Adversarially Robust Few-shot Learning through Simple Transfer

Akshayvarun Subramanya
University of Maryland, Baltimore County
akshayv1@umbc.edu

Hamed Pirsiavash
University of California, Davis
hpirsiav@ucdavis.edu

Abstract

Few-shot image classification, where the goal is to generalize to tasks with limited labeled data, has seen great progress over the years. However, the classifiers are vulnerable to adversarial examples, posing a question regarding their generalization capabilities. Previous works have tried to combine meta-learning approaches with adversarial training to improve the robustness of few-shot classifiers. We show that a simple transfer-learning based approach can be used to train adversarially robust few-shot classifiers. We also present a method for novel classification task based on calibrating the centroid of the few-shot category towards the base classes. We show that standard adversarial training on base categories along with calibrated centroid-based classifier in the novel categories, outperforms or is on-par with previous methods on standard benchmarks for few-shot learning. Our method is simple, easy to scale, and with little effort can lead to robust few-shot classifiers. Code: https://github.com/UCDvision/Simple_few_shot.git

1. Introduction

Few-shot learning presents the challenge of learning quickly from few examples of data, which is generally considered the hallmark of human intelligence. This is an important practical problem due to the scarce availability of fully annotated data in the real world and such a setting can be considered for various real world computer vision tasks. As a result, it is of paramount importance that such safety-critical systems are reliable and robust to input perturbations. Specifically in this work, we consider robustness to adversarial examples - carefully crafted perturbations that when added to inputs, *fool* the classifier. The most common method of improving robustness is by adversarial training [7] which involves training on adversarial examples using adversary of choice. Traditional adversarially robust methods [7] consider a data-rich setting where many examples are available per category. This becomes challenging when the end-user has access to limited amount of annotated data but is interested in building a robust classifier. Such a setting is more practical and it is important to develop methods which can work with minimal effort in the pre-deployment stage.

One of the well known frameworks for few-shot learning is MAML [2] which aims at learning a network initialization using a bi-level optimization procedure, that when finetuned on limited data is able to generalize to the new task. [3, 11] perform adversarial training on top of meta-learners to improve robustness. However, adversarial training on its own is expensive and combining with meta-learning makes the problem computationally intensive. [11] showed that there exists a compromise between training robust meta-learners and performance, motivating the need for a simpler approach.

We consider a simple setting and show that adversarial training along with a centroid-based classifier can outperform previous methods in terms of robustness. Such a setting is practically relevant, since the adversarial training is done just once and robustness for few-shot classes can be achieved without creating adversarial examples. We believe it also becomes easier to incorporate new approaches to robustness, such as verifiably robust classifiers and can bring together robust methods for both large and limited dataset settings.

2. Method

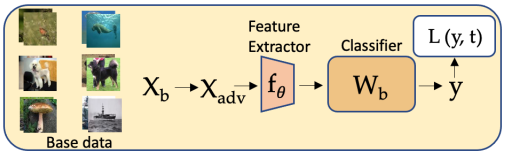
Here we introduce notation and provide a description of our method. Our first objective is to learn a feature extractor f_{θ_b} and linear classifier C_{ω_b} using the abundantly-labeled base dataset X_b. Previous approaches consider multiple few-shot tasks sampled from X_b for meta-learning, whereas we consider a standard mini-batch based training.

At the next stage, when a N-way K-shot task is sampled from the non-overlapping novel dataset X_n, we use only the feature extractor f_{θ_b} and learn a new classifier C_{ω_n} that can generalize to novel categories.We divide our approach into two stages: (1) Robust Base training and (2) Novel training.

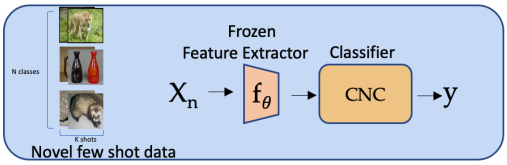
2.1. Robust Base Training

Given a base dataset X_b, we perform adversarial training using an iterative adversary such as PGD [7]. Specifically, we solve the min-max objective

$$\theta^* = \min_{\theta} \mathbb{E}_{(x,y)\in X_b} \left[\max_{||\delta||_p<\epsilon} \mathcal{L}(\theta, x+\delta, y) \right] \quad (1)$$



(a) Robust Base Training



(b) Novel Training

Figure 1. Our method is divided into two phases (a) Robust Base training involves standard adversarial training with base dataset consisting of many examples and categories (b) Novel Training involves transferring or adapting the network for novel few-shot data using a Calibrated Nearest Centroid (CNC) classifier. Note that the feature extractor remains frozen in the second phase.

