A Fourier-based Framework for Domain Generalization: Supplementary Material

Qinwei Xu[1] Ruipeng Zhang[1] Ya Zhang[1,2] ✉ Yanfeng Wang[1,2] Qi Tian[3]

[1] Cooperative Medianet Innovation Center, Shanghai Jiao Tong University
[2] Shanghai Artificial Intelligence Laboratory [3] Huawei Cloud & AI

{qinweixu, zhangrp, ya_zhang, wangyanfeng}@sjtu.edu.cn, tian.qi1@huawei.com

1. More examples of amplitude-only and phase-only reconstruction

We present more examples of amplitude-only reconstructed images, phase-only reconstructed images, as well as their corresponding original images in Fig. 1. As we can see, the general visual structures of different objects are preserved in the phase-only reconstructed images, while the amplitude-only reconstructed images mainly contain low-level statistics without clear semantic meanings.

2. Implementation details

2.1. Experiments on three DG benchmarks

Network details: We closely follow the implementations of [2, 16]. For Digits-DG, we use the same backbone network as [16]. For PACS and OfficeHome, we use ImageNet pre-trained ResNet18 and ResNet50 as the backbone.

Optimization details: For all the datasets, we train our network using the nesterov-momentum SGD with a momentum of 0.9 and weight decay of 5e-4. For Digits-DG and PACS, we train the model for 50 epochs. For OfficeHome, we train the model for 30 epochs. The initial learning rate for Digits-DG is 0.05 and decayed by 0.1 every 20 epochs. For PACS and OfficeHome, the initial learning rate is 0.001 and decayed by 0.1 at 80% of the total epochs. The batch size is set to 128 for Digits-DG and 16 for PACS and OfficeHome.

Data Augmentation details: The term "augmentation" here refers to the typical data augmentation techniques. For Digits-DG, we only use simple augmentations composed of random flipping and resizing. The input image size is 32×32. For PACS and OfficeHome, we use the standard augmentation protocol as in [2], which consists of randomly cropping the images to retain between 80% to 100%, randomly applied horizontal flipping and randomly color jittering with magnitude of 0.4. The input image size is 224×224.

Model-specific details: For all experiments, we set the momentum m for the teacher model to 0.9995 and the temperature T to 10. The weight β of the consistency loss is set to 2 for Digits-DG and PACS, and 200 for OfficeHome. We also use a sigmoid ramp-up [13] for β with a length of 5 epochs. The augmentation strength of AM is chosen as 1.0 for Digits-DG and PACS, and 0.2 for OfficeHome. For the convenience of applying amplitude mixing, when an origi-

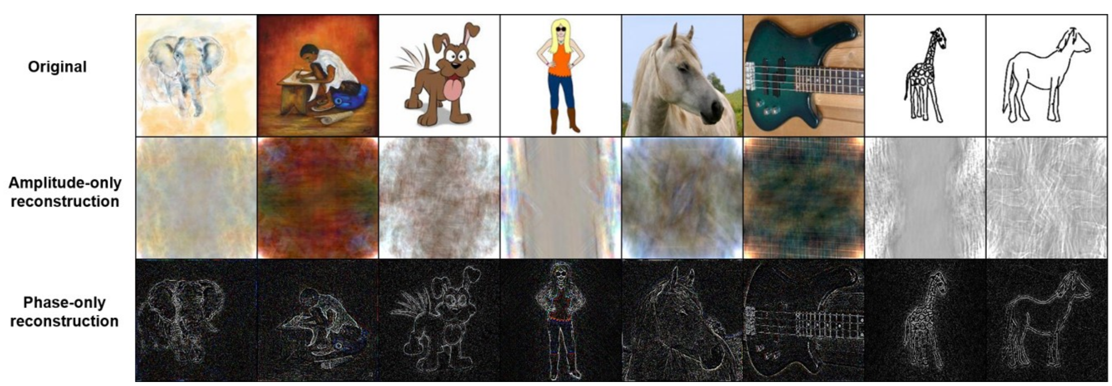

Figure 1. Examples of amplitude-only and phase-only reconstruction.

Table 1. Leave-one-domain-out results on Digits-DG.

