Supplementary Material

Prithvijit Chattopadhyay[1], Yogesh Balaji[2], and Judy Hoffman[1]

[1] Georgia Institute of Technology
[2] University of Maryland
{prithvijit3,judy}@gatech.edu, yogesh@cs.umd.com

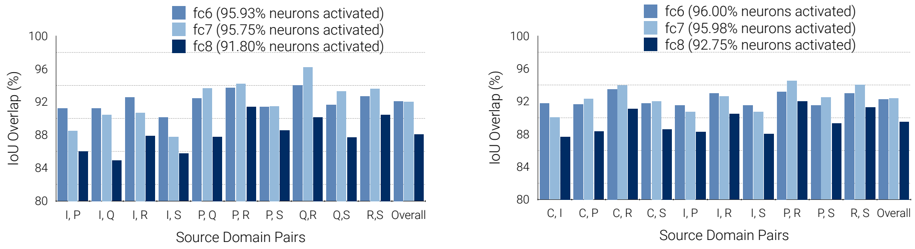
In this supplement, we further discuss the specificity of the obtained domain-specific masks (Section. 0.1). Following this, we discuss how sparsity as an incentive compares with `sIoU` in terms of learning a balance between specificity and invariance and in terms of performance (Section. 0.2). In Section. 0.3, we discuss alternative techniques for directly ensembling masks instead of the output predictions in response to each mask. In Section. 0.4, we provide more extensive comparisons to prior work on the PACS [10] dataset. Finally, in Section. 0.5, we describe in detail the implementation and other details associated with our experiments. We use C, I, P, Q, R, S to denote the domains – *clipart*, *infograph*, *painting*, *quickdraw*, *real* and *sketch* respectively on the DomainNet [17] dataset.

0.1 Domain Specificity

As discussed in Section. 3.2 (main paper), we incentivize domain specificity by optimizing the *soft*-IoU (`sIoU`) objective (see Eqn. 2 in main paper). To understand the extend of domain-specificity achieved at convergence, we measure the Jaccard Similarity Coefficient [7] (also known as IoU) among pairs of *discrete* source domain masks, which we obtain by thresholding the soft-mask values per-domain at 0.5, i.e., $m = \mathbf{1}_{\mathbf{m}^d > 0.5}$ for domain d.

Fig. 1 shows the IoU among pairs of source domain masks in addition to the overall average on DomainNet for the I,P,Q,R,S→C and C,I,P,R,S→Q shifts with AlexNet as the backbone architecture ($\lambda_O = 0.1$ during training). Note that the above metric provides information about the fraction of overlapping neurons which are shared among pairs of source domains but only considers them among the ones which are activated (turned *on*) based on the discrete masks m. Therefore, in addition to the IoU statistics (as represented by the bars), we also report the fraction of activated neurons on average. We note that domain specificity does emerge by learning masks in the manner described in Sec. 3.2 of the main paper, as evident by the IoU measures across pairs being lower than – (1) ∼96% for the maximal pairwise IoU and (2) ∼92% for overall IoU measures across both the shifts. Fig. 2 shows how the layerwise overall IoU measure evolves as λ_O increases. While at lower values of λ_O, the amount of specificity is relatively low and similar across layers, at higher values of λ_O we see an increase of varying degrees across layers – the relative ordering among layers in terms of IoU being `fc6>fc7>fc8`, indicating the importance of having *more* shared neurons in the earlier layers.

Finally, note that since the pairwise IoU measures indicate the fraction of neurons which are shared among the neurons which are turned *on*, upon convergence we can essentially categorize the neurons present in the task network



(a) I,P,Q,R,S → C (O.O.D. Acc 50.06%) (b) C,I,P,R,S → Q (O.O.D. Acc 13.07%)

Fig. 1: **Emergence of domain-specificity in AlexNet with $\lambda_O = 0.1$.** We show the IoU overlap among pairs of discrete source domain masks for the two shifts (a) I,P,Q,R,S→C and (b) C,I,P,R,S →Q on DomainNet [17] with out-of-domain accuracies 48.70% and 12.7% respectively. We find that domain-specificity does indeed emerge, as indicated by the IoU measures.

