Learning to Balance Specificity and Invariance for In and Out of Domain Generalization

Prithvijit Chattopadhyay[1], Yogesh Balaji[2], Judy Hoffman[1]
[1] Georgia Institute of Technology, [2] University of Maryland
{prithvijit3,judy}@gatech.edu
yogesh@cs.umd.com

Abstract

*We introduce **D**omain-specific **M**asks for **G**eneralization, a model for improving both in-domain and out-of-domain generalization performance. For domain generalization, the goal is to learn from a set of source domains to produce a single model that will best generalize to an unseen target domain. As such, many prior approaches focus on learning representations which persist across all source domains with the assumption that these domain agnostic representations will generalize well. However, often individual domains contain characteristics which are unique and when leveraged can significantly aid in-domain recognition performance. To produce a model which best generalizes to both seen and unseen domains, we propose learning domain specific masks. The masks are encouraged to learn a balance of domain-invariant and domain-specific features, thus enabling a model which can benefit from the predictive power of specialized features while retaining the universal applicability of domain-invariant features. We demonstrate competitive performance compared to naive baselines and state-of-the-art methods on both PACS and DomainNet.*

1. Introduction

The success of deep learning has propelled computer vision systems from purely academic endeavours to key components of real-world products. This deployment into unconstrained domains has forced researchers to focus attention beyond a closed-world supervised learning paradigm, where learned models are only evaluated on held-out in-domain test data, and instead produce models capable of generalizing to diverse test time data distributions.

This problem has been formally studied and progress measured in the *domain generalization* literature [5, 13]. Most prior work in domain generalization focuses on learning a model which generalizes to unseen domains by either directly optimizing for domain invariance [13] or designing regularizers that induce such a bias [1], the idea being that features which are present across multiple training distributions are more likely to persist in the novel distributions. However, in practice, as the number of training time data sources increases it becomes ever more likely that at least some of the data encountered at test time will be very similar to one or more source domains. In such a situation, ignoring features specific to only a domain or two may artificially limit the efficacy of the final model. However, leveraging a balance between "*invariance*" – features that are shared across domains – and "*specificity*" – features which are specific to individual domains – might actually aid the model in making a better prediction.

In this paper, we propose DMG: **D**omain-specific **M**asks for **G**eneralization, an algorithm for automatically learning to balance between domain-invariant and domain-specific features producing a single model capable of simultaneously achieving strong performance across multiple distinct domains. At a high-level, we cast this problem of *balanced* feature selection as one of learning distribution-specific binary masks over features of a shared deep convolutional network (CNN). Specifically, for a given layer in the CNN, we associate domain-specific mask parameters for each neuron which decide whether to turn that neuron *on* or *off* during a forward pass and learn these masks end-to-end via backpropagation along with the network parameters. We demonstrate that DMG achieves competitive out-of-domain performance on the commonly used PACS [9] benchmark and on the challenging DomainNet [14] dataset. Additionally, we demonstrate that our model can be used as a drop-in replacement for an aggregate model when evaluated on in-domain test samples, or can be trivially converted into a high performing domain-specific model given a known test time domain label.

2. Approach

Problem Setup. Domain generalization involves training a model on data, denoted as $\mathcal{X}$, sampled from p source distributions that generalizes well to q unknown target distributions which lack training data. We focus on the classification case, where the goal is to learn a model which maps inputs to the desired output label, $M : \mathcal{X} \rightarrow \mathcal{Y}$ (the source and target distributions share the same label space).

