MORGAN: Meta-Learning-based Few-Shot Open-Set Recognition via Generative Adversarial Network
Supplementary Material

Debabrata Pal[1, 2], Shirsha Bose[3], Biplab Banerjee[1], Yogananda Jeppu[2]
[1]Indian Institute of Technology, Bombay, [2]Honeywell Technology Solutions, India,
[3]Technical University of Munich, Germany
{deba.iitbcsre19, shirshabosecs, getbiplab, yvjeppu}@gmail.com

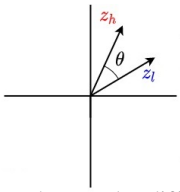
1. Notation table and associated descriptions

Table 1. Notation Table

Notations	Descriptions
Input data and processed features	
H, W, B	Height, width and spectral bands of input 3D HSI patches
$x^s, y^s; \mathcal{S}, \mathcal{S}_f$	Support sample and label, Support set and its features
$\mathcal{S}^k$	Support set of k^{th} *known* class
$x^q, y^q; \mathcal{Q}, \mathcal{Q}_f$	Query sample and label, Query set and its features
$\mathcal{Q}_{aug}$	Unknown query features with *pseudo-unknown* samples
$\mathcal{Q}_{dist}$	Euclidean distance of each query from the prototypes
$\mathcal{K}, \mathcal{U}$	Number of *known* and *pseudo-unknown* classes in an episode
m, N	Number of support, queries per class for training in an episode
s_l, s_h	*Pseudo-known*, *unknown* samples generated in adversarial optimization
$\mathcal{P}, \mathcal{P}_k$	*Known* prototype set, prototype for single *known* class
$\mathcal{P}_y, \mathcal{P}_{\mathcal{U}}$	True prototype of a query, prototype for open-space
z_l, z_h	Noise vectors for generating *pseudo-known*, *unknown* samples
$\mathcal{N}(0, \sigma)$	General isotropic Gaussian with mean=0, standard deviation σ
$\sigma L, \sigma H$	Standard deviations for generating *pseudo-known*, *unknown* samples
Network components and associated hyperparameters	
$f_\varphi, \mathcal{O}_\xi$	Feature extractor, outlier detector
$G_{\mathcal{H}\theta}, D_{\mathcal{H}\phi}$	*Pseudo-unknown* sample generator and discriminator
$G_{\widehat{\mathcal{H}\theta}}, D_{\widehat{\mathcal{H}\phi}}$	Clone copy of $G_{\mathcal{H}\theta}, D_{\mathcal{H}\phi}$
$\nabla GH, \nabla DH$	Gradients for *pseudo-unknown* sample generator and discriminator
$G_{\mathcal{L}\theta}, D_{\mathcal{L}\phi}$,	*Pseudo-known* sample generator and discriminator
$G_{\widehat{\mathcal{L}\theta}}, D_{\widehat{\mathcal{L}\phi}}$	Clone copy of $G_{\mathcal{L}\theta}, D_{\mathcal{L}\phi}$
$\nabla GL, \nabla DL$	Gradients for *pseudo-known* sample generator and discriminator
$\beta L, \beta H$	Learning rates for updating original GAN networks
$\alpha L, \alpha H$	Learning rates for updating cloned GAN networks
$\mathcal{I}_o, \mathcal{I}_i$	Iterations to update cloned GAN network parameters
τ	Range of sampled episodes
Distance, loss functions and regularizer	
d	Euclidean distance
$\mathcal{L}_l, \mathcal{L}_h$	Adversarial losses for generating *pseudo-known* and *unknown* samples
$\mathcal{L}_{Kc}$	Compaction loss to increase *known* class data density
$\mathcal{L}_{Os}$	Outlier scattering loss to increase separation from closed-boundary
γ	Hyperparameter (positive) to weight the distance of a outlier from $\mathcal{P}_y$
$\mathcal{L}_{Oc}$	Outlier calibration loss for outlier detector
$\mathcal{L}_{FE}$	Loss for feature extractor
λ_{AOL}	Anti-overlap latent regularizer
ϵ	Small positive constant to ensure $\lambda_{AOL} >= 0$

2. Analysis of AOL regularizer

In MORGAN, two noise vectors z_l and z_h are sampled from isotropic gaussian distribution with varying standard deviations, $[\mathcal{N}(0, \sigma L)]$ and $[\mathcal{N}(0, \sigma H)]$, respectively, where $\sigma L < \sigma H$. The angular difference (θ) or similarity between z_l and z_h impacts the metric distance between the generated pseudo *known* s_l and *unknown* s_h samples. Mathematically, θ controls the range of the λ_{AOL} regularizer to penalize *pseudo-unknown* generator $G_{\mathcal{H}\theta}$ in producing non-overlapping pseudo outliers and is shown in Fig 1.



