Supplementary Material to 'Exact Feature Distribution Matching for Arbitrary Style Transfer and Domain Generalization'

Yabin Zhang[1], Minghan Li[1], Ruihuang Li[1], Kui Jia[2], Lei Zhang[1*]
[1]Hong Kong Polytechnic University [2]South China University of Technology
{csybzhang,csrhli,cslzhang}@comp.polyu.edu.hk, liminghan0330@gmail.com, kuijia@scut.edu.cn

The following materials are provided in this supplementary file:

- More visualization of high-order style statistics (*cf*. Section 3.2 in the main paper).
- More visualization of different methods on arbitrary style transfer (AST) (*cf*. Section 4.1 in the main paper).
- More domain generalization (DG) results on category classification (*cf*. Section 4.2 in the main paper).
- More discussions (*cf*. Section 4.3 in the main paper).

A. More visualization on high-order style statistics

The t-SNE [31] visualization of the fourth standardized moment-kurtosis [15, 34] and infinite norm are illustrated in Fig. A1a and Fig. A1b, respectively. These evidences, together with the third standardized moment-skewness in Figure 4 of the main paper, verify that the style information can be represented by high-order statistics beyond mean and standard deviation.

(a) Fourth standardized moment-kurtosis (b) Infinite norm

Figure A1. t-SNE [31] visualization of (a) the fourth standardized moment-kurtosis [15, 34] and (b) infinite norm, which clearly show that the style information can be represented by high-order statistics. The visualized features are extracted from the 1st residual block of ResNet-18 [11] trained on the dataset of four domains [18].

B. More visualization on AST

More visualization of style-transferred images on the classical style transfer [12] and the more challenging photorealistic style transfer [23] tasks are illustrated in Fig. A2 and Fig. A3, respectively. Our EFDM preserves the image structures and details more faithfully while transferring the style, leading to more stable performance.

C. More comprehensive DG results

We report more results of category classification by following the two experimental strategies in [9, 40], which are detailed as follows.

*Corresponding author

Figure A2. Additional results on style transfer [12]. Results of 'Gatys' [7] and 'CMD' [16] are obtained with official codes.

Results following DomainBed [9]. DomainBed [9] is a novel testbed for domain generalization tasks, where fair comparisons are expected under the strictly controlled datasets, network architectures, and model selection criteria[1]. The style data in EFDMix are introduced with the random shuffle strategy for its convenience. Results with different model selection criteria are illustrated in Tabs. A1 to A3. One can see that our EFDMix generally outperforms the competitors with different model selection criteria. Particularly, as illustrated in Tab. A3, our proposed EFDMix consistently outperforms the strong ERM baseline on all sub-tasks and achieves new state-of-the-art performance under the model selection strategy of test-domain validation set (oracle).

