Robust Few-Shot Learning Without Using Any Adversarial Samples

Gaurav Kumar Nayak, Ruchit Rawal, Inder Khatri, and Anirban Chakraborty, Member, IEEE

Abstract—The high cost of acquiring and annotating samples has made the “few-shot” learning problem of prime importance. Existing works mainly focus on improving performance on clean data and overlook robustness concerns on the data perturbed with adversarial noise. Recently, a few efforts have been made to combine the few-shot problem with the robustness objective using sophisticated meta-learning techniques. These methods rely on the generation of adversarial samples in every episode of training, which further adds to the computational burden. To avoid such time-consuming and complicated procedures, we propose a simple but effective alternative that does not require any adversarial samples. Inspired by the cognitive decision-making process in humans, we enforce high-level feature matching between the base class data and their corresponding low-frequency samples in the pretraining stage via self distillation. The model is then fine-tuned on the samples of novel classes where we additionally improve the discriminability of low-frequency query set features via cosine similarity. On a one-shot setting of the CIFAR-FS dataset, our method yields a massive improvement of 60.55% and 62.05% in adversarial accuracy on the projected gradient descent (PGD) and state-of-the-art auto attack, respectively, with a minor drop in clean accuracy compared to the baseline. Moreover, our method only takes 1.69× of the standard training time while being ≈ 5× faster than the state-of-the-art adversarial meta-learning methods. The code is available at https://github.com/vcl-iisc/robust-few-shot-learning.

Index Terms—Adversarial defense, adversarial robustness, few-shot learning, Fourier transform, self distillation.

I. INTRODUCTION

TRAINED deep models deployed in the real world are expected to yield reliable predictions. However, these models remain vulnerable to malicious attacks even after attaining very high performance on the target task. In other words, even a slight perturbation in the input image, which is human- imperceptible, changes the predictions of these models. Such perturbed images are often called “adversarial images” [1], [2], [3], [4], [5]. The perturbations added to the input image are called “adversarial noise,” which are carefully crafted (via adversarial attack) with an objective to “fool” the trained network. Such attacks are threats to trained deep neural networks and can lead to catastrophic consequences in applications such as self-driving cars [6] and biometric authentication [7]. Thus, there is an urgent need to defend the deep models against these attacks.

Adversarial training (AT) [8], [9], [10], [11] is one of the most effective and popular techniques for defense against adversarial attacks. However, its efficacy is heavily dependent on the amount of training data [12], [13]. The adversarially trained models fail to resist adversarial attacks when scarce data or few training samples are present. Their performance gets worse, especially in the case of few-shot learning where the samples in the support set can be as low as one sample per class (also known as one shot) [14].

Popular methods of few-shot learning [15], [16], [17] only focus on improving performance on a clean query set. Their performance drops drastically when query samples are perturbed via adversarial attacks. Yin et al. [18] showed that even the sophisticated meta-learning algorithms for few-shot learning such as relation networks [19], matching networks [20], and MAML [21] are susceptible to adversarial attacks. They proposed an adversarial meta learner, which improved the adversarial accuracy but only against weaker attacks. Their method depended on the MAML framework, and their overall robustness was still poor. Recently, Goldblum et al. [14] combined AT with meta-learning to overcome these problems. Importantly, their framework is agnostic to any specific meta-learning algorithm and showed better robustness. However, they require additional $n+1$ SGD steps in every episode of pretraining compared to standard meta-learning. In each episode, they need to generate adversarial samples through an optimization procedure which runs for $n$ iterations. Overall, these methods that are designed to include robustness in few-shot learning are computationally expensive and dependent on meta-learning, which involves complicated optimization. Also, these methods need to construct adversarial samples for every batch of training data, which itself needs several iterations of backpropagation, adding a further computational overhead.

This work handles the above concerns by proposing a simple but effective alternative that avoids complicated training like meta-learning and does not require any adversarial samples in its training. Our approach for robust few-shot learning is inspired by human cognition. Humans primarily rely on...
Fig. 1. Robust acc. (on PGD) versus Clean acc. (left) and Robust acc. versus total time taken (right) (includes pretraining, fine-tuning, and inference time for a given support and query set) for one-shot five-way setting on CIFAR-FS dataset. Our method obtains a large gain in robust acc. without compromising much on clean performance and takes significantly lesser computational overhead compared to robust few-shot methods.

low frequencies of a given image to associate it with one of the target labels [22], [23]. However, deep models are tuned to obtain high accuracy, and in that process, they become sensitive to high-frequency details of the image. Hence, perturbations in those regions can easily impact the model’s decision while remaining human imperceptible. These high-frequency details get heavily contaminated by adversarial attacks (refer Section IV-A in supplementary). Hence, to resist these models from being fooled by adversarial images and enforce similar behavior to humans in decision-making, we explicitly force the network to learn on low frequencies. However, high frequencies allow deep models to be more discriminative, necessary for attaining good accuracy on “clean” (i.e., unperturbed) samples. Hence, by ensuring that the models trained on low frequencies are as discriminative as those trained with original images, the right framework can achieve good accuracy on both clean and adversarial images. With this intuition, our proposed approach has broadly two stages: pretraining and fine-tuning with evaluation.

In the pretraining stage, we first train a teacher network on the original samples of base classes, which is used for self-distillation. The student network is then trained on low frequencies of original samples while matching discriminative features of the teacher network on corresponding original data via cosine similarity. Similarly, we also apply this loss on the query set and their low-frequency counterparts while fine-tuning on novel classes that further boost the robustness.