θ	λ_{AOL}
0°	$2 \times max(\epsilon, cos(s_l, s_h))$
90°	$1 \times max(\epsilon, cos(s_l, s_h))$
180°	0

Figure 1. The angular difference between two noise vectors, z_l and z_h, determines the Anti-overlap regularizer's value

We define the AOL regularization term in (1) and explain the following scenarios based on the value of θ.

$$\lambda_{AOL} = (1 + cos(z_l, z_h)).max(\epsilon, cos(s_l, s_h)) \quad (1)$$

Where $cos(i, j) = \frac{i.j}{\|i\|_2 \|j\|_2}$ represents cosine similarity between vectors i, j and ϵ is small positive constant (0.00001). ϵ helps to prevent λ_{AOL} to become negative.

Case-1 $\{\theta = 0^\circ\}$: The role of the λ_{AOL} regularizer is to penalize the *pseudo-unknown* generative network ($G_{\mathcal{H}\theta}$) heavily when the feature representation of s_h becomes equivalent to that of s_l produced by the *pseudo-known* generative network $G_{\mathcal{L}\theta}$. If the cosine similarity of two noise vectors, z_l and z_h is very high, the generated adversarial samples can have identical feature representation or reside very closely in the metric space. Then, the sole purpose of using a $G_{\mathcal{H}\theta}$ to generate s_h is lost. Due to high similarity between z_l and z_h, $cos(z_l, z_h)$ becomes one and it makes λ_{AOL} to have a value of $2 \times max(\epsilon, cos(s_l, s_h)$. Further, if the generated samples are highly similar, λ_{AOL} becomes two as $cos(s_l, s_h)$ is one. This value is added with the loss objective, $\mathcal{L}_h$, to penalize the $G_{\mathcal{H}\theta}$.

Case-2 $\{\theta = 90^\circ\}$: Similarly, for low similarity between z_l and z_h or angle between z_l, z_h is 90°, $cos(z_l, z_h)$ becomes zero and it makes λ_{AOL} to have a value of $1 \times max(\epsilon, cos(s_l, s_h)$. However, If the generated samples are still highly similar, λ_{AOL} becomes one as $cos(s_l, s_h)$ is one. It is added with the loss objective, $\mathcal{L}_h$, to penalize the $G_{\mathcal{H}\theta}$.

Case-2 $\{\theta = 180^\circ\}$: In case z_l and z_h are sampled from

Table 2. 5-shot FSOSR performance comparison using the proposed outlier scattering loss and the reciprocal loss for four HSI datasets

Reciprocal loss	Indian Pines			Pavia University			Salinas			Houston-2013		
	ClosedOA	OpenOA	AUROC	ClosedOA	OpenOA	AUROC	ClosedOA	OpenOA	AUROC	ClosedOA	OpenOA	AUROC
Reciprocal Loss [2]	93.21±0.14	82.49±0.08	81.59±0.12	80.54±0.16	80.81±0.21	82.62±0.09	84.33±0.17	85.50±0.14	89.52±0.12	81.71±0.22	83.55±0.15	85.49±0.12
Scattering Loss (Ours)	**95.09±0.18**	**90.43±0.24**	**95.59±0.12**	**81.11±0.19**	**92.18±0.18**	**92.98±0.14**	**86.64±0.09**	**93.70±0.11**	**94.99±0.11**	**83.64±0.21**	**92.18±0.14**	**95.17±0.16**

quite separated data points in the gaussian distribution, or ($\sigma H - \sigma L$) is relatively high, θ can be considered as $180°$. Intuitively, it creates the feature representations of s_l and s_h completely different. Also, λ_{AOL} becomes zero as $cos(z_l, z_h)$ is -1. No further penalization happened on *pseudo-unknown* generator, optimized for $\mathcal{L}_h$.

