ConstStyle: Robust Domain Generalization with Unified Style Transformation

Supplementary Material

A. Details of the Training Process

Details of unified domain determination algorithm, training and inference processes are presented in Algorithms 1, 2 and 3, respectively.

B. Proofs

B.1. Proof of Lemma 1.

Let us start with $L^{\mathcal{S}_k}$, we have:

$$\begin{aligned} L^{\mathcal{S}_k} &= \frac{1}{|\mathcal{S}_k|} \sum_{(x,y)\in\mathcal{S}_k} [l(\omega(x), y)] \\ &= \frac{1}{|\mathcal{S}_k|} \sum_{(x,y)\in\mathcal{S}_k} l(\zeta(\theta_f(z_x)), y) \\ &= \frac{1}{|\mathcal{S}_k|} \sum_{(x,y)\in\mathcal{S}_k} l(\zeta(\theta_f(\sigma_x * \frac{z_x-\mu_x}{\sigma_x} + \mu_x)), y) \\ &= \frac{1}{|\mathcal{S}_k|} \sum_{(x,y)\in\mathcal{S}_k} f(\mu_x, \sigma_x, \frac{z_x-\mu_x}{\sigma_x}, y). \end{aligned} \tag{7}$$

Similarly, we have:

$$\begin{aligned} L^{\mathcal{S}_k^T} =& \frac{1}{|\mathcal{S}_k|} \sum_{(x,y)\in\mathcal{S}_k} [l(\omega^T(x), y)] \\ =& \frac{1}{|\mathcal{S}_k|} \sum_{(x,y)\in\mathcal{S}_k} l(\zeta(\theta_f(z_x^T)), y) \\ =& \frac{1}{|\mathcal{S}_k|} \sum_{(x,y)\in\mathcal{S}_k} l(\zeta(\theta_f(\sigma^T * \frac{z_x-\mu_x}{\sigma_x} + \mu^T)), y) \\ =& \frac{1}{|\mathcal{S}_k|} \sum_{(x,y)\in\mathcal{S}_k} f(\mu^T, \sigma^T, \frac{z_x-\mu_x}{\sigma_x}, y). \end{aligned} \tag{8}$$

By subtracting 8 from 7, we obtain:

$$\begin{aligned} L^{\mathcal{S}_k^T} - L^{\mathcal{S}_k} = \frac{1}{|\mathcal{S}_k|} \sum_{(x,y)\in\mathcal{S}_k} \Big(& f(\mu^T, \sigma^T, \frac{z_x-\mu_x}{\sigma_x}, y) \\ & - f(\mu_x, \sigma_x, \frac{z_x-\mu_x}{\sigma_x}, y) \Big). \end{aligned} \tag{9}$$

Using the Taylor approximation for a function with two variables, we derive:

$$\begin{aligned} & f(\mu^T, \sigma^T, \frac{z_x-\mu_x}{\sigma_x}, y) - f(\mu_x, \sigma_x, \frac{z_x-\mu_x}{\sigma_x}, y) \\ & \approx (\mu^T - \mu_x) \cdot \nabla_{\mu_x} f \quad + (\sigma^T - \sigma_x) \cdot \nabla_{\sigma_x} f. \end{aligned} \tag{10}$$

Let $\mathcal{D}_\mu(\mathcal{T}, \mathcal{S}_k)$ denote the distance between means of the unified instance style $\mathcal{T}$ and seen instance style $\mathcal{S}_k$, while $\mathcal{D}_\sigma(\mathcal{T}, \mathcal{S}_k)$ represents the distance between standard deviations. Let $||v||$ denote the L2-norm of vector v. Assume f is a β-Lipschitz function, we can suppose $\sup_{x\in\mathcal{S}_k} ||\nabla_{\mu_x} f|| = \beta_\mu$, $\sup_{x\in\mathcal{S}_k} ||\nabla_{\sigma_x} f|| = \beta_\sigma$, we have:

