Main Results

Our main results are in **Table 1**. We compare the default MAML steps, which we take as our baseline (denoted as SGD), with our full method (denoted as SGD+All), which consists of combining USA, UFSGM, EnAug, and AT ($\lambda_{\text{AT}} = \lambda_{\text{Aug}} = 1, \lambda_\alpha = 0$ in **Algorithm 1**). In terms of absolute performance (Top-1 row), our method outperforms the baseline on cross-domain tasks (Birds,

[2]We determine the ranges of stepsize based on training performance on *mini*ImageNet and keep it the same for other datasets. We selected the minimum and maximum stepsize where performance decreased drastically.

[3]All is the average accuracy over all stepsizes. Top-1 is the accuracy of the best performing stepsize. Top-40% is the average of the top 40% accuracies among all the stepsizes

Table 1: Main results. We compare the default MAML steps (SGD) with our method (SGD+All) on same-domain and cross-domain benchmarks.

Metric	Dataset	5-way 1-shot		5-way 5-shot		10-way 1-shot	
		SGD	SGD+All	SGD	SGD+All	SGD	SGD+All
All	Mini	**37.21**±0.22	36.24±0.22	38.14±0.39	**43.48**±0.38	21.31±0.37	**22.07**±0.4
	Birds	**30.07**±0.52	29.83±0.46	33.29±0.29	**38.07**±0.33	17.54±0.06	**17.81**±0.12
	Flowers	40.54±0.48	**40.86**±0.45	44.55±0.43	**53.10**±0.55	26.79±0.41	**28.07**±0.41
	Signs	34.19±0.76	**34.76**±0.61	45.39±0.45	**51.02**±0.5	23.77±0.13	**24.88**±0.13
	Avg.	**35.50**±0.49	35.42±0.44	40.34±0.39	**46.42**±0.44	22.35±0.24	**23.21**±0.26
Top-1	Mini	**46.71**±0.81	46.62±0.58	**62.46**±0.68	62.16±0.66	31.10±0.66	**31.42**±0.82
	Birds	36.03±0.48	**36.43**±0.58	50.71±0.54	**51.82**±0.64	23.16±0.21	**23.61**±0.39
	Flowers	51.97±0.79	**54.36**±1.04	70.63±0.76	**74.52**±0.69	38.61±0.49	**40.68**±0.58
	Signs	43.44±0.99	**44.10**±1.03	72.52±0.84	**74.13**±0.90	35.10±0.21	**35.11**±0.26
	Avg.	44.54±0.77	**45.37**±0.81	64.08±0.71	**65.66**±0.72	31.99±0.39	**32.70**±0.51
Top-40%	Mini	45.07±0.52	**45.79**±0.44	55.38±0.46	**60.70**±0.91	28.65±0.64	**30.56**±0.77
	Birds	34.88±0.55	**35.77**±0.55	45.50±0.49	**51.03**±0.63	21.78±0.07	**22.87**±0.30
	Flowers	49.98±0.69	**53.16**±0.73	65.16±0.73	**73.55**±0.86	36.14±0.53	**39.75**±0.61
	Traffic	41.81±1.02	**43.39**±1.01	67.34±0.43	**71.45**±0.90	33.56±0.11	**34.27**±0.17
	Avg.	42.94±0.70	**44.52**±0.68	58.34±0.53	**64.18**±0.82	30.03±0.34	**31.86**±0.46

Flowers, Signs), while the performance is comparable on same-domain tasks (Mini). In terms of robustness to the base stepsize (All and Top-40% rows), our method outperforms the baseline over a large range of stepsizes for 5-way 5-shot and 10-way 1-shot tasks, while for 5-way 1-shot the results are either comparable (All) or better (Top-40%) depending on the metric considered. More results can be found in **Appendix A.6**.

4.2 Discussion

Ablation Study. We perform an ablation study for the 5-way 1-shot case in **Table 2** and plot the accuracy at different base stepsizes for the Flowers dataset in **Figure 3**. Comparing SGD to SGD+AT in Table 2, we observe that adversarial training is beneficial over the baseline (Top1 and Top-40%), except for large stepsizes with SGD, which is reflected in the All metric and in the dip in performance in Figure 3a. Notice also how the use of AT flattens the accuracy curve near the optimal stepsize. Comparing SGD to SGD+USA, we observe a very small but consistent improvement over the baseline in the vicinity of the optimal stepsize (Top-1 and Top-40%). Comparing SGD+USA to SGD+USA+UFGSM, we observe improvement over a wide range of stepsizes, which is reflected in the curves and in All and Top-40% metrics. Comparing SGD+USA+UFGSM+EnAug+AT to SGD+USA+UFGSM+EnAug shows that AT+EnAug is beneficial in terms of absolute performance (Top-1) as well as creating robustness to the choice of stepsize (Top-40%). The drop for the All metric is explained by the dip in the curve for the largest stepsizes. Overall, the best absolute performance (Top-1) is always obtained through some use of *adversarial training*, while using the full method consistently results in *increased robustness* to the choice of stepsize (best Top-40% performance). **Appendix A.5** shows various AT results.

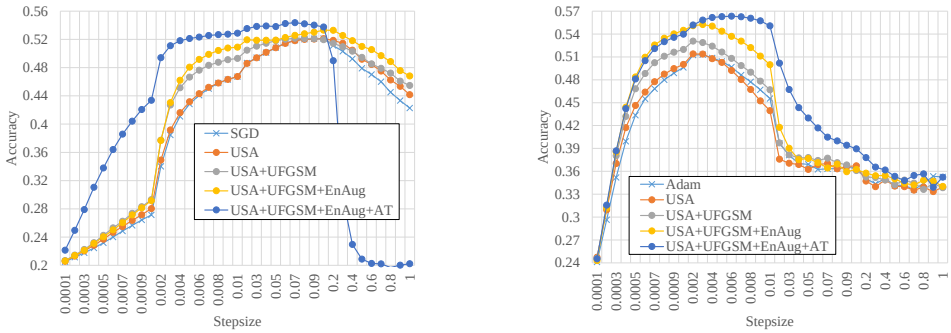
UFGSM vs. FGSM. We compare UFGSM to FGSM (Goodfellow et al., 2015) in **Table 3**. The results show that our uncertainty-based approach consistently outperforms FGSM, which suggests that the uncertainty information extracted from the support set was useful.

SGD vs. Adam. Our method can be used with SGD and Adam, however, we have mainly focused on SGD throughout the paper. SGD tends to yield the best results (Top-1 and Top-40%), as shown in **Table 4** for the baseline and full method. More results for Adam can be found in Sections A.6 and A.8 of the appendix.

Best checkpoint vs. Overfitted checkpoint. We also consider the problem of repurposing an overfitted checkpoint. We find that our method improves both absolute performance and robustness to stepsize. In fact, the gains are more substantial on the overfitted (worse) checkpoint than the best checkpoint (**Appendix A.8**).