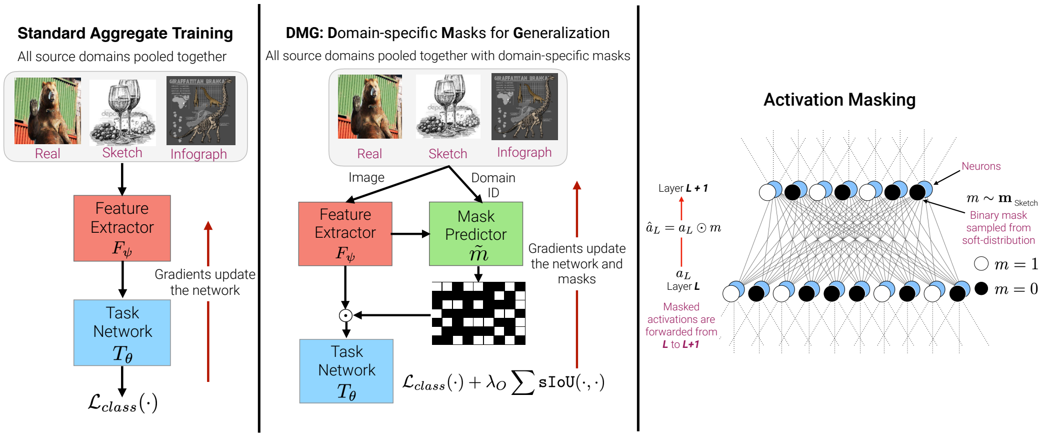
Standard Aggregate Training
All source domains pooled together
Real
Sketch
Infograph
Feature Extractor F_ψ
Task Network T_θ
$\mathcal{L}_{class}(\cdot)$
Gradients update the network
DMG: Domain-specific Masks for Generalization
All source domains pooled together with domain-specific masks
Real
Sketch
Infograph
Image
Domain ID
Feature Extractor F_ψ
Mask Predictor $\tilde{m}$
$\odot$
Task Network T_θ
$\mathcal{L}_{class}(\cdot) + \lambda_O \sum \texttt{sIoU}(\cdot,\cdot)$
Gradients update the network and masks
Activation Masking
Neurons
Layer **L + 1**
$\hat{a}_L = a_L \odot m$
a_L
Layer **L**
$m \sim \mathbf{m}_{\text{Sketch}}$
Binary mask sampled from soft-distribution
$m = 1$
$m = 0$
Masked activations are forwarded from **L** to **L+1**

Figure 1: **Illustration of our approach (DMG)**: We introduce domain-specific activation masks for learning a balance between domain-specific and domain-agnostic features. [Left] Our training pipeline involves incorporating domain-specific masks in the vanilla aggregate training process. [Middle] For an image belonging to *sketch*, we sample a binary mask from the corresponding mask parameters, which is then applied to the neurons of the task-network. [Right] Post feature extraction, an elementwise product of the obtained binary masks is performed with the neurons of the task network layer (L) to obtain the *effective* activations being passed on to the next layer ($L+1$). The mask and network parameters are learned end-to-end based on the standard cross-entropy coupled with the sIoU loss penalizing mask overlap among the source domains.

We decompose the parametric model $M_\Theta : \mathcal{X} \to \mathcal{Y}$, into a feature extractor (F_ψ) and a task-network (T_θ) i.e., $M_\Theta(x) = (T_\theta \circ F_\psi)(x)$ and learn domain specific masks only on the neurons present in the task network. We refer to the set of source domains as D_S and index individual source domains by d.

Activation or Feature Selection via Domain Specific Masks. We cast this problem of capturing a balance between domain-specific and domain-invariant feature components as that of learning binary masks on the neurons of the task network specific to individual source domains. Our masks can be viewed as layer-wise gates (or switches) which decide which activations to turn *on* or *off* during a forward pass through the network.

Given k neurons at some layer L of T_θ, we introduce parameters $\tilde{\mathbf{m}}^d \in \mathbb{R}^k$ for each of the source distributions $d \in D_S$. During training, for instances x_i^d from domain d, we first form mask probabilities $\mathbf{m}^d$ as $\mathbf{m}^d = \sigma(\tilde{\mathbf{m}}^d)$[*]. Then, the binary masks m_i^d are sampled from a bernoulli distribution given by the mask probabilities. i.e., $m_i^d \sim \mathbf{m}^d$, with $m_i^d \in \{0,1\}^k$. Upon sampling masks for individual neurons, the effective activations which are passed on to the next layer $L+1$ are $\hat{a}_L = a_L \odot m_i^d$, i.e., an elementwise product of the obtained feedforward activations and the sampled binary masks (see Fig. 1, right).