Here, $\mathcal{L}(\theta, x, y)$ represents the training objective, which is commonly cross-entropy and $\theta = (\theta_b, \omega_b)$ represents the combination of the feature extractor and base classifier parameters. There are different methods for optimizing the inner maximization in Equation 1. We use the Projected Gradient Descent (PGD) algorithm [7] with $p = \infty$ which corresponds to finding a perturbation δ within an ϵ-bounded hypercube around x that maximizes the objective. Note that adversarial training which is a computationally expensive procedure, needs to be performed just once using base dataset. The **intuition** behind this approach is that the model sees multiple categories across batches rather than episodic data, hence gaining better understanding of semantic categories and robustness which can be beneficial for adaptation. Episodic data corresponds to sampling a N-way, K-shot task from base dataset with data and labels changing across multiple epochs. Since network is trained for multiple epochs and sampling is performed many times, the label associated with a particular category changes across epochs, hence the network does not get a semantic understanding of each category, but is merely tuned for fast adaptation.

Weight averaging: Weight averaging (WA) has been shown to improve generalization [6] in deep networks as it approximates ensembling in temporal fashion and can find flatter optima in loss surface. We perform this for only the feature extractor parameters θ_b which will be used in the next step. Similar to [5], we keep a separate copy of the weights and for every iteration perform exponential moving average method $\theta_b{}' \leftarrow \tau\theta_b{}' + (1-\tau) * \theta_b$ and use $\theta_b{}'$ during the evaluation. We set $\tau = 0.999$ in all our experiments.

2.2. Novel Training

During this stage, we consider the N-way, K-shot novel task and adapt our feature extractor f_{θ_b} using classifier C_{ω_n} .

Linear classifier: The simplest baseline is to learn a linear model on top of the frozen feature extractor using the few shot examples. We find this achieves reasonable performance suggesting that a robust base classifier corresponds to a robust novel classifier. However, this approach alone is not sufficient to achieve improved robustness compared to previous methods, which maybe due to fact that the model can become biased towards the specific few-shot samples and may not capture the true class distribution.

Background on Distribution Calibration (DC) [13]: Previous work [13] has shown that standard accuracy of few-shot classifiers can be improved by using Distribution Calibration. They present a *free-lunch* hallucination-based method where the feature distributions of the novel categories are calibrated using the base dataset, due to the similarity between the base and novel datasets. The mean and covariance of each novel category is calibrated using the statistic of base data and sampling or *hallucination* is done for many points from a Gaussian distribution to learn a logistic regression classifier.

Calibrated Nearest Centroid (CNC): DC method can be computationally expensive and is difficult for large scale settings due to covariance matrix calculation which can be of $\mathcal{O}(N*D^2)$ complexity where D is the feature dimensionality and N is the number of data points. Moreover, sampling from a multivariate Gaussian with non-diagonal covariance is expensive and can be of the order of at least $\mathcal{O}(D^{2.3})$.

To overcome these drawbacks, we present a simple method where we rely only on the **calibrated mean** and classify query sample using a non-parametric Nearest Centroid based algorithm. We call this the **Calibrated Nearest Centroid** (CNC) Classifier. We find the nearest base-category centers to each novel training sample and then average them along with the novel training sample to obtain the new mean or centroid for the novel category. More formally:

$$\mu_j = \frac{1}{m+1}(z_j + \sum_{i \in \mathcal{S}_j} \mu_i^b)$$

where μ_j is the center for the novel category j and $\mathcal{S}_j$ is the set of m base category centers that are closest to z_j, μ_i^b is the mean of base category i in the feature space. In the case of k-shot setting, we calculate a centroid for each sample and average them to get one centroid for each category.

At inference, we simply find the nearest center to the query point and assign its label: $\hat{y} = \{y_j | \arg\max_j \tilde{\mu}_j^T \tilde{z}\}$ where $\tilde{z}$ and $\tilde{\mu}$ represent ℓ_2 normalized version of vectors z and μ respectively, while $\hat{y}$ is the prediction. Note that ℓ_2 normalization is done for both query point and centroids, which ensures that euclidean distance reduces to the cosine form. Note that similar to [13], as a preprocessing step, we transform the embeddings by taking the square root of each dimension so that their distribution gets closer to Gaussian. It can be seen as "Tukey's Transformation" [10] with $\lambda = 0.5$.