Table 2: Ablation Study for 5-way 1-shot classification with SGD optimizer. SGD+AT corresponds to using ($\lambda_{\text{AT}} = \lambda_\alpha = 1, \lambda_{\text{Aug}} = 0$) in Algorithm 1, SGD+USA to ($\lambda_{\text{AT}} = \lambda_{\text{Aug}} = \lambda_\alpha = 0$), SGD+USA+UFGSM to ($\lambda_{\text{AT}} = \lambda_\alpha = 0, \lambda_{\text{Aug}} = 1$) and SGD+USA+UFGSM+EnAug+AT to ($\lambda_{\text{AT}} = \lambda_{\text{Aug}} = 1, \lambda_\alpha = 0$).

Metric	Dataset	5-way 1-shot				
		SGD	SGD+AT	SGD+USA	SGD+USA +UFGSM	SGD+USA +UFGSM +EnAug+AT
All	Mini	37.21±0.22	34.60±0.18	37.53±0.27	**38.71±0.28**	36.24±0.22
	Birds	30.07±0.52	28.81±0.45	30.39±0.53	**31.07±0.50**	29.83±0.46
	Flowers	40.54±0.48	38.80±0.44	41.09±0.49	**42.33±0.60**	40.86±0.45
	Signs	34.19±0.76	33.42±0.59	35.02±0.82	**36.10±0.48**	34.76±0.61
	Avg.	35.50±0.49	33.91±0.41	36.01±0.36	**37.05±0.27**	35.42±0.44
Top-1	Mini	46.71±0.81	46.58±0.65	**46.72±0.71**	46.66±0.77	46.62±0.58
	Birds	36.03±0.48	36.40±0.48	36.08±0.49	36.02±0.48	**36.43±0.58**
	Flowers	51.97±0.79	**54.55±0.84**	52.01±0.77	52.30±0.77	54.36±1.04
	Signs	43.43±0.99	43.98±0.98	43.42±1.06	43.96±0.64	**44.10±1.03**
	Avg.	44.54±0.77	**45.38±0.74**	44.56±0.44	44.73±0.41	45.37±0.81
Top-40%	Mini	45.07±0.52	45.53±0.50	45.23±0.59	45.76±0.62	**45.79±0.44**
	Birds	34.88±0.55	35.57±0.60	35.17±0.55	35.30±0.48	**35.77±0.55**
	Flowers	49.98±0.69	52.92±0.68	50.39±0.71	51.07±0.67	**53.16±0.73**
	Signs	41.83±1.02	43.19±0.96	42.08±1.06	43.18±0.73	**43.39±1.01**
	Avg.	42.94±0.70	44.30±0.68	43.22±0.43	43.83±0.39	**44.52±0.68**



(a) With SGD optimizer

(b) With Adam optimizer

Figure 3: Ablation study for 5-way 1-shot classification on Flowers dataset.

Table 3: Comparing UFGSM against FGSM (Goodfellow et al., 2015) on 5-way 1-shot tasks. For all metrics and datasets, the proposed uncertainty-based method is better.

Metric	Dataset	SGD+USA	
		+UFGSM	+FGSM
All	Mini	**38.71**±0.28	36.97±0.54
	Birds	**31.07**±0.50	30.05±0.55
	Flowers	**42.33**±0.60	41.17±0.55
	Signs	**36.10**±0.48	34.78±0.50
	Avg.	**37.05**±0.27	35.74±0.27
Top-1	Mini	**46.66**±0.77	43.93±0.85
	Birds	**36.02**±0.48	34.52±0.74
	Flowers	**52.30**±0.77	50.63±0.84
	Signs	**43.96**±0.64	42.62±1.00
	Avg.	**44.73**±0.41	42.93±0.47
Top-40%	Mini	**45.76**±0.62	43.18±0.87
	Birds	**35.30**±0.48	33.88±0.75
	Flowers	**51.07**±0.67	49.47±0.69
	Signs	**43.18**±0.73	41.92±0.90
	Avg.	**43.83**±0.39	42.11±0.46

Table 4: Comparing SGD vs. Adam for default MAML steps (baseline) and our method (baseline+all) on 5-way 1-shot classification.

Metric	Dataset	Baseline		Baseline+All	
		SGD	Adam	SGD	Adam
All	Mini	**37.21**±0.22	31.87±0.20	**36.24**±0.22	33.57±0.29
	Birds	**30.07**±0.52	28.02±0.39	**29.83**±0.46	29.51±0.46
	Flowers	**40.54**±0.48	40.33±0.57	40.86±0.45	**44.58**±0.70
	Signs	**34.19**±0.76	33.98±0.43	34.76±0.61	**35.10**±0.43
	Avg.	**35.50**±0.49	33.55±0.22	35.42±0.44	**35.69**±0.29
Top-1	Mini	**46.71**±0.81	43.08±0.10	**46.62**±0.58	44.72±0.35
	Birds	**36.03**±0.48	34.00±0.63	**36.43**±0.58	35.45±0.67
	Flowers	**51.97**±0.79	51.28±0.47	54.36±1.04	**56.04**±1.14
	Signs	**43.43**±0.99	42.53±0.79	**44.10**±1.03	43.01±0.85
	Avg.	**44.54**±0.77	42.72±0.32	**45.37**±0.81	44.80±0.54
Top-40%	Mini	**45.07**±0.52	40.18±0.14	**45.79**±0.44	42.92±0.35
	Birds	**34.88**±0.55	32.38±0.61	**35.77**±0.55	34.54±0.66
	Flowers	**49.98**±0.69	48.18±0.44	53.16±0.73	**54.08**±0.76
	Signs	**41.83**±1.02	40.90±0.70	**43.39**±1.01	42.26±0.77
	Avg.	**42.94**±0.70	40.41±0.29	**44.52**±0.68	43.45±0.40

Validating Proposal 1 and 2. Our method is built on the hypotheses that we should take *small* stepsizes on *high* uncertainty parameters and add *more* adversarial perturbation on *high* uncertainty input gradients. To validate those choices, we have tried taking *large* stepsizes on *high* uncertainty parameters, and using *less* adversarial perturbation on *high* uncertainty input gradients, which resulted in lower accuracy and robustness to the stepsize (**Appendix A.7**).

Freezing most uncertain layers. We observed that the batch normalization (BN) layers have the most uncertainty (**Figure 1**). In Proposal 1, we argued that the higher uncertainty component amplifies the error when we use bigger stepsize. As suggested by an anonymous reviewer, this motivates an additional experiment that we perform where we "freeze" the BN layer during meta-testing (i.e. we manually set the stepsize for the BN scale and shift parameters to zero). We performed the BN layer freezing experiment on SGD+All (see detailed results in **Table 10** and **Figure 12** in **Appendix A.9**). It turns out that SGD+All (w/ freezing BN) outperforms SGD+All (w/o freezing BN) on the All metric, with most improvement in the higher stepsize range (>0.1). This is consistent with our intuition that updating high-uncertainty layers (such as BN) with large stepsizes can be harmful.

5 Conclusion

In this paper we considered the novel problem of repurposing pretrained MAML checkpoints for out-of-domain few-shot learning. Our method uses deep ensembles to estimate model parameter and input gradient uncertainties over the support set, and builds upon the default MAML gradient steps through the addition of uncertainty-based adversarial training and adaptive stepsizes. Our experiments over popular few-shot benchmarks show that our method yields increased accuracy and robustness to the choice of base stepsize. More generally, our results motivate the use of adversarial learning as a data augmentation scheme for improving few-shot generalization. In the future, it would be interesting to apply our method to related settings such as domain adaption and transfer learning.