$$\begin{aligned} L^{\mathcal{S}_k^T} - L^{\mathcal{S}_k} \leq & \frac{1}{|\mathcal{S}_k|} \sum_{(x,y)\in\mathcal{S}_k} (\beta_\mu * ||\mu^T - \mu_x|| + \\ & \beta_\sigma * ||\sigma^T - \sigma_x||) \\ = & \beta_\mu * \frac{1}{|\mathcal{S}_k|} \sum_{(x,y)\in\mathcal{S}_k} ||\mu^T - \mu_x|| + \\ & \beta_\sigma * \frac{1}{|\mathcal{S}_k|} \sum_{(x,y)\in\mathcal{S}_k} ||\sigma^T - \sigma_x|| \\ = & \beta_\mu * \mathcal{D}_\mu(\mathcal{T}, \mathcal{S}_k) + \beta_\sigma * \mathcal{D}_\sigma(\mathcal{T}, \mathcal{S}_k) \end{aligned}$$

Let $\beta = \max(\beta_\mu, \beta_\sigma)$, then:

$$L^{\mathcal{S}_k^T} - L^{\mathcal{S}_k} \leq \beta \times (\mathcal{D}_\mu(\mathcal{T}, \mathcal{S}_k) + \mathcal{D}_\sigma(\mathcal{T}, \mathcal{S}_k)) \tag{11}$$

B.2. Proof of Theorem 1

According to B.1, for the seen domains $\{\mathcal{S}_k\}_{k=1}^N$, the total empirical loss across N seen domains is bounded as follows:

$$\sum_{k=1}^{N} L^{\mathcal{S}_k{}^T} \leq \sum_{k=1}^{N} L^{\mathcal{S}_k} + \beta * \sum_{k=1}^{N} (\mathcal{D}_\mu(\mathcal{T}, \mathcal{S}_k) + \mathcal{D}_\sigma(\mathcal{T}, \mathcal{S}_k)) \tag{12}$$

It can be observed that the upper bound of this loss depends on the total distance from the unified domain to N seen domains $\mathcal{S}_k$. Therefore, to minimize the loss over the seen domains, we aim to reduce the distance between the unified domain $\mathcal{T}$ and N seen domains $\mathcal{S}_k$. Consequently, the unified domain style $\mathcal{N}^T = (\mu^T, \Sigma^T)$ is the barycenter of N seen domain styles.

B.3. Proof of Theorem 2

The loss function of the model trained on seen domains, obtained by ConstStyle, and test on unseen domain is given by:

Algorithm 1: ConstStyle Training Process

Input: Seen data $\mathcal{S} = \{(x, y)\}$, Model $\omega = \zeta(\theta_f(\theta_s(.))$, the update interval γ, the number of epochs E, the learning rate η, and the number of clusters N';
Output: Optimal model ω^*, the final unified domain $\mathcal{N}^T$;
Algorithm:
for *epoch* $\leq E$ **do**
  $\varepsilon \leftarrow \emptyset$; // Set of style features
  for $x \in \mathcal{S}$ **do**
    if *epoch* $\leq \xi$ **then**
      $z_x = \theta_s(x)$;
      $p(x) = \zeta(\theta_f(z_x))$;
    else
      $z_x = \theta_s(x)$;
      $\epsilon_s \sim \mathcal{N}^T$; // sample style features
      $\mu_s, \sigma_s = split(\epsilon_s)$;
      $z_x^T = \sigma_s * \frac{z_x - \mu_x}{\sigma_x} + \mu_s$; // project to the unified domain
      $p(x) = \zeta(\theta_f(z_x^T))$;
    $l = \sum_{c \in \mathcal{C}} y_c . \log(p_c(x))$;
    $\omega = \omega - \eta . \nabla_\omega l$; // Update model
    if *epoch* % γ == *0* **then**
      $\mu_{x_c} = \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} z_{x_{c,h,w}}, \sigma_{x_c} = \sqrt{\frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} (z_{x_{c,h,w}} - \mu_{x_c})^2}$;
      $\epsilon_x = concat(\mu_x, \sigma_x)$; // extract style features
      $\varepsilon = \varepsilon \cup \epsilon_x$; // store style features
  if *epoch* % γ == *0* **then**
    $\mathcal{N}(\epsilon^T, \Sigma^T)$ =Unified Domain Determination(ε, N');
    $\mathcal{N}^T \leftarrow \mathcal{N}(\epsilon^T, \Sigma^T)$; // get unified domain style
  $\omega^* = \omega$;
return $\omega^*, \mathcal{N}^T$

