Supplementary Material

Learning Transferable Adversarial Robust Representations via Multi-view Consistency

A Experimental Details[1]

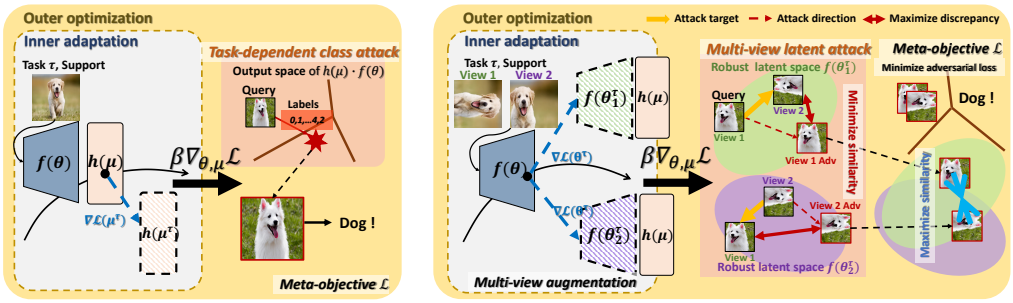
A.1 Dataset

For meta-training, we utilize CIFAR-FS (Bertinetto et al., 2019) and Mini-ImageNet (Russakovsky et al., 2015). CIFAR-FS and Mini-ImageNet each consist of 100 classes, with 64 classes for meta-training, 16 classes for meta-validation, and 20 classes for meta-testing. To evaluate our model on few-shot classification tasks, we utilize 6 benchmark few-shot datasets: CIFAR-FS (Bertinetto et al., 2019), Mini-ImageNet (Russakovsky et al., 2015), Tiered-ImageNet (Finn et al., 2017), Cars (Krause et al., 2013), CUB (Welinder et al., 2010), and Flower (Nilsback & Zisserman, 2008). Additionally, for assessing robust transferability, we employ 3 additional benchmark standard image classification datasets: CIFAR-10, CIFAR-100, and STL-10. CIFAR-10 and CIFAR-100 consist of 50,000 training images and 10,000 test images each, with 10 and 100 classes, respectively. All images are resized to a resolution of $32\times32\times3$ (width, height, and channel) for meta-training. Specifically, we leverage the *TorchMeta*[2] library to load the few-shot datasets into our frameworks.

A.2 Meta-train

We use ResNet12 and ResNet18 as the base encoder network for CIFAR-FS and Mini-ImageNet. All models are trained using tasks that consist of a 5-way 5-shot support set images and a 5-way 15-shot query set images. They are then validated using clean tasks, which consist of a 5-way 1-shot support set images and a 5-way 15-shot query set images. Specifically, the model is trained with randomly selected 200 tasks and validated with randomly selected 100 tasks. To optimize the models, we train them for 100,000 steps using the SGD optimizer with a weight decay of 1e-4. For data augmentation, we apply random crop with a size ranging from 0.08 to 1.0, color jitter with a probability of 0.8, horizontal flip with a probability of 0.5, grayscale with a probability of 0.2, gaussian blur with a probability of 0.0, and solarization with a probability ranging from 0.0 to 0.2. Normalization is excluded for adversarial training.

In the case of adversarial learning, we employ 3 steps and 7 steps for our task-agnostic latent adversarial attack. To generate adversaries using the query set images, we take a gradient step within an l_∞ norm ball with $\epsilon = 8.0/255.0$ and $\alpha = 2.0/255.0$. To obtain robust representations, we utilize an original meta-training objective, a multi-view instance-wise adversarial training objective, and a cosine distance loss with a regularization hyperparameter λ of 6.0 for adversarial training. The overall model figure of MAVRL is shown in Figure 5.



Previous approach MAVRL

Figure 5: Concept of MARVRL compared to previous approach (AQ).

[1]Code is available in the Supplementary zip folder.

[2]https://github.com/tristandeleu/pytorch-meta

A.3 HYPERPARAMETER DETAILS OF META-LEARNING FRAMEWORK

We use a single step for the inner optimization of meta-training and meta-testing to improve computational efficiency, with an inner learning rate of 0.005. For the outer optimization, we employ an outer learning rate of 0.005 for CIFAR-FS. In the case of Mini-ImageNet, we use the same step size as CIFAR-FS but with a different inner learning rate of 0.001 and an outer optimization learning rate of 0.001. Both datasets utilize a batch size of 16. The training time on CIFAR-FS takes approximately 33 hours using a single NVIDIA GeForce RTX-3090.

A.4 META-TEST

The trained models are evaluated using 400 randomly selected unseen tasks from the test set. Each task is composed of a 5-way 5-shot support set images and a 5-way 15-shot query set images. In the evaluation process, we employ a single step for the inner optimization. It is important to note that we use the same learning rate and meta-step size as the model was trained with during meta-training.