Methods	MNIST	MNIST-M	SVHN	SYN	Avg.
Jigen [2]	96.5	61.4	63.7	74.0	73.9
L2A-OT [17]	96.7	63.9	68.6	83.2	78.1
DeepAll [16]	95.8±0.3	58.8±0.5	61.7±0.5	78.6±0.6	73.7
CCSA [9]	95.2±0.2	58.2±0.6	65.5±0.2	79.1±0.8	74.5
MMD-AAE [7]	96.5±0.1	58.4±0.1	65.0±0.1	78.4±0.2	74.6
CrossGrad [11]	96.7±0.1	61.1±0.5	65.3±0.5	80.2±0.2	75.8
DDAIG [16]	96.6±0.2	64.1±0.4	68.6±0.6	81.0±0.5	77.6
FACT (*ours*)	**97.9±0.2**	**65.6±0.4**	**72.4±0.7**	**90.3±0.1**	**81.5**

Table 2. Leave-one-domain-out results on OfficeHome.

Methods	Art	Clipart	Product	Real	Avg.
Jigen [2]	53.04	47.51	71.47	72.79	61.20
RSC [5]	58.42	47.90	71.63	74.54	63.12
L2A-OT [16]	60.60	50.10	74.80	77.00	65.60
DeepAll	57.88±0.20	52.72±0.50	73.50±0.30	74.80±0.10	64.72
CCSA [9]	59.90±0.30	49.90±0.40	74.10±0.20	75.70±0.20	64.90
MMD-AAE [7]	56.50±0.40	47.30±0.30	72.10±0.30	74.80±0.20	62.70
CrossGrad [11]	58.40±0.70	49.40±0.40	73.90±0.20	75.80±0.10	64.40
DDAIG [16]	59.20±0.10	52.30±0.30	**74.60±0.30**	76.00±0.10	65.50
Jigen (*our imple.*)	57.95±0.62	49.21±0.35	72.61±0.45	74.90±0.25	63.67
RSC (*our imple.*)	57.67±0.51	48.48±0.44	72.62±0.31	74.16±0.48	63.23
FACT ((ours))	**60.34±0.11**	**54.85±0.37**	74.48±0.13	**76.55±0.10**	**66.56**

nal image is generated, we sample another image from the whole dataset, and then mixing the amplitude spectrums of these two images to produce two augmented counterparts. We then pass these two original images as well as their augmented counterparts to the model. Therefore, within a single iteration, the number of input images is $4\times$ batch size.

2.2. Single domain evaluations on PACS

Here we present the experimental details about the single domain evaluations in the **Discussion** of main paper. Specifically, we train ResNet18 using the original images, phase-only reconstructed images and amplitude-only reconstructed images, respectively, on the training splits from a single domain, and then evaluate on the validation splits of all domains. The phase-only reconstructed images are generated by setting the amplitude component as a constant of 20000 when reconstructing the images via inverse FFT. Through this way, the amplitude information is eliminated in the phase-only images so that the model only relies on the phase information for classification. Similar operations are applied to get the amplitude-only reconstructed images. Since the distributions of phase-only and amplitude-only images differ drastically from the original images, we train the networks from scratch in order to remove the impacts of ImageNet pre-training. This ensures a fair comparison between the performances of the original images, phase-only reconstructed images and amplitude-only reconstructed images. Other basic settings are kept the same with the above DG experiments on PACS.

3. Complete results on DG benchmarks

We present the complete results in the form of mean±std on Digits-DG, PACS, OfficeHome in Table 1, Table 2, Table 3, respectively. Note that unlike most previous work, Dou *et al.* directly report the best accuracy on the target domain [3]. For fair comparison, we also report our results under this protocol in Table 3.

4. Additional results of AlexNet

To further verify the flexibility of our framework with different backbone networks, we experiment on PACS by incorporating AlexNet into our FACT framework. Specifically, we use the Caffe-version of ImageNet pretrained AlexNet[1]. We train the network with nesterov-momentum SGD, batch size of 32 and weight decay of 5e-4 for 50 epochs. The initial learning rate is 0.001 and decayed by 0.1 at 80% of the total epochs. We also use the standard

[1] https://drive.google.com/file/d/1wUJTH1Joq2KAgrUDeKJghP1Wf7Q9w4z-/view

Table 3. Leave-one-domain-out results on PACS. †: results are reported based on the best models on test splits.

