1 Proposed Algorithm

Algorithm 1: DEGAA: Offline Training Paradigm

Data. $S = \cup_{i=1}^{n} S_i, T = \cup_{j=1}^{m} T_j, D = S \cup T$

Step 1: *Domain Embedding Extraction*

Input. Number of sampled domains N_t, Number of support points N_s and query points N_q

Initialize. Feature Extractor $G(\mathbf{x}; \psi)$ initialized with pre-trained ImageNet weights

for *t = 1 To N* **do**
- $D_t \leftarrow \textsc{Randomly Sample}(D, N_t)$; `// sample` N_t `domains`
- **for** *d in D_t* **do**
  - $S_d, S_q \leftarrow \textsc{Randomly Sample}(D_d, N_s), \textsc{Randomly Sample}(D_d, N_q)$;
- **end**
- $\hat{\mu}_{D_t} \leftarrow \textsc{Kme}(S_d, G(\mathbf{x}; \psi))$;
- $J_\psi \leftarrow \textsc{Prototypical Loss}(\hat{\mu}_{D_t}, S_t, G(\mathbf{x}; \psi))$; `// Following [1]`
- $\psi \leftarrow \textsc{Sgd}(J(t), \psi)$;

end

Output. $d_e \leftarrow \textsc{Kme}(D, G(\mathbf{x}; \psi))$;

Step 2: *Warm - Up*

Input. Domain Embedding d_e, source images and labels $(\mathbf{x}_j^{S_i}, y_j^{S_i})_{j=1}^{p_i} \in (S_i)_{i=1}^{n}$, training images $(\mathbf{x}_j^{\widehat{S}}, y_j^{\widehat{S}})_{j=1}^{n'} \in \widehat{S} \subset S$

Initialize. Feature Extractor $F(\mathbf{x}; \psi)$ initialized with pre-trained ImageNet weights

for *t = 1 To N'* **do**
- $J_\theta \leftarrow \textsc{Cross Entropy}(y_j^{\widehat{S}}, F(\textsc{Concat}(\mathbf{x}_j^{\widehat{S}}, d_e)))$; `// Supervised training`
- $\theta \leftarrow \text{SGD}(J(t), \theta)$;

end

Step 3: *Compute Centroids*

for *i = 1 To n* **do**
- **for** *j = 1 To p_i* **do**
  - $\mathcal{T} \leftarrow F(\textsc{Concat}(\mathbf{x}_j^{S_i}, d_e); \theta)$;
- **end**

end

$\mathcal{C} \leftarrow \textsc{Centroids}(\mathcal{T})$; `// Per - class centroid from source feature maps`

Step 4: *Pseudo - Labelling and Adaptation Stage*

Input. Domain Embedding d_e, trained backbone $F(\mathbf{x}; \theta)$, number of episodes per batch K, batch for K episodes $(\mathbf{x}_i^{S_K}, y_i^{S_K})_{i=1}^{n'} \in S_K \subset S, (\mathbf{x}_j^{T_K})_{j=1}^{m'} \in T_K \subset T$, target loss weight λ

for *t = 1 To N"* **do**
- $S_K, T_K \leftarrow \textsc{Randomly Sample}(S, n'), \textsc{Randomly Sample}(T, m')$;
- **for** *k = 1 To K* **do**
  - **for** *j = 1 To m'* **do**
    - $D'_t \leftarrow F(\textsc{Concat}(\mathbf{x}_j^{T_K}, d_e); \theta)$; `// Concatenated feature maps(target)`
  - **end**
  - $D'_k, D'_u \leftarrow \textsc{Lof}(D'_t)$; `// Known classes` D'_k `and unknown` D'_u
  - $Y_{pseudo} \leftarrow \textsc{Knn}(D'_k, \mathcal{C})$; `// Assign nearest centroid class`
  - **for** *i = 1 To n'* **do**
    - $D'_s \leftarrow F(\textsc{Concat}(\mathbf{x}_i^{S_K}, d_e); \theta)$; `// Concatenated feature maps(source)`
    - $Y_s \leftarrow y_i^{S_K}$;
  - **end**
  - $\widehat{Y}_s, \widehat{Y}_t \leftarrow \textsc{Softmax}(\textsc{Gaa}(D'_s, D'_k))$; `// Source and target labels using GAA`
  - $J_\theta \leftarrow \textsc{Cross Entropy}(\widehat{Y}_s, Y_s) + \lambda \textsc{Cross Entropy}(\widehat{Y}_t, Y_{pseudo})$;
  - $\theta \leftarrow \text{SGD}(J(t), \theta)$;
- **end**

