DOMAIN ADAPTATION VIA ACTIVATION SHAPING

https://github.com/Truvella99/Activation-Shaping-AML

Davide Giovanni Freni
s305571@studenti.polito.it

Domenico Gagliardo
s310454@studenti.polito.it

Gaetano Roberto
s318989@studenti.polito.it

Politecnico di Torino, Italy
DAUIN

Abstract

It is widely known that deep models do not behave consistently well with any data met at test time. Not surprisingly, the problem of domain shift represents one of the most investigated topics in computer vision field. In this work, to address this issue and enhance the model's robustness to domain shift, Activation Shaping (ASH) has been exploited. Activation Shaping is the process of modifying activation maps from a layer within an architecture using either some user-defined or learned rules. In particular, we evaluated how ASH can be applied to Unsupervised Domain Adaptation setting. Also random activation maps have been considered, and turning off some outputs in a random way has proven to be effective. Our experiments, utilizing the PACS dataset, show that Activation Shaping is highly helpful for detecting out-of-distribution (OOD) data, with the best results achieved when applying ASH to the middle/latest layers of the network. This approach offers a simple yet effective strategy for improving model reliability in diverse deployment scenarios.

1. Introduction

Despite their advanced capabilities, deep learning models often struggle with domain shift, where differences in data distributions (different illumination conditions, visual style, background, etc.) between training sets and real-world scenarios lead to decreased model performance.

This problem in the computer vision community is widely studied and addressed through Unsupervised Domain Adaptation (UDA) and Domain Generalization (DG).

UDA focuses on training a model that works well on the target distribution in settings with a labeled Source Domain (Training Set) and an unlabeled Target Domain (Test Set). In DG, the approach is extended to include multiple labeled Source Domains and a single Target Domain, used only at test time.

Note that current methods for detecting out-of-distribution (OOD) data typically require additional training, extra data, or significant changes to existing network structures. Our work, instead, adopts a novel strategy by simply integrating an Activation Shaping Module (ASM) (see Figure 1) into domain adaptation scenarios. As a matter of fact, ASH just consists in adjusting the activation maps from a layer within an architecture according to some user-defined or learned rules. In this way, it is possible to develop a model robust to domain shifts, taking into account both single and multiple Source Domains scenarios.

Our empirical findings demonstrate the efficacy of ASH, particularly when applied in the second half of our network (middle and last part of the network). As ASH is extended towards the network's initial layers, we document a notable reduction in accuracy, reaffirming the critical nature of strategic layer selection.

To rigorously test our approach, we employ the PACS dataset, an image classification dataset featuring seven different object categories (Dog, Elephant, Giraffe, Guitar, Horse, House, and Person) across four visual domains (Photos, Art Paintings, Cartoons, and Sketches). The variety of domains in this dataset is ideal for conducting our experiments, testing ASH's effectiveness in different scenarios.

2. Related Works

2.1. Extremely simple activation shaping for out-of-distribution detection

In 2023, the Activation Shaping method [3] emerged as an innovative approach for Out-of-Distribution (OOD) detection. By pruning and adjusting activations in a neural network's later layers, specifically within a ResNet-50 model, ASH enhances OOD detection without significantly impacting in-distribution accuracy. ASH operates on the

principle that neural networks often exhibit redundancy in their learned features. By sparsifying the feature representation through pruning (ASH-P), binary activation (ASH-B), or scaling (ASH-S), ASH helps the model focus on what's really important for good performance. Tested on well-known datasets like CIFAR and various OOD datasets, ASH has demonstrated state-of-the-art results.