The overall contributions are summarized as follows.
1) We are the first to propose a computationally cheaper non-meta-learning approach for robust few-shot learning that does not require any adversarial samples.
2) We train networks to enforce similar decision-making as human cognitive processes to incorporate robustness. For this, we perform pretraining using self-distillation via cosine similarity to make the feature representation of low-frequency samples close to original samples of base classes. Similarly, we also apply this loss on the query set and their low-frequency counterparts while fine-tuning on novel classes that further boost the robustness.
3) The model trained with low frequencies obtained via FFT at low or high radius results in poor performance in either clean or adversarial samples. To avoid this, we progressively apply cosine similarity loss on the low-frequency samples obtained by varying radius from an easy high radius (favoring more high frequencies) to gradual hard low radii (favoring less high frequencies).
4) We demonstrate the efficacy of our proposed framework via extensive ablations and experiments on benchmark datasets for different few-shot settings.
II. RELATED WORKS

A. Adversarial Defense

AT [2], [9], [10], [11] has proved to be the most effective and reliable technique, which generates adversarial samples during training using min-max optimization. Goodfellow et al. [2] proposed the first AT method that crafted the adversarial examples by fast gradient sign method (FGSM) attack. Following this, Madry et al. [8] reported improvement in robustness performance by using a stronger attack, namely, projected gradient descent (PGD) in AT. Later on, Zhang et al. [25] further regularized AT by using a KL divergence loss term to enforce similar output distribution on clean and adversarial data. These additional regularizations in AT methods can be viewed similar to self-distillation ($S \rightarrow S$ in Section V-B) but rely on adversarial samples. With explicit use of large teacher ($T$) network, robustness is transferred to lightweight student ($S$) in [4], [26]. However, the network $T$ is assumed to be adversarially trained and robust to adversarial attacks unlike ours where $T$ is a nonrobust network trained with standard cross entropy and $S$ has identical architecture as $T$. The AT methods usually have high computational complexity and also require high-capacity networks [26]. An alternative to these, some non-AT-based methods like JARN [27], BPFC [28], and GCE [29] have been proposed. They do not rely on adversarial samples and thus take less training time. However, they are not robust to a wide range of attacks and their performance is lower when faced with stronger attacks [30].

B. Few-Shot Classification

One of the earliest and simplest baselines for few-shot classification consists of pretraining the model on the base classes and then fine-tuning on the novel-class support set [31], [32]. Dhillon et al. [24] demonstrated that transductive fine-tuning outperforms various state-of-the-art techniques across various benchmark datasets. Recently, the few-shot-learning community has shifted its attention to meta-learning, which attempts to “learn” the “learning-algorithm” itself over a distribution of tasks, such that it can quickly adapt to novel tasks using fine-tuning [15], [16], [17], [21], [33], [34].

C. Robust Few-Shot Learning

Current state-of-the-art adversarial robustness techniques prove to be ineffective in a few-shot setting due to their high dependence on large-labeled datasets [14], [18]. Yin et al. [18] proposed a novel meta-learning algorithm that facilitates the learning of accurate and robust few-shot learners. Goldblum et al. [14] proposed an algorithm-agnostic adversarial robustness technique [adversarial querying (AQ)] that can be combined with the existing state-of-the-art meta-learning techniques. Liu et al. [35] proposed a robust meta-learning [long-term cross adversarial training (LCAT)] that halves the training time compared to current AQ while maintaining similar performance.

The aforementioned techniques provide robustness by generating adversarial samples at the training stage, which leads to high-computational costs. Moreover, these methods are primarily designed for meta-learning-based solutions that involve complicated optimization (e.g., difficult to optimize for larger architectures [24]). More recently, Dong et al. [36] proposed a non-meta-learning-based few-shot robustness method that learns a robust feature extractor as well as a postprocessing feature purifier network using AT. On similar lines, Subramanya and Pirsiavash [37] train a robust network on base samples using AT and classify samples from novel classes by finding the nearest base-category centroids in the feature space of the network. In contrast to existing works, our proposed method is a non-meta-learning AT-free approach, which is not only computationally cheaper but also more effective than prior works.

Recently, several works [22], [32] observed that deep neural networks rely heavily on the high-frequency spectrum for prediction, making them highly sensitive to any adversarial perturbations in those regions. Li et al. [38] further observed that many publicly available robust models showed a preference for low-frequency spectrum. Inspired by the above findings and to mimic human cognition in deep neural networks, we employ frequency regularization to enforce few-shot learners to rely on the low-frequency spectrum while ensuring model features to highly discriminative, resulting in correct predictions.

III. PRELIMINARIES

A. Notations

In the pretraining stage, we use base class data $D_B = \{(x^B_i, y^B_i)\}_{i=1}^{b}$ containing $b$ samples each belonging to one of the $c_1$ classes. The novel class data $D_N = \{D_{N_i}, D_{N_q}\}$ whose samples are from $c_2$ different classes are used in the fine-tuning stage. $D_{N_i} = \{(x^N_{N_i}, y^N_{N_i})\}_{i=1}^{n_i}$ is the support set consisting of $n_i$ labeled samples. $D_{N_q} = \{x^N_{N_q}\}_{i=1}^{n_q}$ is the query set with $n_q$ unlabeled samples.

The teacher and student models are represented by $T$ and $S$ whose outputs $T(x; \theta)$ and $S(x; \theta)$ are the logits for any input image $x$ that is fed to the networks $T$ and $S$, respectively. We denote the softmax function by $P_{soft}(.)$ and its output for a particular class $i$ by $P_{soft}(i)$. $A_{adv}$ denote a set of $m$ distinct adversarial attacks, i.e., $A_{adv} = \{A_{k}\}_{k=1}^{m}$. An $i$th adversarial image of query set, i.e., $\tilde{x}^N_{N_i}$, is obtained by using an adversarial attack $A_k \in A_{adv}$ such that it fools the model $S$. FT($.)$ and $FT^{-1}(.)$ denote fourier and inverse fourier transforms. For any $i$th sample $(x_i)$ in the spatial domain, its corresponding representation in the frequency domain is represented by $z_i$. The low and high frequencies of $z_i$ separated at a radius $r$ are denoted by $z_{lr}$ and $zh_{lr}$ whose spatial representations are $xl_{ir}$ and $xh_{ir}$, respectively.

B. Adversarial Attacks

An attack $A_k \in A_{adv}$ carefully crafts noise ($\delta$) such that this noise when added to the input image ($x$) is human- imperceptible but the network predictions on them change. In other words, a model $M$ on input $(\tilde{x})$ gets fooled when $\text{argmax}(M(x)) \neq \text{argmax}(M(\tilde{x}))$, where $\tilde{x} = x + \delta$ and $\|\delta\| \leq \epsilon$. We consider the popular $l_\infty$ threat model in our
setup, i.e., the $l_\infty$-norm of the perturbations ($\|\delta\|_\infty$) is within the $\epsilon$ ball.