Methods	Art	Cartoon	Photo	Sketch	Avg.
		ResNet18			
JiGen [2]	79.42	75.25	96.03	71.35	80.51
Epi-FCR [6]	82.10	77.00	93.90	73.00	81.50
MMLD [8]	81.28	77.16	96.09	72.29	81.83
InfoDrop [12]	80.27	76.54	96.11	76.38	82.33
L2A-OT [17]	83.30	78.20	96.20	73.60	82.80
RSC [5]	83.43	80.31	95.99	80.85	85.15
DeepAll	77.63±0.84	76.77±0.33	95.85±0.20	69.50±1.26	79.94
MetaReg [1]	83.70±0.19	77.20±0.31	95.50±0.24	70.30±0.28	81.70
DDAIG [16]	84.20±0.30	78.10±0.60	95.30±0.40	74.70±0.80	83.10
CSD [10]	78.90±1.10	75.80±1.00	94.10±0.20	76.70±1.20	81.40
EISNet [14]	81.89±0.88	76.44±0.31	**95.93±0.06**	74.33±1.37	82.15
RSC (*our imple.*)	80.55±0.78	**78.60±0.38**	94.43±0.01	76.02±1.68	82.40
FACT (*ours*)	**85.37±0.29**	78.38±0.29	95.15±0.26	**79.15±0.69**	**84.51**
MASF [3]†	80.29±0.18	77.17±0.08	94.99±0.09	71.69±0.22	81.04
FACT (*ours*)†	**85.90±0.27**	**79.35±0.03**	**96.61±0.17**	**80.89±0.26**	**85.69**
		ResNet50			
RSC [5]	87.89	82.16	97.92	83.35	87.83
DeepAll [5]	84.94±0.66	76.98±1.13	**97.64±0.10**	76.75±0.41	84.08
MetaReg [1]	87.20±0.13	79.20±0.27	97.60±0.31	70.30±0.18	83.60
EISNet [14]	86.64±1.41	81.53±0.64	97.11±0.40	78.07±1.43	85.84
RSC (*our imple.*)	83.92±1.02	79.52±2.17	95.15±0.10	82.20±1.28	85.20
FACT (*ours*)	**89.63±0.51**	**81.77±0.19**	96.75±0.10	**84.46±0.78**	**88.15**
MASF [3]†	82.89±0.16	80.49±0.21	95.01±0.10	72.29±0.15	82.67
FACT (*ours*)†	**90.89±0.19**	**83.65±0.12**	**97.78±0.05**	**86.17±0.14**	**89.62**

Table 4. Leave-one-domain-out results on PACS with AlexNet as backbone. †: results are reported based on the best models on test splits.

Methods	Art	Cartoon	Photo	Sketch	Avg.
DeepAll [2]	66.68	69.41	89.98	60.02	71.52
JiGen [2]	67.63	71.71	89.00	65.18	73.38
Epi-FCR [6]	64.70	72.30	86.10	65.00	72.00
MMLD [8]	69.27	72.83	88.98	66.44	74.38
RSC [5]	71.62	75.11	90.88	66.62	76.05
DeepAll	65.60±0.34	70.88±0.29	87.16±0.19	66.43±0.68	72.52
EISNet [14]	70.38±0.37	71.59±1.32	91.20±0.00	70.25±1.36	75.86
MetaVIB [4]	71.94±0.34	**73.17±0.21**	**91.93±0.23**	65.94±0.24	75.74
FACT (*Ours*)	**75.50±0.52**	71.16±0.24	89.10±0.20	**71.65±0.39**	**76.85**
MASF [3]†	70.35±0.33	72.46±0.19	**90.68±0.12**	67.33±0.12	75.21
FACT (*Ours*)†	**76.46±0.28**	**72.57±0.39**	90.24±0.26	**73.56±0.08**	**78.21**

augmentation protocol as in [2], which consists of random resized cropping, horizontal flipping and color jittering. We set the momentum m for the teacher model to 0.9995 and the temperature T to 10. The weight β of the consistency loss is set to 2. We also use a sigmoid ramp-up [13] for β with a length of 5 epochs. The augmentation strength of AM is chosen as 1.0.