A.5 ADVERSARIAL EVALUATION

We evaluate the robustness of our trained models against two types of attacks: PGD (Madry et al., 2018) and AutoAttack (Croce & Hein, 2020). For all l_∞ PGD attacks, we conduct them within a norm ball size of $\epsilon = 8./255.$, with a step size of $\alpha = 8./2550.$, and using 20 steps for inner maximization. AutoAttack[3] is a combination of four different types of attacks (APGD-CE, APGD-T, FAB-T, and Square). During test time, we utilize the standard version of AutoAttack.

B ADVERSARIAL TRAINING

Many existing works aim at enhancing the adversarial robustness of models trained using supervised learning (Goodfellow et al., 2015; Carlini & Wagner, 2017; Papernot et al., 2016), such as adversarial training (AT) and regularized Kullback-Leibler divergence (KLD) loss. AT uses project gradient descent (PGD) (Madry et al., 2018) to maximize loss in inner-maximization loops while minimizing overall loss on adversarial samples. TRADES (Zhang et al., 2019) have theoretically shown that KLD loss enhances robustness by enforcing consistency in predictive distribution between clean and adversarial examples. Transfer learning (Shafahi et al., 2019) can also be used to transfer learned robust representations to new domains with few data. One of the most similar adversarial learning methods to ours is RoCL (Kim et al., 2020), which proposes to adversarially train a robust neural network without *labeled data*, by instance-wise adversarial attack. However, we found that the simple application of instance-wise attacks on few-shot learning is not effective (Table 6).

C ADDITIONAL ABLATION EXPERIMENTAL RESULTS

C.1 MAVRL VS NAÏVE COMBINATION OF SSL AND AML.

Our framework consists of three novel technical components: 1) Bootstrapping multi-view encoders 2) task-agnostic multi-view latent adversarial attack and 3) meta-adversarial multi-view representation learning. To provide more detailed ablation experiments on our approach, we demonstrate the results of each ablation experiment along with the figure and algorithms.

As discussed in Section 3.3, a naive combination of self-supervised learning (SSL) and adversarial meta-learning (AML) fails to achieve transferable robust representation learning. We investigate two cases of this naïve combination. First, we incorporate task-agnostic instance-wise attacks for adversarial training with a *single encoder* during the outer optimization phase. We generate adversarial examples following previous works (Kim et al., 2020) using a *single encoder* (parameters θ^τ), as shown in Eq. 7. We then minimize the adversarial loss in the outer optimization, as indicated in Eq. 8 [1], i.e., $\mathcal{L}_{\texttt{abl}}[1]$. However, as demonstrated in Table 6, without multi-view encoders, the model fails to generate strong adversarial examples, resulting in insufficient robustness even within the seen domain. Additionally, when we incorporate representation learning loss in the outer optimization using Eq. 8 [2], i.e., $\mathcal{L}_{\texttt{abl}}[2]$. the model exhibits slightly improved transferable robustness but still

[3] https://github.com/fra31/auto-attack

performs poorly. The difference is illustrated as blue text in Eq. 7, 8. In conclusion, a simple combination of self-supervised learning and adversarial meta-learning, as presented in Algorithm 3, leads to representation collapse due to the small batch size, rendering the task-agnostic adversarial attack ineffective in leveraging transferable robustness in unseen domains.

$$\begin{aligned}\delta_1^{\mathrm{i}+1} &= \prod_{B(x,\epsilon)}\Big(\delta_1^{\mathrm{i}} + \gamma\,\mathtt{sign}\big(\nabla_{\delta_1^{\mathrm{i}}}\mathcal{L}_{\mathtt{sim}}(f_{\theta^\tau}(t_1(x^q)+\delta_1^{\mathrm{i}}), f_{\theta^\tau}(t_2(x^q)), \{f_{\theta^\tau}(x^q_{\mathtt{neg}})\})\big)\Big),\\ \delta_2^{\mathrm{i}+1} &= \prod_{B(x,\epsilon)}\Big(\delta_2^{\mathrm{i}} + \gamma\,\mathtt{sign}\big(\nabla_{\delta_2^{\mathrm{i}}}\mathcal{L}_{\mathtt{sim}}(f_{\theta^\tau}(t_2(x^q)+\delta_2^{\mathrm{i}}), f_{\theta^\tau}(t_1(x^q)), \{f_{\theta^\tau}(x^q_{\mathtt{neg}})\})\big)\Big),\end{aligned} \tag{7}$$