Algorithm 2: Unified Domain Determination

Input: Set of all style features $\varepsilon = \{\epsilon_x | x \in S\}$, Number of clusters N';
Output: Unified Domain Style $\mathcal{N}(\epsilon^T, \Sigma^T)$;
Algorithm:
$\{\mathcal{C}_k \sim \mathcal{N}(\epsilon_{\mathcal{C}_k}, \Sigma_{\mathcal{C}_k}) | k = 1..N'\} \leftarrow BayesGMM(\varepsilon, N')$ $\epsilon^T = \frac{1}{N'} \sum_{k=1}^{N'} \epsilon_{\mathcal{C}_k}$;
$\Sigma^T = \frac{1}{N'} \sum_{k=1}^{N'} \Sigma_{\mathcal{C}_k}$;
$\mathcal{N}^T = \mathcal{N}(\epsilon^T, \Sigma^T)$;
return $\mathcal{N}^T$

$$
\begin{aligned}
L^{\mathcal{U}^T} &= \frac{1}{|\mathcal{U}|} \sum_{(u,y) \in \mathcal{U}} [l(\omega^T(u), y)] \\
&= \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \frac{1}{|\mathcal{U}_c|} \sum_{u \in \mathcal{U}_c} [l(\zeta(\theta_f(z_u^T))), y_c] \\
&= \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \frac{1}{|\mathcal{U}_c|} \sum_{u \in \mathcal{U}_c} [l(\zeta(\theta_f(\sigma_u^T * \frac{z_u - \mu_u}{\sigma_u} + \mu_u^T), y_c] \\
&= \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \frac{1}{|\mathcal{U}_c|} \sum_{u \in \mathcal{U}_c} [f(\mu_u^T, \sigma_u^T, \frac{z_u - \mu_u}{\sigma_u}, y_c)] \quad (13)
\end{aligned}
$$

Similarly, We have:

$$
\begin{aligned}
L^{\mathcal{S}^T} &= \frac{1}{|\mathcal{S}|} \sum_{(x,y) \in \mathcal{S}} [l(\omega^T(x), y)] \\
&= \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \frac{1}{|\mathcal{S}_c|} \sum_{x \in S_c} [l(\zeta(\theta_f(z_x^T))), y_c] \\
&= \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \frac{1}{|\mathcal{S}_c|} \sum_{x \in \mathcal{S}_c} [l(\zeta(\theta_f(\sigma^T * \frac{z_x - \mu_x}{\sigma_x} + \mu^T), y_c] \\
&= \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \frac{1}{|\mathcal{S}_c|} \sum_{x \in \mathcal{S}_c} [f(\mu^T, \sigma^T, \frac{z_x - \mu_x}{\sigma_x}, y_c)]. \quad (14)
\end{aligned}
$$

Assume that the cardinality of seen domain S and unseen domain U are the same for all classes, i.e, $|\mathcal{S}_c| = |\mathcal{U}_c| = d = \frac{|\mathcal{U}|}{|\mathcal{C}|} = \frac{|\mathcal{S}|}{|\mathcal{C}|}, \forall c \in \mathcal{C}$. From Equations 13 and 14, we have:

