Deep Stable Learning for Out-Of-Distribution Generalization

Mingda Zhang

School of Data Science
The Chinese University of Hong Kong, Shenzhen, China

May 19, 2022

Stable learning

- $X = \{S, V\}, Y$ We know that $Y = \mathbf{S}^e \beta_S + \mathbf{V}^e \beta_V + \varepsilon^e$ where $\beta_V = 0$
- For simple OLS, we can get

$$\hat{\beta}_{V_{OLS}} = \beta_V + \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{V}_i^T \mathbf{V}_i \right)^{-1} \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{V}_i^T \mathbf{S}_i \right) \left(\beta_S - \hat{\beta}_{S_{OLS}} \right)$$

$$\hat{\beta}_{S_{OLS}} = \beta_S + \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{S}_i^T \mathbf{S}_i \right)^{-1} \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{S}_i^T \mathbf{V}_i \right) \left(\beta_V - \hat{\beta}_{V_{OLS}} \right)$$

Stable learning

- $X = \{S, V\}, Y$ We know that $Y = \mathbf{S}^e \beta_S + g(S) + \mathbf{V}^e \beta_V + \varepsilon^e$ where $\beta_V = 0$
- For simple OLS, we can get

$$\begin{aligned}
\hat{\beta}_{V_{OLS}} &= \beta_V + \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{V}_i^T \mathbf{V}_i \right)^{-1} \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{V}_i^T g\left(\mathbf{S}_i\right) \right) \\
&+ \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{V}_i^T \mathbf{V}_i \right)^{-1} \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{V}_i^T \mathbf{S}_i \right) \left(\beta_S - \hat{\beta}_{S_{OLS}} \right) \\
\hat{\beta}_{S_{OLS}} &= \beta_S + \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{S}_i^T \mathbf{S}_i \right)^{-1} \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{S}_i^T g\left(\mathbf{S}_i\right) \right) \\
&+ \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{S}_i^T \mathbf{S}_i \right)^{-1} \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{S}_i^T \mathbf{V}_i \right) \left(\beta_V - \hat{\beta}_{V_{OLS}} \right)
\end{aligned}$$

Stable learning

- **Two assumptions:**

Assumption 1.

There exists a stable function $f(s)$ such that for all environment $e \in \mathcal{E}, \mathbb{E}\left(Y^e \mid \mathbf{S}^e = s, \mathbf{V}^e = v\right) = \mathbb{E}\left(Y^e \mid \mathbf{S}^e = s\right) = f(s)$

  - This assumption can be guaranteed by $Y \perp \mathbf{V} \mid \mathbf{S}$.

Assumption 2.

All stable features $\mathbf{S}$ are observed.

- **One purpose:**

Proposition 1.

If $\mathbf{X}$ are mutually independent with mean 0, then $\mathbb{E}\left(\mathbf{V}^T g(\mathbf{S})\right) = 0$ and $\mathbb{E}\left(\mathbf{V}^T \mathbf{S}\right) = 0$.

Stable learning

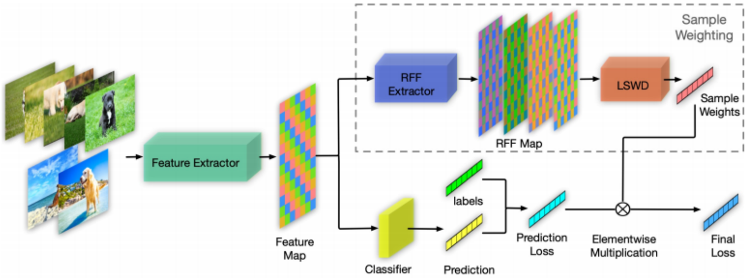
- **coefficient estimation:** With the learned sample weights $\hat{W}$ from variable decorrelation, one can run weighted least square (WLS) to estimate the regression coefficient β as:

$$\hat{\beta}_{WLS} = \arg\min_{\beta} \sum_{i=1}^{n} \hat{W}_i \cdot \left(Y_i - \mathbf{X}_{i,\beta}\right)^2 .$$

- **sample weight:** We know variables $\mathbf{X}_{,j}$ and $\mathbf{X}_{,k}$ are independent if $\mathbb{E}\left[\mathbf{X}_{,j}^{a}\mathbf{X}_{,k}^{b}\right] = \mathbb{E}\left[\mathbf{X}_{,j}^{a}\right]\mathbb{E}\left[\mathbf{X}_{,k}^{b}\right]$ for all $a, b \in \mathbb{N}$.

