Eliminating Domain Bias for Federated Learning in Representation Space

Jianqing Zhang\textsuperscript{1}  
Yang Hua\textsuperscript{2}  
Jian Cao\textsuperscript{1}  
Hao Wang\textsuperscript{3}

Tao Song\textsuperscript{1}  
Zhengui Xue\textsuperscript{1}  
Ruhui Ma\textsuperscript{1}  
Haibing Guan\textsuperscript{1}

\textsuperscript{1}Shanghai Jiao Tong University  
\textsuperscript{2}Queen's University Belfast  
\textsuperscript{3}LSU
Federated Learning (FL)

- In practice, clients generate their specific private data, as shown by the colorful icons here.
Statistical Heterogeneity Issue

- Client-specific private data brings the *statistical heterogeneity* issue
Statistical Heterogeneity Issue

• With heterogeneous data, clients’ local training turns the received global model to client-specific local models
Representation bias phenomenon

- After local training, the feature representations are **biased** to client-specific domains

![t-SNE visualization for representations before/after local training in FedAvg.](image)

We use *color* and *shape* to distinguish *labels* and *clients*, respectively. Representations form *client-specific domains* after local training.
Representation degeneration phenomenon

• At the same time, representations’ quality is also *degenerated*

Per-layer MDL (bits) for representations before/after local training in FedAvg.

A large MDL value means low representation quality.
Personalized FL (pFL)

- pFL methods learn personalized modules, but
Personalized FL (pFL)

• pFL methods learn personalized modules, but
• feature extractors are still trained with only biased local data domains on clients, leading to

![Diagram showing biased and not biased feature extractors before local training.](image-url)
Personalized FL (pFL)

• pFL methods learn personalized modules, but
• feature extractors are still trained with only biased local data domains on clients, leading to
• representation bias and representation degeneration during local training.
Our Domain Bias Eliminator (DBE)

• Thus, we propose DBE to *eliminate domain bias in representation space* via two modules:
Our Domain Bias Eliminator (DBE)

- Thus, we propose DBE to *eliminate domain bias in representation space* via two modules:
  - Personalized Representation Bias Memory (PRBM)
  - Mean Regularization (MR)
Personalized Representation Bias Memory (PRBM)

- PRBM stores personalized (biased) representation information ($\tilde{z}_i^p$) for each client, and
- make the remaining information ($z_i$) to be global.

![Diagram of Local model (original)]

![Diagram of Local model (with PRBM)]
Personalized Representation Bias Memory (PRBM)

- PRBM stores personalized *(biased)* representation information \( (\mathbf{z}_i^P) \) for each client, and
- make the remaining information \( (\mathbf{z}_i^G) \) to be global.

Local model (with PRBM)
Personalized Representation Bias Memory (PRBM)

- PRBM stores personalized (biased) representation information ($z_i^p$) for each client, and
- make the remaining information ($z_i^g$) to be global.
- Formally,

\[
\mathcal{L}_{D_i}(\theta) := \mathbb{E}_{(x_i, y_i) \sim D_i} [\ell(f(x_i; \theta^f), \theta^h; y_i)]
\]
Personalized Representation Bias Memory (PRBM)

- **PRBM** stores personalized *(biased)* representation information \((\bar{z}_i^p)\) for each client, and
- make the remaining information \((z_i^p)\) to be global.
- Formally,

Local loss (original):

\[
\mathcal{L}_{D_i}(\theta) := \mathbb{E}_{(x_i,y_i) \sim D_i} [\ell(h(f(x_i; \theta^f); \theta^h), y_i)]
\]

Local loss (with **PRBM**):

\[
\mathcal{L}_{D_i}(\theta_i) := \mathbb{E}_{(x_i,y_i) \sim D_i} [\ell(h(f(x_i; \theta^f) + \bar{z}_i^p; \theta^h), y_i)]
\]
Personalized Representation Bias Memory (PRBM)

