A Implementation of Method

A.1 Proof of Theorem 1

Theorem 2. *Let the global optimal representation for class c denote $g_c^* = [a_{c,1}^*, ..., a_{c,d}^*]$, and $z_k^{c,t}$ denote the representation of sample x in the class c of the k-th client. Assuming that $\forall i$, both $|a_{c,i}^*|$ and $|z_{k,i}^{c,t}|$ are upper bounded by G, and all dimensions are disentangled. Then, in round t, the i-th dimension of local representation $z_k^{c,t}$ satisfies*

$$|z_{k,i}^{c,t} - a_{c,i}^*| \leq 2(1 - \hat{p}_k^c \Gamma)G + \delta\Gamma,$$

where $\hat{p}_k^c$ is the accumulation regarding the i-th dimension of the class-c prototype, $\Gamma = \frac{1-(p_k^c)^t}{1-p_k^c}$ and δ is the maximum margin induced by the optimization of the inter-loss term.

Proof. If the global optimum satisfies:

$$f_1(x_c; w_{*,1}) = g_c^* = [a_{c,1}^*, ..., a_{c,d}^*],$$

we could define the local optimum of class c in client k:

$$f_1(x_c; w_{k,1}^*) = z_k^c = [a_{c,1}^*, ..., a_{c,s}^*, \sigma_{k,s+1}^c, ..., \sigma_{k,s+r}^c],$$

where $\{a_{c,1}^*, ..., a_{c,s}^*\}$ denotes the dimensions relative to classifying class c with other seen classes in client k. Since the intra-class loss decorrelates each dimensions of the feature space, $\{\sigma_{k,s+1}^c, ..., \sigma_{k,s+r}^c\}$ are free dimensions and irrelevant to the seen class. By minimizing the inter-class loss

$$D_{c_i,c_j} = \frac{1}{N_k^c} \sum_{n=1}^{N_k^c} \max\{||z_{k,n}^{c_i} - g_{c_i}^t|| - ||z_{k,n}^{c_i} - g_{c_j}^t)||, 0\}.$$

Since all dimensions are irrelevant (achieved by intra-class loss), we can reach the optimum of each dimension of feature representations in the limit, which has the following property:

$$||(z_k^{c_i})_m - g_{c_i,m}^t|| \leq \min\{||(z_k^{c_i})_m - g_{c_j,m}^t||\}, \ \forall i \neq j,$$

where m denote one of the dimensions. In terms of the representation space of a certain class c, we have the dimensions $\{a_{c,1}^*, ..., a_{c,s}^*\}$ relevant to the seen classes in the local client reach the optimal. In this case, we can rewrite the locality of $\{\sigma_{k,s+1}^c, ..., \sigma_{k,s+r}^c\}$ by introducing a slack variable as follows:

$$\sigma_{k,s+j}^{c,t+1} - g_{c,s+j}^t = \xi^t,$$

where ξ^t is the induced slack variable. The $s+j$ -th element of class prototypes of class c can be written as:

$$g_{c,s+j}^t = \hat{p}_k^c a_{c,s+j}^* + p_k^c \sigma_{k,s+j}^t + \sum_{k'} p_{k'}^c \sigma_{k',s+j}^t,$$

where $\hat{p}_k^c$ denotes the proportions of a subset of clients that can provide the support information of these dimensions. Note that, such support is related to class c to distinguish unseen classes in client k. $p_{k'}$ is the clients can not provide the support information except for client k ($\hat{p}_k^c + p_{k'}^c + p_k^c = 1$). Putting all things together, we could have the following equation:

$$\sigma_{k,s+j}^{c,t+1} = \hat{p}_k^c a_{c,s+j}^* + p_k^c \sigma_{k,s+j}^t + \sum_{k'} p_{k'}^c \sigma_{k',s+j}^{c,t} + \xi^t.$$