Method	Backbone	1-shot		5-shot	
		Standard Acc.	Robust Acc.	Standard Acc.	Robust Acc.
AQ	ResNet18	41.48 ± 0.56	20.52 ± 0.45	59.32 ± 0.53	32.18 ± 0.50
Ours (Linear)	ResNet18	42.63 ± 0.56	19.56± 0.45	**61.35 ± 0.51**	30.63 ± 0.52
Ours (CNC)	ResNet18	**44.98 ± 0.59**	**21.38 ± 0.46**	**61.30 ± 0.55**	**33.41 ± 0.51**
AQ	ResNet12	47.95 ± 0.63	21.71 ± 0.47	**69.69 ± 0.51**	35.55 ± 0.53
Ours (Linear)	ResNet12	47.81 ± 0.60	22.81 ± 0.50	65.83 ± 0.53	35.29 ± 0.53
Ours (CNC)	ResNet12	**49.49 ± 0.63**	**25.32 ± 0.52**	66.48 ± 0.53	**38.83 ± 0.57**
AQ	WRN-50-2	38.99 ± 0.55	22.09 ± 0.45	57.11 ± 0.51	33.62 ± 0.50
Ours (Linear)	WRN-50-2	43.14 ± 0.54	19.94 ± 0.43	62.93 ± 0.50	30.52 ± 0.52
Ours (CNC)	WRN-50-2	**46.71 ± 0.62**	**23.04 ± 0.50**	**63.60 ± 0.55**	**36.06 ± 0.54**
AQ	WRN-28-10	44.17 ± 0.60	**23.81 ± 0.48**	62.41 ± 0.54	33.62 ± 0.50
Ours (Linear)	WRN-28-10	52.36 ± 0.62	22.23±0.52	**72.11 ± 0.51**	32.29 ± 0.59
Ours (CNC)	WRN-28-10	**53.22 ± 0.66**	22.91 ± 0.51	70.13 ± 0.52	**35.40 ± 0.58**
AQ	DenseNet121	38.32 ± 0.55	10.19 ± 0.32	56.65 ± 0.51	17.77 ± 0.41
Ours (Linear)	DenseNet121	39.77 ± 0.56	18.16±0.42	57.45± 0.54	27.89 ±0.52
Ours (CNC)	DenseNet121	**42.05± 0.60**	**20.21 ± 0.45**	**58.59±0.55**	**32.24±0.56**
AQ	DenseNet161	37.35 ± 0.52	9.80 ± 0.31	55.97 ± 0.53	16.69 ± 0.38
Ours (Linear)	DenseNet161	40.75 ± 0.55	17.44 ± 0.41	59.84 ± 0.53	27.11± 0.50
Ours (CNC)	DenseNet161	**43.48 ± 0.60**	**20.63 ± 0.45**	**60.92 ± 0.54**	**33.87 ± 0.53**

Table 1. **Results on Mini-ImageNet dataset**. Our CNC method outperforms other approaches which becomes clear as we move to larger architectures. We can also see that our linear classifier serves as a strong baseline and can be used to learn robust few-shot classifier.

Method	Backbone	1-shot		5-shot	
		Standard Acc.	Robust Acc.	Standard Acc.	Robust Acc.
AQ	ResNet18	45.41± 0.68	21.76 ± 0.59	**64.98 ± 0.58**	34.24 ± 0.65
Ours (Linear)	ResNet18	44.76 ± 0.63	21.01 ± 0.58	62.23 ± 0.63	31.60± 0.66
Ours (CNC)	ResNet18	**48.89± 0.71**	**27.16 ± 0.66**	**64.36± 0.61**	**39.13 ± 0.71**

Table 2. **Results on CIFAR-FS dataset**. We can see our CNC method outperforms compared to previous approaches.