$$\text{[1]}\ \min_{\theta,\phi,\alpha}\mathbb{E}_{p_{\mathcal{D}}(\tau)}\Big[\mathbb{E}_{\mathcal{Q}}\big[\overbrace{\big(\mathcal{L}_{\mathtt{ce}}(g_\phi\circ f_{\theta^\tau}(t_j(x^q)), y^q\big)}^{\text{original meta-learning objective}} + \underbrace{\lambda\mathcal{L}_{\mathtt{kl}}(g_\phi\circ f_{\theta^\tau}(t_j(x^q)^{\mathtt{adv}}), g_\phi\circ f_{\theta^\tau}(t_j(x^q))))}_{\text{multi-view adversarial training}}\big]\Big],$$

$$\begin{aligned}\text{[2]}\ \min_{\theta,\phi,\alpha}\mathbb{E}_{p_{\mathcal{D}}(\tau)}\Big[\mathbb{E}_{\mathcal{Q}}\big[&\overbrace{\big(\mathcal{L}_{\mathtt{ce}}(g_\phi\circ f_{\theta^\tau}(t_j(x^q)), y^q\big)}^{\text{original meta-learning objective}} + \underbrace{\lambda\mathcal{L}_{\mathtt{kl}}(g_\phi\circ f_{\theta^\tau}(t_j(x^q)^{\mathtt{adv}}), g_\phi\circ f_{\theta^\tau}(t_j(x^q))))}_{\text{multi-view adversarial training}}\\ &+ \underbrace{\mathcal{L}_{\mathtt{cos}}(f_{\theta^\tau}(t_1(x^q)^{\mathtt{adv}}), f_{\theta^\tau}(t_2(x^q)^{\mathtt{adv}}))}_{\text{multi-view consistency loss}}\big]\Big].\end{aligned} \tag{8}$$

Algorithm 2: MAVRL.

Input: Meta-training distribution $p_{\mathcal{D}}(\tau)$, random data augmentations $t_1(\cdot), t_2(\cdot)$, feature encoder $f_\theta(\cdot)$, classifier $g_\phi(\cdot)$, meta-learning rate β

Output: Adversarially meta-trained parameters θ, ϕ, α

while *not converged* **do**
- Sample M different meta-training tasks $\{\tau\} = \{(\mathcal{S}, \mathcal{Q})\} \sim p_{\mathcal{D}}(\tau)$
- **for** $i = 1, \cdots, M$ **do**
  - /* Bootstrapped multi-view encoders. */
  - $\theta_j^\tau \leftarrow \theta - \alpha\nabla_\theta\mathbb{E}_{\mathcal{S}}[\mathcal{L}_{\mathtt{ce}}(g_\phi\circ f_\theta(t_j(x^s)), y^s)]$, for $j = 1, 2$
  - /* Generate multi-view latent adversaries. */
  - $t_1(x^q)^{\mathtt{adv}}, t_2(x^q)^{\mathtt{adv}} \leftarrow t_1(x^q)+\delta_1^q, t_2(x^q)+\delta_2^q$ // δ_1^q, δ_2^q are obtained by Eq. 5
  - /* Our loss. */
  - $\mathcal{L}^\tau_{\mathtt{ours}} \leftarrow \mathbb{E}_{\mathcal{Q}}\big[\sum_{j=1,2}(\mathcal{L}_{\mathtt{ce}}(\cdot,\cdot) + \lambda\mathcal{L}_{\mathtt{kl}}(\cdot,\cdot)) + \mathcal{L}_{\mathtt{cos}}(\cdot,\cdot)\big]$ // See details in Eq. 6
- /* Update meta-parameters */
- $[\theta, \phi, \alpha] \leftarrow [\theta, \phi, \alpha] - \beta\nabla_{\theta,\phi,\alpha}\sum_{\{\tau\}}\mathcal{L}^\tau_{\mathtt{ours}}/M$

return meta-parameters θ, ϕ, α

Algorithm 3: Naïve combination.

Input: Meta-training distribution $p_{\mathcal{D}}(\tau)$, random data augmentations $t_1(\cdot), t_2(\cdot)$, feature encoder $f_\theta(\cdot)$, classifier $g_\phi(\cdot)$, meta-learning rate β