During training, we sample such binary masks corresponding to the source distribution the instance being fed through the network belongs to, thereby making feedforward predictions by *only* using domain-specific masks. Thus, the prediction made by the entire network M_Θ for an instance $x_i^d \in d$ can be expressed as $\hat{y}_i = M_\Theta(x_i^d; m_i^d)$ where m_i^d denotes the sampled mask (for domain d) being applied to all neurons in the task-network T_θ.[†] We learn the mask-parameters in addition to the network parameters during training via backpropagation. Note that since the mask-parameters $\tilde{\mathbf{m}}^d$ cannot be updated directly using back-propagation (the sampled binary masks being discrete), we approximate gradients through sampled discrete masks using the straight-through estimator [2], i.e., we use a discretized m_i^d during a forward pass but use the continuous version $\mathbf{m}^d$ during the backward pass by approximating $\nabla_{\{\theta,\tilde{\mathbf{m}}^d\}} m_i^d \approx \nabla_{\{\theta,\tilde{\mathbf{m}}^d\}} \mathbf{m}^d$. Even though the hard sampling step is non-differentiable, gradients with respect to m_i^d serve as a noisy estimator of $\nabla \mathbf{m}^d$.

Incentivizing Domain-Specificity. To ensure the masks capture neurons that are specific to individual source domains (to incentivize domain-specificity), we introduce an additional *soft*-overlap loss that ensures masks associated with each of the source distributions overlap minimally. We quantify overlap amongst masks associated with different source distributions via a soft analog [15] of the Jaccard Similarity Coefficient [7] (IoU score) among pairs of source domain masks (IoU score itself non- differentiable). We minimize the following *soft*-overlap loss for every pair of source domain masks $\{\mathbf{m}^{d_i}, \mathbf{m}^{d_j}\}$ at a layer L as,

$$\texttt{sIoU}(\mathbf{m}^{d_i}, \mathbf{m}^{d_j}) = \frac{\mathbf{m}^{d_i} \cdot \mathbf{m}^{d_j}}{\sum_k (\mathbf{m}^{d_i} + \mathbf{m}^{d_j} - \mathbf{m}^{d_i} \odot \mathbf{m}^{d_j})} \quad (1)$$

where $\mathbf{m}^{d_i} \cdot \mathbf{m}^{d_j}$ approximates the intersection of masks for the pair of source domains as the inner product of the mask distributions, $\odot$ denotes the elementwise product and k denotes the number of neurons in layer L. During training, sIoU encourages that predictions for instances from different source domains are made using distinct sub-networks

[*] $\sigma(x) = \frac{1}{1+\exp(-x)}$ is the sigmoid function.

[†] Note that, akin to dropout [16], these domain-specific masks identify *domain-specific* sub-networks – for an instance x_i^d, the sampled binary mask m_i^d identifies a specific “thinner” subnetwork.

(as identified by the domain-specific binary masks). To summarize, for a set of source domains D_S the overall objective we optimize during training ensures – (1) good predictive performance on the discriminative task at hand and (2) minimal overlap among source-domain masks,

$$\mathcal{L}(\theta, \psi, \tilde{\mathbf{m}}^{d_1}, .., \tilde{\mathbf{m}}^{d_{|D_S|}}) = \sum_{d \in D_S} \sum_{x_i^d \in d} \mathcal{L}_{\texttt{class}}(\theta, \psi, m_i^d) \\ + \lambda_O \sum_{L \in T_\theta} \sum_{(d_i, d_j) \in D_S} \texttt{sIoU}(\mathbf{m}^{d_i}, \mathbf{m}^{d_j}) \tag{2}$$

where $m_i^d \sim \mathbf{m}_i^d$ for every instance x_i^d of the source domain d and $\mathcal{L}_{\texttt{class}}(\cdot)$ denotes the standard cross entropy loss. Fig. 1 summarizes our training pipeline in context of a standard aggregation method where a CNN backbone architecture is trained on data from all the source domains.