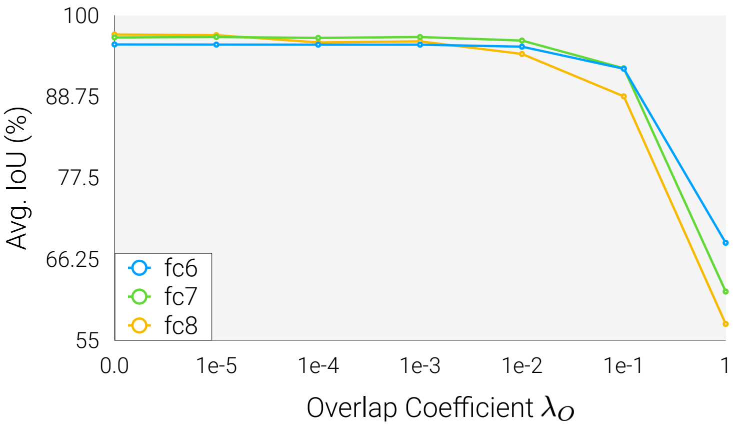
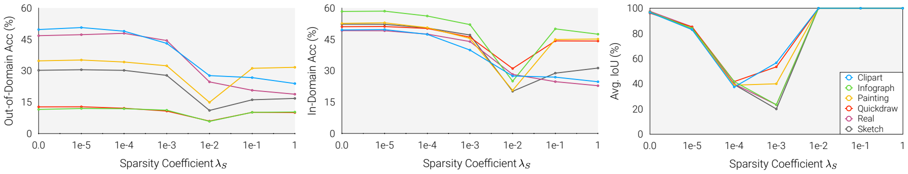


Fig. 2: **Layerwise IoU sensitivity to λ_O.** The average IoU score among pairs of source domain masks decreases as λ_O increases, indicating the degree to which domain-specificity emerges in individual layers (fc6, fc7, fc8).

into three categories – (1) *equally useless* – neurons turned *off* across all the source domain masks, (2) *equally useful* or *shared* – neurons turned *on* across all the source domain masks and (3) *domain-specific* – neurons turned *on* only for specific source domains.

0.2 Choice of Incentive: sIoU vs Sparsity

As described in Section. 3.2 (main paper), to ensure feature selection, we impose a *soft*-IoU loss in addition to standard cross-entropy training to penalize overlap among pairs of source domain masks. However, in practice, one could also impose a sparsity constraint on the domain-sepcific masks being learned ensure minimality in the number of features or neurons selected during learning. However, just incorporating a sparsity constraint does not explicitly incentivize domain-specificity – masks corresponding to all the source domains could just end up



(a) Out-of-Domain Acc (b) In-Domain Acc (c) Average IoU

Fig. 3: **Sensitivity to λ_S.** We replace the `sIoU` with a differentiable sparsity term (coefficient λ_S) – L1-norm of the *soft*-source domain masks, i.e., $\sum_{D_i \in D_S} ||\mathbf{m}_i||_1$ – and study the sensitivity to λ_S as measured by out-of-domain accuracy (a), in-domain accuracy (b) and average IoU score measured among pairs of source domain masks. The legends in (b) indicate the target domain in the corresponding multi-source shift. We find that predictive performance and specificity (Avg. IoU) is *very* sensitive to λ_S.

picking the same set of neurons, which is equivalent to learning a bottleneck layer during training.

We investigate the consequences of incorporating a sparsity regularizer in Figure. 3 on all the multi-source shifts of the DomainNet dataset using AlexNet as our backbone architecture. Specifically, instead of the `sIoU` loss, we penalize the L1-norm of the soft-mask values, i.e., $\mathbf{m}^d$ for all the source domains – $\sum_{d \in D_S} ||\mathbf{m}^d||_1$[‡]. We run a sweep over different values of the coefficient (λ_S) of this sparsity incentive from 0 to 1 in logarithmic increments. Fig. 3 (a) and (b) show how out-of-domain and in-domain generalization performances and Fig. 3 (b) shows how the pairwise IoU measure among the source domain masks – indicating domain-specificity, vary with λ_S. Unlike λ_O (see Sec. 5, main paper), we find that generalization performance is quite sensitive to the choice of λ_S, with both out-of-domain and in-domain accuracies degrading significantly at relatively high values of λ_S. We find performance comparable to our approach only at values of $\lambda = 10^{-5}$. For the pairwise IoU measures, we observe that while specificity increases to some extent till $\lambda_S = 10^{-3}$, but decreases sharply with further increase in λ_S. At high-values of λ_S, we observe that the source domain masks are extremely sparse and have high overlap indicating the fact that the masks essentially encourage learning just a bottleneck layer. This further demonstrates the efficacy of the `sIoU` loss in maintaining a reasonable balance between encouraging specificty while retaining predictive performance.