Case-3 $\{\theta = others\}$: For any other angle between the 0, and 180 degrees, the value of the λ_{AOL} regularizer is determined as per (1) and is used to penalize the *pseudo-unknown* generative network.

Thus the angle between the z_l and z_h vectors determines how much the λ_{AOL} regularizer will penalize the *pseudo-unknown* generator for generating adversarial outliers so that they can be dissimilar to the *pseudo-known* samples.

3. Analysis on Outlier Scattering Loss

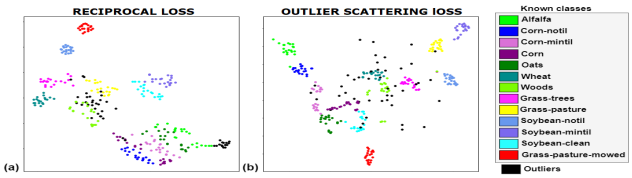


Figure 2. t-SNE visualization of the metric space due to optimization using (a) Reciprocal Points classification loss [2] (b) our Outlier Scattering Loss over Indian Pines dataset

Our Oulier Scattering Loss is philosophically similar to the reciprocal points classification loss [3], where the reciprocal relation ensures that the higher the distance of an unknown query $\in \mathcal{Q}_{aug}$ from a *known* class true prototype $\mathcal{P}_y$, the lesser the probability of that query belonging to that *known* class. The reciprocal points [2] are external learnable parameters representing the extra-class space information of each *known* class. Minimizing the reciprocal points classification loss, a query is classified as *known* or *unknown* based on the otherness with reciprocal points. During reciprocal points optimization, the *known* classes are spread towards the feature space periphery, limiting the *unknown* samples in a bounded open space. However, in multi-class classification, reciprocal points repel the outliers and creates a fixed cluster in open-space closer to the closed-set distribution reducing separation margin and lag transferable knowledge over the meta-learning episodes. Hence, unlike [2], we pull these outliers towards an open-space prototype to learn transferable knowledge over the episodes and maximize the separation margin from closed-set. Also, reciprocal points do not help feature extractor to extract suitable support and query features in metric learning. In contrast, we compute the Outlier Scattering Loss on the set of *known* class prototypes $\mathcal{P}$ and the prototype representing the open space, $\mathcal{P}_\mathcal{U}$. When multiple *known* class and open set samples come in interaction, the fine-grained outlies are scattered in open space and pulled towards $\mathcal{P}_\mathcal{U}$, increasing the margin from the closed boundaries. Combined with compaction loss and outlier calibration loss, the Outlier Scattering Loss is also used to optimize the feature extractor to generate relevant support query features in metric learning.

In Fig. 2, when we use the traditional reciprocal points in our methodology, we see that *known-unknown* samples remain unscattered, forming a cluster like *known* classes and residing in very close vicinity of one of the *known* classes in metric space. It hampers the outlier detection capability and produces reduced OpenOA and AUROC values. Nevertheless, when we use prototype-based scattering loss, the *known-unknown* and pseudo-unknown data points are scattered uniformly over the metric space and eventually help in better outlier detection, thus boosting OpenOA and AUROC. Table 2 shows that our prototype-based Outlier Scattering Loss implementation gives a higher OpenOA and AUROC than the reciprocal loss. We gain 7.49%, 11.37%, 8.2%, 8.63% OpenOA, and 14%, 10.36%, 5.47%, 9.68% AUROC over the Indian Pines, University of Pavia, Salinas, and Houston-2013 datasets, respectively.

4. Experimental results on mini-imageNet

Even though MORGAN applies to generic image datasets, we evaluated it on the hyperspectral imaging (HSI) datasets, given their enormous real-life applications, costly annotations, and absence of a land-cover map for the entire globe. Nonetheless, the table below shows that MORGAN beats the existing methods convincingly on mini-Imagenet.

