

Supplementary material: “Mode-Guided Feature Augmentation for Domain Generalization”

Muhammad Haris Khan¹
muhammad.haris@mbzuai.ac.ae

Talha Zaidi²
tzaidi@ksu.edu

Salman Khan¹
salman.khan@mbzuai.ac.ae

Fahad Shehbaz Khan¹
fahad.khan@mbzuai.ac.ae

¹ Computer Vision Department
Mohamed Bin Zayed University of
Artificial Intelligence
Abu Dhabi, UAE

² Computer Science Department
Kansas State University
Kansas, USA

Results on Office-Home dataset. Table 1 summarizes the results for Office-Home dataset [11] using ResNet-18. Proposed method (MGFA) provides a gain of 1.08% over the baseline model in terms of overall(average) accuracy. In comparison to other methods, MGFA shows the second-best performance in overall accuracy with a minor difference in accuracy of 0.3% with the best performing method (CrossGrad). MGFA records the best recognition accuracy of 74.3% in A,C,R \rightarrow P shift. Moreover, MGFA delivers the second-best performance in all other domain shifts.

OH	Art (A)	ClipArt (C)	Product (P)	Real (R)	Average
MMD-AAE[10]	56.5	47.3	72.1	74.8	62.7
JiGen[10]	52.15	45.86	70.86	73.15	60.51
CrossGrad[10]	58.4	49.4	73.9	75.8	64.4
AGG(Baseline)	57.26	48.01	72.7	74.12	63.02
Ours	58.09	48.63	74.30	75.41	64.10

Table 1: Domain generalization results on Office-Home (OH) dataset [11] with recognition accuracy (%) using ResNet-18 [10] architecture. Numbers in red and blue denote the best and the second-best performance, respectively.

t-SNE visualizations. We visualize the deep features using t-SNE algorithm when a model is trained in single source settings (Fig. 1 & Fig. 2). In both figures, the left column displays feature embedding extracted from the baseline, and the right column visualizes feature embedding extracted from our method (MGFA). In Fig. 1 we can see that, in MGFA, the deep features from all classes make compact and concentrated clusters. However, in baseline model, the deep features from at least three different classes form clusters that have intersecting segments.

Fig. 3 visualizes deep features using t-SNE algorithm [10] when a model is trained in multi-source settings (art-painting, photo, sketch). The left column displays feature embedding extracted from the baseline, and the right column visualizes feature embedding extracted from our method (MGFA). In case of MGFA, we notice that the deep features from

different classes make compact clusters. So, they are possibly more separable from each other and in turn could facilitate the learning of a more generalizable classifier across domains. On the contrary, in baseline, features from different classes form clusters with many overlapping segments.

Fig. 4 visualizes the same deep features as in Fig. 3 but from the perspective of domain labels. Each color denotes a particular domain in PACS dataset [9]. We can observe that, compared to baseline, in MGFA, the deep features from different domains are better overlapped with each other. This shows that the proposed method attempts to remove domain differences by potentially bridging the domain gap through performing mode-guided feature augmentations.