0.3 Ensembling Choices at Test-time

In Section. 3.2 (main paper), we describe how we follow a soft-scaling scheme akin to dropout [18] at test-time. Specifically, we obtain predictions corresponding to neurons in the task network soft-scaled by individual source domain masks and average them (call this `Pred-Ens`). In this section, we further investigate if the

[‡] Since the soft-mask probabilities ($\mathbf{m}^d$) are positive, $||\mathbf{m}^d||_1$ is essentially the sum of mask probabilities per-neuron and is therefore differentiable and can be optimized using gradient descent.

	Method	Clipart	Infograph	Painting	Quickdraw	Real	Sketch	Overall
	Out-of-Domain							
AlexNet	Aggregate	47.17	10.15	31.82	11.75	44.35	26.33	28.60
	Aggregate-SGD$^{\mp}$	42.30	12.42	31.45	9.52	42.76	29.34	27.97
	Multi-Headed	45.96	10.56	31.07	12.05	43.56	25.93	28.19
	MetaReg [1]$^{\mp}$	42.86	**12.68**	32.47	9.37	43.43	29.87	28.45
	DMG (`Pred-Ens`)	50.06	12.23	**34.44**	13.07	**46.98**	**30.13**	**31.15**
	DMG (`Mask-Ens`)	**50.10**	12.17	34.38	**13.14**	46.79	30.01	31.10
	In-Domain							
AlexNet	Aggregate	48.56	57.24	51.38	49.60	47.48	50.72	50.83
	Aggregate-SGD$^{\mp}$	48.14	54.93	50.55	48.33	47.57	49.98	49.92
	Multi-Headed	48.16	56.73	51.31	49.75	47.65	50.82	50.74
	MetaReg [1]$^{\mp}$	48.87	56.06	51.23	49.60	48.66	50.12	50.76
	DMG (`Pred-Ens`)	49.63	58.47	52.88	51.33	49.07	52.42	52.30
	DMG (`Mask-Ens`)	49.49	58.38	52.81	51.16	48.90	52.29	52.17
	DMG-KnownDomain	**51.91**	**61.01**	**54.93**	**53.84**	**51.08**	**54.47**	**54.54**

Table 1: **Ensembling Choices at Test-time**. We study how different ensembling choices at test-time – (1) `Mask-Ens`: ensemble predictions from all the source domain masks and (2) `Pred-Ens`: combine masks and then make a prediction – compare in terms of in [bottom-half] an out-of-domain [top-half] performance. Using AlexNet as the backbone architecture on the DomainNet [17] dataset, we find that `Mask-Ens` leads to very minor ($< 1\%$) drop in both in and out-of-domain performance compared to `Pred-Ens` at test-time. The columns identify the held out sixth domain for each of the multi-source shifts.$^{\mp}$We were unable to optimize the MetaReg [1] objective with Adam [8] as the optimizer and therefore, we also include comparisons with Aggregate and MetaReg trained with SGD.

choice of ensembling method at test-time matters. We compare `Pred-Ens` with the setting where we average the soft masks ($\mathbf{m}^d$ for source domain d) and draw a single prediction by scaling neurons with the averaged soft-mask – `Mask-Ens`.

In Table. 1, we compare DMG (`Pred-Ens`) and DMG (`Mask-Ens`) in terms of both in and out-of-domain performances on all the multi-source shifts on DomainNet using AlexNet as the backbone architecture. We observe that `Mask-Ens` performs comparatively with `Pred-Ens`, with the margin of difference being within ∼1%.