Algorithm 3: ConstStyle Inference Process

Input: Unseen data $\mathcal{U} = \{u|u \sim \mathcal{U}\}$, Optimal model ω^*, Unified domain $\mathcal{N}^T$;
Output: Prediction set $L_{\mathcal{U}}$;
Algorithm:
$L_{\mathcal{U}} = \emptyset$;
for $u \in \mathcal{U}$ **do**
  $z_u = \theta_s(u)$;
  $\mu^T, \sigma^T = split(\epsilon^T)$;
  $z_u^T = (\alpha.\sigma_u + (1-\alpha).\sigma^T).\frac{z_u - \mu_u}{\sigma_u} + (\alpha.\mu_u + (1-\alpha).\mu^T)$;
  $p(u) = \zeta(\theta_f(z_u^T))$;
  $y_u = \arg\max(softmax(p(u))$;
  $L_{\mathcal{U}} = L_{\mathcal{U}} \cup y_u$;
return $L_{\mathcal{U}}$

$$L^{\mathcal{U}^T} - L^{\mathcal{S}^T} = \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}}\frac{1}{d}\sum_{u\in U_c, x\in S_c}[f(\mu_u^T, \sigma_u^T, \frac{z_u-\mu_u}{\sigma_u}, y_c) - f(\mu^T, \sigma^T, \frac{z_x-\mu_x}{\sigma_x}, y_c)] \quad (15)$$

By applying the Taylor approximation for three variables, we obtain:

$$\begin{aligned}
&f(\mu_u^T, \sigma_u^T, \frac{z_u-\mu_u}{\sigma_u}, y_c) - f(\mu^T, \sigma^T, \frac{z_x-\mu_x}{\sigma_x}, y_c)\\
&\approx (\mu_u^T - \mu^T)\nabla_{\mu^T} f + (\sigma_u^T - \sigma^T)\nabla_{\sigma^T} f\\
&\quad + (\frac{z_u-\mu_u}{\sigma_u} - \frac{z_x-\mu_x}{\sigma_x})\nabla_{\frac{z_x-\mu_x}{\sigma_x}} f\\
&= (\alpha * \mu_u + (1-\alpha) * \mu^T - \mu^T)\nabla_{\mu^T} f\\
&\quad + (\alpha * \sigma_u + (1-\alpha) * \sigma^T - \sigma^T)\nabla_{\sigma^T} f\\
&\quad + (\frac{z_u-\mu_u}{\sigma_u} - \frac{z_x-\mu_x}{\sigma_x})\nabla_{\frac{z_x-\mu_x}{\sigma}} f\\
&= \alpha * (\mu_u - \mu^T) * \nabla_{\mu^T} f + \alpha * (\sigma_u - \sigma^T) * \nabla_{\sigma^T} f\\
&\quad + (\frac{z_u-\mu_u}{\sigma_u} - \frac{z_x-\mu_x}{\sigma})\nabla_{\frac{z_x-\mu_x}{\sigma}} f \qquad (16)
\end{aligned}$$

Denote $||v||$ as the L2-norm of tensor v. Suppose that $\sup_{x\in\mathcal{S}}(||\nabla_{\mu^T} f||, ||\nabla_{\sigma^T} f||) = \beta$ and $\sup_{x\in S}\nabla_{\frac{z_x-\mu_x}{\sigma}} f = \xi$, then we have:

$$\begin{aligned}
&L^{\mathcal{U}^T} - L^{\mathcal{S}^T}\\
&\leq \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}}\frac{1}{d}\sum_{u\in\mathcal{U}_c, x\in\mathcal{S}_c}(\alpha \times (\beta \times ||\mu^T - \mu_u||\\
&\quad + \beta \times ||\sigma^T - \sigma_u||) + \xi \times ||\frac{z_u-\mu_u}{\sigma_u} - \frac{z_x-\mu_x}{\sigma}||)\\
&\leq \alpha \times \beta \times \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}(||\mu^T - \mu_u|| + ||\sigma^T - \sigma_u||)\\
&\quad + \xi \times \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}, x\in\mathcal{S}}||\frac{z_u-\mu_u}{\sigma_u} - \frac{z_x-\mu_x}{\sigma}||. \qquad (17)
\end{aligned}$$