The results are presented in Table 4, which have shown that FACT with AlexNet as backbone is still able to outperform the state-of-the-arts, by exceeding both EISNet [14] and MetaVIB [4] by around 1% in terms of the average performance. The largest performance gain of FACT comes from the generalization tasks on art and sketch domain, both of which bear a large distribution shift from the pre-training domain of ImageNet. This demonstrates the effectiveness of our method when generalizing to unknown out-of-domains.

5. Variants of Fourier data augmentation

The form of Fourier-based data augmentation is not restricted to amplitude swap (AS) or amplitude mix (AM) mentioned in the paper. Here we further propose three more variants of Fourier-based data augmentation:

- **Amplitude CutMix (AC)**: We define a pixel-level mixing strategy based on the CutMix [15]. Specifically, we firstly sample a binary mask $\mathbf{s}$ from Bernoulli distribution in the shape of the input image. We then linearly mix the amplitude components of two images to generate the augmented images:

$$\hat{\mathcal{A}}(x_i^k) = (1 - \mathbf{s}) \cdot \mathcal{A}(x_i^k) + \mathbf{s} \cdot \mathcal{A}(x_{i'}^{k'}) \tag{1}$$

  where $[\cdot]$ denotes element-wise prodution. Note that AS can be seen as a special case of AC, where the entries of $\mathbf{s}$ in the center area are fixed as 1 and 0 for the remaining area.

- **Amplitude Jittering (AJ)**: We can directly perturb the amplitude information in an image with random noises. Suppose a Gaussian noise $\mathbf{n} \sim \mathcal{N}(0, \sigma)$, we can generate the perturbed amplitude spectrum as:

$$\hat{\mathcal{A}}(x_i^k) = (1 + \mathbf{n}) \cdot \mathcal{A}(x_i^k) \tag{2}$$

  where the strength of noise jittering is controlled by σ.

- **Amplitude Elimination (AE)**: The above Fourier-based data augmentation are all based on amplitude perturbation. Nevertheless, we can directly use the phase-only reconstruction as augmented versions of the original images. In this way, the amplitude information in original images is completely eliminated.

We report the performances of all different augmentations incorporated in the baseline DeepAll and our FACT framework in Table 5 and 6 respectively. Among all the augmentation types, AM performs best in terms of the average performance. AJ with a larger σ ($0.5 \sim 0.7$) can also reach a relatively good performance. Interestingly, when incorporated in FACT, the performance of AJ shows a clear trade-off on different domains as the parameter σ changes. With a relatively smaller σ ($0.1 \sim 0.3$), AJ performs better when generalizing to the cartoon and photo domain, while with a relatively larger σ ($0.5 \sim 0.7$), AJ is better at generalizing to the art and sketch domain. Further increasing the value of σ (*e.g.* σ = 0.9) will degrade the performance, mainly due to the augmentation strategy is too aggressive. Nevertheless, the AM strategy shows a better trade-off on all the four cases, thus is a more general choice than AJ.

On the other hand, AC gains a moderate performance among all the perturbation-based variants, which means a simple linear interpolation strategy of AM is a better choice. However, the AE strategy which directly eliminates the amplitude information performs relatively worse than all the other augmentation variants. This may attribute to the large distribution discrepancy of the phase-only reconstructed images compared with the original image domain, which may increase the difficulty of model learning. Another issue is that the phase-only images also follows a different distribution with the pre-trained dataset ImageNet, which means the model may benefit less from ImageNet pre-training. Furthermore, as we mentioned in the **Discussion** of main paper, desirable performances on domains like photo may also require the presence of amplitude information, as these domains contain rich low-level details, therefore totally eliminating amplitude information and overly highlighting phase information may not be a good choice.

All the above augmentation types are just part of the instantiations of Fourier-based data augmentation. In the future, other more effective instantiations of Fourier-based data augmentation may be proposed. Composition of different augmentation operations is also a topic to be studied.