2.2. Domain-adversarial training of neural networks

In 2016, a significant advancement in domain adaptation was introduced, focusing on models that effectively operate across various domains by learning features that cannot discriminate between the training (source) and test (target) domains. Useful for this purpose was the domain-adversarial neural network (DANN), which combines domain adaptation and deep feature learning within one training process [4]. The DANN, utilizing both standard layers and new gradient reversal layers, proved adaptable to various feed-forward models. This approach marked a notable achievement in tasks like sentiment analysis, image classification, and person re-identification, delivering state-of-the-art results on datasets such as MNIST. Its success underlined the importance of domain-invariant feature learning for effective domain adaptation, a concept that was groundbreaking in 2016 and has since influenced subsequent research in the field.

2.3. Deeper, broader and artier domain generalization

In the field of domain generalization, the focus has shifted towards creating models that are domain-agnostic and that can be applied to various unseen domains. In particular, there are domain adaptation methods, that use (un)labeled target data to adapt source model(s) to a specific Target Domain, and domain generalization approaches, that learn a domain agnostic model from multiple sources that can be applied to any Target Domain. In 2017, to improve domain generalization, a low-rank parameterized CNN model has been developed in order to counter the risk of overfitting caused by an increasing number of Source Domains [5]. For their analysis, by intersecting the classes found in Caltech256, Sketchy, TU-Berlin and Google Images,the authors created the PACS dataset, which encompasses various domains like photos, sketches, cartoons, and paintings. The goal here is to achieve deep semantic understanding across different domains, ensuring a consistent classification regardless of the domain, just like a human would do.

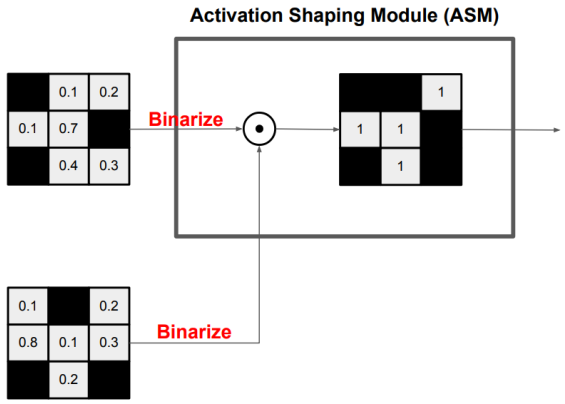


Figure 1. **Illustration of the Activation Shaping Module.** The two input activation maps (A and M) undergo a binarization process (with a binarization threshold set to 0), converting the original floating-point values into binary values (0s and 1s), and are then multiplied element-wise.

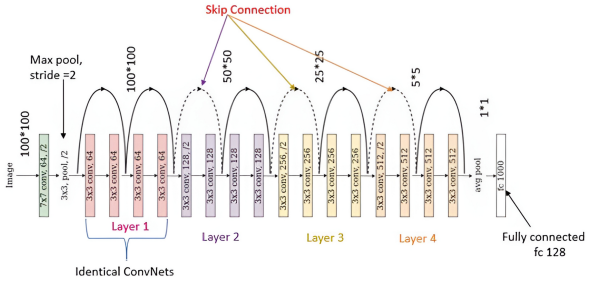
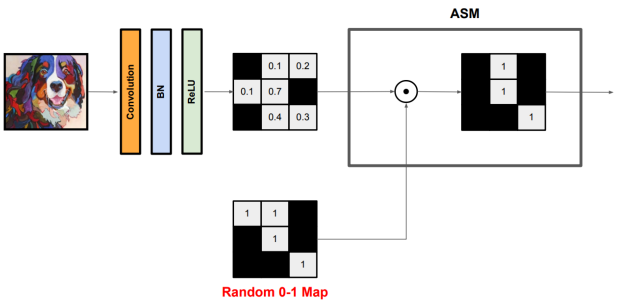


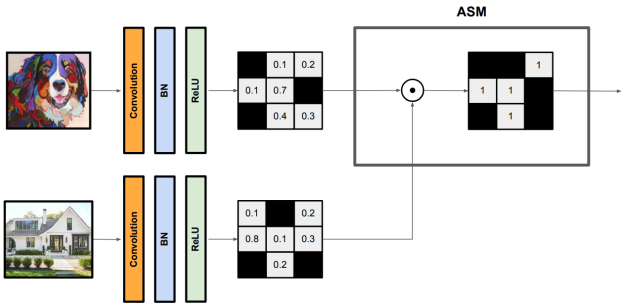
Figure 2. Diagram of the ResNet-18 Architecture.