$$\min_{W} \sum_{a=1}^{\infty}\sum_{b=1}^{\infty} \left\| \mathbb{E}\left[\mathbf{X}_{,j}^{a^T} \boldsymbol{\Sigma}_W \mathbf{X}_{,k}^{b}\right] - \mathbb{E}\left[\mathbf{X}_{,j}^{a^T} W\right]\mathbb{E}\left[\mathbf{X}_{,k}^{b^T} W\right] \right\|_2^2$$

StableNet


Sample Weighting
RFF Extractor
RFF Map
LSWD
Sample Weights
Feature Extractor
Feature Map
labels
Classifier
Prediction
Prediction Loss
Elementwise Multiplication
Final Loss

decorrelation

- **purpose:** gets rid of both linear and non-linear dependencies between features
- DWR [KXC+20] proposed to decorrelate the every two features, i.e.,

$$w(\mathbf{X}) = \arg\min_{w_0(\mathbf{X})} \sum_{1\leq i,j\leq d, i\neq j} \left(\mathrm{Cov}\left(X_i, X_j; w_0\right)\right)^2$$

restriction: The loss function focuses on the linear correlation only.

SRDO [SCZK20] proposed to learn $w(\mathbf{X})$ by estimating the density ratio of the training distribution P^{tr} and a specific target distribution $\tilde{P}$. The target distribution $\tilde{P}$ is $\tilde{P}\left(X_1, X_2, \ldots, X_d\right) = \prod_{i=1}^{d} P^{\mathrm{tr}}\left(X_i\right)$.

$$w(\mathbf{X}) = \frac{P(Z=1 \mid \mathbf{X})}{1 - P(Z=1 \mid \mathbf{X})}.$$

Here $P(Z=1 \mid \mathbf{X})$ means the probability of a sample $\mathbf{X}$, which is drawn from the balanced mixture of P^{tr} and $\tilde{P}$, belonging to P^{tr}.

restriction: The density ratio should be estimated accurately.

HSIC for decorrelation

- Use kernel method for different data $X_i^{\Phi} = K\left(X_i, \cdot\right)^*$
- HSIC criterion:

$$\begin{aligned} HSIC(x,y) =& [E_{p(x,y)} < k(x,\cdot) - E_{p(x)}k(x,\cdot), k(y,\cdot) - E_{p(y)}k(y,\cdot) >]^2 \\ =& [E_{p(x,y)}k(x,y)]^2 + [E_{p(x)p(y)}k(x,y)]^2 \\ & - 2(E_{p(x,y)}k(x,y))(E_{p(x)p(y)}k(x,y)) \end{aligned}$$

- **restriction:** The kernel matrix is hard to calculate with large batch-size.

RFF

- We can find a measure which is the Fourier transform of k.

$$k(x-y) = \int_{R^d} p(\omega) e^{j\omega'(x-y)} d\omega = E_{\omega \sim p(w)} \left[\zeta_\omega(x) \zeta_\omega(y)^* \right]$$

where $\zeta_\omega(x) = e^{jw^T x}$. And we only need the real part. So, we let $z_w(x) = \sqrt{2} \cos\left(w^T x + b\right)$

$$k(x, y) = E_{\omega \sim p(w)} \left[z_w(x) z_w(y) \right]$$

where $w \sim p(w), b \sim \mathrm{Uniform}(0, 2\pi)$.

Covariance matrix estimate

- Covariance matrix:

$$\hat{\Sigma}_{AB} = \tfrac{1}{n-1} \textstyle\sum_{i=1}^{n} \Big[\Big(\mathbf{u}\left(A_i\right) - \tfrac{1}{n} \textstyle\sum_{j=1}^{n} \mathbf{u}\left(A_j\right) \Big)^T \cdot \\ \Big(\mathbf{v}\left(B_i\right) - \tfrac{1}{n} \textstyle\sum_{j=1}^{n} \mathbf{v}\left(B_j\right) \Big) \Big]$$

where

$$\mathbf{u}(A) = \left(u_1(A), u_2(A), \ldots u_{n_A}(A)\right), u_j(A) \in \mathcal{H}_{\mathrm{RFF}}, \forall j$$

$$\mathbf{v}(B) = \left(v_1(B), v_2(B), \ldots v_{n_B}(B)\right), v_j(B) \in \mathcal{H}_{\mathrm{RFF}}, \forall j$$

and

$$\mathcal{H}_{\mathrm{RFF}} = \{\{: x \to \sqrt{2}\cos(\omega x + \phi) \mid \\ \omega \sim N(0,1), \phi \sim \mathrm{Uniform}(0, 2\pi)\},$$