6. Sensitivity to different hyperparameters

In this section we carry out detailed ablation studies about the sensitivity to different hyperparameters related to our method. If not specifically mentioned, all the experiments below are conducted based on the ResNet18 backbone on PACS. When investigating the sensitivity to a specific hyperparameter, other hyperparameters are fixed to their default values, *i.e.*, m = 0.9995, T = 10, $(\eta, \beta) = (1.0, 2)$ for PACS and $(\eta, \beta) = (0.2, 200)$ for OfficeHome.

6.1. Sensitivity to the momentum m

The results are shown in Table 7. Basically, a larger momentum value is expected to enhance the effect of the teacher model, thus induce a better performance. Therefore, in all the remaining experiments, we set the momentum value m to 0.9995.

6.2. Sensitivity to the temperature T

The results are shown in Table 8. FACT is not sensitive to the change of the temperature value. Generally, a temperature $T > 1$ would lead a good performance. More specifically, a relatively smaller value of T (*e.g.* $T = 1 \sim 5$) would result in a better performance on the sketch domain, but at the sacrifice of the performances on other target domains. While a relatively larger value of T (*e.g.* $T = 10$)

Table 5. Leave-one-domain-out results on PACS with different variants of Fourier-based data augmentation. The backbone network is ResNet18. The performances are reported from *DeepAll* trained with the augmented images.

Augmentation	Art	Cartoon	Photo	Sketch	Avg.
AS-partial	82.00±0.13	76.19±0.15	93.89±0.18	77.27±1.18	82.34
AS-full	83.50±0.73	76.07±0.30	94.49±0.50	77.13±2.19	82.80
AC	82.63±0.50	77.15±0.48	94.94±0.03	75.01±0.70	82.43
AJ ($\sigma = 0.1$)	80.96±0.97	76.20±0.59	94.55±0.36	76.61±0.13	82.08
AJ ($\sigma = 0.3$)	81.18±0.22	77.71±0.91	95.22±0.26	79.10±0.04	83.30
AJ ($\sigma = 0.5$)	81.62±0.65	77.34±0.67	94.49±0.40	79.24±1.92	83.17
AJ ($\sigma = 0.7$)	80.71±0.28	**77.72±0.53**	94.96±0.32	**80.22±0.49**	83.40
AJ ($\sigma = 0.9$)	80.50±0.66	76.58±1.07	94.01±0.21	78.12±1.30	82.30
AE	80.08±0.47	76.12±0.81	93.57±0.45	78.50±2.26	82.07
AM	**83.90±0.50**	76.95±0.45	**95.55±0.12**	77.36±0.71	**83.44**

Table 6. Leave-one-domain-out results on PACS with different variants of Fourier-based data augmentation. The backbone network is ResNet18. The performances are reported from *FACT* trained with the augmented images.

Augmentation	Art	Cartoon	Photo	Sketch	Avg.
AS-partial	81.61±0.06	76.95±0.14	93.83±0.61	78.30±0.80	82.67
AS-full	83.46±0.28	77.37±0.86	94.10±0.34	78.63±0.61	83.39
AC	83.32±0.79	77.79±0.38	94.70±0.14	79.24±1.04	83.76
AJ ($\sigma = 0.1$)	80.66±0.49	**78.45±0.99**	**95.45±0.11**	77.59±0.60	83.04
AJ ($\sigma = 0.3$)	82.62±0.68	77.94±0.82	95.24±0.27	78.74±0.63	83.64
AJ ($\sigma = 0.5$)	82.47±0.85	77.70±0.64	95.12±0.43	80.83±0.54	84.03
AJ ($\sigma = 0.7$)	83.09±0.12	77.22±0.88	94.97±0.13	80.80±1.16	84.02
AJ ($\sigma = 0.9$)	82.29±0.30	77.23±0.87	94.54±0.34	**81.16±0.35**	83.80
AE	81.43±0.42	76.17±0.22	93.78±0.47	79.57±0.77	82.74
AM	**85.37±0.29**	78.38±0.29	95.15±0.26	79.15±0.69	**84.51**

Table 7. Sensitivity to the momentum m. Results are reported based on the ResNet18 backbone on PACS.