2.4. Our work

Differentiating from "Extremely simple activation shaping for out-of-distribution detection": While the previous work used ResNet-50 pretrained on ImageNet for Activation Shaping, we adopted a ResNet-18 (see Figure 2), also pretrained on ImageNet. This choice, despite the smaller architecture, has provided high accuracy, particularly with 'Photos' as the Target Domain. In contrast to the original study which achieved optimal results by applying ASH to the later layers of the network and noted a decline in both accuracy and OOD detection when moving ASM towards the beginning, our research shows that the best results are obtained when ASM is positioned in the middle layers. However, even when applied at the end of the network, our approach still shows good results. Furthermore, we applied ASH within the contexts of UDA in an innovative way with different custom activation shaping modules, whereas the original study focused primarily on pruning and adjusting activations.



(a) By slightly modifying the ASM, random maps ablation can be performed.



(b) The ASM can receive activation maps from both the Source Domain (Art Paintings) and the Target Domain (one between Photos, Cartoons or Sketches).

Figure 3. ASM shows a really flexible behaviour and can be easily adapted to receive inputs of different nature.

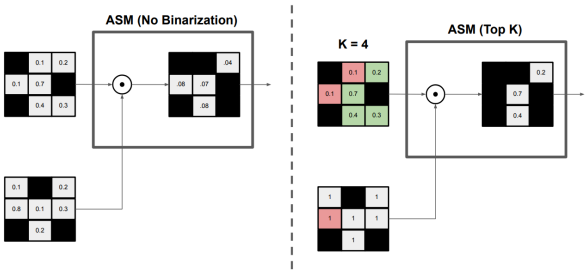


Figure 4. Additional variations of the custom Activation Shaping Module (ASM) can be achieved by omitting the binarization step and implementing a top-K approach.

Diversity with respect to "Domain-adversarial training of neural networks": Our approach to domain adaptation diverges from the domain-adversarial neural network (DANN) methodology. Instead of employing DANN, our model utilizes Activation Shaping Module, previously presented, to address domain adaptation challenges. This shift from DANN to ASM represents a significant change in the strategy for mitigating domain shift.

Comparison with "Deeper, broader and artier domain generalization": While the discussed paper on domain generalization employed a variety of datasets, our study concentrated only on the PACS dataset that they created. Regarding domain generalization, we suppose that a modified version of the Activation Shaping Module that accepts more than two activation maps could offer a new, different approach, from previous studies. However, we have not delved directly into the domain generalization setting, but starting from PACS data and its four different domains is possible to go down this road.

3. Method

Our study introduces an Activation Shaping Module designed for domain adaptation in convolutional neural networks, which aims to mitigate the domain shift problem by emphasizing content-specific features over style-specific ones. The ASM operates by taking two activation maps as inputs: a main activation map (A) and an auxiliary activation map (M), both of which are subject to binarization. We tried also different threshold values with a fixed configuration, but with poor results. So, we kept using 0 as binarization threshold. Then, the element-wise product of these 2 activation maps is taken. This process effectively simplifies the activation maps, enabling the model to focus on generalizable features that enhance cross-domain applicability. We integrated the ASM (implemented as a function hooked using PyTorch forward hooks [2], which receives in input activation maps and processes them before passing to the next module) into a ResNet-18 architecture to examine the module's influence when inserted at different layers, exploring various configurations and evaluating the resultant effects on domain adaptation performance. In the context of Unsupervised Domain Adaptation (UDA), we trained the network using labeled data from the Source Domain and unlabeled data from the Target Domain. The forward pass through the ASM was performed with images from both domains, and the network's output was used to compute a simple classification loss (cross-entropy loss) that we optimized. Essentially, we take 1 picture from the Source Domain (with the corresponding label), 1 picture from the Target Domain and then we perform the forward pass. Additionally, we conducted random maps ablation studies by varying the ratio of 0s and 1s within the activation maps to understand the influence of activation sparsity on the model's capability to generalize from the Source Domain to the Target domain. By manipulating these ratios, we investigated the robustness of feature representations under varying degrees of activation pruning (see Figure 3).