Table 3. 5-shot comparison on miniImageNet dataset

Model	mini-Imagenet (5-way)			
	1-shot		5-shot	
	ClosedOA	AUROC	ClosedOa	AUROC
OpenMax [1]	63.69	62.64	80.56	62.2
PEELER [6]	65.86	60.57	80.61	67.35
SnaTCHer [5]	67.60	69.40	82.36	76.15
OCN [8]	66.89	69.73	82.33	74.97
MORGAN [ours]	**69.22**	**72.65**	**84.87**	**79.61**

5. Consequence of noise variations

Case-1 $\sigma L < \sigma H$: In the presence of moderate standard deviation $\sigma = \sigma H - \sigma L$ of an in isotropic Gaussian distribution in Fig. 3a, pseudo-fine-grained outliers are generated surrounding the closed boundary. Hence, we obtain good closedOA, openOA and AUROC results.

Case-2 $\sigma L \approx \sigma H$: Many *pseudo-known* samples erroneously get recognized as outliers due to very low marginal seperation $\sigma = \sigma H - \sigma L$ for *pseudo-unknown* sample generation. It causes a reduced closedOA value. (Fig. 3b)

Table 4. 1, 5, 10, 15-shot FSOSR performance comparison of the proposed MORGAN over four benchmark hyperspectral datasets

Shot	Indian Pines			Pavia University			Salinas			Houston-2013		
	ClosedOA	OpenOA	AUROC	ClosedOA	OpenOA	AUROC	ClosedOA	OpenOA	AUROC	ClosedOA	OpenOA	AUROC
1-shot	91.47±0.14	87.42±0.08	90.83±0.12	79.88±0.16	85.22±0.21	90.11±0.09	83.24±0.17	90.22±0.14	91.96±0.12	77.15±0.22	89.20±0.15	91.06±0.12
5-shot	95.09±0.18	90.43±0.24	95.59±0.12	81.11±0.19	92.18±0.18	92.98±0.14	86.64±0.09	93.70±0.11	94.99±0.11	83.64±0.21	92.18±0.14	95.17±0.16
10-shot	96.12±0.26	92.37±0.27	96.21±0.15	84.05±0.25	93.65±0.23	93.71±0.33	88.39±0.21	94.52±0.35	95.77±0.21	85.85±0.27	93.72±0.31	95.88±0.19
15-shot	**96.95±0.22**	**93.45±0.19**	**96.84±0.19**	**86.53±0.17**	**95.10±0.12**	**94.11±0.21**	**90.25±0.12**	**95.96±0.15**	**96.29±0.15**	**87.96±0.37**	**94.51±0.19**	**95.99±0.17**

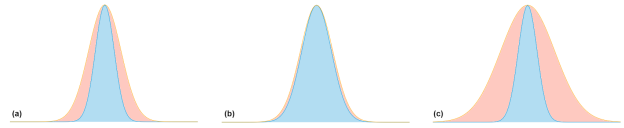


Figure 3. Effect of the noise variance on two generators responsible for generating *pseudo-known* and outlier samples, when (a) $\sigma L < \sigma H$ (b) $\sigma L \approx \sigma H$ (c) $\sigma L << \sigma H$. The blue region indicates the region for sampling a noise vector to generate a *pseudo-known* sample. Also, another noise vector is sampled from the surrounding orange zone to generate a *pseudo-unknown* sample.

Case-3 $\sigma L << \sigma H$: Finally in Fig. 3(c) due to $\sigma L << \sigma H$, pseudo-outliers mix with other *known* class samples, reducing OpenOA.

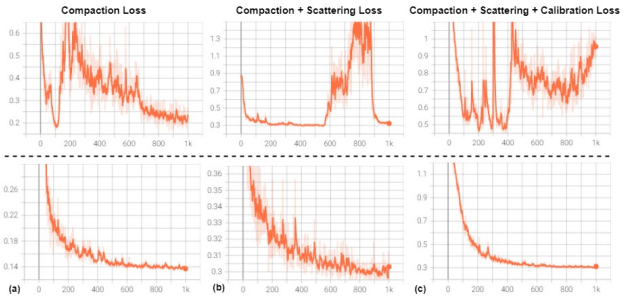


Figure 4. Feature extractor loss decay comparison for 5-shot FSOSR over Salinas dataset using MAML [4] (row1) and Reptile [7] (row2)-based pseudo feature generation, (a) only compaction loss decay b) optimization with compaction and scattering loss (c) three-fold loss optimization using compaction, scattering, and outlier calibration loss.