[1] https://github.com/facebookresearch/DomainBed

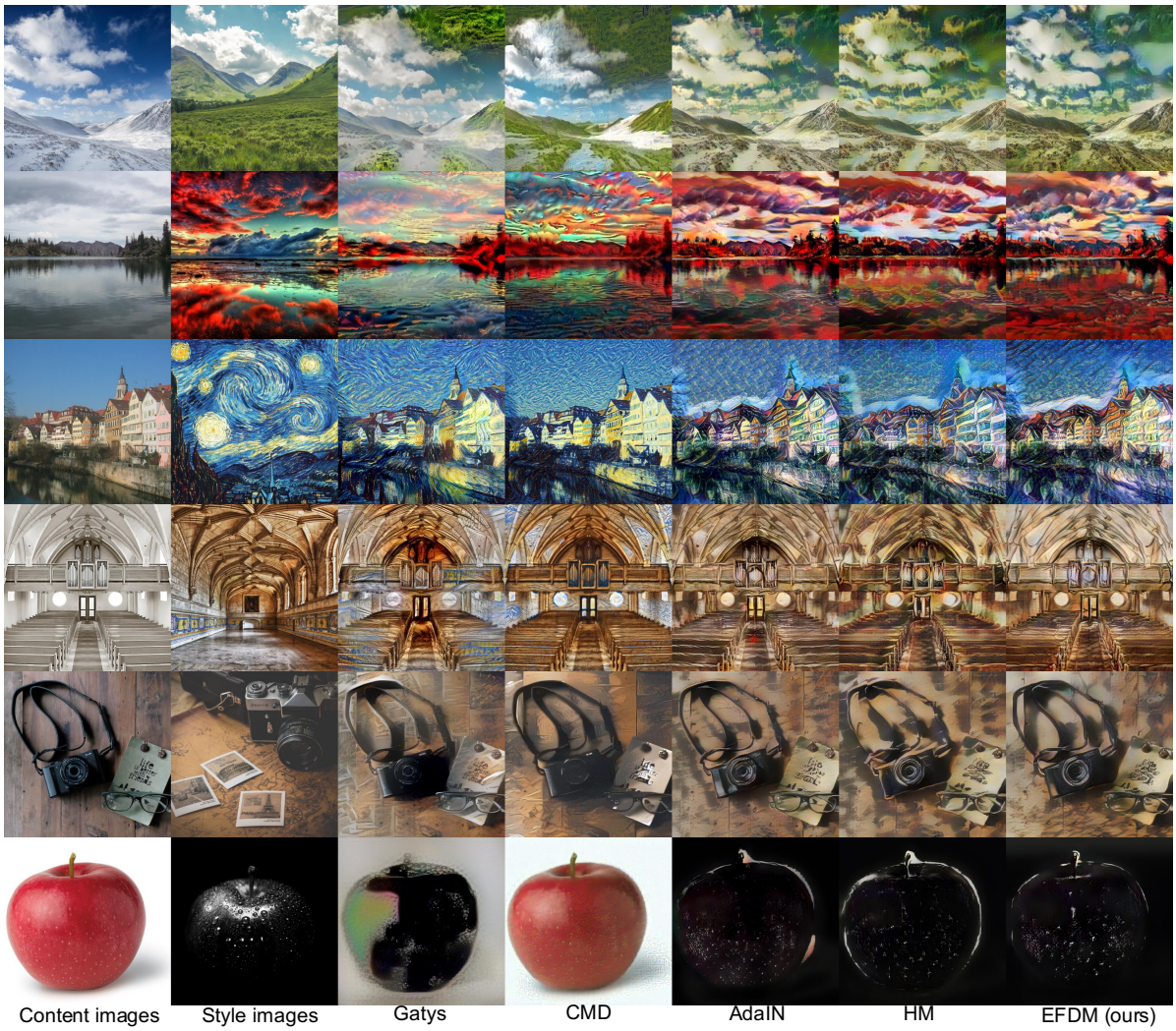

Figure A3. Additional results on photorealistic style transfer [23]. Results of 'Gatys' [7] and 'CMD' [16] are obtained with official codes.