Baseline	Average Accuracy Across All Domains
	63.63%

Table 1. **Average Accuracy Across All Domains for the baseline.** The average accuracy of 63.63% across all domains provides a reference for evaluating the performance improvements achieved by the various experiments.

Lastly, we analyzed the impact of binarization on the network's performance, since there is the possibility that binarization may potentially distort the feature representation too much and alter the network behaviour in an undesired way (one thing that could be altered are the batch normalization layers for instance). To address this aspect, we evaluated the network's performance with and without the binarization step, and we also investigated the effects of retaining only the top K values in the activation maps (see Figure 4). While Domain Generalization (DG) presents a further extension to this work, allowing for a model to process inputs from multiple labeled Source Domains, our study did not focus on this aspect. However, future research could benefit from exploring this pathway, which holds the potential to yield a model with enhanced generalization capabilities.

4. Experiments

As a starting point we took the GitHub repository [1] and trained the unmodified ResNet-18 and the Object Classifier, minimizing the cross-entropy loss computed on the Source Domain (Art Painting). Then, we tested the model on the three Target Domains (Cartoon, Sketch, Photo) separately and averaged the results (see Tab. 1), in order to establish a baseline for comparing the results of our experiments. However, an important note has to be done for what that regards all the experiments that use Photo as Target Domain. These ones, indeed, always yielded excellent results in our analysis. This phenomenon is not merely coincidental or indicative of the exceptional power of our method, but is a consequence of ResNet-18's pretraining on the ImageNet dataset, which comprises millions of labeled photos spanning thousands of categories, many of which closely resemble examples in the Photo domain. Using Cartoon or Sketch as Target Domain, instead, we have patterns in data that are really different from those of ImageNet examples and, so, we will get lower accuracy values. For this reason, all the results of the experiments involving the Target Domain Photo will be greater than 90% in terms of accuracy, but will be around 40%-60% of accuracy for Cartoon and Sketch used as Target Domains. Moving to the experiments, at the beginning we focused on using the ASM in the context of UDA.

4.1. Unsupervised Domain Adaptation

The main idea behind dealing with the domain shift problem using activation shaping is that in our network there could be some specific paths that are more content-specific (focus on the object in the picture) and some other paths that carry style-specific information. With this experiment, we are asking ourselves: "can we discard style-specific paths in order to retain only content-specific paths, which are the ones that result useful to us to accomplish correctly (and with a higher accuracy) our classification task?". For this purpose, in the context of UDA the activation maps in input to our ASM come from both the Source and the Target Domain. That said, once we implemented our custom ASM, we tried to insert it in several different layers. First of all, we started working on the 4 macrolayer of the ResNet-18, integrating one single ASM at the time after one of these layers. Then, we moved to a lower level and decided to consider also the various microlayers to integrate the ASM, conducting several experiments to observe if we can improve the results. We also attempted to attach our custom ASM in several different layers at the same time, but this approach tendentially worsened the results in all the cases. We also tried inserting the ASM after each ReLU activation, one ASM every 3 ReLU activation and performed several other experiments. However, due to the bad results of the insertion of ASM after multiple layers, in the following analysis we focused only on adding the ASM on a single layer at the time, since adding two or more modules seems to be altering our network not in a positive way. A further selection then has been made to elect some candidates as best layers on which applying the ASM. In particular, we obtained the best result for UDA when inserting our module in the layer3.0.downsample.0 (64.85% as average accuracy), followed by the layer4.0.downsample.0 (64.05% as average accuracy), as we can see from Figure 5. However, the layer4.0.downsample.0 performed much worse in the following experiments, so for this reason we decided to discard it. Generally, inserting the ASM in the middle seems to lead to improvements in terms of accuracy with respect to the baseline and it makes sense if we consider that it compress a bit the representation. Instead, inserting the ASM on the first layers or in the last layer does not produce good results, which is reasonable if we consider that at the beginning our model has learned only low level features, whereas at the end we have the last layer which is responsible for classification.