3. Experiments

In this section, we describe our experiments and provide implementation details. We evaluate our method on benchmark datasets such as Mini-ImageNet , CIFAR-FS. Due to lack of space, we refer the readers to supplementary material for more implementation details and additional experiments.
We use Pytorch and NVIDIA 2080Ti GPUs for all our experiments. We report the accuracy for 5-way, 1-shot and 5-way, 5-shot settings averaged over 1000 different trials as well as the 95% confidence intervals. Since our goal is to build robust models, we mainly focus on improving Robust Accuracy for which we use 20 iterations of PGD. We compare our results with [3] ,which we refer to as Adversarial Querying (AQ) where adversarial examples are created for query data. We refer to training a linear classifier as **Linear** and the Calibrated Nearest Centroid classifier as **CNC**.
Results: We present our main results on Mini-ImageNet in Table 1 and CIFAR-FS in Table 2. We observe that **CNC** method outperforms other approaches in Robust Accuracy under most settings and boosts standard accuracy as well. We consider the ResNet12 network [9] containing additional regularization such as DropBlock which leads to improved results.We also show results on large-scale architectures such as WideResNets and DenseNets. The difference becomes clear as we move to larger architectures. Our Linear classifier also serves as a strong baseline for robust few-shot settings. Our method is straightforward to scale for large architectures since it is equivalent to standard adversarial training. However, we observed that scaling meta-learning combined with adversarial training is difficult. As a comparison, we trained both AQ and our model on 4 NVIDIA TITAN RTX GPUs using WideResNet-28-10 backbones. AQ method took 1.7 hour/epoch and required 60 epochs while our method took 0.36 hour/epoch for 250 epochs. The total training time for AQ was around 100 hours whereas our method took around 90 hours, showcasing the scalability of our approach.

Analysis: Here, we would like to study the effect of different combinations of the base and novel training, allowing for a careful analysis. We consider Mini-ImageNet dataset on ResNet18 backbone and the results are presented in Table 3. We observe that the simple baseline of robust base training and training a linear classifier during novel training can be a considered a strong baseline (Exp Id 1,4). We also observe that DC algorithm [13] improves standard accuracy but introduces a drop in robustness (Exp Id 2,5). Note that DC involves hallucination of examples at a feature level and

	Base Training	Novel Training			1-shot		5-shot	
Exp. Id	WA	DC	CNC	Linear	Standard Acc.	Robust Acc.	Standard Acc.	Robust Acc.
1	✗	✗	✗	✓	41.40 ± 0.56	18.25 ± 0.45	$\mathbf{59.30 \pm 0.54}$	27.96 ± 0.50
2	✗	✓	✗	✗	$\mathbf{43.72 \pm 0.57}$	14.18 ± 0.38	58.04 ± 0.52	13.97 ± 0.39
3	✗	✗	✓	✗	42.56 ± 0.60	$\mathbf{19.57 \pm 0.45}$	58.22 ± 0.53	$\mathbf{30.42 \pm 0.50}$
4	✓	✗	✗	✓	42.63 ± 0.56	19.56 ± 0.45	$\mathbf{61.35 \pm 0.51}$	30.63 ± 0.52
5	✓	✓	✗	✗	44.73 ± 0.59	15.29 ± 0.41	59.78 ± 0.53	19.49 ± 0.49
6	✓	✗	✓	✗	$\mathbf{44.98 \pm 0.59}$	$\mathbf{21.38 \pm 0.46}$	61.30 ± 0.55	$\mathbf{33.41 \pm 0.51}$

Table 3. Illustration of different configurations of Base and Novel training. Here we show results on ResNet18 backbone on Mini-ImageNet. WA represents Weight Averaging, DC represents Distribution Calibration and CNC corresponds to the Calibrated Nearest Centroid classifier.

	1-shot	
Method	Standard Acc.	Robust Acc.
Linear No Adv	42.63 ± 0.56	19.56 ± 0.45
Linear 7-PGD	42.00 ± 0.56	18.83 ± 0.42
Linear 20-PGD	42.03 ± 0.58	19.01 ± 0.42
Ours (CNC)	$\mathbf{44.98 \pm 0.59}$	$\mathbf{21.38 \pm 0.46}$

Table 4. Experiment where adversarial training is performed on few-shot data when learning the linear classifier using ResNet18 backbone on MiniImageNet. **No Adv** refers to clean examples, **7-PGD** refers to 7-step PGD and **20-PGD** refers to 20-step PGD used in training. Robust Accuracy is calculated using 20-step PGD.