Prediction at Test-time. To obtain a prediction at test-time, we follow a soft scaling scheme similar to Dropout [16] – we scale every neuron by the associated domain-specific *soft*-mask[‡] $\mathbf{m}^d$ instead of turning neurons *on* or *off* based on a discrete mask $m \sim \mathbf{m}^d$ and average the predictions obtained from obtained by applying $\mathbf{m}^d$ for all the source domains to the task network.

3. Experiments

We conduct experiments on the commonly used PACS [9] dataset and the large-scale DomainNet [14] dataset. PACS consists of 4 domains – *photo*, *art-painting*, *cartoon* and *sketch* – and DomainNet consists of 6 domains – *real*, *clipart*, *sketch*, *painting*, *quickdraw* and *infograph*. In addition to out-of-domain performance, we also report in-domain generalization performance on DomainNet. For both the datasets, we use the standard `test` splits as provided by the authors in [9] and [14]. Following standard practice, we conduct leave-one-out domain generalization experiments by treating one domain as the target and everything else as the source distributions. For DomainNet, we use C, I, P, Q, R, S to denote the domains – *clipart*, *infograph*, *painting*, *quickdraw*, *real* and *sketch* respectively. On PACS, we use A, C, P and S to denote the domains – *art-painting*, *cartoon*, *photo* and *sketch* respectively. We experiment with AlexNet [8], ResNet-18 [6] and ResNet-50 [6] backbone architectures pretrained on the ImageNet [4] dataset. For AlexNet, we apply domain-specific masks on the input activations of the last three fully-connected layers (our task networks T_θ) and turn dropout [16] *off* while learning the domain-specific masks. For ResNet-18 and 50, we apply domain specific masks on the input activations of the last residual block and the first fully connected layer.[§] We first compare with two simple baselines (treating dropout [16] as usual if present in the backbone CNN) – (1) Aggregate - the CNN backbone trained on data accumulated from all the source domains and (2) Multi-Headed - the CNN backbone with different classifier heads corresponding to each of the source domains; at test-time we average predictions from all the heads. We also compare with recently proposed domain generalization approaches. Table 1 and 2 summarize out of domain generalization results on the DomainNet and PACS datasets, respectively. Table 3 summarizes in-domain generalization results on DomainNet.

	Method	Clipart	Infograph	Painting	Quickdraw	Real	Sketch	Overall
AlexNet	Aggregate	44.82	9.88	29.78	12.61	41.31	25.35	27.29
	Multi-Headed	44.25	9.66	29.35	11.77	41.02	24.21	26.71
	MetaReg [1]	42.86	**12.68**	32.47	9.37	43.43	**29.87**	28.45
	DMG (Ours)	**49.27**	10.92	**33.74**	**13.30**	**44.78**	29.05	**30.18**
RN-18	Aggregate	56.19	16.18	40.99	14.91	53.68	40.01	37.13
	Multi-Headed	56.28	16.56	40.89	12.18	52.20	36.64	35.79
	DMG (Ours)	**59.48**	**17.52**	**43.48**	**15.10**	**53.86**	**40.59**	**38.34**
RN-50	Aggregate	62.18	19.94	45.47	13.81	57.45	44.36	40.53
	Multi-Headed	61.42	20.24	44.17	13.59	57.90	45.68	40.50
	DMG (Ours)	**63.89**	**21.03**	**47.19**	**15.93**	**58.22**	**45.62**	**41.98**

Table 1: **Out of Domain Generalization Results on DomainNet**. We report performance (accuracy %) of our baselines and our model (DMG) in the standard domain generalization setting where we train on five source domains and report performance on the held-out sixth domain (identified by the column headers).