Algorithm	A	C	P	S	Avg
ERM [32]	84.7 ± 0.4	**80.8** ± 0.6	97.2 ± 0.3	79.3 ± 1.0	85.5
IRM [1]	84.8 ± 1.3	76.4 ± 1.1	96.7 ± 0.6	76.1 ± 1.0	83.5
GroupDRO [27]	83.5 ± 0.9	79.1 ± 0.6	96.7 ± 0.3	78.3 ± 2.0	84.4
Mixup [36]	86.1 ± 0.5	78.9 ± 0.8	**97.6** ± 0.1	75.8 ± 1.8	84.6
MLDG [19]	85.5 ± 1.4	80.1 ± 1.7	97.4 ± 0.3	76.6 ± 1.1	84.9
CORAL [29]	**88.3** ± 0.2	80.0 ± 0.5	97.5 ± 0.3	78.8 ± 1.3	86.2
MMD [21]	86.1 ± 1.4	79.4 ± 0.9	96.6 ± 0.2	76.5 ± 0.5	84.6
DANN [6]	86.4 ± 0.8	77.4 ± 0.8	97.3 ± 0.4	73.5 ± 2.3	83.6
CDANN [22]	84.6 ± 1.8	75.5 ± 0.9	96.8 ± 0.3	73.5 ± 0.6	82.6
MTL [3]	87.5 ± 0.8	77.1 ± 0.5	96.4 ± 0.8	77.3 ± 1.8	84.6
SagNet [26]	87.4 ± 1.0	80.7 ± 0.6	97.1 ± 0.1	80.0 ± 0.4	**86.3**
ARM [37]	86.8 ± 0.6	76.8 ± 0.5	97.4 ± 0.3	79.3 ± 1.2	85.1
VREx [17]	86.0 ± 1.6	79.1 ± 0.6	96.9 ± 0.5	77.7 ± 1.7	84.9
RSC [13]	85.4 ± 0.8	79.7 ± 1.8	**97.6** ± 0.3	78.2 ± 1.2	85.2
EFDMix	86.7 ± 1.1	80.3 ± 0.9	96.3 ± 0.6	**80.8** ± 1.3	86.0

Table A1. Domain generalization results of category classification on PACS by following DomainBed [9], where the model selection strategy is training-domain validation set.

Algorithm	A	C	P	S	Avg
ERM [32]	83.2 ± 1.3	76.8 ± 1.7	**97.2** ± 0.3	74.8 ± 1.3	83.0
IRM [1]	81.7 ± 2.4	77.0 ± 1.3	96.3 ± 0.2	71.1 ± 2.2	81.5
GroupDRO [27]	84.4 ± 0.7	77.3 ± 0.8	96.8 ± 0.8	75.6 ± 1.4	83.5
Mixup [36]	85.2 ± 1.9	77.0 ± 1.7	96.8 ± 0.8	73.9 ± 1.6	83.2
MLDG [19]	81.4 ± 3.6	77.9 ± 2.3	96.2 ± 0.3	76.1 ± 2.1	82.9
CORAL [29]	80.5 ± 2.8	74.5 ± 0.4	96.8 ± 0.3	78.6 ± 1.4	82.6
MMD [21]	84.9 ± 1.7	75.1 ± 2.0	96.1 ± 0.9	76.5 ± 1.5	83.2
DANN [6]	84.3 ± 2.8	72.4 ± 2.8	96.5 ± 0.8	70.8 ± 1.3	81.0
CDANN [22]	78.3 ± 2.8	73.8 ± 1.6	96.4 ± 0.5	66.8 ± 5.5	78.8
MTL [3]	85.6 ± 1.5	78.9 ± 0.6	97.1 ± 0.3	73.1 ± 2.7	83.7
SagNet [26]	81.1 ± 1.9	75.4 ± 1.3	95.7 ± 0.9	77.2 ± 0.6	82.3
ARM [37]	85.9 ± 0.3	73.3 ± 1.9	95.6 ± 0.4	72.1 ± 2.4	81.7
VREx [17]	81.6 ± 4.0	74.1 ± 0.3	96.9 ± 0.4	72.8 ± 2.1	81.3
RSC [13]	83.7 ± 1.7	**82.9** ± 1.1	95.6 ± 0.7	68.1 ± 1.5	82.6
EFDMix	**86.3** ± 0.7	79.9 ± 0.9	96.3 ± 0.4	**79.0** ± 0.6	**85.4**

Table A2. Domain generalization results of category classification on PACS by following DomainBed [9], where the model selection strategy is leave-one-domain-out cross-validation.