Let $\sigma_{k,s+j}^{c,t+1}$ be r_{t+1} for simplicity. We will have

$$r_{t+1} = \hat{p}_k^c a_{k,s+j}^* + p_k^c r_t + \sum_{k'} p_{k'}^{c,t} + \xi^t.$$

With the recursive iteration to r_0, we can compute that

$$r_{t+1} = \frac{1-(p_k^c)^t}{1-p_k^c}\hat{p}_k^c a_{k,s+j}^* + (p_k^c)^t r_0 + A + B,$$

where A is defined as

$$A = \sum_{k'} p_{k'}^{c,t}\sigma_{k',s+j}^{c,t} + ... + (p_k^c)^t \sum_{k'} p_{k'}^{c,0}\sigma_{k',s+j}^{c,0},$$

and B is defined as

$$B = \xi^t + p_k^c\xi^{t-1} + ... + (p_k^c)^t\xi^0,$$

If $|\xi^t|$ is bounded by δ and $\sigma_{k',s+j}^{c,t}$ is bounded by G for all t, we have the inequality:

$$|A| \leq \Gamma(1-\hat{p}_k^c - p_k^c)G,$$

and

$$|B| \leq \Gamma\delta,$$

where Γ is defined as:

$$\Gamma = \frac{1-(p_k^c)^t}{1-p_k^c}.$$

Therefore, we have the following bound

$$\begin{aligned}
|r_{t+1} - a_{c,s+j}^*| =& |\Gamma\hat{p}_k^c a_{k,s+j}^* - a_{c,s+j}^* + (p_k^c)^t r_0 + A + B| \\
\leq& (1-\hat{p}_k^c\Gamma)G + (p_k^c)^t G + |A| + |B| \\
\leq& 2(1-\hat{p}_k^c\Gamma)G + \Gamma\delta.
\end{aligned}$$

In the above first inequality, we use $|a+b+c| \leq |a| + |b| + |c|$. And in the above second inequality, we combine the similar terms in A and B. As $z_{k,i}^{c,t} = g_{c,i}^*$ for i=1,2,...,s, we universally have

$$|z_{k,i}^{c,t} - a_{c,i}^*| \leq 2(1-\hat{p}_k^c\Gamma)G + \delta\Gamma,$$

which completes the proof. □

A.2 Proof of Lemma 1

Lemma 2. *Assuming a covariance matrix $M \in \mathbf{R}^{d\times d}$ computed from the feature of each sample with the standard normalization, and its eigenvalues $\{\lambda_1, \lambda_2, ..., \lambda_d\}$, we will have the following equality that satisfied*

$$\sum_{i=1}^{d}(\lambda_i - \frac{1}{d}\sum_{j=1}^{d}\lambda_j)^2 = ||M||_F^2 - d.$$

Proof. On the right-hand side, we have

$$\begin{aligned}
||M||_F^2 - d =& Tr((M)^T M - d) \\
=& Tr(U\sum U^T U\sum U^T) - d \\
=& \sum_{i=1}^{d}\lambda_i^2 - d.
\end{aligned}$$

As we have applied the standard normalization on M, we have the the characteristic of eigenvalues $\frac{1}{d}\sum_{j=1}^{d}\lambda_j = 1$. Then for the right-hand side, we can have the deduction

$$\begin{aligned}\sum_{i=1}^{d}(\lambda_i - \frac{1}{d}\sum_{j=1}^{d}\lambda_j))^2 &= \sum_{i=1}^{d}(\lambda_i - 1)^2 \\ &= \sum_{i=1}^{d}(\lambda_i^2 - 2\lambda_i + 1) \\ &= \sum_{i=1}^{d}\lambda_i^2 - d.\end{aligned} \tag{6}$$

This constructs the equality of the left-hand side and the right-hand side, which completes the proof. □

B Simulation and Visualization

To show the dimensional collapse and verify our method, we simulate data for four categories, and samples of each category are generated from a circle with a different center and a radius of 0.5. We set centers (1,1), (1,-1), (-1,1) and (-1,-1) for class 1 to 4, respectively. The input is the two-dimensional position, and we adopt a three-layer MLP model with hidden size 128 and 3. We visualize the outputs of the second layer (3 dimensions) and project them onto an unit sphere. In the simulation, we adopt SGD with learning rate 0.1 and generate 5000 samples for each category, and the number of iterations and batch size are set to 50 and 128. In the paper, we train the model and show the visualization of models trained on four classes of samples, two classes of samples and two classes of samples with FedMR to show the global feature space, collapsed feature space and reshaped feature space, respectively.