Momentum	Art	Cartoon	Photo	Sketch	Avg.
$m = 0.9$	84.41±0.65	76.36±0.84	94.64±0.16	78.99±1.11	83.60
$m = 0.99$	84.07±1.05	77.22±0.88	94.61±0.28	**79.48±0.98**	83.84
$m = 0.999$	84.61±0.06	77.84±0.35	95.14±0.35	79.05±0.71	84.16
$m = 0.9995$	**85.37±0.29**	**78.38±0.29**	**95.15±0.26**	79.15±0.69	**84.51**

can reach a better trade-off between all the four leave-one-domain-out cases. Continuously increase the value of T (*e.g.* $T = 20$) would degrades the average performance again, mainly because the discriminality in predictions is over-smoothed, thus confusing the model in decision making. A similar trend can also be found on other datasets. Therefore, for convenience, we set the temperature $T = 10$ for all the experiments.

6.3. Sensitivity to the perturbation strength η

Recall that the mix coefficient λ in the AM strategy are sample from a uniform distribution $U(0, \eta)$, thus the value of η controls the strength of the amplitude perturbation. We present the impact of different values of η on PACS and OfficeHome in Table 9. Note that when $\eta = 0$, no Fourier-based data augmentation is applied, and the difference between the original image and its augmented counterpart[2] is only induced by the randomness in basic augmentations (*i.e.*, flipping, cropping, and color jittering).

As we can see, the value of η has different effects on different datasets. On PACS, a larger η would result in a better performance, and the best performance is achieved at

[2]More exactly, "the original image and its augmented counterpart" are two differently augmented versions of the same image.

Table 8. Sensitivity to temperature T. Results are reported based on the ResNet18 backbone on PACS.

Temperature	Art	Cartoon	Photo	Sketch	Avg.
$T = 1$	84.06±0.02	76.52±0.28	93.47±0.12	**81.04±0.64**	83.77
$T = 2$	84.23±0.52	78.11±0.12	94.05±0.29	80.20±0.92	84.15
$T = 5$	84.46±0.29	77.87±0.37	95.11±0.25	80.49±0.14	84.48
$T = 10$	**85.37±0.29**	**78.38±0.29**	**95.15±0.26**	79.15±0.69	**84.51**
$T = 20$	84.91±0.11	77.96±0.41	94.88±0.21	79.16±0.52	84.23

Table 9. Sensitivity to the perturbation strength η. Results are reported based on the ResNet18 backbone. For PACS, the consistency loss weight β is fixed as 2, and for OfficeHome, β is fixed as 200.

PACS	Art	Cartoon	Photo	Sketch	Avg.
$(\eta = 0.0, \beta = 2)$	82.68±0.44	78.06±0.39	95.35±0.44	74.76±0.67	82.71
$(\eta = 0.2, \beta = 2)$	82.29±0.51	77.73±0.68	**96.33±0.25**	75.10±1.33	82.86
$(\eta = 0.4, \beta = 2)$	83.51±0.38	77.68±0.46	96.23±0.08	76.48±0.72	83.48
$(\eta = 0.6, \beta = 2)$	83.96±0.51	**78.57±0.16**	95.95±0.11	78.06±0.85	84.14
$(\eta = 0.8, \beta = 2)$	84.18±0.15	78.33±0.41	95.72±0.23	78.16±0.10	84.10
$(\eta = 1.0, \beta = 2)$	**85.37±0.29**	78.38±0.29	95.15±0.26	**79.15±0.69**	**84.51**
OfficeHome	Art	Clipart	Product	Real	Avg.
$(\eta = 0.0, \beta = 200)$	59.18±0.40	54.03±0.61	73.91±0.22	76.10±0.10	65.81
$(\eta = 0.2, \beta = 200)$	**60.34±0.11**	54.85±0.37	**74.48±0.13**	**76.55±0.10**	**66.56**
$(\eta = 0.4, \beta = 200)$	59.65±0.34	55.09±0.21	73.87±0.32	76.19±0.18	66.20
$(\eta = 0.6, \beta = 200)$	58.51±0.38	**55.10±0.13**	73.22±0.23	75.48±0.25	65.58
$(\eta = 0.8, \beta = 200)$	57.13±0.63	55.03±0.58	72.93±0.15	74.68±0.13	64.94
$(\eta = 1.0, \beta = 200)$	57.86±0.14	53.25±0.06	72.70±0.31	74.42±0.26	64.56