- PRBM stores personalized (biased) representation information ($\bar{z}_i^p$) for each client, and
- make the remaining information ($z_i^p$) to be global.
- Formally,

Local loss (original):

$$\mathcal{L}_{D_i}(\theta) := \mathbb{E}_{(x_i,y_i) \sim D_i} [\ell(h(f(x_i;\theta^f);\theta^h), y_i)]$$

Local loss (with PRBM):

$$\mathcal{L}_{D_i}(\theta_i) := \mathbb{E}_{(x_i,y_i) \sim D_i} [\ell(h(f(x_i;\theta^f) + \bar{z}_i^p;\theta^h), y_i)]$$

View the PRBM as a personalized translation transformation $\text{PRBM} : \mathcal{Z} \mapsto \bar{\mathcal{Z}}$:
**Personalized Representation Bias Memory (PRBM)**

- PRBM stores personalized *(biased)* representation information \((\tilde{z}_i^p)\) for each client, and
- make the remaining information \((z_i^p)\) to be global.
- Formally,

  Local loss (original):
  \[
  \mathcal{L}_{D_i}(\theta) := \mathbb{E}_{(x_i,y_i) \sim D_i} [\ell(h(f(x_i; \theta^f); \theta^h), y_i)]
  \]

  Local loss (with PRBM):
  \[
  \mathcal{L}_{D_i}(\theta_i) := \mathbb{E}_{(x_i,y_i) \sim D_i} [\ell(h(f(x_i; \theta^f) + \tilde{z}_i^p; \theta^h), y_i)]
  \]

  View the PRBM as a personalized translation transformation **PRBM** : \(\mathcal{Z} \mapsto \mathcal{Z}\):

  \[
  \mathcal{L}_{D_i}(\theta_i) := \mathbb{E}_{(x_i,y_i) \sim D_i} [\ell(h(\text{PRBM}(f(x_i; \theta^f); \tilde{z}_i^p); \theta^h), y_i)]
  \]
Personalized Representation Bias Memory (PRBM)

• We make PRBM to be trainable to learn personalized representation information
Personalized Representation Bias Memory (PRBM)

- We make PRBM to be trainable to learn personalized representation information
- However, trainable PRBM requires guidance to recognize the global and personalized information
DBE: PRBM + Mean Regularization (MR)

- MR explicitly guides the local feature extractor to generate $z_i^g$ with global information, by
- further regularize $z_i^g$ to a globally shared *client-invariant mean* $\bar{z}^g$
DBE: PRBM + Mean Regularization (MR)

- MR explicitly guides the local feature extractor to generate $z_i^g$ with global information, by
- further regularize $z_i^g$ to a globally shared *client-invariant mean* $z^g$

A consensus obtained during the initialization period before FL

Local model (with PRBM and MR)
DBE: **PRBM + Mean Regularization (MR)**

- MR explicitly guides the local feature extractor to generate $z^g_i$ with global information, by
- further regularize $z^g_i$ to a globally shared **client-invariant mean** $\hat{z}^g$
DBE: PRBM + Mean Regularization (MR)

- MR explicitly guides the local feature extractor to generate $z_i^g$ with global information, by
- further regularize $z_i^g$ to a globally shared \textit{client-invariant mean} $\bar{z}^g$
- Formally,

Local loss (with PRBM):

$$L_{D_i}(\theta_i) := \mathbb{E}_{(x_i, y_i) \sim D_i}[\ell(h(\text{PRBM}(f(x_i; \theta^f); \bar{z}_i^p); \theta^h), y_i)]$$
DBE: PRBM + Mean Regularization (MR)

• MR explicitly guides the local feature extractor to generate $z_i^g$ with global information, by
• further regularize $z_i^g$ to a globally shared client-invariant mean $\bar{z}^g$
• Formally,