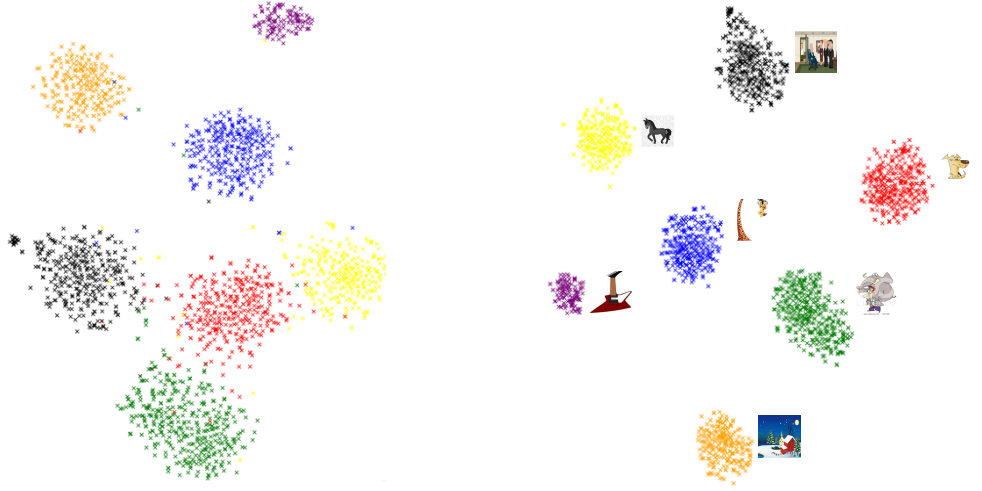


Figure 1: Visualization of deep features using t-SNE algorithm [9] when trained in single source settings (cartoon). Each color denotes a particular class in PACS dataset [9]. Left: feature embedding extracted from the baseline. Right: feature embedding extracted from our method (MGFA).

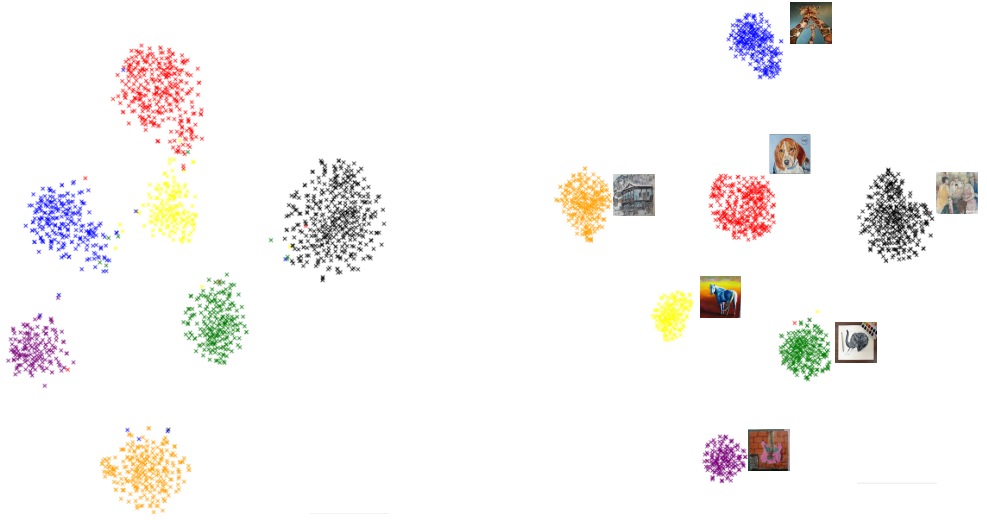


Figure 2: Visualization of deep features using t-SNE algorithm [6] when trained in single source settings (art-painting). Each color denotes a particular class in PACS dataset [4]. Left: feature embedding extracted from the baseline. Right: feature embedding from MGFA.

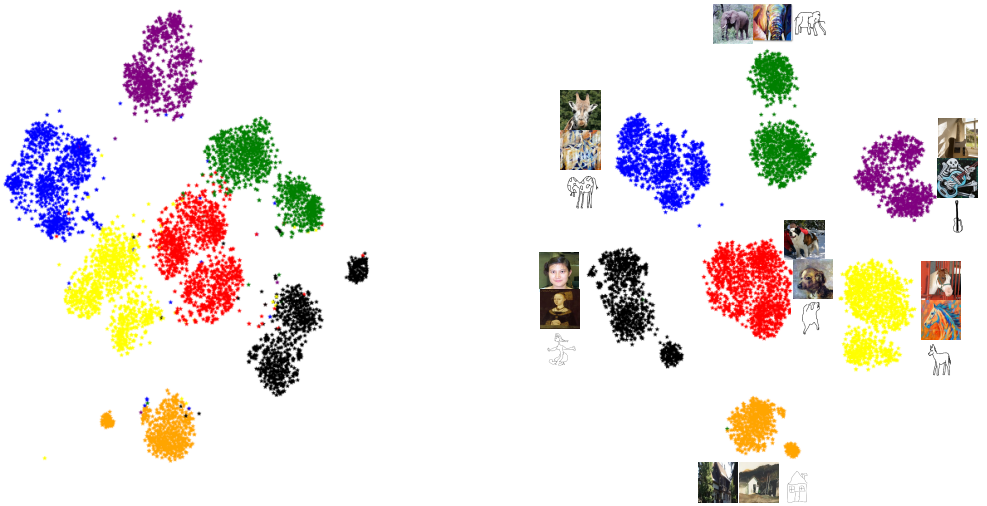


Figure 3: Visualization of deep features using t-SNE algorithm [6] when trained in multi-source settings (art-painting, photo, sketch). Each color denotes a particular class in PACS dataset [4]. Left: feature embedding extracted from the baseline. Right: feature embedding extracted from our method (MGFA).