REFERENCES

Hilal Asi and John C. Duchi. The importance of better models in stochastic optimization. In *National Academy of Sciences*, 2019.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. In *Machine Learning*, 2010.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, 2015.

Aymeric Dieuleveut, Alain Durmus, Francis Bach, et al. Bridging the gap between constant step size stochastic gradient descent and markov chains. In *Annals of Statistics*. Institute of Mathematical Statistics, 2020.

Sayna Ebrahimi, Mohamed Elhoseiny, Trevor Darrell, and Marcus Rohrbach. Uncertainty-guided continual learning with bayesian neural networks. In *International Conference on Learning Representations*, 2020.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. In *International Conference on Learning Representations*, 2018.

Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In *International Joint Conference on Neural Networks*, 2013.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *arXiv preprint arXiv:1412.6980*, 2014.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, 2017.

Yoonho Lee and Seungjin Choi. Gradient-based meta-learning with learned layerwise metric and subspace. In *International Conference on Machine Learning*, 2018.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018.

Seyed Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *Computer Vision and Pattern Recognition*, 2016.

Saeid Motiian, Quinn Jones, Seyed Iranmanesh, and Gianfranco Doretto. Few-shot adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, 2017.

Maria Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 2008.

Yoshio Nishi and Robert Doering. Handbook of semiconductor manufacturing technology. CRC press, 2000.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2017.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International Conference on Machine Learning*, 2016.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 2017.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *Computer Vision and Pattern Recognition*, 2018.

Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *International Conference on Learning Representations*, 2020.

Hung-Yu Tseng, Hsin-Ying Lee, Jia-Bin Huang, and Ming-Hsuan Yang. Cross-domain few-shot classification via learned feature-wise transformation. In *International Conference on Learning Representations*, 2020.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Conference on Computer Vision and Pattern Recognition*, 2017.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, 2016.

Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010.

Yeming Wen, Paul Vicol, Jimmy Ba, Dustin Tran, and Roger Grosse. Flipout: Efficient pseudo-independent weight perturbations on mini-batches. In *International Conference on Learning Representations*, 2018.

Garrett Wilson and Diane J. Cook. A survey of unsupervised deep domain adaptation. In *ACM Transactions on Intelligent Systems and Technology*, 2020.

Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L. Yuille, and Quoc V. Le. Adversarial examples improve image recognition. In *Computer Vision and Pattern Recognition*, 2020.

Chengxiang Yin, Jian Tang, Zhiyuan Xu, and Yanzhi Wang. Adversarial meta-learning. In *arXiv preprint arXiv:1806.03316*, 2018.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, 2014.

Yabin Zhang, Hui Tang, Kui Jia, and Mingkui Tan. Domain-symmetric networks for adversarial domain adaptation. In *Computer Vision and Pattern Recognition*, 2019.

An Zhao, Mingyu Ding, Zhiwu Lu, Tao Xiang, Yulei Niu, Jiechao Guan, Ji Rong Wen, and Ping Luo. Domain-adaptive few-shot learning. In *arXiv*, 2020.

A APPENDIX

A.1 NOTATIONS AND ACRONYMS

Notations and acronyms used repeatedly throughout the paper are summarized below.

Notation	Meaning
USA	Uncertainty-based stepsize adaption
I/USA	Inverse USA
FGSM	Fast gradient sign method optimization (Goodfellow et al., 2015)
UFGSM	Uncertainty-based FGSM
I/UFGSM	Inverse UFGSM
EnAug	Ensemble augmentation
T	Number of test-time gradient steps
M	Size of deep ensemble
θ_0	Pretrained MAML checkpoint to repurpose
θ_t^m	Ensemble element m at time t
θ_T	Model parameters after meta-testing
α	Base stepsize
α^{adap}	Uncertainty-based stepsize (parameter-wise)
D^{Spt}	Support set (test-time)
D^{Aug}	Adversarial examples based on D^{Spt}

A.2 DESIGN CHOICES FOR USA

Here, we introduce the design choices of the Algorithm 2.

Line 3: We calculate the maximum $\text{Max}(u)$ and minimum $\text{Min}(u)$ values among the estimated standard deviations u for each parameter. As proposal 1, in order to assign to low stepsize to high uncertainty, the standard deviation of each parameter is subtracted from the maximum value $\text{Max}(u) - u$ for inverse-relationship. At this time, the largest value among parameter values becomes 0. To prevent this, we add the smallest $\text{Max}(u) - u + \text{Min}(u)$. Note that all values of u are non-negative. We also tried alternatives ways like non-inverse-relationship and relative standard deviation, but it did not seem to work better. To keep the same scale of the uncertainty before and after the transformation, we chose the inverse-relationship using $\text{Max}(u) - u + \text{Min}(u)$.

Line 4: In this study, we use layer-wise adapted stepsize. In order to assign the same stepsize for each layer, we calculate the average of the parameters of each layer and the values of the parameters of the corresponding layer are replaced with the average value. It repeats all layers and performs the corresponding operation.

Line 5: To more easily compare the effectiveness of USA with the baseline, we set the average stepsize equals to the base stepsize, i.e. $1/|\alpha^{\text{adap}}| \sum \alpha^{\text{adap}} = \alpha$. If USA is applied without this rescaling, than the average stepsize can change, which makes it difficult to distinguish the effect of using an overall different stepsize (for constant SGD e.g.) vs. our adaptation of stepsizes.

A.3 EXPERIMENTAL SETUP AND PRETRAINED MODEL SELECTION

Our baseline model was trained using the same hyperparameters with *mini*ImageNet training of MAML except the inner loop stepsize. The inner loop stepsize was set to 0.1 for reproducing the 5-way 1-shot accuracy reported in the original MAML paper. We trained the model for 150,000 iterations. A pretrained model was selected with the validation accuracy among *mini*ImageNet training checkpoints. We used the checkpoint with the highest validation accuracy as the pretrained model θ_0. The highest test accuracy we achieved was 49.24% during meta-training. For the checkpoint that we chose with the highest validation performance, the test accuracy was 47.58%. Note that the cross-domain performance highly depended on which checkpoint we used. See Fig.4 in Appendix A.4.

The size of ensemble is $M = 5$ which is the same as the deep ensembles (Lakshminarayanan et al., 2017). The parameter for the FGSM is $\epsilon = 0.05$. The scale value a of Gaussian random perturbation for ensemble model training is $\sigma = 0.05$. The gradient step is $T = 10$ which is same with *mini*ImageNet test of MAML.

We do not split those datasets except *mini*ImageNet, because we do not use the datasets in the meta-training. We used the same splits as Ravi & Larochelle (2017) for *mini*ImageNet. Tseng et al. (2020) used a randomly split manner for the cross-domain test and Triantafillou et al. (2020) used all traffic signs dataset for the test.

A.4 Cross-domain accuracy while meta-training MAML on miniImageNet

The performance of the cross-domain significantly differed depending on the selected checkpoint. Figure 4 shows the performance for every 200 iterations. We selected the checkpoint based on miniImageNet validation performance, which is the highest at the 57,200 iteration.