Local loss (with PRBM):
\[
\mathcal{L}_{D_i}(\theta_i) := \mathbb{E}_{(x_i, y_i) \sim D_i} [\ell(h(\text{PRBM}(f(x_i; \theta^f); \bar{z}_i^p); \theta^h), y_i)]
\]

Local loss (with PRBM and MR):
\[
\mathcal{L}_{D_i}(\theta_i) := \mathbb{E}_{(x_i, y_i) \sim D_i} [\ell(h(\text{PRBM}(f(x_i; \theta^f); \bar{z}_i^p); \theta^h), y_i)] + \kappa \cdot \text{MR}(\bar{z}_i^g, \bar{z}_g)
\]
DBE: PRBM + Mean Regularization (MR)

- MR explicitly guides the local feature extractor to generate $z^g_i$ with global information, by
- further regularize $z^g_i$ to a globally shared *client-invariant mean*.
- Formally,

Local loss (with PRBM):
\[
\mathcal{L}_{D_i}(\theta_i) := \mathbb{E}_{(x_i, y_i) \sim D_i} [\ell(h(\text{PRBM}(f(x_i; \theta^f_i); \tilde{z}_i^p); \theta^h_i), y_i)]
\]

Local loss (with PRBM and MR):
\[
\mathcal{L}_{D_i}(\theta_i) := \mathbb{E}_{(x_i, y_i) \sim D_i} [\ell(h(\text{PRBM}(f(x_i; \theta^f_i); \tilde{z}_i^p); \theta^h_i), y_i)] + \kappa \cdot \text{MR}(\tilde{z}_i^g, \tilde{z}^g)
\]

Final loss for client $i$
Improved Bi-directional Knowledge Transfer

• DBE can promote *bi-directional knowledge transfer* between server and client with
• *Theoretical guarantee*
Improved Bi-directional Knowledge Transfer

- **Local-to-global** knowledge transfer

**Corollary 1.** Consider a local data domain $D_i$ and a virtual global data domain $D$ for client $i$ and the server, respectively. Let $D_i = \langle U_i, c^* \rangle$ and $D = \langle U, c^* \rangle$, where $c^*: \mathcal{X} \mapsto \mathcal{Y}$ is a ground-truth labeling function. Let $H$ be a hypothesis space of VC dimension $d$ and $h: \mathcal{Z} \mapsto \mathcal{Y}, \forall h \in H$. When using DBE, given a feature extraction function $F^g: \mathcal{X} \mapsto \mathcal{Z}$ that shared between $D_i$ and $D$, a random labeled sample of size $m$ generated by applying $F^g$ to a random sample from $U_i$ labeled according to $c^*$, then for every $h^g \in H$, with probability at least $1 - \delta$:

$$
\mathcal{L}_D(h^g) \leq \mathcal{L}_{D_i}(h^g) + \sqrt{\frac{4}{m}(d \log \frac{2em}{d} + \log \frac{4}{\delta})} + d_H(\tilde{U}_i^g, \mathcal{U}^g) + \lambda_i,
$$

where $\mathcal{L}_{D_i}$ is the empirical loss on $D_i$, $e$ is the base of the natural logarithm, and $d_H(\cdot, \cdot)$ is the $H$-divergence between two distributions. $\lambda_i := \min_{h^g} \mathcal{L}_D(h^g) + \mathcal{L}_{D_i}(h^g)$, $\tilde{U}_i^g \subseteq \mathcal{Z}$, $\mathcal{U}^g \subseteq \mathcal{Z}$, and $d_H(\tilde{U}_i^g, \mathcal{U}^g) \leq d_H(\tilde{U}_i, \mathcal{U})$. $\tilde{U}_i^g$ and $\mathcal{U}^g$ are the induced distributions of $U_i$ and $U$ under $F^g$, respectively. $U_i$ and $U$ are the induced distributions of $U_i$ and $U$ under $F$, respectively. $F$ is the feature extraction function in the original FedAvg without DBE.
Improved Bi-directional Knowledge Transfer