C. Fourier Transform

This transformation establishes a mapping between spatial and frequency domains. To obtain the frequency representation of an i\textsuperscript{th} sample from its spatial domain, FT(\cdot) is applied, i.e., $z_i = \text{FT}(x_i)$. The frequencies in $z_i$ can be further split into low and high at a radius $r$ (note that different strategies for selecting the value of radius $r$ are also discussed in Section V-D) using Hadamard product with corresponding low- and high-frequency masks as shown in the following:

$$z_{l,r} = z_i \circ F^{\text{mask}_l}, \quad z_{h,r} = z_i \circ F^{\text{mask}_h}. \quad (1)$$

The value of $F^{\text{mask}_l}$ at a $p$\textsuperscript{th} row and $q$\textsuperscript{th} column is calculated as

$$F^{\text{mask}_l}_{pq} = I \left( \sqrt{\left( \frac{p-d}{2}\right)^2 + \left( q - d/2 \right)^2} < r \right) \quad (2)$$

where $I$ is an indicator function. The zero frequency of the sample ($z_i$) obtained via FT(\cdot) is shifted to the center $(d/2,d/2)$ where $d \times d$ is the dimension of $z_i$. The value of low-frequency mask at a location $(p,q)$ is 1 if its euclidean distance from the center is less than $r$; otherwise, the value is 0. $F^{\text{mask}_h}$ is the complement of $F^{\text{mask}_l}$. To get the spatial representation of the above frequencies

$$x_{l,r} = \text{FT}^{-1}(z_{l,r}), \quad x_{h,r} = \text{FT}^{-1}(z_{h,r}). \quad (3)$$

D. Self Distillation

Transfer of knowledge from network $T$ to network $S$ having identical architectures.

E. Few-Shot Setting

During the training stage, the model is trained from scratch on $D_B$ data containing base classes. During testing, the pretrained model is fine-tuned on a small-sized support set $D_S$, $(n_s \ll b)$ that has samples from novel classes. For a k-way n-shot setup, the following holds: $c_z = k$ and $n_s = k \cdot n$. We follow the setup described by Goldblum et al. [14], where the fine-tuned model is susceptible to adversarial attacks and the query set is perturbed with an objective to fool the model. The robustness of the fine-tuned model is then quantified by evaluating it on both the clean query set $\{x_i^B\}_{i=1}^{n_q}$ and the query perturbed samples $\{x_i^{B,i}\}_{i=1}^{n_q}$.

IV. PROPOSED APPROACH

Our proposed approach is broadly divided into two stages: 1) pretraining and 2) fine-tuning and evaluation. In pretraining stage, we first train the teacher network on $D_B$ and then use it for self distillation. The student network is trained on low-frequency samples of $D_B$ along with the feature responses of the teacher model on original samples. In second stage, the pretrained model is fine-tuned transductively on $D_N$ along with our additional loss to improve discriminativeness on low-frequency query samples. The fine-tuned model is then evaluated on both the clean and adversarially perturbed query samples using the logits obtained at different low frequencies of the input. The proposed method is also described in Fig. 2. Next, we discuss each of these stages in detail.

A. Stage 1 (Pretraining)

In the first step, we train the network $T$ with trainable parameters $\theta_t$ from scratch on $D_B$ by minimizing the cross entropy loss ($L_{ce}$) as shown in the following:

$$\theta_t^* = \min_{\theta_t} L_{ce} = \min_{\theta_t} \frac{1}{b} \sum_{i=1}^{b} -\log \left( P_{\text{eo}} \left( T(x_i^B) \right) \right). \quad (4)$$

$L_{ce}$ loss ensures that network $T$ learns highly discriminative features on the base class data. In the second step, we use the trained network $T$ from step 1 as a teacher model to train the network $S$ via self distillation [39], [40], [41]. Networks $T$ and $S$ are of identical architectures where $S$ is initialized with $\theta_t^*$. The network $T$ is used to obtain logit responses on samples of $D_B$. The network $S$ is trained on low frequencies of $D_B$ samples to match the logit responses via maximization of cosine similarity loss ($L_{cs}$) as mentioned in the following along with $L_{ce}$ loss on $D_B$:

$$\theta_s^* = \min_{\theta_s} (L_{ce} - L_{cs}); \quad L_{cs} = \frac{1}{b} \sum_{i=1}^{b} \left( S(x_i^B)^T \cdot T(x_i^B) \right). \quad (5)$$

Distinguishing class labels using low-frequency samples is a significantly challenging task compared to vanilla classification as low-frequency samples inherently contain less information. Hence, directly training the student network $S$ on low-frequency images using $L_{ce}$ would lead to poor performance. We overcome this challenge by using the $L_{cs}$ loss, which encourages network $S$ to be as discriminative on low-frequency samples as the network $T$ is for original samples in the feature (logits) space. To obtain $x_i^B$ in order to compute this loss above, we first convert the spatial data into frequency domain using FT, i.e., $z_i^B = \text{FT}(x_i^B), \forall i \in [1 \ldots b]$. Next, low-frequency samples $(x_i^{B,i})$ are obtained by superimposing a low-frequency mask (at a radius $r$) on the frequency-domain representation followed by FT$^{-1}$. Refer to Fourier transform operations in Section III.