end

2 Dataset Details

- **Office31:** The Office31 dataset [2] contains 31 object categories in three domains: Amazon, DSLR and Webcam. The object categories include everyday objects such as keyboards, laptops and file cabinets. Amazon Domain contains 2817 images captured against clean background and at a unified scale. The DSLR domain contains 498 high resolutions images while the WebCam contains 795 low resolutions images.
- **OfficeHome:** The OfficeHome dataset [3] contains images in 4 different domains consisting of 65 object categories found typically in Office and Home settings. Total 15,500 images are present with images in each class vary between 70 images to a maximum of 99 images.
- **VisDA-2017:** This large scale dataset [4] contains over 280,000 images across 12 categories. The training images are generated from the same object under different conditions while the validations images are sourced from MSCOCO.
- **DomainNet:** The DomainNet dataset [5] contains over half a million images in 6 different domains, each consisting of 345 categories of objects. The domains include clipart, real world photos, sketches, infograph, QuickDraw and paintings

3 Tables

Following previous works **OS** indicates normalized accuracy for all the classes including the unknown as one class and **OS*** shows normalized accuracy only on known classes.

Table 1: Classification Accuracy (%) of open set domain adaptation tasks on Office-31 (ResNet-50)

Method	**A→W**		**D→W**		**W→D**		**A→D**		**D→A**		**W→A**		**Avg**	
	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*
ResNet [6]	82.5	82.7	85.2	85.5	94.1	94.3	96.6	97.0	71.6	71.5	75.5	75.2	84.2	84.4
ATI-λ [7]	87.4	88.9	84.3	86.6	93.6	95.3	96.5	98.7	78.0	79.6	80.4	81.4	86.7	88.4
OSBP [8]	86.5	87.6	88.6	89.2	97.0	96.5	97.9	98.7	88.9	90.6	85.8	84.9	90.8	91.3
STA [9]	89.5	92.1	93.7	96.1	97.5	96.5	99.5	99.6	89.1	93.5	87.9	87.4	92.9	94.1
JPOT [10]	92.8	92.2	95.2	96.0	98.1	96.2	99.5	98.6	93.0	94.1	88.9	88.4	94.6	94.3
Ours	-													

Table 2: Classification accuracy (%) of open set domain adaptation tasks on Office-Home (ResNet-50)

Method	**Ar→Cl**	**Pr→Cl**	**Rw→Cl**	**Ar→Pr**	**Cl→Pr**	**Rw→Pr**	**Cl→Ar**	**Pr→Ar**	**Rw→Ar**	**Ar→Rw**	**Cl→Rw**	**Pr→Rw**	**Avg.**
ResNet [6]	53.4	52.7	51.9	69.3	61.8	74.1	61.4	64.0	70.0	78.7	71.0	74.9	65.3
ATI-λ [7]	55.2	52.6	53.5	69.1	63.5	74.1	61.7	64.5	70.7	79.2	72.9	75.8	66.1
OSBP [8]	56.7	51.5	49.2	67.5	65.5	74.0	62.5	64.8	69.3	80.6	74.7	71.5	65.7
STA [9]	58.1	53.1	54.4	71.6	69.3	81.9	63.4	65.2	74.9	85.0	75.8	80.8	69.5
JPOT [10]	59.6	54.2	54.6	72.3	70.1	82.1	62.9	68.3	75.1	84.8	77.4	81.2	70.2
PGL [11]	61.6	58.4	65.0	77.1	72.0	83.0	68.8	72.2	78.6	85.9	82.8	82.6	74.0
Ours	-												