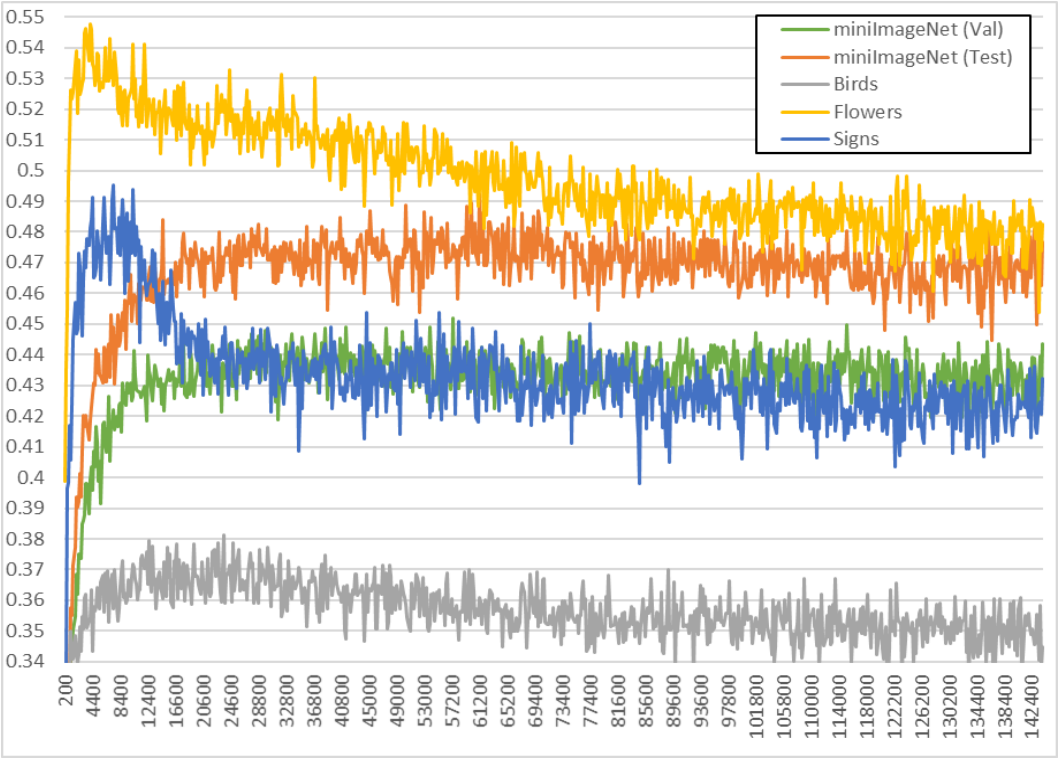


Figure 4: Cross-domain performance while meta-training on *mini*ImageNet. X-axis is training iteration on meta-training on *mini*ImageNet and y-axis is classification accuracy.

A.5 Additional results for AT-only setting.

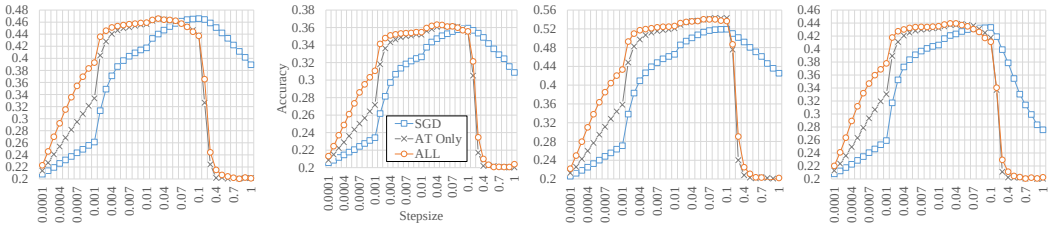
To evaluate the effect of AT, we only applied AT in MAML ($\lambda_\alpha = \lambda_{\text{AT}} = 1, \lambda_{\text{Aug}} = 0$). **Table 5**, **Figure 5** and **6** show AT results on 5-way 1-shot with SGD and Adam. Adam is worse than SGD in the results. However, we found that AT and Adam are good combination for a meta-test training.

A.6 Additional results for 5-way 5-shot and 10-way 1-shot classification.

We present some additional results for 5-way 5-shot and 10-way 1-shot classification in this section. **Table 6** shows the 5-way 5-shot and 10-way 1-shot classification results. Especially in the case of the Top-40% Avg our method significantly increased the performance about 5.84% and 1.83% in 5-way 5-shot and 10-way 1-shot respectively. It means that it can increase the probability to select of

Table 5: 5-way 1-shot classification results. In order to verify the effectiveness of AT, we applied only AT among the proposed method and the all proposed method.

Metric	Dataset	Baseline SGD	Ours (SGD) AT Only	Ours (SGD) All	Baseline Adam	Ours (Adam) AT Only	Ours (Adam) All
All	Mini	**37.21**$\pm$**0.22**	34.60$\pm$0.18	36.24$\pm$0.22	31.87$\pm$0.20	33.40$\pm$0.33	**33.57**$\pm$**0.29**
	Birds	**30.07**$\pm$**0.52**	28.81$\pm$0.45	29.83$\pm$0.46	28.02$\pm$0.39	29.37$\pm$0.51	**29.51**$\pm$**0.46**
	Flowers	40.54$\pm$0.48	38.80$\pm$0.44	**40.86**$\pm$**0.45**	40.33$\pm$0.57	44.53$\pm$0.68	**44.58**$\pm$**0.70**
	Signs	34.18$\pm$0.76	33.42$\pm$0.59	**34.76**$\pm$**0.61**	33.98$\pm$0.43	34.96$\pm$0.47	**35.10**$\pm$**0.43**
	Avg.	**35.50**$\pm$**0.49**	33.91$\pm$0.41	35.42$\pm$0.44	33.55$\pm$0.22	35.57$\pm$0.29	**35.69**$\pm$**0.29**
Top-1	Mini	**46.71**$\pm$**0.81**	46.58$\pm$0.65	46.62$\pm$0.58	43.08$\pm$0.10	**44.80**$\pm$**0.40**	44.72$\pm$0.35
	Birds	36.03$\pm$0.48	36.40$\pm$0.48	**36.43**$\pm$**0.58**	34.00$\pm$0.63	35.28$\pm$0.66	**35.45**$\pm$**0.67**
	Flowers	51.97$\pm$0.79	**54.55**$\pm$**0.84**	54.36$\pm$1.04	51.28$\pm$0.47	**56.10**$\pm$**1.06**	56.04$\pm$1.14
	Signs	43.44$\pm$0.99	43.98$\pm$0.98	**44.10**$\pm$**1.03**	42.53$\pm$0.79	**43.13**$\pm$**0.83**	43.01$\pm$0.85
	Avg.	44.54$\pm$0.77	**45.38**$\pm$**0.74**	45.37$\pm$0.81	42.72$\pm$0.32	**44.83**$\pm$**0.51**	44.80$\pm$0.54
Top-40%	Mini	45.07$\pm$0.52	45.53$\pm$0.50	**45.79**$\pm$**0.44**	40.18$\pm$0.14	42.59$\pm$0.43	**42.92**$\pm$**0.35**
	Birds	34.88$\pm$0.55	35.57$\pm$0.60	**35.77**$\pm$**0.55**	32.38$\pm$0.61	34.26$\pm$0.73	**34.54**$\pm$**0.66**
	Flowers	49.98$\pm$0.69	52.92$\pm$0.68	**53.16**$\pm$**0.73**	48.18$\pm$0.44	53.88$\pm$0.72	**54.08**$\pm$**0.76**
	Signs	41.81$\pm$1.02	43.19$\pm$0.96	**43.39**$\pm$**1.01**	40.90$\pm$0.70	42.06$\pm$0.74	**42.26**$\pm$**0.77**
	Avg.	42.94$\pm$0.70	44.30$\pm$0.68	**44.52**$\pm$**0.68**	40.41$\pm$0.29	43.20$\pm$0.42	**43.45**$\pm$**0.40**