Generating the low-frequency mask at a suitable radius $r^*$ is crucial for achieving good robustness as, if $r^*$ is too low, it could lead to lower performance due to less discriminability (even if no adversarial contamination is there) and if $r^*$ is too high it could lead to lower performance due to adversarial contamination (even if more discriminability is there). However, empirically estimating $r^*$ is impractical without generating adversarial samples as it highly depends on the dataset, model, etc. In such a scenario, an intuitive approach could be to select the lowest radius possible (to ensure less adversarial contamination) while maintaining discriminativeness (to ensure decent clean accuracy). For that, we perform a weighted sampling over the radii defined on the range $[r_{max}, r_{min}]$, where weights form a long-tailed distribution (described in the following) by initially assigning a higher weight to $r_{max}$ as it is relatively...
Fig. 2. Detailed steps of our proposed approach for robust few-shot learning. In pretraining stage, the teacher network \( T \) is trained on base class data \( D_B \) (step 1). The student network \( S \) is trained (step 2) via self distillation using the trained network \( T \). The progressive learning (PL) module uses an initial long tail weight distribution to sample radius. The distribution is shifted if student performance on the peak radius \( r_p \) crosses the threshold \( \ell \). In the linear programming (LP) module, the low-pass filtered version of an input image is computed using a low-frequency mask (at sampled radius \( r^{\text{lp}} \)) in the Fourier transform (FT), followed by \( FT^{-1} \). In fine-tuning stage (step 1), apart from the \( L_{\text{ce}} \) on support set \( \{x^{s}N_q\} \) and \( L_{\text{ce}} \) loss on query set \( \{x^{q}N_q\} \), we additionally apply \( L_{\text{ls}} \) loss on \( \{x^{s}N_q\} \). The weighted distribution \( \hat{w}^{\text{range}} \) over radii via PL module obtained at the end of pretraining is used in this stage to sample radii at which low frequencies are obtained via LP module. Finally, in the evaluation step, the logits obtained at each radius \( r_{\text{max}} \) to \( r_{\text{min}} \) are weighted using the weight distribution \( \hat{w}^{\text{range}} \). The complete training in both the stages does not require generation of any adversarial sample, making the method computationally efficient.

easier to learn at the high radius:

\[
u^{\text{range}}_r = \lambda^i, \text{ such that } \lambda \in (0, 1), \quad r = (r^{\text{range}})_i \quad \text{and} \quad r^{\text{range}} = [r_{\text{max}}, r_{\text{max}} - 1, \ldots, r_{\text{min}}]. \tag{6}\]

Here, \( i \) is the index of radius \( r \in r^{\text{range}} \). The stochastic-weighted sampling also encourages the model to avoid overfitting on samples of any particular radius. Once the training accuracy of the model on low-frequency samples (at \( r_{\text{max}} \)) exceeds the threshold \( \ell \), we shift the weighting scheme by having the highest weight at radius \( r_{\text{max}} - 1 \).

We repeat this process till \( r_p \) (radius corresponding to the highest weight) is equal to \( r_{\text{min}} \). We term this strategy PL, as we learn discriminative features on low-frequency samples by gradually moving from high to low radius

\[
L_{\text{ce}} = 1/n_s \sum_{i=1}^{n_s} -\log P_{\text{softmax}} \left( S(x_i^{s}N_q) \right) y_i^{s} \\
L_{e} = 1/n_q \sum_{i=1}^{n_q} \sum_{j=1}^{k} P_{\text{softmax}} \left( S(x_i^{e}N_q) \right) \log P_{\text{softmax}} \left( S(x_i^{e}N_q) \right) \\
L_{\text{ce}} = 1/n_q \sum_{i=1}^{n_q} \frac{S(x_i^{N_q})^T S(x_i^{N_q})}{\|S(x_i^{N_q})\|^2}.
\tag{7}\]

B. Stage 2 (Fine-Tuning and Evaluation)

A new classification layer having cosine softmax with \( k \) class nodes for a \( k \)-way \( n \)-shot setting is added on the top of the logits of pretrained model \( S \) obtained from the previous stage. The weights of this layer are initialized with class mean features of the support set where the mean feature vector corresponding to each class is taken as a row in the weight matrix with biases as zeros. This layer’s inputs and weights are normalized before taking softmax on their dot product.

We perform fine-tuning on this model using samples of \( D_N \) with losses in (7).

The discriminative low-frequency features learned during pretraining on \( D_B \) might not be directly useful during fine-tuning on \( D_N \) as \( D_B \) and \( D_N \) consist of disjoint sets of classes. Thus, we apply \( L_{\text{ce}} \) loss on the query-set samples in the fine-tuning stage to improve discriminativeness of low-frequency features of novel class samples. Further, evaluating the performance on low-frequency samples of a small-sized support set is not a good indicator for updating the weight distribution. Even if the query data are relatively substantial in quantity, we cannot use it to evaluate the performance and compare it with the threshold, as it is unlabeled. Hence, rather than following a similar PL strategy as pretraining, we employ the weight distribution \( \hat{w}^{\text{range}} \) obtained at the end of pretraining to generate the corresponding low-frequency samples.

The fine-tuned model \( S \) with optimal parameters (i.e., \( Q' = \min \{L_{\text{ce}} + L_e - L_{\text{ce}}\} \) is used to predict labels on the query set \( D_q \). During evaluation on a query sample, we perform a weighted average of the logits predicted on low-frequency input at each radius \( r \in [r_{\text{min}}, r_{\text{max}}] \).

\[
\text{Pred} = \arg \max \left( \sum_{r=r_{\text{min}}}^{r_{\text{max}}} (\log t_r \cdot \hat{w}_r^{\text{range}}) \right). \tag{8}\]
Here, $\hat{w}_r$ is the weight corresponding to radius $r \in r^{\text{range}}$ using the weight distribution obtained at the end of pretraining and logit, is the logits when low-frequency query sample ($x_{l,S}$) obtained at the radius $r$ is fed to the fine-tuned model $S$. All the steps related to different stages (pretraining and fine-tuning with evaluation) that are discussed here are also structurally summarized to provide an overall algorithm which is presented in the supplementary material (Section V). Also, qualitatively demonstrate the working of our method on clean and adversarially perturbed input in Section IV-B of supplementary material. Next we validate our proposed method through extensive experiments including different ablations and comparison with the state-of-the-art.

