Learning to Learn Single Domain Generalization
Supplementary Materials

Fengchun Qiao
University of Delaware
fengchun@udel.edu

Long Zhao
Rutgers University
lz311@cs.rutgers.edu

Xi Peng
University of Delaware
xipeng@udel.edu

1. Experimental Details

Task models: We design specific task models and employ different training strategies for the three datasets according to their characteristics.

In Digits dataset, the model architecture is *conv-pool-conv-pool-fc-fc-softmax*. There are two 5 $\times$ 5 convolutional layers with 64 and 128 channels respectively. Each convolutional layer is followed by a max pooling layer with the size of 2 $\times$ 2. The size of the two Fully-Connected (FC) layers is 1024 and the size of the softmax layer is 10.

In CIFAR-10-C [2], we use Wide Residual Network (WRN) [11] with 16 layers and the width is 4. The first layer is a 3$\times$3 convolutional layer. It converts the original image with 3 channels to feature maps of 16 channels. Then the features go through three groups of 3$\times$3 convolutional layers. Each group consists of two blocks and each block is composed of two convolutional layers with the same number of channels. And their channels are $\{64, 128, 256\}$ respectively. Each convolutional layer is followed by batch normalization (BN) [3]. An average pooling layer with the size of 8 $\times$ 8 is appended to the output of the third group. Finally, a softmax layer with the size of 10 predicts the distribution over classes.

In SYTHIA [7], we use FCN-32s [5] with the backbone of ResNet-50 [1]. The model begins with ResNet-50. 1$\times$1 convolutional layer with 14 channels is appended to predict scores for each class at each of the coarse output locations. A deconvolution layer is followed to up-sample the coarse outputs to the original size through bilinear interpolation.

Wasserstein Auto-Encodes: We follow [8] to implement WAEs but slightly modifying architectures for the three datasets according to their characteristics.

In Digits dataset, the encoder and decoder are built with FC layers. The encoder consists of two FC layers with the size of 400 and 20 respectively. Accordingly, the decoder consists of two FC layers with the size of 400 and 3072 respectively. The discriminator consists of two FC layers with the size of 128 and 1 respectively. The architecture of is shown in Fig. 1 (a).

In CIFAR-10-C [2], the encoder begins with four convolutional layers with the channels of $\{16, 32, 32, 32\}$. And two FC layers with the size of 1024 and 512 are followed. Accordingly, the decoder begins with two FC layers with the size of 512 and 1024 respectively. And four deconvolution layers with the channels of $\{32, 32, 16, 3\}$ are followed. Each layer is followed by BN [3] except for the final layer of the decoder. The discriminator consists of two FC layers with the size of 128 and 1 respectively. The architecture is shown in Fig. 1 (b).

In SYTHIA [7], the encoder begins with three convolutional layers with the channels of $\{32, 64, 128\}$. And two FC layers with the size of $\{3840, 512\}$ are followed. Accordingly, the decoder begins with two FC layers with the size of $\{512, 3840\}$. And three deconvolution layers with the channels of $\{64, 32, 3\}$ are followed. Each layer is followed by BN [3] except for the final layer of the decoder. The discriminator consists of three FC layers with the size of $\{512, 512, 1\}$. The architecture is shown in Fig. 1 (c).

We apply the Adam optimizer in training WAEs. The learning rate is 0.001 for Digits and 0.0001 for both CIFAR-10-C and SYTHIA. The training epoches is 20 for Digits, 100 for CIFAR-10-C [2], and 200 for SYTHIA [7].

2. Additional Experimental Results

2.1. Ablation Study

Validation of meta-learning scheme: The results of four kinds of unseen corruptions are shown in Fig. 2. As seen, M-ADA can significantly reduce variance and yield better performance across all levels of severity. The experimental results prove that the meta-learning scheme plays a key role to improve the training stability and classification accuracy. This is extremely important when performing adversarial domain augmentation in challenging conditions.

Hyper-parameter tuning of K, α, and β: We study the effect of three important hyper-parameters of M-ADA: the number of augmented domains (K), the distance between the source and augmented domain in the embedding space (α), and the deviation between the source and augmented

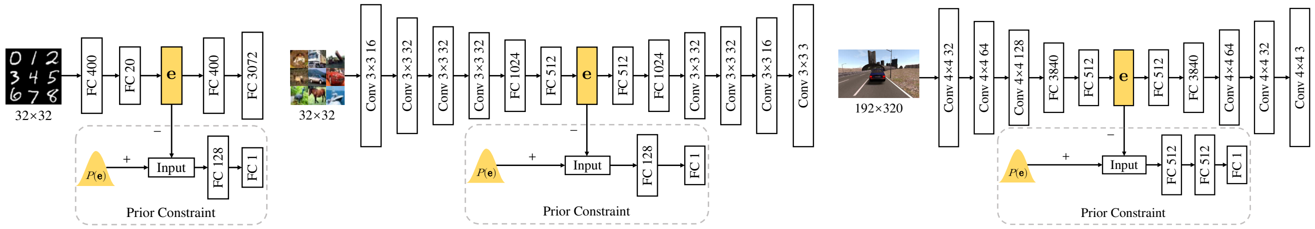

Figure 1. Architectures of WAEs. **From left to right:** (a) WAE for *Digits* ; (b) WAE for *CIFAR-10-C* [2]; and (c) WAE for *SYTHIA* [7]. Note that "**+**": positive samples for discriminator; "**-**": negative samples for discriminator.