(a) Mini (b) Birds (c) Flowers (d) Signs

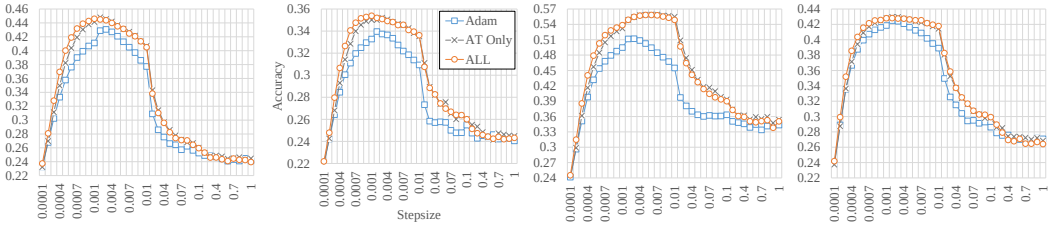
Figure 5: 5-way 1-shot classification results with AT on SGD.

optimal stepsize for a new task. As shown in Figure 7, there are more flatten curve near the highest accuracy in both tasks.

A.7 Validating proposals 1 and 2 with "Inverse" counterparts.

To validate proposal 1, we evaluate the inverse strategy of USA, denoted I/USA, which assigns higher stepsizes to layers with higher uncertainty. Specifically, we flip the stepsizes by replacing **line 3** of Algorithm 2 with $c \leftarrow u$. To measure the effect of USA, we only applied USA in MAML ($\lambda_\alpha = \lambda_{\text{AT}} = \lambda_{\text{Aug}} = 0$). The results for 5-way 1-shot classification are in **Table 7**. USA outperforms the baselines over broad ranges of stepsizes compared to Adam and SGD. I/USA significantly decreased the performance of all metrics. Therefore, USA reflected useful knowledge well into the stepsize for a new task. See accuracy by stepsizes in Fig 8. Figure 8 shows the results for verifying of Proposal 1. USA shows more flatten curve in Top-1 accuracy ranges than SGD and I/USA. I/USA degraded the performance through all ranges of stepsizes.

For verifying the proposal 2, we evaluated three methods UFGSM , I/UFGSM and FGSM on USA. I/UFGSM is inversely implemented method of UFGSM. FGSM uses examples generated by FGSM instead of UFGSM. To measure the effect of UFGSM, we only applied UFGSM in MAML with USA ($\lambda_\alpha = \lambda_{\text{AT}} = 0, \lambda_{\text{Aug}} = 1$). Table 7 shows 5-way 1-shot classification results. UFGSM outperforms all the other methods for every metric. In spite of I/UFGSM had stronger adversarial perturbation than UFGSM, the performance was worse than UFGSM. Note that I/UFGSM and FGSM showed almost similar performances. The reason is that most of the input pixels have an uncertainty close to 0; therefore, when scaling after inverse, most pixels have a value of 1. Therefore, the generated examples are almost similar to the adversarial example in which FGSM is applied. See accuracy by stepsizes in Fig 9. Figure 9 shows the result for verifying of proposal 2. UFGSM



(a) Mini (b) Birds (c) Flowers (d) Signs

Figure 6: 5-way 1-shot classification results with AT on Adam.

Table 6: 5-way 5-shot and 10-way 1-shot classification results using SGD. Our method outperformed both tasks. Especially 5-way 5-shot classification performance significantly improved than baseline.

Metric	Dataset	5-way 5-shot Baseline SGD	5-way 5-shot Ours (SGD) SGD+AT	5-way 5-shot Ours (SGD) SGD+All	10-way 1-shot Baseline SGD	10-way 1-shot Ours (SGD) SGD+AT	10-way 1-shot Ours (SGD) SGD+All
All	Mini	38.14±0.39	41.06±0.35	**43.48±0.38**	21.31±0.37	20.97±0.37	**22.07±0.40**
	Birds	33.29±0.29	36.04±0.28	**38.07±0.33**	17.54±0.06	17.08±0.07	**17.81±0.12**
	Flowers	44.55±0.43	48.94±0.48	**53.10±0.55**	26.79±0.41	26.50±0.42	**28.07±0.41**
	Signs	45.39±0.45	47.98±0.55	**51.02±0.50**	23.77±0.13	23.99±0.11	**24.88±0.13**
	Avg.	40.34±0.39	43.51±0.41	**46.42±0.44**	22.35±0.24	22.14±0.24	**23.21±0.26**
Top-1	Mini	**62.46±0.68**	62.07±0.61	62.16±0.66	31.10±0.66	31.18±0.81	**31.42±0.82**
	Birds	50.71±0.54	**51.93±0.62**	51.82±0.64	23.16±0.21	23.42±0.37	**23.61±0.39**
	Flowers	70.63±0.76	74.14±0.72	**74.52±0.69**	38.61±0.49	**40.84±0.58**	40.68±0.58
	Signs	72.52±0.84	73.66±0.98	**74.13±0.90**	35.10±0.21	34.76±0.17	**35.11±0.26**
	Avg.	64.08±0.71	65.45±0.73	**65.66±0.72**	31.99±0.39	32.55±0.48	**32.70±0.51**
Top-40%	Mini	55.38±0.46	59.97±0.69	**60.70±0.91**	28.65±0.64	30.22±0.74	**30.56±0.77**
	Birds	45.50±0.49	50.48±0.53	**51.03±0.63**	21.78±0.07	22.61±0.22	**22.87±0.30**
	Flowers	65.16±0.73	72.61±0.78	**73.55±0.86**	36.14±0.53	39.40±0.66	**39.75±0.61**
	Traffic	67.34±0.43	70.56±0.87	**71.45±0.90**	33.56±0.11	34.21±0.13	**34.27±0.17**
	Avg.	58.34±0.53	63.40±0.72	**64.18±0.82**	30.03±0.34	31.61±0.44	**31.86±0.46**