V. Experiments

We demonstrate the effectiveness of our technique by performing experiments on two benchmarks datasets in few-shot-learning, specifically CIFAR-FS [16] and Mini-ImageNet [20]. As per the standard protocol, we use 15 as query shot and report the estimate of the mean accuracy evaluated over 1000 few-shot episodes along with the 95% confidence interval of this estimate (value put in parentheses). We report both the clean as well as adversarial performance where the query samples are perturbed via adversarial attack [PGD and state-of-the-art auto attack (AA)]. Refer to Section III in the supplementary for attack parameters and training details. We also evaluate our method in more challenging realistic scenarios, such as considering an unequal amount of samples across classes in a support set, or varying the query set size. In such cases, we also observe similar consistent performance. We use the ResNet-12 [42] architecture as our backbone feature extractor for all the experiments and ablations. Refer to supplementary (Section I-B), where we also perform an ablation over different choices of backbone architectures and observe that our method consistently improves the adversarial accuracy across architectures compared to the respective baselines. Similar to [24], our baseline (vanilla) model consists of pretraining using $L_{ce}$ and mix-up regularization, and fine-tuning using $L_{ce}$ (on support set) and $L_e$ (on query set). The value of radius is kept fixed at 2, i.e., $r = 2$ for ablations pertaining to $L_{ce}$, and the evaluation is performed on the same $r$ unless otherwise specified. In Sections V-A–V-H, we first validate the utility of the various components used in pretraining and fine-tuning stages of our method (Sections V-A–V-D), followed by comparison with the state-of-the-art methods [Section V-E]. For additional training details refer to supplementary (Section III).

A. Frequency Regularization in Pretraining

We motivate the benefit of regularizing the vanilla network (S) to match the feature responses (logits) of the original and low-frequency samples during pretraining in Table I. The vanilla network achieves decent one-shot and five-shot clean accuracies on the CIFAR-FS dataset; however, it is extremely vulnerable to adversarial perturbations. Thus, we regularize $S$ by matching the logits for an original sample as input and its corresponding low-frequency sample using $L_{ce}$. The aim of such frequency-based regularization is to reduce the dependence of the model on high-frequency components. We observe that such frequency-based regularization indeed leads to a nontrivial increase in robust accuracy (across different attacks). However, there is a huge drop in the clean accuracy as low-frequency samples inherently have less discriminative content (at the cost of reducing adversarial perturbations/contamination) compared to original samples.

B. Utility of Teacher Network

In an attempt to increase the model’s discriminability on low-frequency samples, we draw inspiration from the knowledge-distillation literature. Specifically, we first train a teacher model (T) using $L_{ce}$ and employ it in a self-distillation setup in order to increase the student model’s (S) discriminability. $S$ is trained using a combination of $L_{ce}$ and $L_{cs}$ loss as in (5). We further fix the batchnorm layer parameters of the student (S) after initializing S with the teacher model’s (T) weights. The rich fine-grained knowledge from T assists S to learn more discriminative features even for low-frequency samples. In Table II, we observe a significant boost in performance (across all the metrics) for the $T \rightarrow S$ self-distillation setup compared to the previous setting where the low-frequency and original samples logits were obtained from the same S network. Although, at par with most state-of-the-art few-shot robustness techniques, there is still a significant drop in clean accuracy (compared to the vanilla network) for the $T \rightarrow S$ setup.

Once the $S$ model is trained on $D_S$ using the $T \rightarrow S$ self-distillation setup as described in stage 1, we shift our focus to
improvements during fine-tuning stage to further boost clean and robust accuracies.

C. Frequency Regularization in Fine-Tuning

Motivated by the effectiveness of $L_{cs}$ during pretraining, we also apply $L_{cs}$ during fine-tuning. Since $L_{cs}$ can be applied in an unsupervised manner (no label information required), we directly optimize it on the query set, in addition to $L_{ce}$ and $L_e$ losses [as shown in (7)]. In Table III, we observe that adding $L_{cs}$ at fine-tuning stage leads to impressive gains as both the clean and robust accuracies increase by $\approx 15\%$ and $20\%$ in one-shot and five-shot settings, respectively. We also performed additional ablations in supplementary such as different options for applying $L_{cs}$ loss (Section I-C) and choices of loss functions for frequency regularization (Section I-A).

D. Effectiveness of PL

The previously described ablations were performed at a fixed radius $r = 2$. Now, we explain the efficacy of our PL technique, where we set $r_{\text{min}}$ to 2 and $r_{\text{max}}$ to $N/2$ where $N$ is the largest frequency component for a given dataset. As discussed in Section IV, both clean and adversarial performance significantly suffer if $r^*$ is either too high (implies high adversarial contamination) or too low (implies low discriminability). We empirically validate this phenomenon by repeating our proposed frequency regularization method at two distinct radii, i.e., $r = 2$ (low) and $r = 16$ (high) (shown in Table V) and observe that although there is an increase in clean accuracy for $r = 16$ as compared to $r = 2$, there is a huge drop in robust accuracy. This sensitivity of clean and robust accuracies on the radius $r$ motivates us to propose a data-driven method of selecting/sampling $r$ during both pretraining and fine-tuning. We use our PL strategy to perform a dynamic-weighted sampling over a broad range of radii $[r_{\text{max}}, r_{\text{min}}]$ [as per (6)]. We observe that PL-based frequency regularization leads to a further boost in clean and robust accuracy. Most importantly, it automates the process of selecting the radius in a data-driven way. We also experiment with a baseline ensembling technique that uniformly samples radii between $[2, N/2]$ and performs ensembling (equal weight to each radius) on the same range. We note that the uniform sampling-based setting significantly underperforms in comparison to PL-based sampling across both one-shot/five-shot settings, especially on robust accuracies which highlights the efficacy of our PL technique. In addition to these, we also perform sensitivity analysis where we vary the values of $r^*$ range (Section II in supplementary) and hyperparameter $\lambda$ used in weighted sampling [see (6)] (Section V-G).