C Implementation of Experiment

C.1 Models and Datasets

C.1.1 Models

For FMNIST, SVHN and CIFAR10, we adopt modified version of ResNet18 as previous studies (Li et al. (2021b); Zhang et al. (2023b); Ye et al. (2023a); Yao et al. (2022)). For CIFAR100, we adopt wide ResNet (https://github.com/meliketoy/wide-resnet.pytorch). For ISIC2019 dataset, we adopt EfficientNet b0 (Tan & Le (2019)) as the same of Flamby benchmark (Terrail et al. (2022)).

C.1.2 Partition Strategies

Dirichlet distribution (Dir(β) is *not suitable* to split data for pure PCDD problems (some classes are partially missing). As an exemplar simulation in Figure 6, Dirichlet allocation usually generates diverse *imbalance* data coupled with occasionally PCDD. However, we do not focus on the locally-imbalance of each client but the *locally-balance* of each client from limited existing categories. This is the intuition difference from a Dirichlet allocation. Therefore, taking an example of CIFAR100 (P10C30), to simulate pure PCDD situation, we first divide all 100 classes to the client in order: 1-30 categories for client 1, 31-60 categories for client 2, 61-90 categories for client 3, and 91-100 categories for client 4. Then the remain clients that lacking categories (less than 30 classes) random choose categories from 1 to 100. After slicing the categories, the samples of each class are equally divided into clients that have such class. Such partition strategy can ensure the difference of class distributions among clients, all categories are allocated and the sample numbers are roughly the same.

C.1.3 ISIC2019 dataset

Table 9: Best hyper-parameters tuned from 0.000001 to 1 of FedMR and a range of state-of-the-art approaches on four datasets under PCDD partitions. Datasets are divided into ϱ clients and each client has ς classes (denoted as $P\varrho C\varsigma$).

Datasets	Split	FedProx	FedProc	MOON	FedDyn	FedDC	FedMR (μ_1)	FedMR (μ_2)
FMNIST	P5C2	0.01	0.00001	0.001	0.0001	0.001	0.01	0.0001
	P10C2	0.01	0.00001	0.0001	0.0001	0.001	0.01	0.0001
	P10C3	0.01	0.00001	0.001	0.0001	0.0001	0.01	0.0001
	P10C5	0.01	0.001	0.001	0.0001	0.001	0.01	0.0001
	IID	0.01	0.00001	0.01	0.0001	0.01	0.01	0.0001
SVHN	P5C2	0.001	0.0001	0.001	0.000001	0.0001	0.001	0.0001
	P10C2	0.001	0.0001	0.001	0.000001	0.0001	0.001	0.0001
	P10C3	0.001	0.0001	0.001	0.000001	0.0001	0.001	0.0001
	P10C5	0.001	0.0001	0.1	0.000001	0.0001	0.001	0.0001
	IID	0.001	0.0001	0.1	0.00001	0.0001	0.001	0.0001
CIFAR10	P5C2	0.01	0.001	0.01	0.0001	0.0001	0.1	0.001
	P10C2	0.01	0.0001	0.0001	0.0001	0.0001	0.1	0.001
	P10C3	0.01	0.001	0.0001	0.0001	0.0001	0.1	0.0001
	P10C5	0.01	0.001	0.01	0.0001	0.00001	0.1	0.001
	IID	0.01	0.001	1	0.0001	0.001	0.1	0.0001
CIFAR100	P10C10	0.01	0.001	0.1	0.0001	0.0001	0.1	0.0001
	P10C20	0.01	0.001	0.1	0.0001	0.0001	0.1	0.0001
	P10C30	0.01	0.001	0.1	0.0001	0.0001	0.1	0.0001
	P10C50	0.01	0.001	0.1	0.0001	0.0001	0.1	0.0001
	IID	0.01	0.001	0.1	0.0001	0.0001	0.1	0.00001