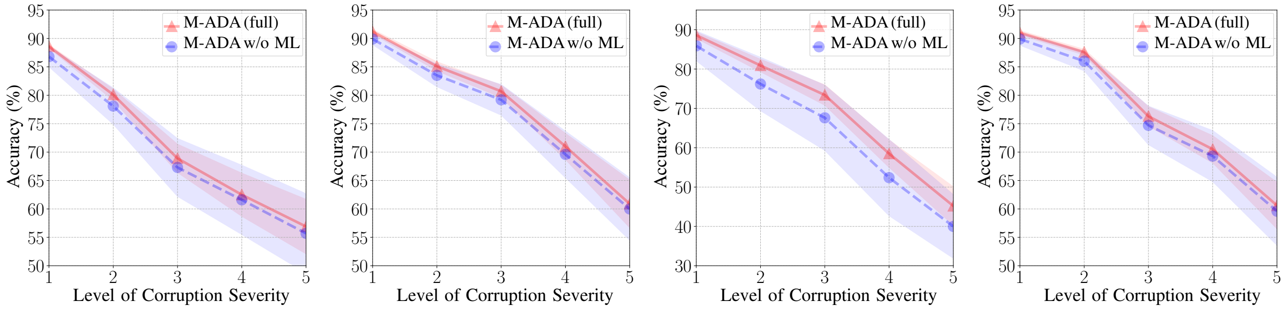

Figure 2. Validation of meta-learning scheme. Five levels of severity are evaluated on each unseen corruption. **From left to right:** (a) *Gaussian Noise*; (b) *Speckle Noise* ; (c) *Impulse Noise*; and (d) *Shot Noise*.

domain (β). We plot the accuracy curve under different K, α, and β in Fig. 3. In Fig. 3 (left), we find that the accuracy reaches the summit when $K = 3$ and keeps falling with K increasing. This is due to the fact that excessive adversarial samples above a certain threshold will increase the instability and degrade the robustness of the model. Since the distance between the augmented and source domain increases as K increases, a large K may break down the constraint of semantic consistency yielding inferior model training. In Fig. 3 (middle), we find that the accuracy reaches the summit when $\alpha = 1.0$ and keeps falling with α increasing. This is because large α will make the source and augmented domain too close in the embedding space, yielding limited domain transportation. In Fig. 3 (right), we observe that the accuracy reaches the summit when $\beta = 2.0 \times 10^3$ and drops slightly when β increases. This is because large β will produce domains too far way from the source $\mathcal{S}$ and even reach out of the manifold in embedding space.

2.2. Comparison on CIFAR-10-C

We train all models on clean data, *i.e.*, CIFAR-10, and test them on corruption data, *i.e., CIFAR-10-C*. In this case, there are totally 19 unseen testing domains. We present the result of each corruption with the highest severity in Tab. 1. We observe that M-ADA substantially outperforms other methods on most corruptions. Specially, in several corruptions such as *Frost*, *Glass blur*, *Gaussian blur*, *Pixelate*, and corruptions related with *Noise*, M-ADA outperforms ERM [4] with more than 10%. More importantly, M-ADA has the lowest values on mCE and relative mCE, indicating its strong robustness against image corruptions.

2.3. Comparison of Different $\mathcal{L}_{\text{relax}}$

WAEs employ Wasserstein metric to measure the distribution distance between the input and reconstruction, which is desirable for domain augmentation. So the reconstruction error $\mathcal{L}_{\text{relax}} = \|\mathbf{x}^+ - V(\mathbf{x}^+)\|^2$ indicates if $\mathbf{x}^+$ lie in the same distribution as $\mathbf{x}$. Using WAE instead of vanilla AE is the key design to achieve this goal (Table 2). Additionally, our experiments indicate that $\|V(\mathbf{x}) - V(\mathbf{x}^+)\|^2$ has better relaxation effect and yields improved accuracy. The distribution distance is more reliable in the reconstruction space where Wasserstein prior has been applied.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, pages 770–778, 2016.

[2] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *ICLR*, 2019.

[3] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015.

[4] Vladimir Koltchinskii. *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems: Ecole d'Eté de Probabilités de Saint-Flour XXXVIII-2008*, volume 2033. Springer Science & Business Media, 2011.