Weight estimate

- Weighted covariance matrix:

$$\hat{\Sigma}_{AB;\mathbf{w}} = \tfrac{1}{n-1} \textstyle\sum_{i=1}^{n} \left[\left(w_i \mathbf{u}\left(A_i\right) - \tfrac{1}{n} \textstyle\sum_{j=1}^{n} w_j \mathbf{u}\left(A_j\right) \right)^T \left(w_i \mathbf{v}\left(B_i\right) - \tfrac{1}{n} \textstyle\sum_{j=1}^{n} w_j \mathbf{v}\left(B_j\right) \right) \right]$$

- Weight estimate:

$$\mathbf{w}^* = \operatorname*{arg\,min}_{\mathbf{w} \in \Delta_n} \sum_{1 \leq i < j \leq m_Z} \left\| \hat{\Sigma}_{\mathbf{Z}_{:,i} \mathbf{Z}_{:,j};\mathbf{w}} \right\|_F^2$$

Learning sample weights globally

- **Local and global integration**

$$\mathbf{Z}_O = \text{Concat}\left(\mathbf{Z}_{G1}, \mathbf{Z}_{G2}, \cdots, \mathbf{Z}_{Gk}, \mathbf{Z}_L\right)$$
$$\mathbf{w}_O = \text{Concat}\left(\mathbf{w}_{G1}, \mathbf{w}_{G2}, \cdots, \mathbf{w}_{Gk}, \mathbf{w}_L\right)$$

- **Train** While training for each batch, we keep $\mathbf{w}_{Gi}$ fixed and only $\mathbf{w}_L$ is learnable.
- **Update global Information** At the end of each iteration of training, we fuse the global information $(\mathbf{Z}_{Gi}, \mathbf{w}_{Gi})$ and the local information $(\mathbf{Z}_L, \mathbf{w}_L)$ as follows:

$$\mathbf{Z}'_{Gi} = \alpha_i \mathbf{Z}_{Gi} + (1 - \alpha_i)\, \mathbf{Z}_L$$
$$\mathbf{w}'_{Gi} = \alpha_i \mathbf{w}_{Gi} + (1 - \alpha_i)\, \mathbf{w}_L$$

unbalanced

- **Introduction:** Domains are split into source domains and target domains.
- **Setting:** We adopt PACS and VLCS for this setting. Three domains are considered as source domains and the other one as target. To make the amount of data from heterogeneous sources clearly differentiated, we set one domain as the dominant domain.

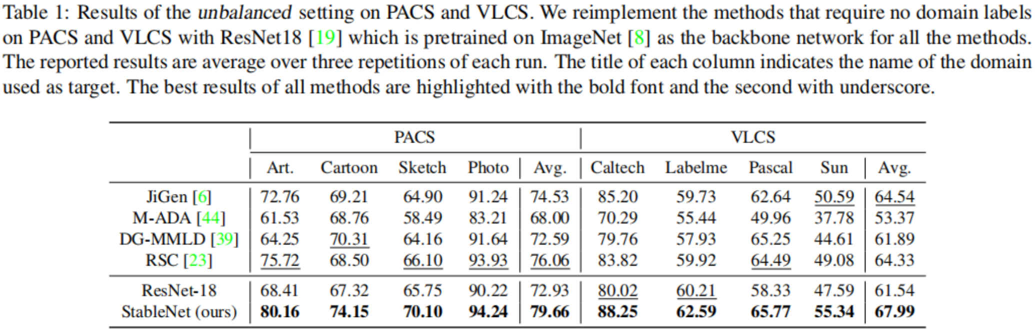


Table 1: Results of the *unbalanced* setting on PACS and VLCS. We reimplement the methods that require no domain labels on PACS and VLCS with ResNet18 [19] which is pretrained on ImageNet [8] as the backbone network for all the methods. The reported results are average over three repetitions of each run. The title of each column indicates the name of the domain used as target. The best results of all methods are highlighted with the bold font and the second with underscore.