- **Global-to-local** knowledge transfer

**Corollary 2.** Let $\mathcal{D}_i$, $\mathcal{D}$, $\mathcal{F}^g$, and $\lambda_i$ defined as in Corollary 1. Given a translation transformation function $\text{PRBM} : \mathcal{Z} \mapsto \mathcal{Z}$ that shared between $\mathcal{D}_i$ and virtual $\mathcal{D}$, a random labeled sample of size $m$ generated by applying $\mathcal{F}'$ to a random sample from $\mathcal{U}_i$ labeled according to $c^*$, $\mathcal{F}' = \text{PRBM} \circ \mathcal{F}^g : \mathcal{X} \mapsto \mathcal{Z}$, then for every $h' \in \mathcal{H}$, with probability at least $1 - \delta$:

$$
\mathcal{L}_{\mathcal{D}_i}(h') \leq \mathcal{L}_{\mathcal{D}}(h') + \sqrt{\frac{4}{m} (d \log \frac{2em}{d} + \log \frac{4}{\delta})} + d_H(\tilde{\mathcal{U}}', \tilde{\mathcal{U}}_i') + \lambda_i,
$$

where $d_H(\tilde{\mathcal{U}}', \tilde{\mathcal{U}}_i') = d_H(\tilde{\mathcal{U}}^g, \tilde{\mathcal{U}}^g_i) \leq d_H(\tilde{\mathcal{U}}, \tilde{\mathcal{U}}_i) = d_H(\tilde{\mathcal{U}}_i, \tilde{\mathcal{U}})$. $\tilde{\mathcal{U}}'$ and $\tilde{\mathcal{U}}_i'$ are the induced distributions of $\mathcal{U}$ and $\mathcal{U}_i$ under $\mathcal{F}'$, respectively.

Please refer to our paper for proofs.
## Extensive Experiments

- How to Split the Model?

Table 1: The MDL (bits, ↓) of layer-wise representations, test accuracy (%, ↑), and the number of trainable parameters (↓) in PRBM when adding DBE to FedAvg on Tiny-ImageNet using 4-layer CNN in the practical setting. We also show corresponding results for the close pFL methods. For FedBABU, “[36.82]” indicates the test accuracy after post-FL fine-tuning for 10 local epochs.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>MDL</th>
<th>Accuracy</th>
<th>Param.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CONV1→CONV2</td>
<td>CONV2→FC1</td>
<td>FC1→FC2</td>
</tr>
<tr>
<td>FedPer [3]</td>
<td>5143</td>
<td>4574</td>
<td>3885</td>
</tr>
<tr>
<td>FedRep [20]</td>
<td>5102</td>
<td>4237</td>
<td>3922</td>
</tr>
<tr>
<td>FedRoD [14]</td>
<td>5063</td>
<td>4264</td>
<td>3783</td>
</tr>
<tr>
<td>FedBABU [61]</td>
<td>5083</td>
<td>4181</td>
<td>3948</td>
</tr>
<tr>
<td><strong>Original (FedAvg)</strong></td>
<td><strong>5081</strong></td>
<td><strong>4151</strong></td>
<td><strong>3844</strong></td>
</tr>
</tbody>
</table>

|                      |                     |                 |         |
| CONV1→DBE→CONV2     | 4650 (-8.48%)       | 4105 (-1.11%)   | 3679 (-4.29%) | 3756 (-3.57%) | 21.81 (+2.35) | 28800 |
| CONV2→DBE→FC1       | 4348 (-14.43%)      | 3716 (-10.48%)  | 3463 (-9.91%) | 3602 (-7.52%) | 47.03 (+27.57) | 10816 |
| FC1→DBE→FC2         | 4608 (-9.31%)       | **3689 (-11.13%)** | 3625 (-5.70%) | 3688 (-5.31%) | 43.32 (+23.86) | 512   |
Extensive Experiments