Algorithm	A	C	P	S	Avg
ERM [32]	86.5 ± 1.0	81.3 ± 0.6	96.2 ± 0.3	82.7 ± 1.1	86.7
IRM [1]	84.2 ± 0.9	79.7 ± 1.5	95.9 ± 0.4	78.3 ± 2.1	84.5
GroupDRO [27]	87.5 ± 0.5	82.9 ± 0.6	97.1 ± 0.3	81.1 ± 1.2	87.1
Mixup [36]	87.5 ± 0.4	81.6 ± 0.7	**97.4** ± 0.2	80.8 ± 0.9	86.8
MLDG [19]	87.0 ± 1.2	82.5 ± 0.9	96.7 ± 0.3	81.2 ± 0.6	86.8
CORAL [29]	86.6 ± 0.8	81.8 ± 0.9	97.1 ± 0.5	82.7 ± 0.6	87.1
MMD [21]	88.1 ± 0.8	82.6 ± 0.7	97.1 ± 0.5	81.2 ± 1.2	87.2
DANN [6]	87.0 ± 0.4	80.3 ± 0.6	96.8 ± 0.3	76.9 ± 1.1	85.2
CDANN [22]	87.7 ± 0.6	80.7 ± 1.2	97.3 ± 0.4	77.6 ± 1.5	85.8
MTL [3]	87.0 ± 0.2	82.7 ± 0.8	96.5 ± 0.7	80.5 ± 0.8	86.7
SagNet [26]	87.4 ± 0.5	81.2 ± 1.2	96.3 ± 0.8	80.7 ± 1.1	86.4
ARM [37]	85.0 ± 1.2	81.4 ± 0.2	95.9 ± 0.3	80.9 ± 0.5	85.8
VREx [17]	87.8 ± 1.2	81.8 ± 0.7	**97.4** ± 0.2	82.1 ± 0.7	87.2
RSC [13]	86.0 ± 0.7	81.8 ± 0.9	96.8 ± 0.7	80.4 ± 0.5	86.2
EFDMix	**87.9** ± 0.7	**83.1** ± 0.1	96.8 ± 0.2	83.8 ± 0.6	**87.9**

Table A3. Domain generalization results of category classification on PACS by following DomainBed [9], where the model selection strategy is test-domain validation set (oracle).

Results following MixStyle [40]. The more comprehensive results of Tab. 1 in the main paper are illustrated in Tab. A4.