	PACS					VLCS				
	Art.	Cartoon	Sketch	Photo	Avg.	Caltech	Labelme	Pascal	Sun	Avg.
JiGen [6]	72.76	69.21	64.90	91.24	74.53	85.20	59.73	62.64	50.59	64.54
M-ADA [44]	61.53	68.76	58.49	83.21	68.00	70.29	55.44	49.96	37.78	53.37
DG-MMLD [39]	64.25	70.31	64.16	91.64	72.59	79.76	57.93	65.25	44.61	61.89
RSC [23]	75.72	68.50	66.10	93.93	76.06	83.82	59.92	64.49	49.08	64.33
ResNet-18	68.41	67.32	65.75	90.22	72.93	80.02	60.21	58.33	47.59	61.54
StableNet (ours)	**80.16**	**74.15**	**70.10**	**94.24**	**79.66**	**88.25**	**62.59**	**65.77**	**55.34**	**67.99**

unbalanced+flexible

- **Introduction:** Domains for different categories can be various
- **Setting:** For PACS and VLCS, we randomly select one domain as the dominant domain for each class, and another domain as the target. For NICO, there are 10 domains for each class, 8 out of which are selected as the source and 2 as the target.

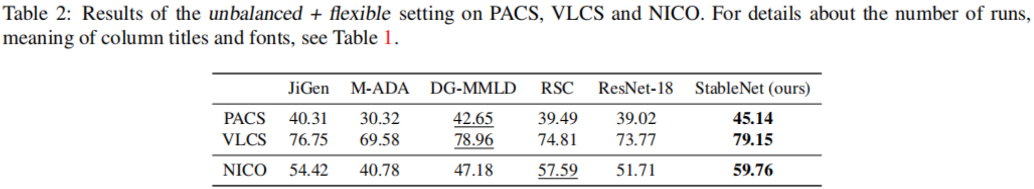


Table 2: Results of the *unbalanced + flexible* setting on PACS, VLCS and NICO. For details about the number of runs, meaning of column titles and fonts, see Table 1.

	JiGen	M-ADA	DG-MMLD	RSC	ResNet-18	StableNet (ours)
PACS	40.31	30.32	42.65	39.49	39.02	**45.14**
VLCS	76.75	69.58	78.96	74.81	73.77	**79.15**
NICO	54.42	40.78	47.18	57.59	51.71	**59.76**

Unbalanced + flexible + adversarial setting

- **Introduction:** The model is under adversarial attack and the spurious correlations between domains and labels are strong and misleading
- **Setting:**
  - For a given category, there is no overlap between the domains in training and testing;
  - There are strong spurious correlations between labels and domains;
  - The ratio of dominant context to other contexts varies from 9.5:1 to 1:1 to generate settings with different levels of distribution shifts.

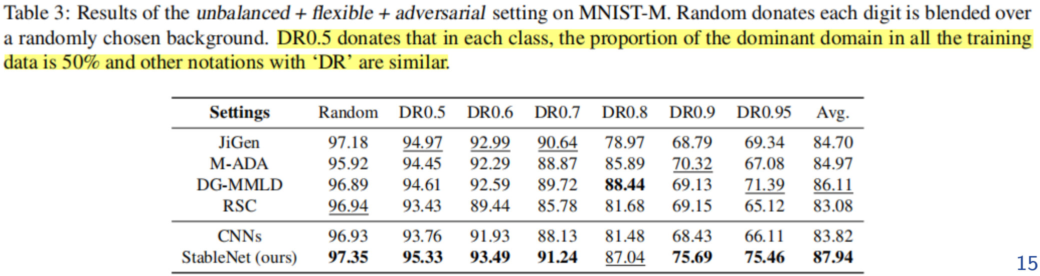


Table 3: Results of the *unbalanced + flexible + adversarial* setting on MNIST-M. Random donates each digit is blended over a randomly chosen background. DR0.5 donates that in each class, the proportion of the dominant domain in all the training data is 50% and other notations with 'DR' are similar.

Settings	Random	DR0.5	DR0.6	DR0.7	DR0.8	DR0.9	DR0.95	Avg.
JiGen	97.18	94.97	92.99	90.64	78.97	68.79	69.34	84.70
M-ADA	95.92	94.45	92.29	88.87	85.89	70.32	67.08	84.97
DG-MMLD	96.89	94.61	92.59	89.72	**88.44**	69.13	71.39	86.11
RSC	96.94	93.43	89.44	85.78	81.68	69.15	65.12	83.08
CNNs	96.93	93.76	91.93	88.13	81.48	68.43	66.11	83.82
StableNet (ours)	**97.35**	**95.33**	**93.49**	**91.24**	87.04	**75.69**	**75.46**	**87.94**

Classic setting

- **Introduction:** Domains are split into source domains and target domains. The capacities of various domains are comparable
- **Setting:** For PACS and VLCS, we utilize three domains as source domains and the remaining one as the target.