E. Comparison With the State-of-the-Art Methods

We compare the efficacy of our complete proposed method against various few-shot adversarial robustness techniques, namely, AQ [14], LCAT [35], LCAT+TRADES [35], Dong et al. [36], and CNC [37]. Dhillon et al. [24] use transductive fine-tuning where a combination of $L_c$ loss (on query set) and traditional $L_{ce}$ loss (on support set) is applied during fine-tuning. To have a fair comparison with them, we follow the same protocol. However, they do not evaluate performance on query perturbed samples; hence, our setup is far more challenging than theirs (baseline). The results are reported in Table IV. Our method obtains significant gains in adversarial accuracy while retaining most of the clean accuracy compared to the baseline. On CIFAR-FS dataset, our method yields significant improvement of $\approx 11\%$–29\% on clean data and $\approx 21\%$–35\% on adversarial data for one-shot while $\approx 8\%$–32\% on clean data and $\approx 20\%$–40\% on adversarial data for five-shot settings), over existing state-of-the-art methods.

We obtain similar observations on large scale MiniImageNet, which is another popular benchmark dataset. Interestingly, we note that, in general, non-meta-learning-based AT approaches (i.e., Dong et al. and CNC)
TABLE V

PERFORMANCE (MEAN ACCURACY IN % WITH CONFIDENCE INTERVAL) COMPARISON OF OUR PROPOSED PL STRATEGY WITH DIFFERENT RADIUS SELECTION TECHNIQUES. OUR PL TECHNIQUE BASED ON DYNAMIC WEIGHTED SAMPLING NOT ONLY OVERCOMES THE SENSITIVITY IN THE MODEL PERFORMANCE ASSOCIATED WITH CHOICE OF A FIXED RADIUS BUT ALSO LEADS TO SIGNIFICANTLY BETTER ADVERSARIAL PERFORMANCE THAN UNIFORM SAMPLING OVER RADIUS.

| Setup  | Method       | Clean   | PGD     | Auto Attack |
|--------|--------------|---------|---------|-------------|
| 1-SHOT | Fixed (r=2)  | 61.49 (0.70) | 60.73 (0.70) | 59.83 (0.70) |
|        | Fixed (r=16) | 63.48 (0.69) | 59.09 (0.19) | 51.19 (0.18) |
|        | ∝ (2, 16) PL on [16, 2] (Ours) | 63.90 (0.70) | 52.83 (0.80) | 55.80 (0.77) |
| 5-SHOT | Fixed (r=2)  | 73.17 (0.60) | 72.26 (0.61) | 72.53 (0.80) |
|        | Fixed (r=16) | 78.81 (0.53) | 3.46 (0.17) | 5.24 (0.23) |
|        | ∝ (2, 16) PL on [16, 2] (Ours) | 79.06 (0.54) | 68.71 (0.71) | 71.69 (0.67) |

TABLE VI

ANALYZING THE SENSITIVITY OF OUR PROPOSED APPROACH TOWARD VARIATIONS IN AMOUNT OF PER-CLASS SAMPLES IN THE SUPPORT SET FOR A FIVE-WAY SETTING ON CIFAR-FS DATASET

| Support Set (Mix, Max. Class Distribution) | Technique | Ours Transductive | Ours Non-Transductive |
|--------------------------------------------|-----------|------------------|-----------------------|
| Clean | PGD | Clean | PGD |
| 1, 5, [3 5 5 5 5] (balanced) | 79.79 (0.55) | 75.93 (0.61) | 73.91 (0.55) | 61.82 (0.65) |
| 1, 5, [4 5 5 5 5] (unbalanced) | 78.99 (0.54) | 74.81 (0.61) | 73.64 (0.55) | 61.46 (0.65) |
| 1, 10, [10 10 10 10 10] (balanced) | 81.85 (0.51) | 78.28 (0.58) | 77.98 (0.51) | 66.94 (0.62) |
| 1, 10, [7 4 8 5 7] (unbalanced) | 82.17 (0.50) | 78.20 (0.57) | 77.67 (0.52) | 66.57 (0.62) |
| 1, 15, [15 15 15 15 15] (balanced) | 83.38 (0.50) | 79.63 (0.57) | 79.92 (0.50) | 69.17 (0.61) |
| 1, 15, [1 1 1 1 1 1] (unbalanced) | 83.10 (0.50) | 79.31 (0.57) | 79.11 (0.50) | 68.49 (0.60) |

TABLE VII

INVESTIGATING THE EFFECT OF VARYING THE SIZE OF QUERY SET FROM FIVE SAMPLES PER CLASS TO 25 SAMPLES PER CLASS FOR FIVE-WAY ONE-SHOT SETTING ON CIFAR-FS DATASET

| Query Set Size (nq) | Technique | Ours Transductive | Ours Non-Transductive |
|---------------------|-----------|------------------|-----------------------|
| Clean | PGD | Clean | PGD |
| 25 | 62.95 (0.84) | 60.47 (0.86) | 55.22 (0.80) | 41.64 (0.79) |
| 50 | 63.24 (0.74) | 59.30 (0.78) | 55.10 (0.70) | 41.77 (0.70) |
| 75 | 65.03 (0.72) | 60.55 (0.76) | 55.00 (0.69) | 41.55 (0.70) |
| 100 | 64.30 (0.70) | 38.90 (0.75) | 55.03 (0.68) | 41.63 (0.68) |
| 125 | 64.94 (0.70) | 39.08 (0.74) | 54.98 (0.66) | 41.55 (0.67) |

Fig. 3. Demonstrating the change in the long-tailed distribution when λ is varied. The distribution approximates to sampling from a fixed radius for very small values of λ, whereas it becomes uniform distribution for very high values of λ (close to 1).

Fig. 4. Variation in performance of our proposed approach with respect to λ on CIFAR-FS. λ is varied for different intermediate values starting from 0.5. Both the clean and robust performances exhibit little sensitivity when λ takes values between 0.5 and 0.8. PGD and AA refer to the adversarial attacks using projected gradient descent and auto attack, respectively.

outperform meta-learning-based AT approaches (AQ, LCAT, and LCAT+TRADES). However, we substantially outperform all the existing state-of-the-art robust few-shot methods on both clean and adversarial accuracies. Thus, it clearly highlights the utility of our frequency-regularized pretraining and fine-tuning. Note that in contrast to existing few-shot robustness techniques, we do not generate adversarial samples during pretraining or fine-tuning. This allows us to significantly reduce the training time while simultaneously outperforming them across both clean and robust accuracies (refer Fig. 1).