6. Influence of incrementing support set size

In Table 4, we have shown the results of MORGAN by gradually incrementing the support set size with a value of 5. We can see that the overall MORGAN performance for the three evaluation metrics increases over all the four HSI datasets due to higher support set information about *known* classes. It increases closed-set data density by extracting relevant *known* support features and directly helps in boosting closedOA. Also, due to compact abating closed bound generation, fine-grained *pseudo-unknowns* and *known-unknowns* become better discriminable. Thus the openOA and AUROC also increase. For the Indian Pines dataset, ClosedOA and AUROC values almost got saturated from 10-shot to 15-shot with very high performance $\approx 97\%$. Similarly, we observe approximately 1.5% gain on all evaluation metrics over other datasets for 10-shots to 15-shots evaluation.

7. Effect of different loss components based on inner-loop optimization algorithm

Fig. 4 represents the loss curves due to the different combinations of objective functions in training the MORGAN feature extractor. In MORGAN, two generative networks generate the *pseudo-known* and *unknown* samples internally in an episode as per Algorithm 2, and they are optimized by either using MAML [4] or Reptile [7]. However, we observe that the optimization choice between MAML or Reptile for adversarial sample generation and augmentation with support and query features to enhance the open-closed data density has a profound consequence in optimizing the feature extractor.

While we use only compaction loss to optimize the feature extractor, we see a sudden dip in the loss curve for the MAML based pseudo sample generation during the initial training epochs. Afterward, the loss increased and gradually converged to a higher value than the Reptile-based feature generation, which converged quite smoothly. The Reptile-based adversarial feature generation converged with slight fluctuations when we added scattering loss to the compaction loss. Whereas for MAML based generation, the loss suddenly increased sharply after 500 epochs, continues fluctuations, and then decays at 900 epochs. Finally, when we add the outlier loss to the compaction and scattering loss, we see a very smooth convergence with a minor fluctuation in Reptile based feature hallucination. Still, MAML continues suffering from highly oscillating loss decay.

Based on these observations, we can envisage that since the MAML is a second-order optimization technique, the model parameters fail to converge at a global optimum quickly and suffers from gradient explosion. The parameters mostly converge to saddle points in the loss plane, where slight perturbations to the model parameters cause fluctuations with high dip and rise. Sometimes, it fails to converge at all. On the other hand, Reptile based optimization helps update generative components quickly due to its first-order optimization. Interestingly, Reptile-produced pseudo-open samples lie closely in metric space to their *known* counterpart. Hence we observe smooth decay when augmenting pseudo features with support query features.

8. MORGAN model parameters

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 11, 11, 30, 1)]	0	[]
conv3d (Conv3D)	(None, 11, 11, 30, 8)	224	['input_1[0][0]']
conv3d_2 (Conv3D)	(None, 11, 11, 30, 8)	1736	['conv3d[0][0]']
cbam_3d_1 (CBAM_3D)	(None, 11, 11, 30, 8)	729	['conv3d_2[0][0]']
conv3d_4 (Conv3D)	(None, 11, 11, 30, 8)	1736	['cbam_3d_1[0][0]']
add (Add)	(None, 11, 11, 30, 8)	0	['conv3d[0][0]', 'conv3d_4[0][0]']
max_pooling3d (MaxPooling3D)	(None, 6, 6, 8, 8)	0	['add[0][0]']
conv3d_5 (Conv3D)	(None, 6, 6, 8, 16)	3472	['max_pooling3d[0][0]']
cbam_3d_2 (CBAM_3D)	(None, 6, 6, 8, 16)	835	['conv3d_5[0][0]']
conv3d_7 (Conv3D)	(None, 6, 6, 8, 16)	6928	['cbam_3d_2[0][0]']
cbam_3d_3 (CBAM_3D)	(None, 6, 6, 8, 16)	835	['conv3d_7[0][0]']
conv3d_9 (Conv3D)	(None, 6, 6, 8, 16)	6928	['cbam_3d_3[0][0]']
add_1 (Add)	(None, 6, 6, 8, 16)	0	['cbam_3d_2[0][0]', 'conv3d_9[0][0]']
max_pooling3d_1 (MaxPooling3D)	(None, 3, 3, 4, 16)	0	['add_1[0][0]']
cbam_3d_4 (CBAM_3D)	(None, 3, 3, 4, 16)	835	['max_pooling3d_1[0][0]']
conv3d_11 (Conv3D)	(None, 1, 1, 2, 32)	13856	['cbam_3d_4[0][0]']
flatten (Flatten)	(None, 64)	0	['conv3d_11[0][0]']