$\eta = 1.0$. While on OfficeHome, a smaller η does better and the best performance is achieved at $\eta = 0.2$. The different behavior of η on different datasets can be attribute to the different extent of domain discrepancy. On PACS, the discrepancy between the four domains is much larger than that on OfficeHome. If we treat the Fourier-based augmentation as a kind of regularization, then a larger η may induce an overly-regularized model for OfficeHome, considering a vanilla baseline can already perform well due to the small domain discrepancy. Therefore, a more aggressive augmentation strategy with a larger η will do better on PACS, while a more conservative augmentation strategy with a smaller η is more suitable for OfficeHome.

6.4. Sensitivity to the consistency loss weight β

The results of the impact of different values of the consistency loss weight β are shown in Table 10. Note that when $\beta = 0$, the model is trained without any consistency constraint. The average performance of FACT on PACS is quite stable when β is in the range of $1.0 \sim 20$, and the best performance is achieved at $\beta = 2.0$. A further smaller or larger value of β will either induce a too weak or too strong constraint. On the other hand, FACT performs better with a larger value of β on OfficeHome. The performance is stable when β is within the range $20 \sim 500$, and the best performance is achieved at $\beta = 200$ on OfficeHome.

An interesting finding is that there seems to be a trade-off between the value of η and β. A larger η together with a smaller β is better for PACS, while a smaller η together with a larger β is better for OfficeHome. It seems that η and β will compensate for each other in terms of regularization power. When equipped with a large η, using a large β may be too aggressive for the model to learn. On the other hand, when equipped with a small η, still using a small β may not fully develop the power of consistency constraint.

In conclusion, for datasets with a large domain discrepancy (*e.g.* PACS, Digits-DG), a larger value of η (*e.g.* $\eta = 1.0$) together with a smaller value of β (*e.g.* $\beta = 2.0$) is desired, while for datasets with a small domain discrepancy (*e.g.* OfficeHome), we suggest a smaller value of η (*e.g.* $\eta = 0.2$) together with a larger value of β (*e.g.* $\beta = 200$).

7. Sampling strategies for amplitude mixing

When implementing the amplitude mixing (AM) for a specific image, the amplitude spectrum from another image is sampled randomly from the whole dataset. Nevertheless, we can restrict the sampling image to be taken from the same or different domain. In other words, we can develop an intra-domain AM operation or an inter-domain AM operation. To study the impact of different sampling strategies,

Table 10. Sensitivity to the consistency loss weight β. Results are reported based on the ResNet18 backbone. For PACS, the perturbation strength η is fixed as 1.0, and for OfficeHome, η is fixed as 0.2.