Observed that, $\frac{z_u-\mu_u}{\sigma_u}, \frac{z_x-\mu_x}{\sigma} \sim \mathcal{N}(0, I)$, where I is the identity matrix size $C \times H \times W$, where C, H, W are the channel, height, and width dimensions of z_x. When the cardinality of seen domains $\mathcal{S}$, unseen domain $\mathcal{U}$ is sufficiently large, we can approximate:

$$\frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}, x\in\mathcal{S}}||\frac{z_u-\mu_u}{\sigma_u} - \frac{z_x-\mu_x}{\sigma}|| = \mathbb{E}[||U - X||], \quad (18)$$

where U and X are two random multivariate variables over $\mathbb{R}^{C\times H\times W}$ drawn from standard Gaussian distribution, $U, X \sim \mathcal{N}(0, I)$. We have:

$$\begin{aligned}
&0 \leq \mathbb{V}[||U - X||] = \mathbb{E}[||U - X||^2] - (\mathbb{E}[U - X])^2\\
&\rightarrow \mathbb{E}[U - X] \leq \sqrt{\mathbb{E}[||U - X||^2]} = \sqrt{Tr(2I)}\\
&\rightarrow \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}, x\in\mathcal{S}}||\frac{z_u-\mu_u}{\sigma_u} - \frac{z_x-\mu_x}{\sigma}|| \leq \sqrt{Tr(2I)}
\end{aligned}$$

Let $\mathcal{D}_\mu(\mathcal{T},\mathcal{U})$ and $\mathcal{D}_\sigma(\mathcal{T},\mathcal{U})$ be the distance between mean and standard deviation of unified domain $\mathcal{T}$ and unseen domain $\mathcal{U}$, respectively. From Equation (17), we obtain:

$$\begin{aligned}
L^{\mathcal{U}^T} - L^{\mathcal{S}^T} \leq \alpha \times \beta \times (\mathcal{D}_\mu(\mathcal{U}, \mathcal{T}) + \mathcal{D}_\sigma(\mathcal{U}, \mathcal{T}))\\
+ \xi \times \sqrt{2.Tr(I)}
\end{aligned}$$

C. Experiment Setup

Image classification: We train a ResNet18 pretrained on ImageNet for 200 epochs with learning rate of 0.001. Batch size is set to 32 for PACS dataset with 3 integrated ConstStyle layers, and 128 with 1 ConstStyle layer for Digit5 dataset.

Image Corruption: We use WideResNet with a single ConstStyle layer as a backbone, training for 200 epochs with a learning rate of 0.05 and batch size of 512.

Instance Retrieval: We train a model with ResNet50 pretrained on ImageNet as the backbone for 80 epochs with a learning rate of 0.0035. We integrate 3 ConstStyle layers