Output: Adversarially meta-trained parameters θ, ϕ, α

while *not converged* **do**
- Sample M different meta-training tasks $\{\tau\} = \{(\mathcal{S}, \mathcal{Q})\} \sim p_{\mathcal{D}}(\tau)$
- **for** $i = 1, \cdots, M$ **do**
  - /* Single encoder. */
  - $\theta^\tau \leftarrow \theta - \alpha\nabla_\theta\mathbb{E}_{\mathcal{S}}[\mathcal{L}_{\mathtt{ce}}(g_\phi\circ f_\theta(x^s), y^s)]$
  - /* Generate multi-view latent adversaries. */
  - $t_1(x^q)^{\mathtt{adv}}, t_2(x^q)^{\mathtt{adv}} \leftarrow t_1(x^q)+\delta_1^q, t_2(x^q)+\delta_2^q$ // δ_1^q, δ_2^q are obtained by Eq. 7
  - /* Ablation loss. */
  - [1] $\mathcal{L}^\tau_{\mathtt{abl}} \leftarrow \mathbb{E}_{\mathcal{Q}}\big[\mathcal{L}_{\mathtt{ce}}(\cdot,\cdot) + \lambda\mathcal{L}_{\mathtt{kl}}(\cdot,\cdot)\big]$
  - [2] $\mathcal{L}^\tau_{\mathtt{abl}} \leftarrow \mathbb{E}_{\mathcal{Q}}\big[\mathcal{L}_{\mathtt{ce}}(\cdot,\cdot) + \lambda\mathcal{L}_{\mathtt{kl}}(\cdot,\cdot) + \mathcal{L}_{\mathtt{cos}}(\cdot,\cdot)\big]$ // See details in Eq. 8
- /* Update meta-parameters */
- $[\theta, \phi, \alpha] \leftarrow [\theta, \phi, \alpha] - \beta\nabla_{\theta,\phi,\alpha}\sum_{\{\tau\}}\mathcal{L}^\tau_{\mathtt{abl}}/M$

return meta-parameters θ, ϕ, α

We conducted ablation experiments on each component of our framework, as summarized in Table 6. Each component significantly contributes to improving robustness in both seen and unseen domains. Notably, when we incorporate bootstrapping multi-view encoders, the model achieves substantially enhanced robustness in the unseen domains. These results highlight the crucial role of each of our novel components in achieving robustness in unseen domains.

Table 6: Results of adversarial robustness for 5-way 5-shot classification tasks on unseen and seen domains. All adversarial meta-learning methods are trained on CIFAR-FS. Rob. stands for accuracy (%) calculated with PGD-20 attack ($\epsilon = 8./255.$, $\gamma = \epsilon/10$). Ablation condition is as follow: [1]: bootstrap multi-view encoders, [2]: task-agnostic adversarial attack, [3]: cosine distance loss.

	CIFAR-FS →			Mini-ImageNet		Tiered-ImageNet		CUB		Flower		Cars		Avg.		CIFAR-FS	
	[1]	[2]	[3]	Clean	Rob.	Clean	Rob.	Clean	Rob.	Clean	Rob.	Clean	Rob.	Clean	Rob.	Clean	Rob.
Naive Combination	-	✓	-	20.30	17.99	21.70	18.66	21.59	18.19	24.77	21.33	21.74	19.30	22.02	19.09	22.64	19.84
	-	✓	✓	20.01	19.24	20.02	18.39	20.00	18.68	20.01	19.56	19.98	19.66	20.00	19.11	20.04	18.52
Ablation	✓	✓	-	40.42	16.60	54.55	28.93	50.01	21.92	69.47	39.79	40.52	16.79	50.99	24.81	68.08	42.97
	✓	-	✓	45.47	12.63	56.14	27.02	52.78	20.32	72.53	39.05	41.44	15.20	53.67	22.84	70.14	41.75
Ours	✓	✓	✓	45.82	24.12	51.46	30.06	48.56	25.23	66.49	42.16	38.29	19.43	50.32	28.20	67.75	43.42

Table 7: Results of adversarial robustness for 5-way 5-shot classification tasks on unseen and seen domains. All adversarial meta-learning methods are trained on CIFAR-FS. Rob. stands for accuracy (%) calculated with PGD-20 attack ($\epsilon = 8./255.$, $\gamma = \epsilon/10$).

CIFAR-FS →	Mini-ImageNet		Tiered-ImageNet		CUB		Flower		Cars		Avg.		CIFAR-FS	
	Clean	Rob.	Clean	Rob.	Clean	Rob.	Clean	Rob.	Clean	Rob.	Clean	Rob.	Clean	Rob.
RMAML (Wang et al., 2021)	28.05	6.65	29.54	9.30	30.24	5.67	42.91	10.79	31.72	5.56	32.49	7.39	57.95	35.30
Ours-MAML	30.35	15.02	40.12	23.83	37.52	18.67	46.54	29.09	31.48	16.24	37.20	20.57	47.26	31.58
Ours-MetaSGD	45.82	24.12	51.46	30.06	48.56	25.23	66.49	42.16	38.29	19.43	50.32	28.20	67.75	43.42

C.2 Different meta-learning methods and adversarial attack iterations

To demonstrate the efficacy of MAVRL in achieving robust and transferable representations, we conducted experiments across three distinct meta-learning frameworks, including MAML (Finn et al., 2017), FOMAML (Finn et al., 2017) and MetaSGD (Li et al., 2017). Furthermore, we evaluate the resilience of MAVRL by subjecting it to multi-view latent attacks of varying attack iterations, specifically 3-step and 7-step.