Table 3: Classification accuracy (%) of open set domain adaptation tasks on VisDA-2017 (VGGNet)

Method	Bic	Bus	Car	Mot	Tra	Tru	UNK	OS	OS*
AATI-λ [7]	46.2	57.5	56.9	79.1	81.6	32.7	65.0	59.9	59.0
OSBP [8]	51.1	67.1	42.8	84.2	81.8	28.0	85.1	62.9	59.2
STA [9]	52.4	69.6	59.9	87.8	86.5	27.2	84.1	66.8	63.9
PGL [11]	93.5	93.8	75.7	98.8	96.2	38.5	68.6	80.7	82.8
Ours	-								

Table 4: Classification accuracy (%) of Multi Source Open Set domain adaptation tasks on Office-31.

Method	**AD→W**		**AW→D**		**WD→A**		**Avg**	
	OS	OS*	OS	OS*	OS	OS*	OS	OS*
MOSDANET [12]	99.0	98.2	99.4	98.3	81.0	79.3	93.1	91.9
Ours				-				

Table 5: Comparison with the state-of-the-art methods on the DomainNet dataset.

Method	R→S	R→C	R→I	R→P	P→S	P→R	P→C	P→I	**Avg**
CGCT [13]	48.9	60.3	26.9	57.1	43.4	58.8	48.5	21.7	45.7
D-CGCT [13]	48.4	59.6	25.3	55.6	45.3	58.2	51.0	21.7	45.6
DCC [14]	43.1	-	-	50.25	43.66	56.90	-	-	48.5
Ours					-				

References

[1] Abhimanyu Dubey, Vignesh Ramanathan, Alex Pentland, and Dhruv Mahajan. Adaptive methods for real-world domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14340–14349, 2021.

[2] Piotr Koniusz, Yusuf Tas, and Fatih Porikli. Domain adaptation by mixture of alignments of second- or higher-order scatter tensors. *CoRR*, abs/1611.08195, 2016.

[3] Mohammad Mahfujur Rahman, Clinton Fookes, Mahsa Baktashmotlagh, and Sridha Sridharan. Multi-component image translation for deep domain generalization. *CoRR*, abs/1812.08974, 2018.

[4] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Chi Su, Qingming Huang, and Qi Tian. Gradually vanishing bridge for adversarial domain adaptation. *CoRR*, abs/2003.13183, 2020.

[5] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? *CoRR*, abs/2008.11687, 2020.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[8] Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 153–168, 2018.

[9] Hong Liu, Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Qiang Yang. Separate to adapt: Open set domain adaptation via progressive separation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[10] Renjun Xu, Pelen Liu, Yin Zhang, Fang Cai, Jindong Wang, Shuoying Liang, Heting Ying, and Jianwei Yin. Joint partial optimal transport for open set domain adaptation. In *IJCAI*, pages 2540–2546, 2020.

[11] Yadan Luo, Zijian Wang, Zi Huang, and Mahsa Baktashmotlagh. Progressive graph learning for open-set domain adaptation. In *International Conference on Machine Learning*, pages 6468–6478. PMLR, 2020.

[12] S. Rakshit, Dipesh Tamboli, Pragati Shuddhodhan Meshram, Biplab Banerjee, G. Roig, and S. Chaudhuri. Multi-source open-set deep adversarial domain adaptation. In *ECCV*, 2020.

[13] Subhankar Roy, Evgeny Krivosheev, Zhun Zhong, Nicu Sebe, and Elisa Ricci. Curriculum graph co-teaching for multi-target domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5351–5360, 2021.

[14] Guangrui Li, Guoliang Kang, Yi Zhu, Yunchao Wei, and Yi Yang. Domain consensus clustering for universal domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9757–9766, June 2021.