Method	Art	Cartoon	Photo	Sketch	Avg
Leave-one-domain-out generalization results					
MMD-AAE [21]	75.2	72.7	96.0	64.2	77.0
CCSA [24]	80.5	76.9	93.6	66.8	79.4
JiGen [4]	79.4	75.3	96.0	71.6	80.5
CrossGrad [28]	79.8	76.8	96.0	70.2	80.7
Epi-FCR [20]	82.1	77.0	93.9	73.0	81.5
Metareg [2]	83.7	77.2	95.5	70.3	81.7
L2A-OT [39]	83.3	78.2	96.2	73.6	82.8
ResNet-18 [11]	77.0±0.6	75.9±0.6	96.0±0.1	69.2±0.6	79.5
+ Manifold Mixup [33]	75.6±0.7	70.1±0.9	93.5±0.7	65.4±0.6	76.2
+ Cutout [5]	74.9±0.4	74.9±0.6	95.9±0.3	67.7±0.9	78.3
+ CutMix [35]	74.6±0.7	71.8±0.6	95.6±0.4	65.3±0.8	76.8
+ Mixup [36]	76.8±0.7	74.9±0.7	95.8±0.3	66.6±0.7	78.5
+ DropBlock [8]	76.4±0.7	75.4±0.7	95.9±0.3	69.0±0.3	79.2
+ MixStyle w/ random shuffle [40]	82.3±0.2	79.0±0.3	96.3±0.3	73.8±0.9	82.8
+ MixStyle w/ domain label [40]	84.1±0.4	78.8±0.4	96.1±0.3	75.9±0.9	83.7
+ MixStyle w/ random shuffle (our imp.) [40]	82.4±0.2	79.4±0.8	96.2±0.1	72.3±0.6	82.6
+ MixStyle w/ domain label (our imp.) [40]	83.1±0.8	78.6±0.9	95.9±0.4	74.2±2.7	82.9
+ EFDMix w/ random shuffle (ours)	83.2±0.7	**79.8**±0.8	96.5±0.3	74.1±0.6	83.4
+ EFDMix w/ domain label (ours)	**83.9**±0.4	79.4±0.7	**96.8**±0.4	**75.0**±0.7	**83.9**
ResNet-50 [11]	84.4±0.9	77.1±1.4	97.6±0.2	70.8±0.7	82.5
+ MixStyle w/ random shuffle [40]	88.7±0.7	81.4±0.7	98.0±0.2	75.0±0.6	85.8
+ MixStyle w/ domain label [40]	90.3±0.3	82.3±0.7	97.7±0.4	74.7±0.7	86.2
+ EFDMix w/ random shuffle (ours)	88.7±0.6	81.8±0.8	98.0±0.2	**77.7**±1.5	86.6
+ EFDMix w/ domain label (ours)	**90.6**±0.3	**82.5**±0.7	**98.1**±0.2	76.4±1.2	**86.9**
Single source generalization results					
ResNet-18 [11]	58.6±2.4	66.4±0.7	34.0±1.8	27.5±4.3	46.6
+ MixStyle w/ random shuffle [40]	61.9±2.2	71.5±0.8	41.2±1.8	32.2±4.1	51.7
+ EFDMix w/ random shuffle (ours)	**63.2**±2.3	**73.9**±0.7	**42.5**±1.8	**38.1**±3.7	**54.4**
ResNet-50 [11]	63.5±1.3	69.2±1.6	38.0±0.9	31.4±1.5	50.5
+ MixStyle w/ random shuffle [40]	73.2±1.1	74.8±1.1	46.0±2.0	40.6±2.0	58.6
+ EFDMix w/ random shuffle (ours)	**75.3**±0.9	**77.4**±0.8	**48.0**±0.9	**44.2**±2.4	**61.2**

Table A4. More comprehensive domain generalization results of category classification on PACS. Results with 'our imp.' are obtained with official codes. The listed domain is the test domain in the leave-one-domain-out setting, while it is the training one in the single source generalization setting.

D. More discussions

Details of user study. We conduct a user study to compare our method against Gatys [7], CMD [16], AdaIN and HM (cf. Fig. 5) quantitatively. Specifically, we employ 10 content images and 12 style images, producing 120 stylized images by each method. In the user study, we display the five stylized images in random order along with the underlying content and style images. Testers are asked to pick ONE favorite result for each style. A total of 26 testers participate in the study, resulting in 3120 votes. As illustrated in Tab. 3 of the paper, our method receives the most votes for its better stylized performance.

Where to apply EFDMix on DG? We investigate this problem by applying EFDMix to different layers in the model. Specifically, we apply EFDMix after the first residual block, denoted as '+ res-1'; '+ res-1-2' means EFDMix is applied after the first and second residual blocks; other notations are similarly defined. As illustrated in Tab. A5, better results are typically achieved by applying EFDMix to multiple lower-level layers (*e.g.*, results with '+ res-1-2' are generally better than the results with '+ res-1'), while applying EFDMix to high-level layers generally results in performance degradation (*e.g.*, '+ res-1-2-3-4' leads to worse performance than '+ res-1-2-3').