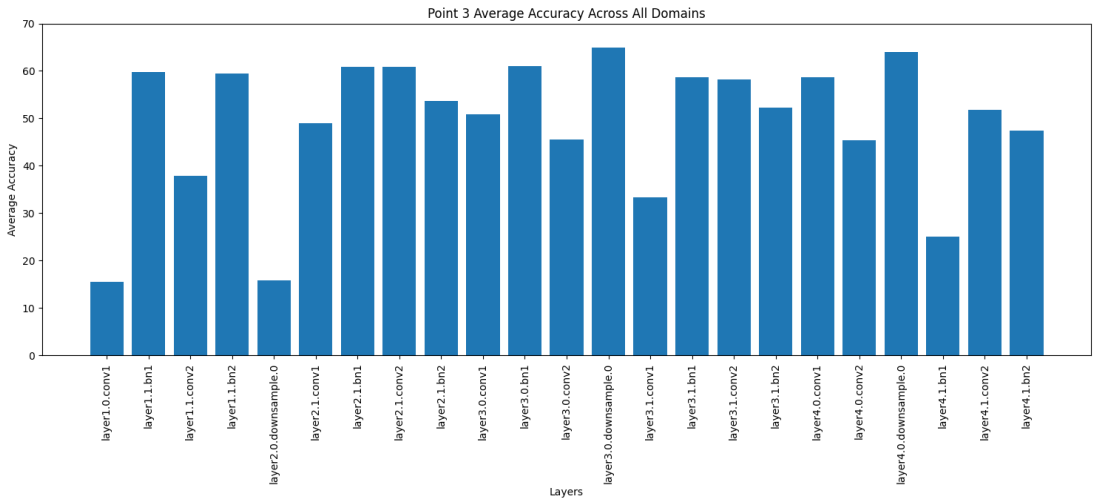


Figure 5. **UDA: Average accuracy across all Target Domains for various layers of the network.** The bars illustrate the average accuracy achieved by inserting the ASM at different layers within the base architecture.

4.2. Random Maps Ablation

After our experiments in the context of UDA, we performed some random maps ablation (see Figure 6) and observed that two of the layers that produced good enough results in the previous experiment, still behaved well in this phase. In this scenario the idea is to make so that each matrix M is a random mask of zeros and ones, whereas our matrix A still comes from our Source Domain. In particular, we experimented trying different one ratio values for matrix M (0.1, 0.4, 0.6 and 0.9), with the goal of deactivating some of the elements of the main activations. What we discovered is that very good results have been obtained with layer3.0.downsample.0 when switching off approximately half of the activations (66.30% of average accuracy across the three Target Domains with one ratio set to 0.6), whereas performances get slightly worse when we prune too much or too little (which is reasonable if we consider that pruning too much, we are throwing away useful information, and that pruning too little we are introducing a very small noise, keeping the representation almost unchanged). Hence, if we take any activation map and we decide to turn off some random outputs, and then we continue the training procedure, the network will adapt in some way and at the end we will obtain almost the same accuracy of the baseline (or even better results). The reason for which we obtained good results is that, similarly to what is achieved with dropout reguralization technique, we are introducing a bit of noise and we are destabilizing the network, preventing our model from memorizing the training data. So, essentially, randomness seems to prevent overfitting.