F. Varying Support and Query Set Size

In this section, we evaluate our proposed method in a more general setting where we also vary the amount of samples per class in the support set and the quantity of query set during inference. This setting is in contrast with the traditional setup (for a k-way n-shot, fixed n across the k classes and fixed query set size) that are typically followed in robust few-shot domain.

We randomly select the support set size of each class instead of fixing it to a class-balanced value like 1 or 5 (as used in the main article). For instance, in a k-way setting, we sample k random numbers between [1, max_shot_value], constraining each class to have minimum one support set sample and maximum max_shot_value. We perform experiments on CIFAR-FS dataset and fix the number of classes, i.e., k as 5 and query set size as 75 (15 query samples per class). In Table VI, we observe that there is only a minor drop for both clean and robust accuracies on our proposed
method (for both transductive and nontransductive setups) in the random sampling (unbalanced) setting compared to the balanced setting. The trend is consistent as we vary the max_shot_value from 5 to 15.

In Table VII, we vary the quantity of the query set from five samples per class (i.e., query set size = 25) to 25 samples per class (i.e., query set size = 125) on CIFAR-FS for five-way one-shot setting. We observe that our proposed method (both transductive and nontransductive fine-tuning) performs decently well as the performance variation in both clean robust accuracy is mild, when the query set size is scaled up.

**G. Sensitivity Analysis: λ Value in Weighted Sampling**

In the PL scheme [see (6)], we use a distribution (i.e., long tailed) over $r^{\text{range}}$ to select the radius stochastically. As the distribution depends on input parameter $\lambda$, it becomes crucial to study the effect of $\lambda$ on the distribution. Fig. 3 shows the changes in the distribution on varying the value of parameter $\lambda$. It can be observed that for the small values (i.e., 0.1) of $\lambda$, the weights are concentrated over a very small range in $r^{\text{range}}$, approximating it as sampling from a fixed radius. On the other hand, for a higher value of $\lambda$ (close to 1), the weights become uniform throughout the $r^{\text{range}}$. Earlier in our experiments (Section V-D), we showed that for both the extremum conditions, i.e., using a fixed radius and the uniform distribution over $r^{\text{range}}$ give poor performance. Hence, we expect to obtain the optimal results for some intermediate values of $\lambda$.

Thus to verify it, in this section, we perform experiments by taking different intermediate values of $\lambda$ starting from 0.5. The obtained results are shown in Fig 4. It can be observed that the clean accuracy of the model over different values of $\lambda$ has minimal variation. In the case of robust accuracy, there is not much variation within the range [0.5–0.8]. However, it starts dropping significantly as the value of $\lambda$ increases after 0.8. This validates our initial intuition where we expected poor performance near $\lambda$ close to 1 (approximating to uniform distribution).

Through this analysis, we found selecting $\lambda$ from an intermediate range, i.e., [0.5–0.8] can provide decent overall performance and the performance is not very sensitive to $\lambda$ within this range. Specifically, even though performance corresponding to $\lambda = 0.6$ seems to be the overall best, $\lambda = 0.8$ also gave competitive performance in all our experiments, which further confirms that the performance is not sensitive to the choice of $\lambda$ in the intermediate range.

**H. t-SNE Visualizations**

In this section, we qualitatively analyze the performance of our proposed approach by visualizing the feature representations obtained at the prefinal layer of the fine-tuned student model. We visualize the feature representations of the query samples using t-SNE algorithm [43] for a five-way five-shot setting on CIFAR-FS dataset. Fig. 5(a) and (c) represent the classwise (i.e., with respect to the predicted class) feature visualization, while Fig. 5(b) and (d) indicates whether the sample was correctly classified or not by the model.

We first visualize the features of clean original and their corresponding low-frequency samples in Fig. 5(a) and (b). The first row represents the visualizations corresponding to the vanilla network (i.e., trained transductively [24] on original samples), which served as a baseline. We observe that although the vanilla network can correctly classify the original samples (represented via symbol “•”) (Fig. 5(b), top row), it is not able to achieve similar performance on the corresponding low-frequency samples (represented via symbol “×”). This
behavior can be primarily attributed to the low-discriminative power (and consequently poor features learned) on low-frequency samples by the vanilla networks (Fig. 5(a), top row). Next, in the second row, we visualize the feature representations of our proposed model on the clean data. We observe that we achieve significantly better performance on the low-frequency samples (Fig. 5(b), bottom row) without sacrificing performance on original samples. As we will see in the subsequent paragraph, this improvement in discriminability on low-frequency samples is key to achieving significant boost in robustness accuracy without much drop in the clean accuracy.

We also visualize the feature representations of adversarially perturbed samples and their corresponding low-frequency counterparts. We note that the vanilla model is highly vulnerable to adversarial attacks as it misclassified all the adversarially perturbed “original” samples (refer Fig. 5(d), top row). However, this trend does not hold for adversarially perturbed low-frequency samples as the vanilla network can predict a fraction of them correctly. Thus, some nontrivial robustness can be achieved for the vanilla model by simply masking out the high-frequency components though the robustness performance is limited by low-discriminative power on the low-frequency samples. Finally, we visualize the feature representations learned using our proposed approach and note that we are able to correctly classify the majority of low-frequency representation of adversarially perturbed samples (refer Fig. 5(d), bottom row). This observation is in line with our empirical results described earlier, where we noticed a significant boost in the robustness accuracy through our proposed approach compared to the vanilla model.

VI. DISCUSSION AND CONCLUSION

In this work, we present for the first time a non-meta-learning method for robust few-shot learning that achieves the state-of-the-art robust accuracy (without compromising much on clean data accuracy) on multiple benchmarks with marginal computational overhead. We primarily achieve this by learning rich discriminative features on low-frequency data in a self-distillation setup during pretraining. In the fine-tuning stage, the student network further benefits by applying additional cosine similarity loss. We also qualitatively verify it by analyzing our proposed approach through visualization of the t-SNE plots of the feature representations of clean and adversarial query samples with their corresponding low-frequency counterparts.