Most importantly, models with EFDMix outperform their counterparts with MixStyle in all settings, justifying the advantages of utilizing high-order statistics for feature augmentations in DG. In practice, we insert EFDMix after the 1st, 2nd,

Methods	MixStyle	EFDMix
ResNet50	82.5	
+ res-1	84.9	85.4
+ res-1-2	**85.8**	**86.6**
+ res-1-2-3	84.6	86.3
+ res-1-2-3-4	78.4	85.6

(a) PACS (ResNet50)

Method	MixStyle	EFDMix
ResNet18	79.5	
+ res-1	79.5	80.3
+ res-1-2	81.3	82.2
+ res-1-2-3	**82.6**	**83.4**
+ res-1-2-3-4	75.6	80.8

(b) PACS (ResNet18)

Method	MixStyle	EFDMix
OSNet	33.3	
+ res-1	**34.5**	34.8
+ res-1-2	33.8	**35.5**
+ res-1-2-3	20.5	31.6
+ res-1-2-3-4	12.7	20.6

(c) re-ID (Market1501→GRID)

Table A5. Ablation experiments on where to apply EFDMix and its comparison to MixStyle. Results are fairly compared under the 'w/ random shuffle' setting.

and 3rd residual blocks in ResNet-18, and after the 1st and 2nd residual blocks in ResNet-50, as suggested in Tab. A5. The OSNet [38] shares a similar architecture to ResNet and we insert EFDMix after its 1st and 2nd residual blocks.

EFDM vs. EFDMix. As illustrated in Fig. A4, EFDMix outperforms EFDM, justifying the efficacy of feature augmentations with mixed styles. Interestingly, models with EFDM achieve better results than these with MixStyle, suggesting that introducing high-order statistics is more effective than generating mixed styles for feature augmentations in DG.

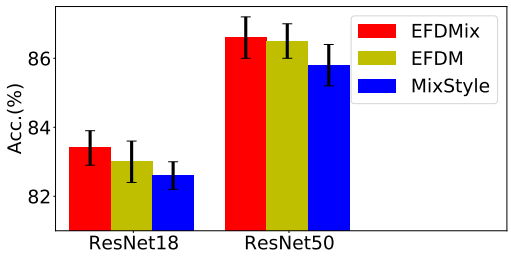

Figure A4. EFDM vs. EFDMix on PACS.

Selection of hyper-parameter α. As illustrated in Fig. A5, a smaller α, whose corresponding value of λ in Eq. (10) of the main paper is close to the extreme value of 0 or 1, tends to produce better results. Therefore, we suggest that $\alpha = 0.1$ is a good starting point.

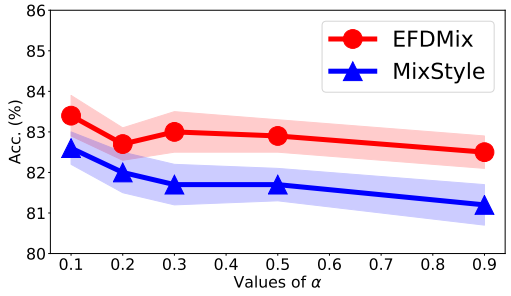

Figure A5. Analyses on hyper-parameter α on PACS.

Loss curves. As illustrated in Fig. A6, EFDM and AdaIN have almost the same convergence curves. Both of them converge to the small loss region rapidly. On the contrary, the model with HM converges slowly, possibly due to its inaccurate matching of eCDFs.

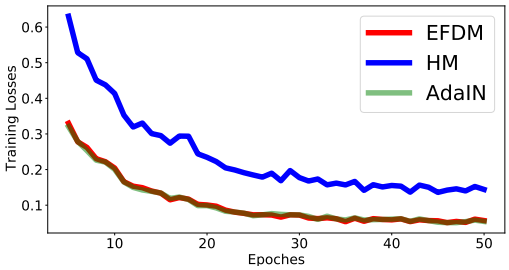

Figure A6. Loss curves on PACS.

The influence of ReLU/PReLU functions. As illustrated in Tab. A6, models with EFDM consistently outperform that with HM, no matter ReLU [25] or PReLU functions [10] are used. Specifically, ReLU introduces more equivalent feature values by setting negative values to zero, leading to larger advantage of EFDM over HM. For models with PReLU, their performance with EFDM also higher than that with HM, since equivalent feature values are common due to their dependency on discrete image pixels, as presented in Fig. 3 of the main paper.