Total params: 38,114
Trainable params: 38,114
Non-trainable params: 0

Figure 5. The layer wise summary of the feature extractor f_φ following R3CBAM in OCN[8]

Model: "generator_high"

Layer (type)	Output Shape	Param #
input_4 (InputLayer)	[(None, 24)]	0
dense_16 (Dense)	(None, 32)	800
dense_17 (Dense)	(None, 48)	1584
dense_18 (Dense)	(None, 64)	3136

Total params: 5,520
Trainable params: 5,520
Non-trainable params: 0

Figure 6. The layer wise summary of $G_{\mathcal{H}\theta}$

Model: "discriminator_high"

Layer (type)	Output Shape	Param #
input_6 (InputLayer)	[(None, 80)]	0
dense_24 (Dense)	(None, 48)	3888
dense_25 (Dense)	(None, 32)	1568
dense_26 (Dense)	(None, 16)	528
dense_27 (Dense)	(None, 8)	136
dense_28 (Dense)	(None, 1)	9

Total params: 6,129
Trainable params: 6,129
Non-trainable params: 0

Figure 7. The layer wise summary of $D_{\mathcal{H}\phi}$

Model: "generator_low"

Layer (type)	Output Shape	Param #
input_3 (InputLayer)	[(None, 24)]	0
dense_13 (Dense)	(None, 32)	800
dense_14 (Dense)	(None, 48)	1584
dense_15 (Dense)	(None, 64)	3136

Total params: 5,520
Trainable params: 5,520
Non-trainable params: 0

Figure 8. The layer wise summary of $G_{\mathcal{L}\theta}$

Model: "discriminator_low"

Layer (type)	Output Shape	Param #
input_5 (InputLayer)	[(None, 80)]	0
dense_19 (Dense)	(None, 48)	3888
dense_20 (Dense)	(None, 32)	1568
dense_21 (Dense)	(None, 16)	528
dense_22 (Dense)	(None, 8)	136
dense_23 (Dense)	(None, 1)	9

Total params: 6,129
Trainable params: 6,129
Non-trainable params: 0

Figure 9. The layer wise summary of $D_{\mathcal{L}\phi}$

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 3)]	0
dense_10 (Dense)	(None, 16)	64
dense_11 (Dense)	(None, 8)	136
dense_12 (Dense)	(None, 2)	18

Total params: 218
Trainable params: 218
Non-trainable params: 0

Figure 10. The layer wise summary of $\mathcal{O}_\xi$

References

[1] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1563–1572, 2016.

[2] Guangyao Chen, Peixi Peng, Xiangqian Wang, and Yonghong Tian. Adversarial reciprocal points learning for open set recognition. *arXiv preprint arXiv:2103.00953*, 2021.

[3] Guangyao Chen, Limeng Qiao, Yemin Shi, Peixi Peng, Jia Li, Tiejun Huang, Shiliang Pu, and Yonghong Tian. Learning open set network with discriminative reciprocal points. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 507–522. Springer, 2020.

[4] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[5] Minki Jeong, Seokeon Choi, and Changick Kim. Few-shot open-set recognition by transformation consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12566–12575, 2021.

[6] Bo Liu, Hao Kang, Haoxiang Li, Gang Hua, and Nuno Vasconcelos. Few-shot open-set recognition using meta-learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2020.

[7] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[8] Debabrata Pal, Valay Bundele, Renuka Sharma, Biplab Banerjee, and Yogananda Jeppu. Few-shot open-set recognition of hyperspectral images with outlier calibration network. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3801–3810, January 2022.