Table 8 highlights that MAVRL outperforms the previous adversarial meta-learning model (Goldblum et al., 2020) in terms of adversarial robustness by more than 10%, irrespective of meta-learning strategies. Furthermore, MAVRL exhibits remarkable robustness with just 3 steps of multi-view latent attacks compared to AQ (Goldblum et al., 2020), which is trained with PGD-7 attacks (i.e., class-wise attack). To emphasize the superiority of multi-view latent attacks over class-wise attacks at the representation level, we calculate feature similarity between clean and adversarial examples using CKA (Kornblith et al., 2019). Notably, the latent attack yielded a lower CKA value than the class-wise attack (as seen in Figure 4b), which means that latent attacks produce perturbations that deviate more significantly from the original clean images, making them more challenging. Through these remarkable results, we underscore that our proposed multi-view latent attack served as a stronger attack that promotes the robust transferability of the model to unseen domains, even with fewer gradient steps of attacks and limited data.

C.3 Consistency Loss regularized to learn Generalized Features

The objective of the proposed meta-adversarial learning framework consists of three different elements, 1) cross-entropy loss, 2) multi-view instance-wise adversarial loss, and 3) cosine distance loss as in Eq. 6. In particular, cosine distance loss enforces the consistency between two maximally dissimilar views of adversaries, leading meta-learners to achieve transferable robustness. We further examine the effectiveness of cosine distance loss by altering it to other consistency loss including contrastive loss and KLD loss. As shown in Table 9, the cosine similarity term was the most effective objective for aligning the multi-view latent spaces, demonstrating the highest unseen domain robustness on average. This is because cosine distance loss explicitly aligns the two latent vectors obtained from multi-view latent attacks while others implicitly enforce the consistency to differently generated adversarial representations.

D MAVRL meta-trained on larger dataset

To provide a more convincing comparison, we additionally conduct experiments where models are meta-trained on a larger dataset, Tiered-ImageNet (Russakovsky et al., 2015). Tiered-ImageNet consists of 779,165 images and 608 classes which are 351, 97, and 160 classes for meta-training, meta-validation, and meta-testing respectively. All images are resized by $3 \times 32 \times 32$ resolution

Table 8: Results of transferable robustness with different meta-learning framework and attack iteration in 5-shot tasks. All models are trained with 5-way 5-shot images on CIFAR-FS and Mini-ImageNet. Rob. stands for accuracy(%) that is calculated with PGD-20 attack ($\epsilon = 8./255.$, step size=$\epsilon/10$). Clean stands for test accuracy(%) of clean images. All models are trained on ResNet12. The number of attack iterations during training is marked in parentheses next to the meta-train dataset. Further, we denote (θ) next to the meta-learning strategies to notice that we update only the encoder parameters during inner optimization.

	CIFAR-FS (3 steps) →	Mini-ImageNet		Tiered-ImageNet		CUB		Flower		Cars	
		Clean	Rob.	Clean	Rob.	Clean	Rob.	Clean	Rob.	Clean	Rob.
MAVRL	+MAML (θ) (Finn et al., 2017)	34.35	15.76	39.06	20.08	42.32	17.46	57.74	32.70	35.78	15.79
MAVRL	+FOMAML (θ) (Finn et al., 2017)	32.06	16.69	37.97	22.15	37.65	17.50	56.68	34.08	36.33	18.45
MAVRL	+MetaSGD (θ) (Li et al., 2017)	44.64	15.75	53.25	28.05	50.78	22.44	70.08	41.52	40.08	16.88
	AQ (Goldblum et al., 2020)	33.79	1.59	36.41	2.27	39.35	2.88	58.69	6.59	37.39	2.30
	CIFAR-FS (7 steps) →	Mini-ImageNet		Tiered-ImageNet		CUB		Flower		Cars	
		Clean	Rob.	Clean	Rob.	Clean	Rob.	Clean	Rob.	Clean	Rob.
MAVRL	+MAML (θ) (Finn et al., 2017)	32.57	16.12	38.90	22.51	39.44	16.52	56.79	32.83	36.58	16.56
MAVRL	+FOMAML (θ) (Finn et al., 2017)	31.71	17.40	37.33	23.28	38.63	18.79	59.57	36.79	37.94	21.34
MAVRL	+MetaSGD (θ) (Li et al., 2017)	45.82	24.12	51.46	30.06	48.56	25.23	66.49	42.16	38.29	19.43
	AQ (Goldblum et al., 2020)	33.09	3.32	37.41	5.05	38.37	4.10	60.14	11.03	36.83	4.47

Table 9: Ablation results of transferable robustness with different consistency loss in MAVRL framework. Rob. stands for accuracy (%) that is calculated with PGD-20 attack ($\epsilon = 8./255.$, step size=$\epsilon/10$). Clean stands for test accuracy (%) of clean images. All models are meta-trained on CIFAR-FS with PGD-7 attacks on ResNet12.