Models	ReLU	PReLU
Acc. (%) with EFDM - Acc. (%) with HM	7.3	1.2

Table A6. The performance gap between models with EFDM and HM. 'ReLU' and 'PReLU' indicate the ResNet18 with ReLU and PReLU activation functions, respectively. We train the models on PACS dataset from scratch.

Methods	Training time	Test time	#param.	Acc.
ResNet18	2.5h	26.3μs	11,180K	79.5%
+ CNSN [30]	+ 0.4h	+ 6.1μs	+ 4K	+ 3.0%
+ MixStyle [40]	+ 0.1h	+ 0μs	+ 0K	+ 3.4%
+ SNR [14]	+ 0.3h	+ 8.9μs	+ 47K	+ 3.5%
+ EFDMix	+ 0.6h	+ 0μs	+ 0K	+ 4.4%

Table A7. Comparison between our EFDMix, CNSN [30], MixStyle [40] and SNR [14] on the PACS dataset. All methods are based on a ResNet18 backbone and run with a GeForce RTX 2080Ti GPU. The test time is averaged over all test examples.

Comparison to related methods on DG. Besides MixStyle, there are also some methods [14, 30] that manipulate feature style via matching mean and standard deviation and can be used in a plug-and-play manner. We fairly compare our method with them in Tab. A7. One can see that our EFDMix achieves the highest classification accuracy with a slight increase in training time. Most notably, our method does not introduce any additional parameters and it maintains fast test speed, demonstrating a clear advantage over CNSN and SNR. These results further validate the effectiveness of exact feature distribution matching.

A detailed analysis on computation time. As stated in the paper, the time complexities of AdaIN and EFDM modules are $O(n)$ and $O(n \log_2 n)$, respectively, where n is the input feature dimension. More specifically, the complexities of calculating mean and standard deviation are $O(n)$ and $O(3n)$, respectively (ignoring the complexity of square root), and thus the total complexity of AdaIN is $O(12n)$, among which $O(8n)$ for computing the mean and standard deviation of content/style features and $O(4n)$ for normalizing the content feature with the mean and standard deviation (cf. Equ.(1)). As for EFDM, the quicksort algorithm, whose average time complexity is $O(n \log_2 n)$, dominates its complexity. The total complexity of EFDM is then $O(2n \log_2 n + 2n)$ (ignoring the complexity of index operation), among which $O(2n \log_2 n)$ for sorting content and style features and $O(2n)$ for the element-wise addition (cf. Equ.(5)).

We also compare the practical runtime of AdaIN and EFDM for various feature dimensions in Fig. A7. When the feature dimension is small, *e.g.*, $n \leq 2^5$, EFDM is faster than AdaIN since $2n \log_2 n + 2n \leq 12n$. With the increase of feature dimension, the time cost ratio of EFDM / AdaIN increases. Since the feature dimension is usually not very big (*e.g.*, $n \leq 2^{12}$), the EFDM module is practically fast. For example, the input feature dimension of AdaIN module is 2^{10} in seminal style transfer [12]. In such a case, the computation time of EFDM module is about 3 times of that of the AdaIN module.

More importantly, compared to the stacked convolution and linear layers, the distribution matching module only occupies a very small amount of the total model complexity. For example, in style transfer models [4], the computation cost of AdaIN

is only 0.004% (*i.e.*, about $\frac{6\text{MFlops}}{142\text{GFlops}}$) of that of the whole model. Therefore, replacing AdaIN with EFDM introduces negligible additional computation time for the whole model, as shown in Tab. 3 in the main paper.

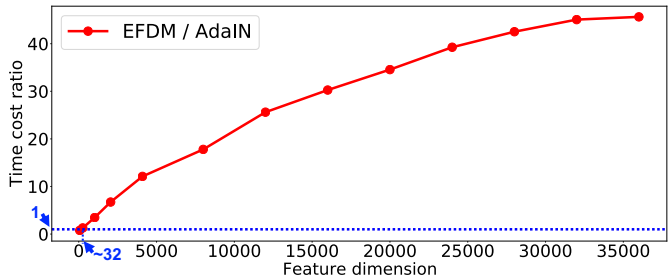

Figure A7. The time cost ratio between EFDM module and AdaIN module vs. the dimension of features.