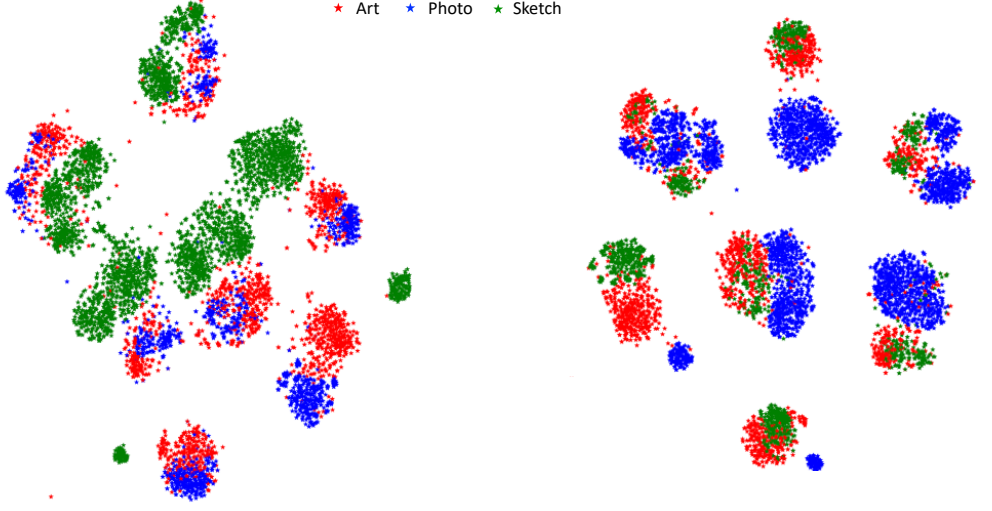


Figure 4: Visualization of deep features using t-SNE algorithm [9] when trained in multi-source settings (art-painting, photo, sketch). Each color denotes a particular domain in PACS dataset [4]. Left: feature embedding extracted from the baseline. Right: feature embeddings from MGFA).

Proof of proposition 1. In the following, we provide proof for proposition 1 stated in the paper.

Proposition 1 Suppose that $\hat{u}_i \sim \mathcal{N}(u_i, \gamma \Sigma_{y_i})$. Then we have an upper bound of \mathcal{L}_∞ , given by [14]:

$$\mathcal{L}_\infty \leq \frac{1}{N} \sum_{i=1}^N -\log\left(\frac{e^{w_{y_i}^T u_i + b_{y_i}}}{\sum_{j=1}^C e^{w_j^T u_i + b_j + \frac{\gamma}{2} z_{jy_i}^T \Sigma_{y_i} z_{jy_i}}}\right) \triangleq \tilde{\mathcal{L}}_\infty, \quad (1)$$

where $z_{jy_i} = w_j - w_{y_i}$.

Proof. Based on the definition of \mathcal{L}_∞ in Eq. 1, we can have:

$$\mathcal{L}_\infty = \frac{1}{N} \sum_{i=1}^N \mathbb{E}_{\hat{u}_i} [\log(\sum_{j=1}^C e^{z_{jy_i}^T \hat{u}_i + (b_j - b_{y_i})})] \quad (2)$$

$$\leq \frac{1}{N} \sum_{i=1}^N \log(\sum_{j=1}^C \mathbb{E}_{\hat{u}_i} [e^{z_{jy_i}^T \hat{u}_i + (b_j - b_{y_i})}]) \quad (3)$$

$$= \frac{1}{N} \sum_{i=1}^N \log(\sum_{j=1}^C e^{z_{jy_i}^T u_i + (b_j - b_{y_i}) + \frac{\gamma}{2} z_{jy_i}^T \Sigma_{y_i} z_{jy_i}}) \quad (4)$$