	Method	Art Painting	Cartoon	Photo	Sketch	Overall
AlexNet	Aggregate [11]	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
	DSN [3]	61.10	66.50	83.30	58.60	67.40
	Fusion [12]	64.10	66.80	**90.20**	60.10	70.30
	MLDG [10]	**66.20**	66.90	88.00	59.00	70.00
	MetaReg [1]	63.50	69.50	87.40	59.10	69.90
	CrossGrad [17]	61.00	67.20	87.60	55.90	67.90
	Epi-FCR [11]	64.70	**72.30**	86.10	65.00	72.00
	DMG (Ours)	64.65	69.88	87.31	**71.42**	**73.32**
ResNet-18	Aggregate [11]	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
	MLDG [10]	79.50	77.30	94.30	71.50	80.70
	MetaReg[†] [1]	79.50	75.40	94.30	72.20	80.40
	CrossGrad [17]	78.70	73.30	94.00	65.10	77.80
	Epi-FCR [11]	**82.10**	77.00	93.90	73.00	**81.50**
	DMG (Ours)	76.90	**80.38**	93.35	**75.21**	81.46
RN-50	Aggregate*	75.49	**80.67**	93.05	64.29	78.38
	Multi-Headed	75.15	76.37	**95.27**	75.26	80.51
	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 [11] as well as our own implementation (indicated as Aggregate*)

DomainNet. On DomainNet, we observe that DMG beats the naive aggregate baseline, the multi-headed baseline and MetaReg [1] based on the AlexNet backbone architecture in terms of overall performance – with an improvement of ∼2% over MetaReg [1]. Interestingly, this corresponds to an almost ∼4% improvement on the I,P,Q,R,S→C and

[‡] Note that sampling from domain-specific *soft*-masks amounts to sampling a "thinned" sub-network from the original task-network. Since it's intractable to obtain predictions from all such possible (exponential) domain-specific sub-networks at test-time, a simple averaging scheme ensures that the *expected* output under the distribution induced by the masks is the same as the actual output at test-time.

[§] For ResNet, the domain-specific masks are trained to *drop* or *keep* specific channels in the input activations as opposed to every spatial feature in every channel which reduces complexity in terms of the number of mask parameters to be learnt.

	Method	Clipart	Infograph	Painting	Quickdraw	Real	Sketch	Overall
AlexNet	Aggregate	45.24	54.74	49.10	47.28	45.88	48.79	48.50
	Multi-Headed	46.19	54.56	49.16	47.59	46.01	47.91	48.57
	MetaReg [1]	48.87	56.06	51.23	49.60	48.66	50.12	50.76
	DMG (Ours)	48.12	57.20	51.18	49.32	47.87	50.59	50.71
	DMG-KnownDomain (Ours)	**50.23**	**59.25**	**53.14**	**51.40**	**49.77**	**52.49**	**52.71**
RN-18	Aggregate	56.19	63.91	58.53	58.32	55.18	57.98	58.35
	Multi-Headed	47.10	55.44	49.74	53.92	45.76	48.20	50.03
	DMG (Ours)	55.80	63.61	58.65	58.61	55.67	58.19	58.42
	DMG-KnownDomain (Ours)	**57.03**	**66.80**	**59.91**	**60.13**	**56.72**	**59.48**	**60.01**
RN-50	Aggregate	59.91	68.39	62.50	62.36	59.18	62.07	62.40
	Multi-Headed	52.31	61.38	55.44	58.22	50.72	53.91	55.33
	DMG (Ours)	59.71	68.38	62.22	62.88	58.57	61.63	62.23
	DMG-KnownDomain (Ours)	**61.46**	**69.90**	**63.61**	**64.47**	**60.10**	**63.20**	**63.79**