	CIFAR-FS →												
Consistency	Mini-ImageNet		Tiered-ImageNet		CUB		Flower		Cars		Avg.	
Loss	Clean	Rob.	Clean	Rob.	Clean	Rob.	Clean	Rob.	Clean	Rob.	Clean	Rob.
KL	21.69	17.56	25.52	20.22	25.78	19.49	33.76	24.80	23.25	18.22	26.00	20.06
Contrastive	43.62	20.17	45.47	22.98	50.59	25.36	70.61	40.12	39.32	18.60	49.92	25.45
Cosine distance	45.82	24.12	51.46	30.06	48.56	25.23	66.49	42.16	38.29	19.43	**50.32**	**28.20**

(channel, width, and height) to validate the model's robust transferability to unseen domain tasks. As demonstrated in Table 10, when the models are meta-trained on a larger dataset, our meta-leaner consistently outperforms the previous adversarial meta-learning method (AQ) for both clean and robust accuracy on unseen domain tasks. This indicates that MAVRL can effectively learn robust representations transferred to unseen domains, regardless of how unseen domains are different from the meta-trained dataset.

E TRANSFERABLE ROBUSTNESS ON NON-RGB DOMAINS

To demonstrate the ability to learn transferable robustness on unseen domain tasks of the proposed framework MAVRL, we further employ unseen domains of non-RGB domains (i.e., ISIC (Codella et al., 2018), CropDisease (Mohanty et al., 2016), and EuroSAT (Helber et al., 2019)), which have much more different distributions from meta-trained RGB dataset (i.e., CIFAR-FS). This experiment can encompass variations such as color scale (RGB, Gray-scale), and distinct image type (i.e., MRI, satellite imagery), enabling more accurate evaluation of the transferable robustness across a wide range of domains. As shown in Table 11, MAVRL exhibits outstanding transferable robustness of 17.47% on average even in non-RGB unseen domain tasks compared to previous adversarial meta-learning method.

F OBFUSCATED GRADIENT

All robust accuracies reported in our paper are calculated using the strength $\epsilon = 8./255.$, step size $\alpha = 8./2550.$, and 20 steps for the ℓ_∞ PGD attacks. In order to assess the presence of obfuscated gradient issues, we conduct experiments with two different settings of ℓ_∞ PGD attacks. Firstly, we apply PGD attacks with an extremely large strength, expecting the robust accuracy to be nearly zero. Secondly, we use the same strength but different step sizes and steps, specifically $4./2550.$ and 40 respectively. In this case, we expect the robust accuracy to remain the same as the robust accuracy from our original evaluation setting. To demonstrate this, we evaluate MAVRL trained on CIFAR-FS with ResNet12 as the base encoder, and built on top of the FOMAML architecture as reported in

Table 10: Results of transferable robustness in 5-way 5-shot unseen domain tasks that are trained on 5-way 5-shot Tiered-ImageNet. Rob. stands for accuracy (%) that is calculated with PGD-20 attack ($\epsilon = 8./255.$, step size=$\epsilon/10$). Clean stands for test accuracy (%) of clean images. All models are trained with PGD-7 attacks on ResNet12.

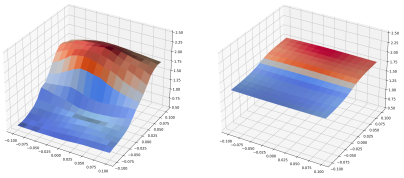
Tiered-ImageNet →	CIFAR-FS		Mini-ImageNet		CUB		Flower		Cars		Avg.	
	Clean	Rob.	Clean	Rob.	Clean	Rob.	Clean	Rob.	Clean	Rob.	Clean	Rob.
AQ (Goldblum et al., 2020)	42.33	2.48	25.91	0.44	36.29	0.31	56.01	1.81	32.64	1.01	38.64	1.21
Ours	**71.11**	**33.68**	**51.16**	**17.40**	**53.48**	**17.75**	**63.58**	**16.12**	**40.14**	**12.00**	**55.89**	**19.39**

Table 11: Results of transferable adversarial robustness in 5-way 15-shot non-RGB unseen domain tasks that are trained on CIFAR-FS.