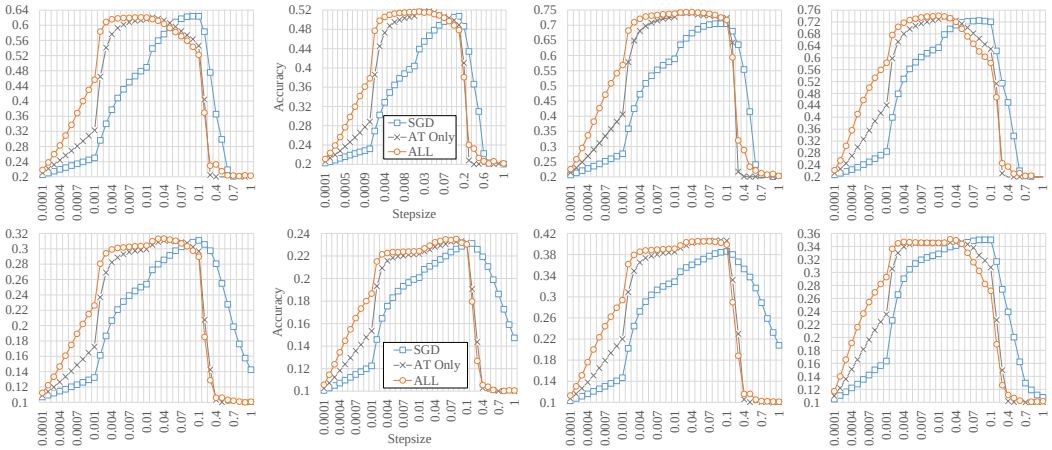
shows flatter curve near the best performing accuracy than SGD, I/UFGSM and FGSM. FGSM and I/UFGSM show very similar trends. We explained the reason why the two methods gave similar results. As can be seen in Fig 10, almost all of the input gradient uncertainty is near zero. When we inverse the value, almost all the value is 1 (See Fig 11). UFGSM improves performance despite less adversarial perturbation than I/UFGSM and FGSM. Through this, even a small change can help model learning if correct (useful) information is reflected.

A.8 5-WAY 5-SHOT CLASSIFICATION RESULTS ON OVERFITTED CHECKPOINT

As can seen in Appendix A.4, overfitted models on *mini*ImageNet degraded performance over all datasets. We investigated the 5-way 5-shot classification results on the checkpoint after meta-training with MAML for 150K iterations. **Table 8** and **9** show the results with SGD and Adam. The number in the parentheses is the difference from the baseline for each checkpoint (e.g. SGD or Adam). The results using the (Last) checkpoint are worse than (Validation) checkpoint due to the overfitting. However the performance boost from using our method is more substantial on the overfitted (Last) checkpoint than the best (Validation) checkpoint, both in terms of absolute performance (Top-1) and robustness to choice of base stepsize (other metrics).

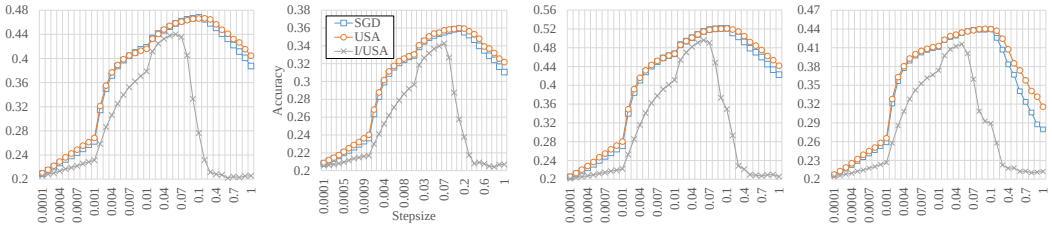
A.9 FREEZING MOST UNCERTAIN LAYERS.

Since some of the BatchNorm parameters have the highest uncertainty (see Figure 1a), we have experimented freezing the BatchNorm parameters during meta-testing (i.e. setting their stepsize to zero). It turns out that SGD+All (w/ freezing BN) outperforms SGD+All (w/o freezing BN) on the All metric (**Table 10**), with most improvement in the higher stepsize range (>0.1) (**Figure 12**). We



(a) Mini (b) Birds (c) Flowers (d) Signs

Figure 7: 5-way 5-shot (Top) and 10-way 1-shot (Bottom) classification with MAML(SGD) and our proposed method. In terms of robustness, our method outperforms MAML(SGD) more effectively. In addition, we show that our method outperforms Top-1 accuracy for Birds, Flowers and Signs. Also our method shows flatter curve near highest accuracy sections. This increases the probability of optimal stepsize selection.



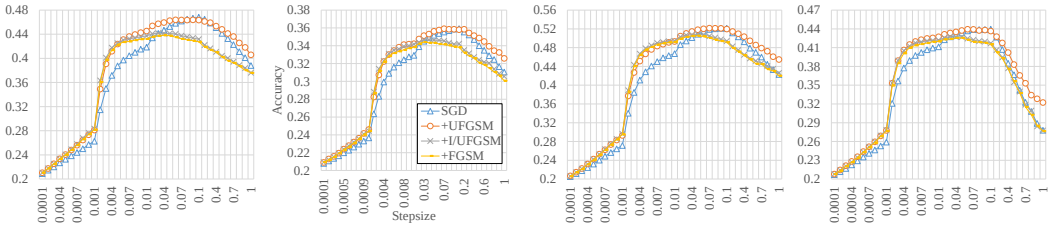
(a) Mini (b) Birds (c) Flowers (d) Signs

Figure 8: 5-way 1-shot classification for verifying proposal 1. We compared to USA and I/USA.

notice a similar trend for 5-way 5-shot SGD, where SGD (w/ freezing BN) also outperforms SGD (w/o freezing BN) on the All metric, with most improvement in the higher stepsize range.

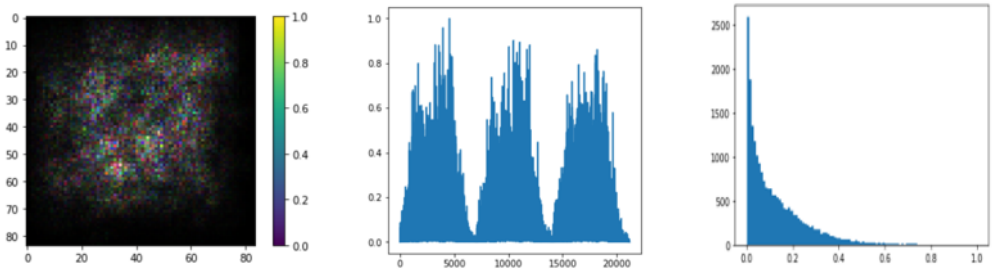
Table 7: 5-way 1-shot classification results for verifying proposal 1 and 2. I/USA and I/UFGSM degraded performance than USA and UFGSM respectively. Our approach which is applied USA and UFGSM outperformed SGD on almost of the metrics.