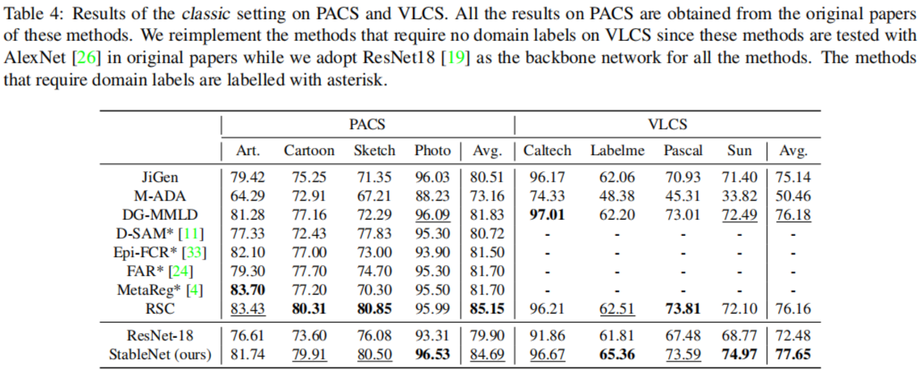
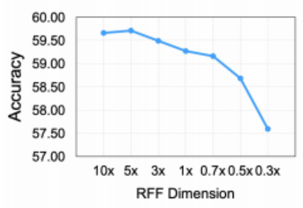


Table 4: Results of the *classic* setting on PACS and VLCS. All the results on PACS are obtained from the original papers of these methods. We reimplement the methods that require no domain labels on VLCS since these methods are tested with AlexNet [26] in original papers while we adopt ResNet18 [19] as the backbone network for all the methods. The methods that require domain labels are labelled with asterisk.

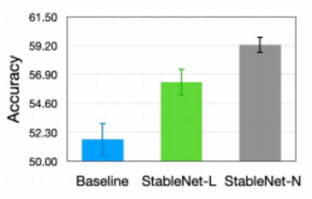
	PACS					VLCS				
	Art.	Cartoon	Sketch	Photo	Avg.	Caltech	Labelme	Pascal	Sun	Avg.
JiGen	79.42	75.25	71.35	96.03	80.51	96.17	62.06	70.93	71.40	75.14
M-ADA	64.29	72.91	67.21	88.23	73.16	74.33	48.38	45.31	33.82	50.46
DG-MMLD	81.28	77.16	72.29	96.09	81.83	**97.01**	62.20	73.01	72.49	76.18
D-SAM* [11]	77.33	72.43	77.83	95.30	80.72	-	-	-	-	-
Epi-FCR* [33]	82.10	77.00	73.00	93.90	81.50	-	-	-	-	-
FAR* [24]	79.30	77.70	74.70	95.30	81.70	-	-	-	-	-
MetaReg* [4]	**83.70**	77.20	70.30	95.50	81.70	-	-	-	-	-
RSC	83.43	**80.31**	**80.85**	95.99	**85.15**	96.21	62.51	**73.81**	72.10	76.16
ResNet-18	76.61	73.60	76.08	93.31	79.90	91.86	61.81	67.48	68.77	72.48
StableNet (ours)	81.74	79.91	80.50	**96.53**	84.69	96.67	**65.36**	73.59	**74.97**	**77.65**

Ablation study

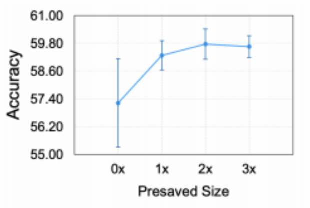
- We exploit the effect of sampling size for Random Fourier Features.
- We further exploit the effect of the size of presaved features and weights.



(a)



(b)



(c)

References I

[KXC^{+}20] Kun Kuang, Ruoxuan Xiong, Peng Cui, Susan Athey, and Bo Li, Stable prediction with model misspecification and agnostic distribution shift, Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 2020, pp. 4485–4492.

[SCZK20] Zheyan Shen, Peng Cui, Tong Zhang, and Kun Kunag, Stable learning via sample reweighting, Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 2020, pp. 5692–5699.