	Weather			Blur					Noise			
	Fog	Snow	Frost	Zoom	Defocus	Glass	Gaussian	Motion	Speckle	Shot	Impulse	Gaussian
ERM [4]	65.92	74.36	61.57	59.97	53.71	49.44	30.74	63.81	41.31	35.41	25.65	29.01
CCSA [6]	66.94	74.55	61.49	61.96	56.11	48.46	32.22	**64.73**	40.12	33.79	24.56	27.85
d-SNE [10]	65.99	75.46	62.25	58.47	53.71	50.48	33.06	63.70	45.30	39.93	27.95	34.02
GUD [9]	68.29	76.75	69.94	62.95	56.41	53.45	38.33	63.93	38.45	36.87	22.26	32.43
M-ADA w/o $\mathcal{L}_{\text{relax}}$	66.99	80.09	74.93	54.15	44.67	60.57	30.53	57.06	59.88	59.18	43.46	55.07
M-ADA w/o ML	67.68	**80.91**	76.20	65.70	56.87	**62.14**	41.20	63.86	60.01	59.63	40.04	55.70
M-ADA (full)	**69.36**	80.59	**76.66**	**68.04**	**61.18**	61.59	**47.34**	64.23	**60.88**	**60.58**	**45.18**	**56.88**

	Digital									
	Jpeg	Pixelate	Spatter	Elastic	Brightness	Saturate	Contrast	Avg.	mCE	RmCE
ERM [4]	69.90	41.07	75.36	72.40	**91.25**	89.09	**36.87**	56.15	1.00	1.00
CCSA [6]	69.68	40.94	77.91	72.36	91.00	**89.42**	35.83	56.31	0.99	0.99
d-SNE [10]	70.20	38.46	73.40	73.33	90.90	89.27	36.28	56.96	0.99	1.00
GUD [9]	74.22	**53.34**	80.27	74.64	89.91	82.91	31.55	58.26	0.97	0.95
M-ADA w/o $\mathcal{L}_{\text{relax}}$	76.45	53.13	80.75	73.85	90.86	87.01	27.83	61.92	0.90	0.86
M-ADA w/o ML	**77.62**	52.49	**81.02**	75.54	90.69	86.58	26.30	64.22	0.85	0.80
M-ADA (full)	77.14	52.25	80.62	**75.61**	90.78	87.62	29.71	**65.59**	**0.82**	**0.77**

Table 1. Full version of Tab. 4 in main paper. The models are generalized from clean data to different corruptions. We report the classification accuracy (%) of 19 corruptions under the corruption level of "5" (severest). We also report the mean Corruption Error (mCE) and relative mCE (RmCE) in the last two columns. The lower the better for mCE and RmCE.

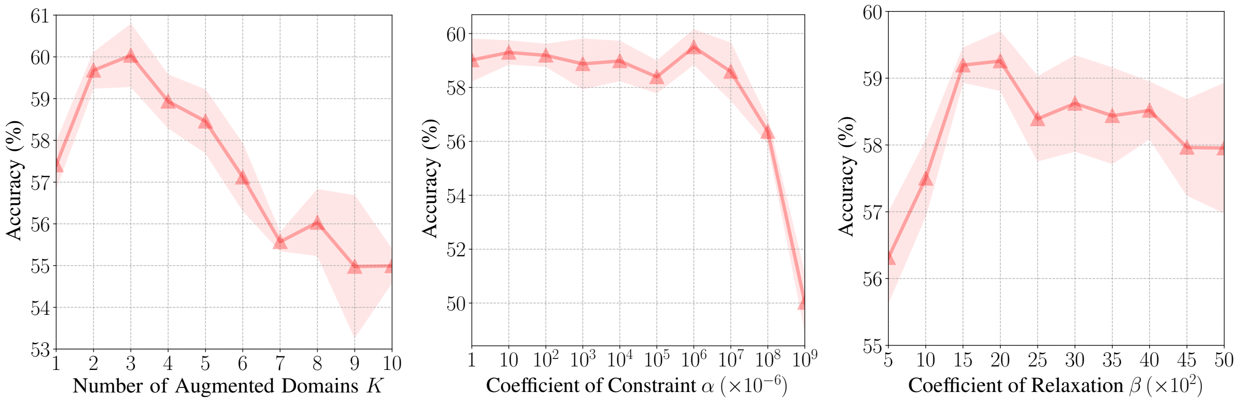

Figure 3. Hyper-parameter tuning of K, α, and β. We set $K = 3$, $\alpha = 1.0$, and $\beta = 2.0 \times 10^3$ according to the best accuracy.

	$\|\mathbf{x} - \mathbf{x}^{+}\|^2$	Vanilla AE	WAE
Digits	55.71%	58.67%	59.49%
CIFAR-10-C	62.03%	63.34%	65.59%

Table 2. Accuracy comparison using different relaxation terms.

[5] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[6] Saeid Motiian, Marco Piccirilli, Donald A. Adjeroh, and Gianfranco Doretto. Unified Deep Supervised Domain Adaptation and Generalization. In *ICCV*, pages 5715–5725, 2017.

[7] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, pages 3234–3243, 2016.

[8] I Tolstikhin, O Bousquet, S Gelly, and B Schölkopf. Wasserstein auto-encoders. In *ICLR*, 2018.

[9] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. In *NeurIPS*, pages 5334–5344, 2018.

[10] Xiang Xu, Xiong Zhou, Ragav Venkatesan, Gurumurthy Swaminathan, and Orchid Majumder. d-sne: Domain adaptation using stochastic neighborhood embedding. In *CVPR*, pages 2497–2506, 2019.

[11] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016.