CIFAR-FS →	EuroSAT		ISIC		CropDisease		Avg.	
	Clean	Rob.	Clean	Rob.	Clean	Rob.	Clean	Rob.
AQ	46.05	4.62	**31.90**	0.62	47.38	0.51	41.78	1.92
Ours	**59.39**	**19.90**	30.77	**5.23**	**57.85**	**27.28**	**49.34**	**17.47**

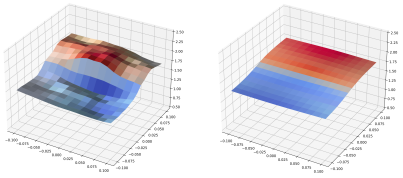
Table 8. As shown in Table 12, we confirm that our models do not exhibit any obfuscated gradient issues.

G VISUALIZATION OF LOSS SURFACE



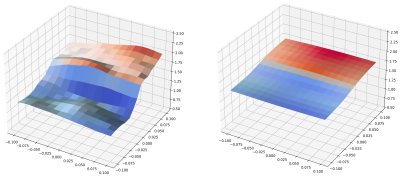
AQ Ours

Figure 6: CIFAR-FS - seen



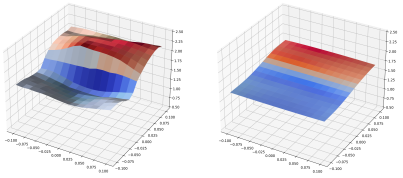
AQ Ours

Figure 7: Mini-ImageNet-*unseen*



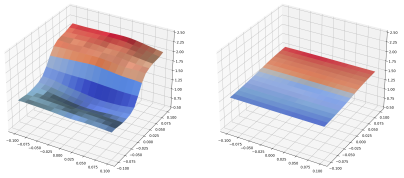
AQ Ours

Figure 8: Tiered-ImageNet-*unseen*



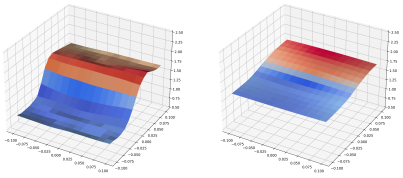
AQ Ours

Figure 9: CUB-*unseen*



AQ Ours

Figure 10: CARS-*unseen*



AQ Ours

Figure 11: Flowers-*unseen*

We visualize the loss surface of our model and baseline AQ (Goldblum et al., 2020) model. As shown in the above Figure our model has a smoother loss surface both in the seen domain and unseen domain while the baseline has a relatively less smooth surface.

Table 12: Test accuracy(%) on multiple benchmark datasets for 5-shots. Robustness is calculated with PGD-20 attack ($\epsilon = 8./255.$, step size=$\epsilon/10$), clean is for clean images. All models are adversarially meta-trained on CIFAR-FS.

				CIFAR-FS		Mini-ImageNet		Tiered-ImageNet		CUB		Cars	
	Strength (ϵ)	Step size (α)	Steps	Clean	PGD ℓ_∞	Clean	PGD ℓ_∞	Clean	PGD ℓ_∞	Clean	PGD ℓ_∞	Clean	PGD ℓ_∞
3 steps	8.0/255.0	8.0/2550.0	20	53.42	35.95	32.06	16.69	37.97	22.15	37.65	17.50	36.33	18.45
	8.0/255.0	4.0/2550.0	40	53.04	35.35	31.70	16.01	38.06	21.98	37.77	18.12	36.10	18.02
	300.0	8.0/2550.0	20	52.72	0.47	31.83	0.92	37.73	0.85	38.14	0.55	36.21	0.44
7 steps	8.0/255.0	8.0/2550.0	20	51.90	36.01	31.71	17.40	37.33	23.28	38.63	18.79	37.94	21.34
	8.0/255.0	4.0/2550.0	40	52.50	36.39	31.95	17.49	38.44	24.22	38.18	18.87	37.41	20.92
	300.0	8.0/2550.0	20	52.20	0.50	31.97	0.59	37.53	0.65	38.78	0.45	37.64	0.48

Table 13: Experiments results for self-supervised robust full-finetuning of MAVRL and the state-of-the-art adversarial self-supervised models on unseen domains. While MAVRL is trained on CIFAR-FS with bilevel attacks , adversarial self-supervised models are trained on full-dataset of CIFAR-100. All models are trained on ResNet18, and evaluated against PGD-20 attacks ($\epsilon = 8./255.$) and AutoAttack (AA) (Croce & Hein, 2020)