Definitions of AdaMean and AdaStd. To ablate the role of mean and standard deviation individually, we implement AdaIN by matching only feature mean and standard deviation, leading to the AdaMean and AdaStd. Specifically, we introduce the AdaMean as:

$$\text{AdaMean: } \boldsymbol{o} = \boldsymbol{x} - \mu(\boldsymbol{x}) + \mu(\boldsymbol{y}). \tag{D.1}$$

The AdaStd is defined as:

$$\text{AdaStd: } \boldsymbol{o} = \frac{\boldsymbol{x} - \mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}\sigma(\boldsymbol{y}) + \mu(\boldsymbol{x}). \tag{D.2}$$

References

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 3, 4

[2] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. *Advances in Neural Information Processing Systems*, 31:998–1008, 2018. 5

[3] Gilles Blanchard, Aniket Anand Deshmukh, Urun Dogan, Gyemin Lee, and Clayton Scott. Domain generalization by marginal transfer learning. *arXiv preprint arXiv:1711.07910*, 2017. 3, 4

[4] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2229–2238, 2019. 5

[5] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 5

[6] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016. 3, 4

[7] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 2, 3, 5

[8] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Dropblock: A regularization method for convolutional networks. *arXiv preprint arXiv:1810.12890*, 2018. 5

[9] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *ICLR*, 2021. 1, 2, 3, 4

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 7

[11] 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. 1, 5

[12] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017. 1, 2, 7

[13] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 124–140. Springer, 2020. 3, 4

[14] Xin Jin, Cuiling Lan, Wenjun Zeng, Zhibo Chen, and Li Zhang. Style normalization and restitution for generalizable person re-identification. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3143–3152, 2020. 7

[15] Derrick N Joanes and Christine A Gill. Comparing measures of sample skewness and kurtosis. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 47(1):183–189, 1998. 1

[16] Nikolai Kalischek, Jan D Wegner, and Konrad Schindler. In the light of feature distributions: moment matching for neural style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9382–9391, 2021. 2, 3, 5

[17] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pages 5815–5826. PMLR, 2021. 3, 4

[18] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017. 1

[19] 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

[20] 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/CVF International Conference on Computer Vision*, pages 1446–1455, 2019. 5

[21] 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. 3, 4, 5

[22] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 624–639, 2018. 3, 4

[23] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep photo style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4990–4998, 2017. 1, 3

[24] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5715–5725, 2017. 5

[25] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010. 7

[26] Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap via style-agnostic networks. *arXiv preprint arXiv:1910.11645*, 2(7):8, 2019. 3, 4

[27] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019. 3, 4

[28] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. *ICLR*, 2018. 5

[29] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016. 3, 4

[30] Zhiqiang Tang, Yunhe Gao, Yi Zhu, Zhi Zhang, Mu Li, and Dimitris N Metaxas. Crossnorm and selfnorm for generalization under distribution shifts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 52–61, 2021. 7

[31] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 1

[32] Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999. 3, 4

[33] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447. PMLR, 2019. 5

[34] Wiki. Kurtosis. https://en.wikipedia.org/wiki/Kurtosis. 1

[35] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019. 5

[36] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 3, 4, 5

[37] Marvin Mengxin Zhang, Henrik Marklund, Nikita Dhawan, Abhishek Gupta, Sergey Levine, and Chelsea Finn. Adaptive risk minimization: A meta-learning approach for tackling group shift. 2020. 3, 4

[38] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3702–3712, 2019. 6

[39] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *European Conference on Computer Vision*, pages 561–578. Springer, 2020. 5

[40] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. *ICLR*, 2021. 1, 4, 5, 7