0.4 More Results

In Table. 2, we present more extensive comparisons of DMG with prior work on the PACS [11] using AlexNet, ResNet-18 and ResNet-50 as the backbone CNN architectures. We now describe briefly the prior approaches we compare to.

DICA [15] is a kernel-based optimization algorithm that aims a learn a transformation that renders representations invariant across domains by minimizing the dissimilarity across the source domains. D-MTAE [6] is an autoencoder

	Method	A	C	P	S	Overall
AlexNet	Aggregate [13]	63.40	66.10	88.50	56.60	68.70
	Aggregate*	56.20	70.69	86.29	60.32	68.38
	Multi-Headed	61.67	67.88	82.93	59.38	67.97
	DICA [15]	64.60	64.50	**91.80**	51.10	68.00
	D-MTAE [6]	60.30	58.70	91.10	47.90	64.50
	DSN [2]	61.10	66.50	83.30	58.60	67.40
	TF-CNN [11]	62.90	67.00	89.50	57.50	69.20
	Fusion [14]	64.10	66.80	90.20	60.10	70.30
	DANN [5]	63.20	67.50	88.10	57.00	69.00
	MLDG [12]	66.20	66.90	88.00	59.00	70.00
	MetaReg [1]	63.50	69.50	87.40	59.10	69.90
	CrossGrad [19]	61.00	67.20	87.60	55.90	67.90
	Epi-FCR [13]	64.70	72.30	86.10	65.00	72.00
	MASF [3]	**70.35**	**72.46**	90.68	67.33	**75.21**
	DMG (Ours)	64.65	69.88	87.31	**71.42**	73.32
ResNet-18	Aggregate [13]	77.60	73.90	94.40	74.30	79.10
	Aggregate*	72.61	78.46	93.17	65.20	77.36
	Multi-Headed	78.76	72.10	94.31	71.77	79.24
	DANN [5]	81.30	73.80	94.00	74.30	80.80
	MAML [4]	78.30	76.50	**95.10**	72.60	80.60
	MLDG [12]	79.50	77.30	94.30	71.50	80.70
	MetaReg† [1]	79.50	75.40	94.30	72.20	80.40
	CrossGrad [19]	78.70	73.30	94.00	65.10	77.80
	Epi-FCR [13]	**82.10**	77.00	93.90	73.00	**81.50**
	MASF [3]	80.29	77.17	94.99	71.68	81.03
	DMG (Ours)	76.90	**80.38**	93.35	**75.21**	81.46
ResNet-50	Aggregate*	75.49	**80.67**	93.05	64.29	78.38
	Multi-Headed	75.15	76.37	**95.27**	75.26	80.51
	MASF [3]	**82.89**	80.49	95.01	72.29	82.67
	DMG (Ours)	82.57	78.11	94.49	**78.32**	**83.37**

Table 2: **Out of Domain Generalization Results on PACS**. We compare performance (accuracy in %) against prior work in the standard domain generalization setting of training on three domains as source and evaluating on the held-out fourth domain (identified by the column headers). We include the aggregate baseline both as reported in [13] as well as our own implementation (indicated as Aggregate*)

based approach which aims to learn invariant representations by cross-domain reconstruction. DSN [2] aims to extract representations that can be partitioned into domain-specific and domain-invariant components. TF-CNN [11] learns a low-rank parameterized CNN for end-to-end domain-generalization training. Fusion [14] fuses predictions from all classifiers trained on all the source domains at test-time. DANN [5] leverages the source domain features extractor from Domain Adversarial Neural Networks to generalize to target domains. MetaReg [1] learns regularizers by modeling domain-shifts within the source set of distributions. MLDG [12] learns network parameters using meta-learning. Epi-FCR [13] is a recently proposed episodic scheme to learn network parameters robust to domain-shift. MASF [3] is a recent approach which introduces complementary losses to explicitly regularize the semantic structure of the feature space via a model-agnostic episodic learning procedure. Cross-Grad [19] uses Bayesian Networks to perturb the input manifold for domain generalization.