4.3. Binarization Ablations

As discussed in the previous section regarding the method, we binarized the main activation map and the auxiliary activation map. However, since there is the possibility that binarization may potentially distort the feature representation too much and alter the network behaviour in an undesired way, we decided to repeat both our experiments without applying binarization (see Figure 7). What we found out is that random map ablations without applying binarization (Ext.2 Var.1 Pt.2) with a one ratio equal to 0.6, performed on layer3.0.downsample.0, gave us 63.69% as average accuracy value. This is almost the same result of the baseline, but the binarization case still behaves better. A slight improvement with respect to the baseline has been obtained with ones ratio = 0.6 on layer3.1.conv2, which gave us 64.90% of average accuracy. Anyway this is still a worst result than the one obtained with binarization for layer3.0.downsample.0 with a one ratio equal to 0.6. In UDA, instead, not binarizing (Ext.2 Var.1 Pt.3) led to a significant accuracy worsening for all the three target domains. This means that binarization is in some way helping the network in giving more importance to features that do not discriminate between the training (source) and test (target) domains.

Then, a further interesting experiment consisted in masking the main activation map retaining only the top K values, where K is an hyperparameter that we tuned opportunely (as shown in Figure 8a). Doing so, the element of the matrix M that are not in the Top-K are discarded. After testing several several values of K (0.1, 0.4, 0.6, and 0.9) across various layers (where for instance 0.1 means that we

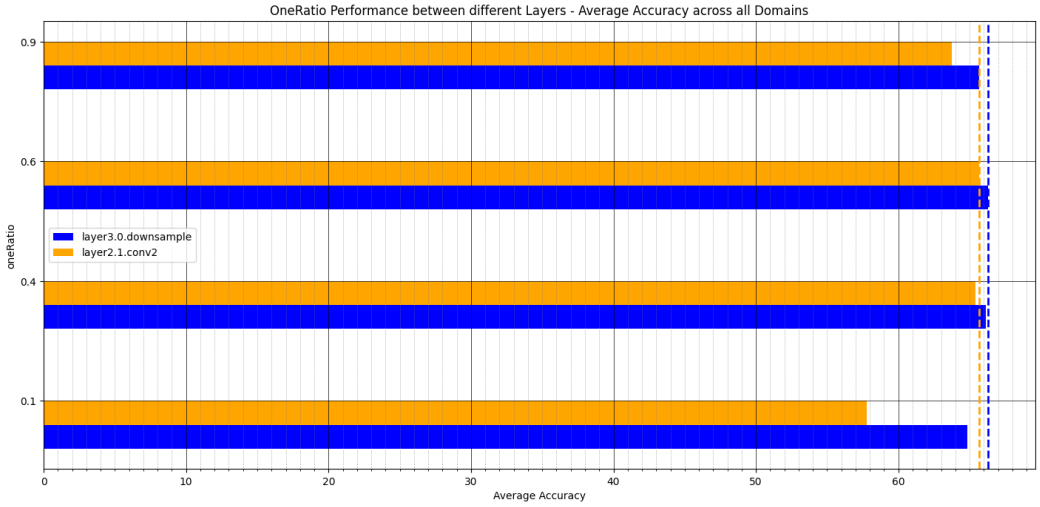
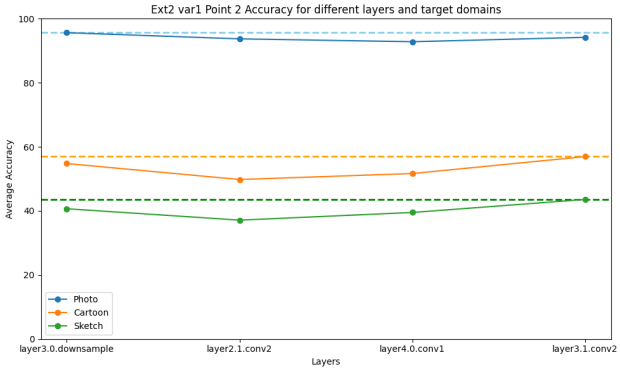
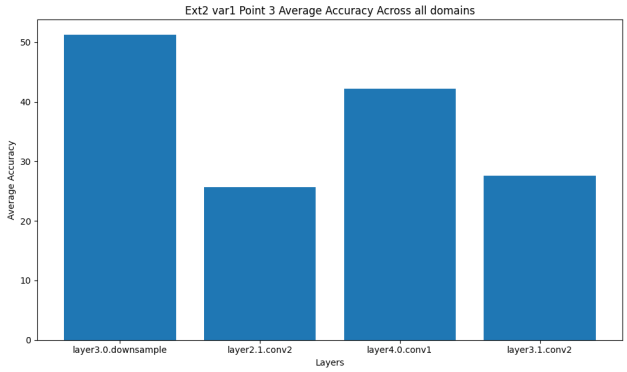