$$= \tilde{\mathcal{L}}_\infty \quad (5)$$

In the above proof, the Inequality (3) follows from Jensen’s inequality $\mathbb{E}[\log X] \leq \log \mathbb{E}[X]$, since the logarithmic function $\log(\cdot)$ is concave. Further, the Eq. 4 is obtained by using the moment-generating function:

$$\mathbb{E}[e^{tX}] = e^{t\mu + \frac{1}{2}\sigma^2 t^2}, X \sim \mathcal{N}(\mu, \sigma^2), \quad (6)$$

this is because $z_{jy_i}^T \hat{u}_i + (b_j - b_{y_i})$ is a Gaussian random variable, i.e.,

$$z_{jy_i}^T \hat{u}_i + (b_j - b_{y_i}) \sim \mathcal{N}(z_{jy_i}^T u_i + (b_j - b_{y_i}), \gamma z_{jy_i}^T \Sigma_{y_i} z_{jy_i}) \quad (7)$$

Reconstructing augmented versions. We brief the procedure to reconstruct the augmented versions into pixel space. Since there is no closed-form inverse function for CNNs like ResNets, we resort to an algorithm [10] that acts in a way similar to [9] and [8], which fixes the weights of network and instead optimizes the network inputs to search the images corresponding to deep features. As our method perturbs deep feature representation of an image to construct the augmented versions, we therefore find it more effective to add a fixed pre-trained generator (BigGAN[11] on ImageNet) to obtain images for classification model and optimize generator inputs. Specifically, the reconstruction procedure comprises of two stages.

In first stage, given we have a random variable l which is normalized with mean and standard deviation to get \tilde{l} and then passed through the generator \mathcal{G} to get $\mathcal{G}(\tilde{l})$. We sample an x_i from a domain of a dataset (such as Photos of PACS). Both $\mathcal{G}(\tilde{l})$ and x_i are passed through a pre-trained CNN (on Photo domain images of PACS dataset) to obtain respective deep feature vectors $f(\mathcal{G}(\tilde{l}))$ and u_i . Now, the first stage searches for l_i corresponding to x_i by jointly minimizing the squared L2 losses in pixel and deep feature spaces:

$$l_i = \arg \min_l \|f(\mathcal{G}(\tilde{l})) - u_i\|_2^2 + \eta \|\mathcal{G}(\tilde{l}) - x_i\|_2^2, \quad (8)$$

where η is a hyperparameter that adjusts the contribution of two losses.

In the second stage, we first produce the augmented version \hat{u}_i corresponding to u_i . Then we search for the corresponding \hat{l}_i using the starting point l_i found in stage 1 through minimizing $\hat{l} \|f(\mathcal{G}(\tilde{l})) - \hat{u}_i\|_2^2$.

Theoretical complexity. We now describe on how we obtain the theoretical computational complexity for showing that our method incurs little extra computational overhead during the training process. Recall that the proposed method builds on two major components: searching for meaningful examples and augmenting them efficiently. In case of a single example, the computational complexity of the first component is $\mathcal{O}(d^2)$ and the second one is $\mathcal{O}(d^2 C)$, where d is the dimensionality of the features produced by the penultimate layer. Compared to our single-source baseline (ResNet-18/50), which is a CNN with L layers, does $\mathcal{O}(d^2 K^2 HWL)$ number of operations, where K is the filter size, and H, W define the resolution of feature maps.

References

- [1] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.
- [2] Fabio M Carlucci, Antonio D’Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the*

- IEEE Conference on Computer Vision and Pattern Recognition*, pages 2229–2238, 2019.
- [3] 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.
 - [4] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5542–5550, 2017.
 - [5] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5400–5409, 2018.
 - [6] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.
 - [7] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015.
 - [8] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. *arXiv preprint arXiv:1804.10745*, 2018.
 - [9] Paul Upchurch, Jacob Gardner, Geoff Pleiss, Robert Pless, Noah Snaveley, Kavita Bala, and Kilian Weinberger. Deep feature interpolation for image content changes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7064–7073, 2017.
 - [10] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017.
 - [11] Yulin Wang, Xuran Pan, Shiji Song, Hong Zhang, Gao Huang, and Cheng Wu. Implicit semantic data augmentation for deep networks. In *Advances in Neural Information Processing Systems*, pages 12635–12644, 2019.