Metric	Dataset	Baseline		SGD		SGD+USA		
		SGD	Adam	+USA	+I/USA	+UFGSM	+I/UFGSM	+FGSM
All	Mini	37.21	31.85	37.53	28.73	**38.71**	37.25	36.97
	Birds	30.07	28.17	30.39	24.98	**31.07**	30.27	30.05
	Flowers	40.54	40.49	41.09	30.52	**42.33**	41.44	41.17
	Signs	34.21	34.28	35.02	27.90	**36.10**	34.98	34.78
	Avg.	35.50	33.68	36.01	28.03	**37.05**	35.98	35.74
Top-1	Mini	46.71	43.07	**46.72**	44.13	46.66	44.31	43.93
	Birds	36.03	34.30	**36.08**	34.20	36.02	34.77	34.52
	Flowers	51.97	51.38	52.01	49.34	**52.30**	50.90	50.63
	Signs	43.44	43.00	43.42	41.03	**43.96**	42.82	42.62
	Avg.	44.54	42.93	44.56	42.17	**44.73**	43.20	42.93
Top-40%	Mini	45.07	40.16	45.23	38.33	**45.76**	43.60	43.18
	Birds	34.88	32.64	35.17	30.28	**35.30**	34.20	33.88
	Flowers	49.98	48.28	50.39	42.19	**51.07**	49.79	49.47
	Signs	41.81	41.32	42.08	35.78	**43.18**	42.17	41.92
	Avg.	42.94	40.60	43.22	36.64	**43.83**	42.44	42.11



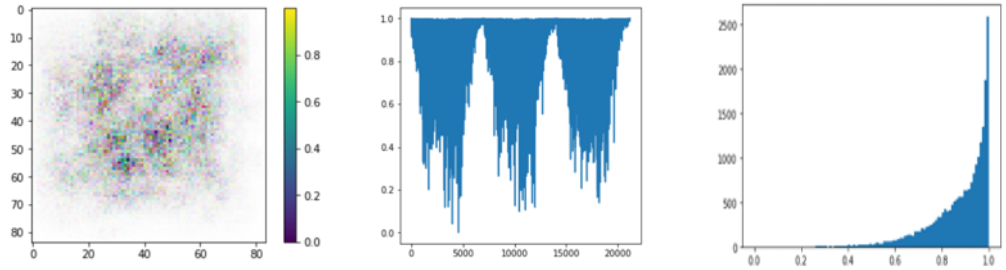
(a) Mini (b) Birds (c) Flowers (d) Signs

Figure 9: 5-way 1-shot classification results to verify proposal 2. We compared to UFGSM, I/UFGSM and FGSM.



(a) Uncertainty map (u) (b) Flatten uncertainty (c) Histogram of uncertainty

Figure 10: (a) shows the re-scaled input gradient uncertainty to generate the UFGSM example. (b) is a flatten plot of the re-scaled input gradient uncertainty. (c) shows a histogram of the uncertainty.



(a) Uncertainty map (u) (b) Flatten uncertainty (c) Histogram of uncertainty

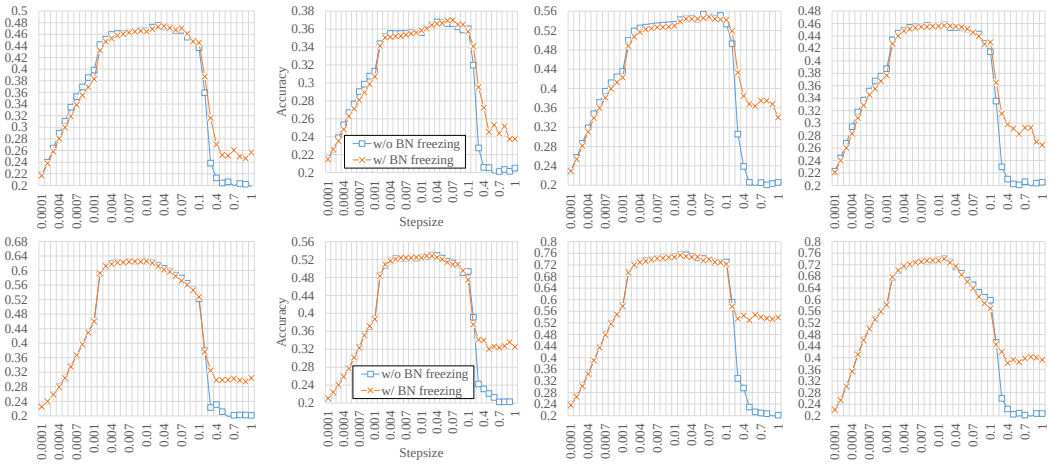
Figure 11: (a) shows the re-scaled inverse input gradient uncertainty to generate the UFGSM example. (b) is a flatten plot of the re-scaled inverse input gradient uncertainty. (c) shows a histogram of the uncertainty.

Table 8: 5-way 5-shot classification results between selected checkpoint with SGD

Metric	Dataset	Checkpoint (Validation)			Checkpoint (Last)		
		Baseline	Ours (SGD)		Baseline	Ours (SGD)	
		SGD	AT Only	All	SGD	AT Only	All
All	Mini	38.14±0.39 (-)	41.06±0.35 (2.92)	**43.48±0.38** (5.34)	34.98±0.54 (-)	38.66±0.52 (3.68)	41.89±0.61 **(6.91)**
	Birds	33.29±0.29 (-)	36.04±0.28 (2.75)	**38.07±0.33** (4.78)	31.33±0.51 (-)	34.34±0.49 (3.01)	36.90±0.5 **(5.57)**
	Flowers	44.55±0.43 (-)	48.94±0.48 (4.39)	**53.10±0.55** (8.55)	40.96±0.67 (-)	45.6±0.6 (4.64)	50.58±0.67 **(9.62)**
	Signs	45.39±0.45 (-)	47.98±0.55 (2.59)	**51.02±0.50** (5.63)	42.05±0.35 (-)	46.01±0.36 (3.96)	49.88±0.35 **(7.83)**
	Avg.	40.34±0.39 (-)	43.51±0.41 (3.17)	**46.42±0.44** (6.08)	37.33±0.52 (-)	41.15±0.49 (3.82)	44.81±0.53 **(7.48)**
Top-1	Mini	**62.46±0.68** (-)	62.07±0.61 (-0.39)	62.16±0.66 (-0.3)	58.49±0.72 (-)	58.44±0.93 (-0.05)	58.72±1.08 **(0.23)**
	Birds	50.71±0.54 (-)	**51.93±0.62** (1.22)	51.82±0.64 (1.11)	46.77±0.53 (-)	48.33±0.72 (1.56)	48.58±0.7 **(1.81)**
	Flowers	70.63±0.76 (-)	74.14±0.72 (3.51)	**74.52±0.69** **(3.89)**	66.31±1.02 (-)	68.99±0.81 (2.68)	69.73±0.71 (3.42)
	Signs	72.52±0.84 (-)	73.66±0.98 (1.14)	**74.13±0.90** (1.61)	67.75±0.57 (-)	70.71±0.48 (2.96)	71.08±0.53 **(3.33)**
	Avg.	64.08±0.71 (-)	65.45±0.73 (1.37)	**65.66±0.72** (1.58)	59.83±0.71 (-)	61.62±0.73 (1.79)	62.03±0.75 **(2.2)**
Top-40%	Mini	55.38±0.46 (-)	59.97±0.69 (4.59)	**60.70±0.91** (5.32)	50.56±0.72 (-)	56.44±0.87 (5.88)	57.22±1 **(6.66)**
	Birds	45.50±0.49 (-)	50.48±0.53 (4.98)	**51.03±0.63** (5.53)	42.13±0.55 (-)	47.22±0.73 (5.09)	47.98±0.73 **(5.85)**
	Flowers	65.16±0.73 (-)	72.61±0.78 (7.45)	**73.55±0.86** (8.39)	59.8±1.02 (-)	67.37±0.83 (7.57)	68.93±0.79 **(9.13)**
	Signs	67.34±0.43 (-)	70.56±0.87 (3.22)	**71.45±0.90** (4.11)	61.95±0.66 (-)	68.03±0.64 (6.08)	69.41±0.5 **(7.46)**
	Avg.	58.34±0.53 (-)	63.40±0.72 (5.06)	**64.18±0.82** (5.84)	53.61±0.74 (-)	59.77±0.77 (6.16)	60.89±0.76 **(7.28)**