Method	Venue	M,MM	M,S	M,SY	M,U	MM,S	MM,SY	MM,U	S,SY	S,U	SY,U	Avg
ERM	-	**80.22**	82.77	92.34	97.46	77.60	74.83	71.67	52.49	77.70	87.86	79.49
Crossgrad	ICLR 2018	79.24	82.95	92.01	97.68	76.64	75.01	73.00	50.77	78.77	84.8	77.02
Mixup	ICLR 2018	75.92	84.88	90.81	96.75	75.87	70.71	67.49	44.03	80.51	82.58	76.95
Cutmix	ICCV 2019	74.86	**85.16**	91.61	97.02	77.78	70.04	68.87	45.51	**80.75**	85.59	77.71
EFDMix	CVPR 2022	76.29	82.87	92.53	97.52	77.65	76.14	73.33	52.34	78.57	85.87	78.88
RIDG	ICCV 2023	79.75	84.48	91.97	97.23	77.8	73.77	71.05	50.73	79.74	86.33	79.28
MixStyle	ICLR 2021	77.96	72.69	83.37	86.82	75.09	62.18	68.15	41.53	58.5	71.88	69.81
DSU	ICLR 2022	78.77	83.83	92.1	**97.81**	78.53	74.78	71.89	53.66	78.14	87.62	79.71
CSU	WACV 2024	78.64	84.29	92.72	97.39	77.27	75.61	72.67	57.28	78.56	88.08	80.25
ConstStyle	Ours	**80.22**	84.69	**92.92**	97.33	**78.73**	**76.27**	**74.19**	**57.58**	80.29	**88.24**	**81.04**
Method	Venue	M,MM,S	M,MM,SY	M,MM,U	M,S,SY	M,S,U	M,SY,U	MM,S,SY	MM,S,U	MM,SY,U	S,SY,U	Avg
ERM	-	80.12	76.39	**71.41**	61.89	83.47	91.07	44.49	77.63	76.87	48.47	71.18
Crossgrad	ICLR 2018	79.59	76.32	71.31	60.37	83.21	91.47	36.57	77.71	74.26	46.55	70.34
Mixup	ICLR 2018	78.35	74.19	69.51	57.22	**85.78**	91.16	34.45	77.11	71.34	41.29	68.04
Cutmix	ICCV 2019	79.82	73.12	68.92	58.28	85.64	91.32	32.52	78.3	72.57	39.92	68.04
EFDMix	CVPR 2022	80.38	76.04	70.13	63.48	83.62	91.96	43.46	77.61	73.94	50.43	71.10
RIDG	ICCV 2023	80.51	74.71	70.45	61.76	84.78	91.41	35.02	78.28	75.74	45.53	69.81
MixStyle	ICLR 2021	78.91	74.97	61.48	57.95	71.44	81.43	42.92	71.44	62.3	40.99	64.38
DSU	ICLR 2022	**80.71**	76.25	70.54	62.35	83.25	91.47	42.87	77.84	76.29	48.31	70.98
CSU	WACV 2024	80.63	76.26	69.50	**64.68**	85.09	91.53	47.31	77.64	75.61	52.73	72.09
ConstStyle	Ours	80.32	**77.93**	70.89	**64.68**	84.88	**92.10**	**48.88**	**79.08**	**77.27**	**53.55**	**72.95**

Table 7. Multiple unseen domain generalization (2 and 3 unseen domains) on Digits5 dataset. Abbrevations: (M: MNIST, MM: MNISTM, S: SVHN, SY: SYN, U: USPS). The best result is colored **purple** and the second best result is colored blue.

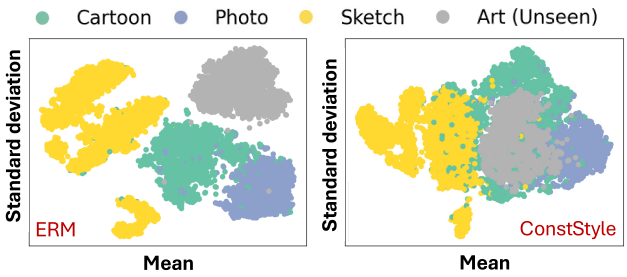

Figure 7. Style statistics of ERM and ConstStyle.

Method	Dataset	
	PACS	Digit5
ConstStyle w/ Pretrained features	86.31	76.61
ConstStyle w/ Domain label	86.73	86.37
ConstStyle	**86.77**	**86.88**

Table 8. Different variants of ConstStyle.

into the model.

Across all experiment scenarios, the number of clusters is fixed to 4. All methods are optimized using SGD optimizer. Optimal hyperparameters are selected based on the performance on the validation dataset.