• Eliminate Representation Bias for the First Level of Representation \textit{after} local training

Figure 3: t-SNE visualization for representations on Tiny-ImageNet (200 labels). “B” and “A” denote “before local training” and “after local training”, respectively. We use \textit{color} and \textit{shape} to distinguish \textit{labels} and \textit{clients}, respectively. \textit{Best viewed in color and zoom-in.}
Extensive Experiments

• DBE can greatly improve existing FL methods in both generalization and personalization abilities
Extensive Experiments

- **DBE** promotes traditional FL methods in both MDL and accuracy by at most
  - -22.35% in MDL (bits) and
  - +32.30 in accuracy (%)

Table 4: The MDL (bits, ↓) and test accuracy (%, ↑) before and after adding DBE to traditional FL methods on Cifar100, Tiny-ImageNet, and AG News in the practical setting. TINY and TINY* represent using 4-layer CNN and ResNet-18 on Tiny-ImageNet, respectively.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>MDL</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cifar100</td>
<td>TINY</td>
</tr>
<tr>
<td>SCAFFOLD [38]</td>
<td>1499</td>
<td>3661</td>
</tr>
<tr>
<td>FedProx [46]</td>
<td>1523</td>
<td>3701</td>
</tr>
<tr>
<td>MOON [45]</td>
<td>1516</td>
<td>3696</td>
</tr>
<tr>
<td>FedGen [96]</td>
<td>1506</td>
<td>3675</td>
</tr>
<tr>
<td>SCAFFOLD+DBE</td>
<td>1434</td>
<td>3549</td>
</tr>
<tr>
<td>FedProx+DBE</td>
<td>1439</td>
<td>3587</td>
</tr>
<tr>
<td>MOON+DBE</td>
<td>1432</td>
<td>3580</td>
</tr>
<tr>
<td>FedGen+DBE</td>
<td>1426</td>
<td>3563</td>
</tr>
</tbody>
</table>
Extensive Experiments

- DBE greatly improves FedAvg at most $+47.40$ on Cifar100 in the pathological setting and
- outperforms the SOTA pFL methods by up to $+11.36$ on Cifar100

Table 5: The test accuracy ($\%$, $\uparrow$) of pFL methods in two statistically heterogeneous settings. Cifar100$\dagger$ represents the experiment with 100 clients and joining ratio $\rho = 0.5$ on Cifar100.