PACS	Art	Cartoon	Photo	Sketch	Avg.
$(\eta = 1.0, \beta = 0.0)$	83.90±0.50	76.95±0.45	**95.55±0.12**	77.36±0.71	83.44
$(\eta = 1.0, \beta = 0.1)$	83.70±0.48	**78.54±0.91**	94.85±0.34	77.03±0.95	83.53
$(\eta = 1.0, \beta = 1.0)$	84.42±0.30	78.30±0.28	95.33±0.30	79.10±0.69	84.29
$(\eta = 1.0, \beta = 2.0)$	**85.37±0.29**	78.38±0.29	95.15±0.26	79.15±0.69	**84.51**
$(\eta = 1.0, \beta = 5.0)$	84.50±1.00	78.13±0.45	94.81±0.22	79.77±0.33	84.30
$(\eta = 1.0, \beta = 10)$	84.26±0.07	78.20±0.38	94.77±0.25	80.02±1.23	84.31
$(\eta = 1.0, \beta = 20)$	84.54±0.49	77.61±0.48	94.62±0.27	80.07±0.72	84.21
$(\eta = 1.0, \beta = 200)$	82.78±0.47	77.13±0.47	92.81±0.12	**80.23±0.70**	83.24
OfficeHome	Art	Clipart	Product	Real	Avg.
$(\eta = 0.2, \beta = 0.0)$	59.22±0.14	52.97±0.42	73.22±0.06	75.36±0.14	65.19
$(\eta = 0.2, \beta = 1.0)$	59.38±0.67	53.22±0.23	73.25±0.19	75.28±0.05	65.28
$(\eta = 0.2, \beta = 2.0)$	59.12±0.28	52.77±0.31	73.66±0.48	75.44±0.30	65.25
$(\eta = 0.2, \beta = 5.0)$	59.52±0.43	52.97±0.00	73.52±0.14	75.36±0.12	65.34
$(\eta = 0.2, \beta = 10)$	59.30±0.64	53.13±0.58	73.74±0.39	75.74±0.42	65.48
$(\eta = 0.2, \beta = 20)$	59.65±0.80	53.46±0.42	73.63±0.22	75.68±0.37	65.60
$(\eta = 0.2, \beta = 50)$	59.87±0.21	54.10±0.42	74.23±0.19	75.82±0.49	66.00
$(\eta = 0.2, \beta = 100)$	60.23±0.43	54.15±0.36	74.41±0.25	76.20±0.41	66.25
$(\eta = 0.2, \beta = 200)$	60.34±0.11	**54.85±0.37**	**74.48±0.13**	**76.55±0.10**	**66.56**
$(\eta = 0.2, \beta = 500)$	**60.39±0.61**	53.63±0.75	74.18±0.19	76.42±0.24	66.16

Table 11. Impact of different sampling strategies for amplitude mixing. Results are reported based on the ResNet18 backbone on PACS.

PACS	Art	Cartoon	Photo	Sketch	Avg.
Intra-domain	83.99±0.10	**78.84±0.38**	**95.39±0.12**	79.01±0.68	84.31
Inter-domain	85.09±0.20	77.43±0.49	94.71±0.17	**79.46±0.92**	84.17
Random	**85.37±0.29**	78.38±0.29	95.15±0.26	79.15±0.69	**84.51**

we carry out experiments by using only intra-domain or inter-domain AM operations. The results are shown in Table 11. As we can see, FACT is not sensitive to the sampling strategies. Whether intra-domain or inter-domain sampling strategy brings a good performance, and using a fully random strategy works best, perhaps because more augmented variants are included through fully random sampling.

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. 3

[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. 1, 2, 3

[3] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *Advances in Neural Information Processing Systems*, pages 6450–6461, 2019. 2, 3

[4] Yingjun Du, J. Xu, H. Xiong, Qiang Qiu, Xiantong Zhen, Cees G. M. Snoek, and L. Shao. Learning to learn with variational information bottleneck for domain generalization. In *ECCV*, 2020. 3

[5] Zeyi Huang, Haohan Wang, E. Xing, and D. Huang. Self-challenging improves cross-domain generalization. In *ECCV*, 2020. 2, 3

[6] 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

[7] 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. 2

[8] Toshihiko Matsuura and Tatsuya Harada. Domain generalization using a mixture of multiple latent domains. In *AAAI*, pages 11749–11756, 2020. 3

[9] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE Inter-

national Conference on Computer Vision, pages 5715–5725, 2017. 2

[10] Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. Efficient domain generalization via common-specific low-rank decomposition. In *International Conference on Machine Learning*, 2020. 3

[11] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. In *International Conference on Learning Representations*, 2018. 2

[12] Baifeng Shi, Dinghuai Zhang, Qi Dai, Zhanxing Zhu, Y. Mu, and J. Wang. Informative dropout for robust representation learning: A shape-bias perspective. In *International Conference on Machine Learning*, 2020. 3

[13] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017. 1, 3

[14] Shujun Wang, Lequan Yu, Caizi Li, Chi-Wing Fu, and Pheng-Ann Heng. Learning from extrinsic and intrinsicsupervisions for domain generalization. In *ECCV*, 2020. 3

[15] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *International Conference on Computer Vision (ICCV)*, 2019. 4

[16] Kaiyang Zhou, Yongxin Yang, Timothy M Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *AAAI*, pages 13025–13032, 2020. 1, 2, 3

[17] K. Zhou, Yongxin Yang, Timothy M. Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *ECCV*, 2020. 2, 3