D. Additional Results

D.1. Multiple Unseen Domains on Digit5 dataset

We perform additional experiment with multiple unseen domains on the Digit5 dataset. The results are shown in Table 7. It can be observed that ConstStyle achieves the best performance in most of the scenarios, and obtains the highest average accuracy.

# of clusters	1	2	3	4	5
PACS	86.07	86.51	86.61	**86.77**	86.61
Digit5	85,80	85,60	85,54	**86.88**	85.93

Table 9. Impacts of the number of clusters.

Batchsize	8	16	32	64	128	256
Accuracy	85.43	86.22	**86.77**	86.33	85.91	85.10

Table 10. Impacts of the batch size on accuracy (PACS dataset).

E. Ablation Studies

In this section, we conduct a more in-depth analysis concerning the impacts hyperparameters in ConstStyle's, which is the number of clusters used during the unified domain determination phase, we additionally perform experiments to explore the influence of training batch size and impact of α in the inference process.

E.1. In-depth analysis of ConstStyle

We first conduct additional experiments to further analyze the behaviors of ConstStyle. Figure 7 illustrates the style statistics for both seen and unseen domains, demonstrating that ConstStyle effectively aligns training and test samples within a unified domain, thereby enhancing performance under distribution shift. Additionally, we evaluate Const-

α	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
PACS	86.00	85.08	85.76	86.34	86.33	86.34	**86.77**	86.46	86.22	86.03	85.86
Digit5	85.96	85.62	85.77	85.91	85.95	**86.88**	86.01	85.99	85.98	86.01	85.96

Table 11. Impact of α to the model performance on different datasets.

Data size	32892	65787	98680	131575
Average training time per epoch (s)	249.2	568.4	857.3	1076.4

Table 12. Scalability of ConstStyle with different number of training data size.

Style with two alternative approaches: **1. Clustering using domain label** and **2. Utilizing pretrained style statistics** with results shown in Table 8. We can observe that while domain labels can produce good performance, they are not always optimal, as some samples have style statistics belonging to other domains; thus, clustering using GMM can form appropriate domain clusters, yielding better performance. Furthermore, using pretrained features for clustering can achieve comparable results if style features are previously learned by the pretrained model, as shown in the PACS dataset in Table 8. However, if the pretrained model has not learned style features, relying on them can significantly degrade ConstStyle's accuracy, as observed in the Digit5 dataset.

E.2. Impacts of the Number of Clusters

We first investigate the impact of the number of clusters during the clustering phase, ranging from one to five. Figure 9 demonstrates that ConstStyle performs consistently across domains, regardless of the number of clusters. This consistency demonstrates ConstStyle's robustness, as the major goal is to construct a single domain by averaging all of the clusters in the visible domains.

E.3. Impacts of the Batch Size

In this section, we investigate the impacts of the batch size on ConstStyle's performance. Experiments are conducted with batch size ranging from 8 to 256, and the results are presented in Table 10. The results suggest that using either very small or very large batch sizes can lead to suboptimal performance, as too few or too many style modifications may disrupt learning stability. The optimal strategy is to use a moderate batch size (about 32), ensuring balanced and steady learning for the model.

E.4. Impacts of Partial Projection

We study the impacts of α on the performance of the proposed method by varying this parameter from 0 to 1, with the results presented in Table 11. It is evident that the impact of α varies significantly across different values, highlighting its important role in achieving optimal performance. When an appropriate value of α is selected, overall performance can improve by up to 0.56% for PACS dataset and up to 0.87% for Digit5 dataset, compared to when no α value is used. This results also highlights the effects of our proposed partial style alignment algorithm (Section 3.5).

E.5. Scalability against larger datasets

ConstStyle has three components: style statistics distribution estimation, unified style determination, and style aligment. The computational complexity of all three components scales linearly with the training data size. As a result, Constlyle is inherently scalable to large datasets. This scalability is also empirically demonstrated in Table 12, which reports the average training time per epoch when varying the training data size.