<table>
<thead>
<tr>
<th>Settings</th>
<th>Pathological setting</th>
<th>Practical setting</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FMNIST</td>
<td>Cifar100</td>
<td>TINY</td>
<td>FMNIST</td>
<td>Cifar100</td>
<td>Cifar100$\dagger$</td>
<td>TINY</td>
<td>TINY$^*$</td>
</tr>
<tr>
<td>Per-FedAvg [22]</td>
<td>99.18</td>
<td>56.80</td>
<td>28.06</td>
<td>95.10</td>
<td>44.28</td>
<td>38.28</td>
<td>25.07</td>
<td>21.81</td>
</tr>
<tr>
<td>pFedMe [67]</td>
<td>99.35</td>
<td>58.20</td>
<td>27.71</td>
<td>97.25</td>
<td>47.34</td>
<td>31.13</td>
<td>26.93</td>
<td>33.44</td>
</tr>
<tr>
<td>Ditto [47]</td>
<td>99.44</td>
<td>67.23</td>
<td>39.90</td>
<td>97.47</td>
<td>52.87</td>
<td>39.01</td>
<td>32.15</td>
<td>35.92</td>
</tr>
<tr>
<td>FedPer [3]</td>
<td>99.47</td>
<td>63.53</td>
<td>39.80</td>
<td>97.44</td>
<td>49.63</td>
<td>41.21</td>
<td>33.84</td>
<td>38.45</td>
</tr>
<tr>
<td>FedRep [20]</td>
<td>99.56</td>
<td>67.56</td>
<td>40.85</td>
<td>97.56</td>
<td>52.39</td>
<td>41.51</td>
<td>37.27</td>
<td>39.95</td>
</tr>
<tr>
<td>FedRoD [14]</td>
<td>99.52</td>
<td>62.30</td>
<td>37.95</td>
<td>97.52</td>
<td>50.94</td>
<td>48.56</td>
<td>36.43</td>
<td>37.99</td>
</tr>
<tr>
<td>FedBABU [61]</td>
<td>99.41</td>
<td>66.85</td>
<td>40.72</td>
<td>97.46</td>
<td>55.02</td>
<td>52.07 **</td>
<td>36.82</td>
<td>34.50</td>
</tr>
<tr>
<td>APFL [21]</td>
<td>99.41</td>
<td>64.26</td>
<td>36.47</td>
<td>97.25</td>
<td>46.74</td>
<td>39.47</td>
<td>34.86</td>
<td>35.81</td>
</tr>
<tr>
<td>APPLE [52]</td>
<td>99.30</td>
<td>65.80</td>
<td>36.22</td>
<td>97.06</td>
<td>53.22</td>
<td>—</td>
<td>35.04</td>
<td>39.93</td>
</tr>
<tr>
<td>FedAvg+DBE</td>
<td><strong>99.74</strong></td>
<td><strong>73.38</strong></td>
<td><strong>42.89</strong></td>
<td><strong>97.69</strong></td>
<td><strong>64.39</strong></td>
<td><strong>63.43</strong></td>
<td><strong>43.32</strong></td>
<td><strong>42.98</strong></td>
</tr>
</tbody>
</table>
Extensive Experiments

- Other experiments also show the **effectiveness** and **efficiency** of our DBE.

<table>
<thead>
<tr>
<th>Items</th>
<th>Heterogeneity</th>
<th>pFL+MR</th>
<th>Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>β = 0.01</td>
<td>β = 0.5</td>
<td>β = 5</td>
</tr>
<tr>
<td></td>
<td>Accuracy</td>
<td>Improvement</td>
<td>Total time</td>
</tr>
<tr>
<td>Per-FedAvg [22]</td>
<td>39.39</td>
<td>21.14</td>
<td>12.08</td>
</tr>
<tr>
<td>pFedMe [67]</td>
<td>41.45</td>
<td>17.48</td>
<td>4.03</td>
</tr>
<tr>
<td>Ditto [47]</td>
<td>50.62</td>
<td>18.98</td>
<td>21.79</td>
</tr>
<tr>
<td>FedPer [3]</td>
<td>51.83</td>
<td>17.31</td>
<td>9.61</td>
</tr>
<tr>
<td>FedRoD [14]</td>
<td>49.17</td>
<td>23.23</td>
<td>16.71</td>
</tr>
<tr>
<td>FedBABU [61]</td>
<td>53.97</td>
<td>23.08</td>
<td>15.42</td>
</tr>
<tr>
<td>APFL [21]</td>
<td>49.96</td>
<td>23.31</td>
<td>16.12</td>
</tr>
<tr>
<td>FedFomo [89]</td>
<td>46.36</td>
<td>11.59</td>
<td>14.86</td>
</tr>
<tr>
<td>APPLE [52]</td>
<td>47.89</td>
<td>24.24</td>
<td>17.79</td>
</tr>
<tr>
<td>FedAvg</td>
<td>15.70</td>
<td>21.14</td>
<td>21.71</td>
</tr>
<tr>
<td>FedAvg+DBE</td>
<td><strong>57.52</strong></td>
<td><strong>32.61</strong></td>
<td><strong>25.55</strong></td>
</tr>
</tbody>
</table>
Eliminating Domain Bias for Federated Learning in Representation Space

Paper with code: https://github.com/TsingZ0/DBE
E-mail: tsingz@sjtu.edu.cn

Thanks!