learning a Logistic Regression Classifier. We believe that one reason for the drop in robustness is because the final classifier becomes biased to the clean hallucinated examples, leading to non-robust margins across different classes. In most configurations we observe that our method matches or even outperforms the DC method and more importantly improves robustness (Exp Id 5,6). This shows that the robustness of the base dataset is transferred to the novel setting. The impact of weight averaging (WA) method can be observed when considering Exp Ids 3 and 6. The temporal ensembling nature of the method helps in finding flatter minima, thereby boosting performance. Previous works [5] have shown that this can be used for standard robust classifiers. We observe this holds for a transfer-learning type setting.
We also conduct another experiment where adversarial training is performed during the novel training stage and results are shown in Table 4. We can see that there is no improvement in Robust Accuracy and our CNC method outperforms without the complex adversarial training procedure

Extension to verifiably robust models: An advantage of our simple framework is that we can incorporate methods from the adversarial examples literature for few-shot learning. Specifically, we consider verifiably robust procedures where the goal is to provide a guarantee on adversarial robustness of the model. Standard adversarial training methods do not lead to provably robust models leading to low verified accuracy, we observe a similar trend in our experiments as well. Many methods have been proposed in this area [1, 8, 14] and we consider one of them - Interval Bound Propagation (IBP) [4]. This allows to train a provably robust model whose accuracy does not drop below the verified accuracy for a given threat model. Here we show that replacing PGD-training with IBP during the robust base training stage described in main paper, can lead to verifiable robustness for few-shot classifiers. We show results for 1-shot setting in Table 5 where we use a ResNet18 backbone on CIFAR-FS dataset. We use the training procedure as described in [12] with $\epsilon = 8/255$ for 1000 epochs. Here Robust Accuracy refers to 20-iteration PGD testing and Verified Accuracy is calculated similar to [4]. We believe this experiment can encourage researchers to incorporate more advanced verification methods in the future for few-shot settings.

Method	1-shot		
	Standard Acc.	Robust Acc.	Verified Acc.
IBP + Linear	37.01 ± 0.65	26.77 ± 0.59	21.79 ± 0.55
IBP + CNC	37.72 ± 0.65	28.12 ± 0.62	23.25 ± 0.61

Table 5. We show that it is possible to train verifiably robust models for few-shot settings. This is an added advantage of our framework due to the similarity to standard classifier training. Results are shown using ResNet18 backbone on CIFAR-FS dataset.

4. Conclusion

We present a simple and scalable approach for improving robustness in few-shot image classifiers. Our method outperforms previous approaches when compared with standard few-shot learning benchmarks on both standard and robust accuracy. Note that our method is similar to traditional adversarial machine learning approaches rather than meta-learning methods, hence, simplifying the algorithm. We believe that the simplicity of our approach would be beneficial to the community, upon which researchers can develop advanced robust few-shot classifiers.

References

[1] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019. 4

[2] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 1

[3] Micah Goldblum, Liam Fowl, and Tom Goldstein. Adversarially robust few-shot learning: A meta-learning approach. *NeurIPS*, 2020. 1, 3

[4] Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018. 4

[5] Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020. 2, 4

[6] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018. 2

[7] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018. 1, 2

[8] Matthew Mirman, Timon Gehr, and Martin Vechev. Differentiable abstract interpretation for provably robust neural networks. In *International Conference on Machine Learning*, pages 3578–3586. PMLR, 2018. 4

[9] Boris N Oreshkin, Pau Rodriguez, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *arXiv preprint arXiv:1805.10123*, 2018. 3

[10] John W Tukey et al. *Exploratory data analysis*, volume 2. Reading, Mass., 1977. 2

[11] Ren Wang, Kaidi Xu, Sijia Liu, Pin-Yu Chen, Tsui-Wei Weng, Chuang Gan, and Meng Wang. On fast adversarial robustness adaptation in model-agnostic meta-learning. In *ICLR*, 2021. 1

[12] Kaidi Xu, Zhouxing Shi, Huan Zhang, Yihan Wang, Kai-Wei Chang, Minlie Huang, Bhavya Kailkhura, Xue Lin, and Cho-Jui Hsieh. Automatic perturbation analysis for scalable certified robustness and beyond. *Advances in Neural Information Processing Systems*, 33, 2020. 4

[13] Shuo Yang, Lu Liu, and Min Xu. Free lunch for few-shot learning: Distribution calibration. In *ICLR*, 2021. 2, 3

[14] Huan Zhang, Hongge Chen, Chaowei Xiao, Sven Gowal, Robert Stanforth, Bo Li, Duane Boning, and Cho-Jui Hsieh. Towards stable and efficient training of verifiably robust neural networks. *arXiv preprint arXiv:1906.06316*, 2019. 4