Our PL module automates the process of selecting the low-frequency mask radius (distribution) in an end-to-end data-driven manner. This is especially important as selecting a fixed radius (e.g., \( r = 2 \)) might work well on certain datasets (e.g., CIFAR-FS) but can yield suboptimal performance on others (e.g., Mini-ImageNet). On the contrary, our PL module learns the weight distribution (i.e., allocation of weights to radii) in a data-driven manner during the pertaining stage. This ensures that the knowledge acquired (regarding the weight distribution) during pretraining is suitable and specific to the dataset-domain and model architecture. We also observed that using weighted ensemble for evaluation usually provided better clean and robust accuracy, compared to evaluating on a peak radius (radius corresponding to highest weight), as it appropriately weights the logit contribution from each radius. As evidenced by experimental results, our method leads to improvement in robust accuracy without compromising much on clean accuracy and is trained with significantly lesser computational overhead as compared to robust meta-learning methods.

We hope that the simplicity yet strong performance of our method would serve as a compelling baseline to robust few-shot learning methods and urge research community to explore more non-meta-learning and adversarial-sample-free robustness methods.

ACKNOWLEDGMENT

The authors conducted the work during their time at the Department of Computational and Data Sciences, Indian Institute of Science, Bengaluru, India.

REFERENCES

[1] C. Szegedy et al., “Intriguing properties of neural networks,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2014, pp. 1–10.
[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in Proc. Int. Conf. Learn. Represent., 2015, pp. 1–11.
[3] Z. Che et al., “SMGSA: A new ensemble adversarial attack powered by long-term gradient memories,” IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 3, pp. 1051–1065, May 2022.
[4] A. Chatnurvedi and U. Garain, “Mimic and fool: A task-agnostic adversarial attack,” IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 4, pp. 1801–1808, Apr. 2021.
[5] M. Duan, Y. Qin, J. Deng, K. Li, and B. Xiao, “Dual attention adversarial attacks with limited perturbations,” IEEE Trans. Neural Netw. Learn. Syst., early access, May 21, 2023, doi: 10.1109/TNNLS.2023.3274142.
[6] J. Yang, A. Boloor, A. Chakrabarti, X. Zhang, and Y. Vorobeychik, “Finding physical adversarial examples for autonomous driving with fast and differentiable image compositing,” 2020, arXiv:2010.08944.
[7] J. Fei, Z. Xia, P. Yu, and F. Xiao, “Adversarial attacks on fingerprint liveness detection,” EURASIP J. Image Video Process., 2020, no. 1, pp. 1–11, Dec. 2020.
[8] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” in Proc. Int. Conf. Learn. Represent., 2018, pp. 1–23.
[9] M. Xu, T. Zhang, Z. Li, and D. Zhang, “InfoAT: Improving adversarial training using the information bottleneck principle,” IEEE Trans. Neural Netw. Learn. Syst., early access, Jun. 22, 2022, doi: 10.1109/TNNLS.2022.3183095.
[10] H. J. Lee, Y. Yu, and M. Ro, “Advancing adversarial training by injecting booster signal,” IEEE Trans. Neural Netw. Learn. Syst., early access, Apr. 14, 2023, doi: 10.1109/TNNLS.2023.3264256.
[11] N. Ye, Q. Li, X.-Y. Zhou, and Z. Zhu, “An annealing mechanism for adversarial training acceleration,” IEEE Trans. Neural Netw. Learn. Syst., vol. 34, no. 2, pp. 882–893, Feb. 2023.
[12] Y. Carmon, A. Raghunathan, L. Schmidt, P. Liang, and J. Duchi, “Unlabeled data improves adversarial robustness,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2019, pp. 1–12.
[13] J.-B. Alayrac, J. Uesato, P.-S. Huang, A. Fawzi, R. Stanforth, and C. Szegedy, “Finding physical adversarial examples for autonomous driving with fast and differentiable image compositing,” 2020, arXiv:2010.08944.
[14] J. Fei, Z. Xia, P. Yu, and F. Xiao, “Adversarial attacks on fingerprint liveness detection,” EURASIP J. Image Video Process., 2020, no. 1, pp. 1–11, Dec. 2020.
[15] M. Duan, Y. Qin, J. Deng, K. Li, and B. Xiao, “Dual attention adversarial attacks with limited perturbations,” IEEE Trans. Neural Netw. Learn. Syst., early access, May 21, 2023, doi: 10.1109/TNNLS.2023.3274142.
[16] J. Yang, A. Boloor, A. Chakrabarti, X. Zhang, and Y. Vorobeychik, “Finding physical adversarial examples for autonomous driving with fast and differentiable image compositing,” 2020, arXiv:2010.08944.
[17] J. Fei, Z. Xia, P. Yu, and F. Xiao, “Adversarial attacks on fingerprint liveness detection,” EURASIP J. Image Video Process., 2020, no. 1, pp. 1–11, Dec. 2020.
[18] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” in Proc. Int. Conf. Learn. Represent., 2018, pp. 1–23.
[19] M. Xu, T. Zhang, Z. Li, and D. Zhang, “InfoAT: Improving adversarial training using the information bottleneck principle,” IEEE Trans. Neural Netw. Learn. Syst., early access, Jun. 22, 2022, doi: 10.1109/TNNLS.2022.3183095.
[20] H. J. Lee, Y. Yu, and M. Ro, “Advancing adversarial training by injecting booster signal,” IEEE Trans. Neural Netw. Learn. Syst., early access, Apr. 14, 2023, doi: 10.1109/TNNLS.2023.3264256.
[21] N. Ye, Q. Li, X.-Y. Zhou, and Z. Zhu, “An annealing mechanism for adversarial training acceleration,” IEEE Trans. Neural Netw. Learn. Syst., vol. 34, no. 2, pp. 882–893, Feb. 2023.
[22] Y. Carmon, A. Raghunathan, L. Schmidt, P. Liang, and J. Duchi, “Unlabeled data improves adversarial robustness,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2019, pp. 1–12.