Table 9: 5-way 5-shot classification results of selected checkpoint with Adam

Metric	Dataset	Checkpoint (Validation) Baseline Adam	Checkpoint (Validation) Ours (Adam) AT Only	Checkpoint (Validation) Ours (Adam) All	Checkpoint (Last) Baseline Adam	Checkpoint (Last) Ours (Adam) AT Only	Checkpoint (Last) Ours (Adam) All
All	Mini	40.08±1.33	41.38±1.07	**41.64±0.97**	37.25±0.64	39.47±0.56	40.36±0.62
		(-)	(1.3)	(1.56)	(-)	(2.21)	**(3.1)**
	Birds	36.68±1.05	**38.79±1**	38.78±0.91	35.39±0.67	37.86±0.72	38.15±0.69
		(-)	(2.11)	(2.10)	(-)	(2.47)	**(2.76)**
	Flowers	58.00±1.55	**60.92±1.24**	60.36±1.14	54.36±0.79	59.09±0.63	59.32±0.69
		(-)	(2.92)	(2.36)	(-)	(4.73)	**(4.96)**
	Signs	**52.65±0.57**	52.37±0.6	52.53±0.64	50.48±0.6	51.13±0.45	51.95±0.49
		(-)	(-0.28)	(-0.12)	(-)	(0.65)	**(1.47)**
	Avg.	46.85±1.12	**48.37±0.98**	48.33±0.91	44.37±0.67	46.89±0.59	47.45±0.62
		(-)	(1.52)	(1.48)	(-)	(2.52)	**(3.08)**
Top-1	Mini	58.89±0.73	**60.35±0.83**	60.08±0.83	55.25±1.02	57.79±0.81	57.86±0.92
		(-)	(1.46)	(1.19)	(-)	(2.54)	**(2.61)**
	Birds	47.92±0.75	**50.88±0.66**	50.84±0.75	46.02±0.85	49.89±1	49.78±0.97
		(-)	(2.96)	(2.92)	(-)	**(3.87)**	(3.76)
	Flowers	73.21±1.19	**77.51±0.94**	77.45±0.89	69.09±0.77	76.61±0.51	76.51±0.71
		(-)	(4.3)	(4.24)	(-)	**(7.52)**	(7.42)
	Signs	70.98±1.07	71.30±1.52	**71.72±1.52**	67.51±0.72	69.75±0.83	70.30±0.79
		(-)	(0.32)	(0.74)	(-)	(2.24)	**(2.79)**
	Avg.	62.75±0.93	65.01±0.99	**65.02±1**	59.47±0.84	63.51±0.79	63.61±0.85
		(-)	(2.26)	(2.27)	(-)	(4.04)	**(4.14)**
Top-40%	Mini	52.17±1.89	54.96±1.18	**55.75±1.07**	46.95±0.82	51.81±0.71	53.29±0.76
		(-)	(2.79)	(3.58)	(-)	(4.86)	**(6.34)**
	Birds	43.88±1.06	47.93±0.91	**48.26±0.87**	41.41±0.8	46.71±0.96	47.11±0.88
		(-)	(4.05)	(4.38)	(-)	(5.3)	**(5.7)**
	Flowers	68.43±1.76	73.86±1.02	**74.03±1.02**	63.44±0.77	72.41±0.79	72.56±0.64
		(-)	(5.43)	(5.60)	(-)	(8.97)	**(9.12)**
	Signs	66.85±0.78	67.36±1.3	**67.89±1.39**	62.95±0.81	65.43±0.68	66.37±0.71
		(-)	(0.51)	(1.04)	(-)	(2.48)	**(3.42)**
	Avg.	57.83±1.37	61.03±1.1	**61.48±1.09**	53.69±0.8	59.09±0.78	59.83±0.75
		(-)	(3.2)	(3.65)	(-)	(5.4)	**(6.14)**



(a) Mini (b) Birds (c) Flowers (d) Signs

Figure 12: 5-way 1-shot (Top) and 5-way 5-shot (Bottom) classification results of freezing BN layers. We tested freezing BN on SGD+All. SGD+All (w/ BN freezing) outperformed SGD+All (w/o BN freezing) in the higher stepsize range (>0.1).

Table 10: 5-way 1-shot and 5-way 5-shot classification results of freezing BN layers. We tested freezing BN on our SGD+All. SGD+All (w/ freezing BN) outperformed in All metric on 5-way 1-shot and 5-way 5-shot.

Metric	Dataset	5-way 1-shot (SGD+All)		5-way 5-shot (SGD+All)	
		w/ freezing BN	w/o freezing BN	w/ freezing BN	w/o freezing BN
All	Mini	**37.13**$\pm$**0.24**	36.24$\pm$0.22	**45.42**$\pm$**0.42**	43.48$\pm$0.38
	Birds	**30.74**$\pm$**0.50**	29.83$\pm$0.46	**40.29**$\pm$**0.47**	38.07$\pm$0.33
	Flowers	**43.88**$\pm$**0.56**	40.86$\pm$0.45	**59.41**$\pm$**0.68**	53.10$\pm$0.55
	Signs	**36.37**$\pm$**0.67**	34.76$\pm$0.61	**54.65**$\pm$**0.56**	51.02$\pm$0.50
	Avg.	**37.03**$\pm$**0.49**	35.42$\pm$0.44	**49.94**$\pm$**0.54**	46.42$\pm$0.44
Top-1	Mini	46.48$\pm$0.54	46.62$\pm$0.58	62.18$\pm$0.66	62.16$\pm$0.66
	Birds	36.41$\pm$0.59	36.43$\pm$0.58	51.90$\pm$0.73	51.82$\pm$0.64
	Flowers	54.16$\pm$0.80	54.36$\pm$1.04	74.59$\pm$0.79	74.52$\pm$0.69
	Signs	44.18$\pm$1.09	44.10$\pm$1.03	73.98$\pm$0.88	74.13$\pm$0.90
	Avg.	45.31$\pm$0.76	45.37$\pm$0.81	65.66$\pm$0.76	65.66$\pm$0.72
Top-40%	Mini	45.44$\pm$0.48	45.79$\pm$0.44	60.62$\pm$0.66	60.70$\pm$0.91
	Birds	35.67$\pm$0.54	35.77$\pm$0.55	51.04$\pm$0.73	51.03$\pm$0.63
	Flowers	52.97$\pm$0.71	53.16$\pm$0.73	73.56$\pm$0.73	73.55$\pm$0.86
	Traffic	43.38$\pm$0.99	43.39$\pm$1.01	71.21$\pm$0.79	71.45$\pm$0.90
	Avg.	44.37$\pm$0.68	44.52$\pm$0.68	64.11$\pm$0.73	64.18$\pm$0.82