Figure 6. **Random maps ablation results with One Ratio tuning.** The bar chart compares the average accuracy obtained inserting the ASM on two of the best layers.



(a) Random activation maps with ones ratio = 0.6: this graph depicts the accuracy for different layers and for the three different Target Domains: Photo, Cartoon, and Sketch. Each line represents the performance trend for one domain, showing how accuracy changes inserting the ASM after the specified layers (layer3.0.downsample, layer2.1.conv2, layer4.0.conv1, and layer3.1.conv2).

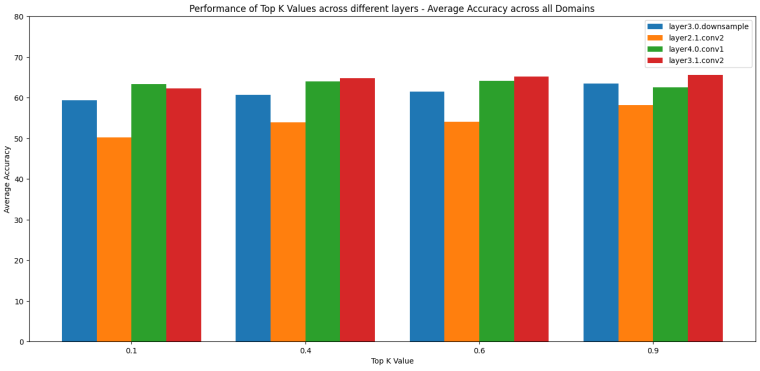


(b) UDA: The bar chart illustrates the average accuracy when inserting the ASM after some of the layers in the case of UDA with no binarization.

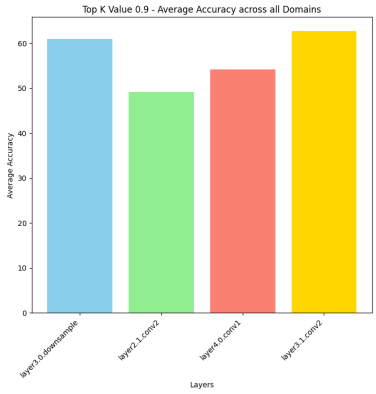
Figure 7. Binarization ablation var1 (with no binarization) results.

maintain different from 0 all the elements of our main activation that represent the 10% of the highest values of the map), what we discovered is that higher values of K generally behaves better, with the best results obtained for K = 0.9 for both random activation maps (Ext.2 Var.2 Pt.2) and UDA (Ext.2 Var.2 Pt.3). Also in this experiment, the layer that worked better has been layer3.1.conv2, followed by layer3.0.downsample.0. (see Figure 8b)

As we can see from the summarized results for random maps ablation, according to Tab. 2, in general binarization could be beneficial, but things may change according to the layer on which we are reasoning on. In any case, the bigger improvement has been obtained by applying binarization. Moving to the summarized results for UDA (shown in Tab. 3), with a value of K that allowed us to preserve the 90% of the elements of our main activation, things get worse without applying binarization. Essentially, binarization plays an important role and seems to be helpful in the context of Out-of-distribution detection.