Table 3: **In Domain Generalization Results on DomainNet**. We report the average performance (accuracy in %) on the test sets for each training source domain. For the case where inputs have known domain label, we can use the corresponding learning mask (DMG-KnownDomain) to achieve the strongest performance without requiring additional models or parameters. Column headers identify the target domains in the corresponding multi-source shifts.

a ∼1.35% improvement on the C,I,P,Q,S→R shifts (see Table 1). Using ResNet-18 as the backbone architecture, we observe that DMG leads to an overall improvement of ∼1% over the naive Aggregate and Multi-headed baselines accompanied by consistent improvements in the individual multi-source shifts. We observe similar trends using ResNet-50, where DMG leads to an overall improvement of ∼1.98% over the baselines. For in-domain evaluation (see Table. 3), we present both standard DMG as well as a version which assumes knowledge of the domain corresponding to each test instance. In this case, we can also report our model using only the mask corresponding to the known domain label and refer to this as DMG-KnownDomain. Notably, for this case where a test instance is drawn from one of the source domains, our single model provides significant performance improvement over a single jointly trained model (Aggregate) (see Table 3). **PACS.** We observe that DMG with AlexNet as the backbone architecture outperforms the listed prior approaches including MetaReg [1], MLDG [10] and Epi-FCR [11]. Notice that this improvement also comes with an almost a ∼6% improvement over Epi-FCR [11] on the A,C,P→S shift. Using ResNet-18 and ResNet-50 as the backbone architectures, we observe that DMG leads to comparable and improved overall performance, with margins of ∼0.04% and ∼2.85% for ResNet-18 and ResNet-50, respectively. On ResNet-18, we compare with prior approaches as listed in [11] and ResNet-50, we compare with the Aggregate and Multi-headed baselines. Due to its increased size, both in terms of number of images and number of categories, DomainNet proves to be a more challenging benchmark than PACS. Likely due to this difficulty, we find that prior domain generalization approaches perform comparably to naive baselines (ex. Aggregate) on DomainNet, indicating there is significant room for improvement. [¶]

Conclusion. We propose DMG: **D**omain-specific **M**asks for **G**eneralization, a method for multi-source domain learning which balances domain-specific and domain-invariant feature representations to produce a single strong model capable of effective domain generalization. Our model, DMG, benefits from the predictive power of features specific to individual domains while retaining the generalization capabilities of components shared across the source domains. We find that DMG achieves competitive out-of-domain performance on the commonly used PACS dataset and competitive in and out-of-domain performance on the challenging DomainNet dataset.

[¶] We find that both in and out-of-domain generalization accuracies are robust to the choice of λ_O with only minor drop(s) in in-domain performance at extreme values (0.1 and 1.0). Furthermore, we observe that as λ_O increases, average pairwise IoU for the source domain masks decreases – indicating an increase in the ”domain-specificity” of the masks involved.

References

[1] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In *Advances in Neural Information Processing Systems*, pages 998–1008, 2018. 1, 3, 4

[2] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. 2

[3] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *Advances in neural information processing systems*, pages 343–351, 2016. 3

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 3

[5] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision*, pages 2551–2559, 2015. 1

[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. 3

[7] Paul Jaccard. 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. 2

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 3

[9] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Deeper, broader and artier domain generalization. In *International Conference on Computer Vision*, 2017. 1, 3

[10] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 3, 4

[11] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M Hospedales. Episodic training for domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1446–1455, 2019. 3, 4

[12] Massimiliano Mancini, Samuel Rota Bulò, Barbara Caputo, and Elisa Ricci. Best sources forward: domain generalization through source-specific nets. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 1353–1357. IEEE, 2018. 3

[13] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18, 2013. 1

[14] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019. 1, 3

[15] Md Atiqur Rahman and Yang Wang. Optimizing intersection-over-union in deep neural networks for image segmentation. In *International symposium on visual computing*, pages 234–244. Springer, 2016. 2

[16] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 2, 3

[17] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. In *Advances in Neural Information Processing Systems*, pages 5334–5344, 2018. 3