		CIFAR-10			CIFAR-100			STL-10			Cars			CUB		
	Method	Clean	PGD ℓ_∞	AA	Clean	PGD ℓ_∞	AA	Clean	PGD ℓ_∞	AA	Clean	PGD ℓ_∞	AA	Clean	PGD ℓ_∞	AA
SSL	RoCL (Kim et al., 2020)	76.76	50.72	45.52	51.91	27.77	22.79	60.44	31.90	27.38	35.00	8.11	5.67	17.21	2.55	1.71
	ACL (Jiang et al., 2020)	75.99	50.35	45.50	51.91	27.77	22.79	63.46	30.24	25.73	30.95	5.86	3.80	17.00	2.33	1.54
	Ours (3 steps)	74.26	49.38	44.31	50.23	27.05	21.96	53.46	32.65	28.96	31.47	9.58	6.19	18.07	4.49	2.73

H Robustness on Unseen Domains with Larger Datasets

To demonstrate the effectiveness of our adversarially transferable meta-trained model, we conduct further evaluations in a standard transfer learning scenario where the encoder, along with its linear layer, is fully trained using the entire dataset. The goal is to assess the generalizable robustness of the learned representations against a self-supervised adversarial learning model trained on a large amount of data. Our evaluations cover both the seen domain, CIFAR-100, and two unseen domains, CIFAR-10 and STL-10. Additionally, we showcase the robust transferability of our models on few-shot image classification benchmark datasets, namely Cars, CUB, and Aircraft. In this case, these datasets are treated as standard image classification tasks with 196, 200, and 100 classes respectively, rather than few-shot image classification tasks like n-way k-shot classification. For these evaluations, we train our models using ResNet18 with latent attacks employing 3 steps, while other self-supervised models are trained with PGD-7 attacks due to computational constraints. The validation process employs the same set of hyperparameters for robust full-finetuning across all datasets, and detailed information about the experimental settings is provided in the following section.

H.1 Baselines for self-supervised adversarial learning approaches

We select baseline models with ACL (Jiang et al., 2020)[4], BYORL (Gowal et al., 2020) and RoCL (Kim et al., 2020)[5] for self-supervised pre-trained baselines. We implement BYORL on top of the BYOL (Grill et al., 2020)[6] framework, following the description in the paper.

H.2 Self-supervised robust linear evaluation

To compare MAVRL with self-supervised pre-trained models, we apply robust full-finetuning, which is the representative evaluation method for demonstrating the quality of the learned representations in self-supervised learning fields. In robust full-finetuning, the parameters of the entire network, including the encoder and the classifier, are trained with adversarial examples. We generate perturbed examples with l_∞ PGD-10 attack with $\epsilon = 8./255.$ and step size $\alpha = 2./255.$ in training. All adversarially full-finetuned models are evaluated against l_∞ PGD-20 attack ($\epsilon = 8./255.$, $\alpha = 8./2550.$) and AutoAttack (Croce & Hein, 2020). Especially, in comparisons with self-supervised models, we pre-train ResNet18 based on FOMAML (Finn et al., 2017), which is the first-order approximation of MAML (Finn et al., 2017), and apply multi-view latent attacks with 3 steps to reduce the computational cost. Other self-supervised models are pre-trained with PGD-7 attacks.

[4] https://github.com/VITA-Group/Adversarial-Contrastive-Learning
[5] https://github.com/Kim-Minseon/RoCLforself-supervisedlearning
[6] https://github.com/lucidrains/byol-pytorch

For optimization, we fine-tune the pre-trained models for 110 epochs with batch size 128 under SGD optimizer with weight decay 5e-4, where Pang et al. (2022) demonstrated as optimal for robust full-finetuning on CIFAR datasets.

H.3 Robustness on unseen domain standard image classification tasks

Although our models utilize only scarce data to train and even apply latent attacks with fewer gradient steps, we show comparable clean and robust accuracy compared to self-supervised pre-trained models which are trained with larger data and stronger attacks with more steps of inner maximization (Table 13). Especially, our methods show a larger gap in robustness on fine-grained datasets (i.e., CUB, Cars), which have highly different distributions from meta-trained domains (i.e., CIFAR-FS). Further, we hope that our models to be robust in real-world adversarial perturbation such as common corruption (Hendrycks & Dietterich, 2019), we evaluate our fully finetuned models with adversarial examples on CIFAR-10, with common corruption datasets on CIFAR-10.

MAVRL also shows comparable accuracy with self-supervised pre-trained models on common corruption tasks (Table 14). From these results, we prove that MAVRL learns good generalized representations with little data effectively. Thus, the experimental results may imply that MAVRL can be used as a means of pretraining the representations to ensure robustness for a variety of applications when the training data is scarce.

Table 14: Test accuracy(%) on common corruption tasks of CIFAR-10-C. All models are adversarially trained on ResNet18, and finetuned on CIFAR-10.

Learning Type	Model	Accuracy
Self-supervised adversarial learning	ACL (Jiang et al., 2020)	68.60
	ROCL (Kim et al., 2020)	66.16
Meta-adversarial learning	MAVRL	67.90