(a) **Tuning of K hyperparameter.** The bar chart shows the average accuracy for different values of K (0.1, 0.4, 0.6, and 0.9) across best layers (layer3.0.downsample, layer2.1.conv2, layer4.0.conv1, and layer3.1.conv2) of the network in which we embedded the ASM. (Pt.2 Ext.2 Var.2)



(b) The best results in terms of average accuracy across all Target Domains have been obtained for K = 0.9. (Pt.3 Ext.2 Var.2)

Figure 8. Binarization ablation Top K (var. 2) results.

		ONE RATIO = 60%	
Source Domain	**Average Accuracy Across All Domains**		
Art Painting	**Pt. 2**	**Ext.2 Var.1 Pt.2**	**Ext.2 Var.2 Pt.2 (TOP-K = 90%)**
layer3.0.downsample.0	66.30 %	63.69 %	63.44 %
layer2.1.conv2	65.67 %	60.20 %	58.23 %
layer4.0.conv1	48.32 %	61.32 %	62.14 %
layer3.1.conv2	53.38 %	64.90 %	65.57 %

Table 2. **Average accuracy across all Domains for various layers in the Source Domain of Art Painting.** The table presents the average accuracy results for different layers under different experimental settings. The experiments Ext.2 Var.1 Pt.2 and Ext.2 Var.2 Pt.2 have been respectively conducted with a one ratio of 60% and a top-K value of 90%.

Source Domain	**Average Accuracy Across All Domains**		
Art Painting	**Pt. 3**	**Ext.2 Var.1 Pt.3**	**Ext.2 Var.2 Pt.3**
			TOP-K = 90%
layer3.0.downsample.0	64.85 %	51.23 %	60.96 %
layer2.1.conv2	60.93 %	25.70 %	49.15 %
layer4.0.conv1	58.62 %	42.21 %	54.22 %
layer3.1.conv2	58.27 %	27.63 %	62.67 %

Table 3. **UDA: Average accuracy across all domains for different layers.**The experiment Ext.2 Var.2 Pt.3 has been conducted with K = 90%.

5. Conclusion

In conclusion, this work has addressed the significant problem of domain shift, which has been formalized with the settings of Unsupervised Domain Adaptation.

The key innovation of the adopted method is the use of activation shaping, that has been shown to be a powerful tool for detecting out-of-distribution (OOD) data, offering a promising solution to mitigate the challenges posed by domain shift. As a matter of fact, by modifying activation maps within the network architecture, activation shaping leverages the insights regarding data distribution discrepancies to enhance model robustness. Furthermore, the ASM can be used to turn off some elements of an activation map, introducing some randomness from which a model can take advantage, preventing overfitting.

In particular, we demonstrated the benefits of the employed methods on the image classification dataset PACS, which is characterized by 4 different visual domains.

We hope this work may represent a significant contribution that unlocks new research directions in the field of image classification, in particular when dealing with different data distributions scenarios.

References

[1] Github project repository. https://github.com/iurada/Activation-Shaping-AML. 4

[2] Stanford university website. https://web.stanford.edu/~nanbhas/blog/forward-hooks-pytorch/. 3

[3] Andrija Djurisic, Nebojsa Bozanic, Arjun Ashok, and Rosanne Liu. Extremely simple activation shaping for out-of-distribution detection. *ICLR*, 2023. https://arxiv.org/pdf/2209.09858.pdf. 1

[4] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, Francois Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 17(1):2096–2030, 2016. https://arxiv.org/pdf/1505.07818.pdf. 2

[5] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Deeper, broader and artier domain generalization. *CoRR*, 2017. https://arxiv.org/pdf/1710.03077.pdf. 2