0.5 Experimental Details

We summarize several experimental details in this section. For all our experiments, we use Adam [8] as the optimizer with a batch size of 64. For PACS, we use an initial learning rate of 10^{-4} for both the network and mask parameters decayed exponentially with a rate of 0.99 every epoch and set weight decay to 10^{-5}. For DomainNet, we use an initial learning rate of 10^{-4} for both the network and mask parameters decayed per-epoch using an inverse learning rate schedule[§] and set weight decay to 0. We conduct a sweep over values of λ_O – coefficient of the `sIoU` loss – in the range $\{0, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$. Our backbone CNN architectures are initialized with ImageNet [9] pretrained checkpoints. We initialize the final linear layer weights (to be learned from scratch) from a zero centered normal distribution ($\mathcal{N}(0, 0.001)$) and a uniform distribution (standard in PyTorch) for DomainNet and PACS respectively. For all our experiments, we initialize the mask parameters from the uniform distribution, i.e., $\tilde{\mathbf{m}}^d \sim \mathtt{U}(0, 1)$. We select the best checkpoints across 50 epochs of training based on overall in-domain validation accuracy. We implement everything in the Pytorch [16] framework[¶]. Our code and data-splits will be made publicly available.

References

1. Balaji, Y., Sankaranarayanan, S., Chellappa, R.: Metareg: Towards domain generalization using meta-regularization. In: Advances in Neural Information Processing Systems. pp. 998–1008 (2018)
2. Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., Erhan, D.: Domain separation networks. In: Advances in neural information processing systems. pp. 343–351 (2016)
3. Dou, Q., de Castro, D.C., Kamnitsas, K., Glocker, B.: Domain generalization via model-agnostic learning of semantic features. In: Advances in Neural Information Processing Systems. pp. 6447–6458 (2019)

[§] $lr_t = \frac{lr_0}{(1+\gamma(t-1))^p}$ where $\gamma = 10^{-4}, p = 0.75$, t identifies the epoch and lr_0 is the initial learning rate.

[¶] The authors of [17] indicated in communication that they used Caffe to implement the multi-source baselines. We re-implement the multi-source baselines in PyTorch [16] to ensure consistency across all our reported results. The subsequent differences in multi-source baseline accuracies can be attributed to the differences in how AlexNet is implemented in PyTorch and Caffe.

4. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 1126–1135. JMLR. org (2017)
5. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. The Journal of Machine Learning Research **17**(1), 2096–2030 (2016)
6. Ghifary, M., Bastiaan Kleijn, W., Zhang, M., Balduzzi, D.: Domain generalization for object recognition with multi-task autoencoders. In: Proceedings of the IEEE international conference on computer vision. pp. 2551–2559 (2015)
7. Jaccard, P.: Etude de la distribution florale dans une portion des alpes et du jura. Bulletin de la Societe Vaudoise des Sciences Naturelles **37**, 547–579 (01 1901). https://doi.org/10.5169/seals-266450
8. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
10. Li, D., Yang, Y., Song, Y.Z., Hospedales, T.: Deeper, broader and artier domain generalization. In: International Conference on Computer Vision (2017)
11. Li, D., Yang, Y., Song, Y.Z., Hospedales, T.M.: Deeper, broader and artier domain generalization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5542–5550 (2017)
12. Li, D., Yang, Y., Song, Y.Z., Hospedales, T.M.: Learning to generalize: Meta-learning for domain generalization. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
13. Li, D., Zhang, J., Yang, Y., Liu, C., Song, Y.Z., Hospedales, T.M.: Episodic training for domain generalization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1446–1455 (2019)
14. Mancini, M., Bulò, S.R., Caputo, B., Ricci, E.: Best sources forward: domain generalization through source-specific nets. In: 2018 25th IEEE International Conference on Image Processing (ICIP). pp. 1353–1357. IEEE (2018)
15. Muandet, K., Balduzzi, D., Schölkopf, B.: Domain generalization via invariant feature representation. In: International Conference on Machine Learning. pp. 10–18 (2013)
16. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in neural information processing systems. pp. 8026–8037 (2019)
17. Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1406–1415 (2019)
18. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research **15**(1), 1929–1958 (2014)
19. Volpi, R., Namkoong, H., Sener, O., Duchi, J.C., Murino, V., Savarese, S.: Generalizing to unseen domains via adversarial data augmentation. In: Advances in Neural Information Processing Systems. pp. 5334–5344 (2018)