Table 10: Best method-specific hyper-parameters (weights of proximal term in FedProx, contrastive loss in MOON and so forth) tuned from 0.000001 to 1 of FedProx, MOON, FedMR and so forth on CIFAR10 and CIFAR100 under larger scale and a real-world application: ISIC2019. Datasets are divided into ϱ clients and each client has ς classes (denoted as $P\varrho C\varsigma$).

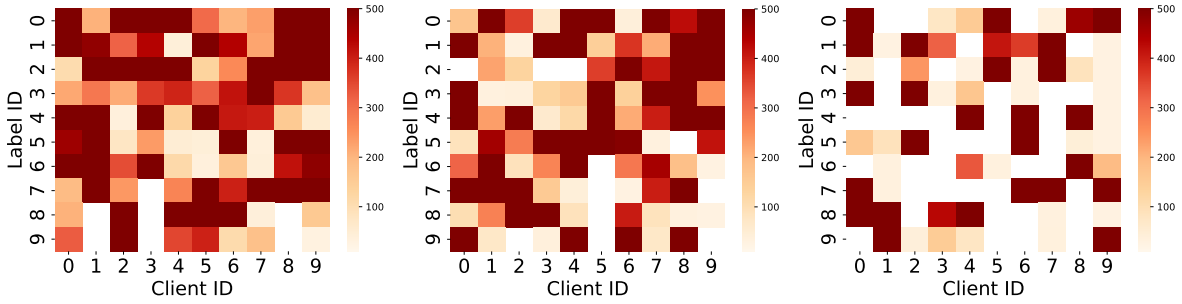
Datasets	Split	FedProx	FedProc	MOON	FedDyn	FedDC	FedMR (μ_1)	FedMR (μ_2)
CIFAR10	P10C3	0.01	0.001	0.0001	0.0001	0.0001	0.1	0.0001
	P50C3	0.0001	0.0001	0.1	0.0001	0.00001	0.01	0.0001
	P100C3	0.0001	0.0001	0.1	0.0001	0.0001	0.01	0.0001
CIFAR100	P10C10	0.01	0.001	0.1	0.0001	0.0001	0.1	0.0001
	P50C10	0.0001	0.001	0.1	0.00001	00001	0.01	0.0001
	P100C10	0.0001	0.0001	0.1	0.00001	0.0001	0.01	0.0001
ISIC2019	Real	0.001	0.0001	0.1	0.0001	0.0001	0.001	0.001

Table 11: Computation times and performance on P5C2 of FMNIST, SVHN and CIFAR10 and P10C10 of CIFAR100.

Dataset	**FedAvg**	**n=0**	**n=10**	**n=50**	**n=128**
FMNIST	67.29/15.52s	71.21/17.78s	76.51/21.43s	75.67/31.56s	75.51/54.07s
SVHN	81.85/22.15s	82.74/24.11s	83.51/28.31s	83.48/57.25s	83.10/88.20s
CIFAR10	67.68/15.74s	72.26/17.70s	73.19/24.56s	73.56/47.26s	74.19/60.07s
CIFAR100	54.31/35.12s	55.38/42.14s	55.57/52.18s	56.35/105.34s	57.27/195.12s

Table 12: Performance of FedAvg, inter-class loss, intra-class loss and both inter-class and inter-class losses (FedMR).

Datasets	Split	FedAvg	Inter	Intra	FedMR
FMNIST	P5C2	0.6729	0.6717	0.7121	0.7497
	P10C2	0.6733	0.6962	0.7028	0.7552
	P10C3	0.8167	0.8241	0.8282	0.8324
	P10C5	0.8953	0.8959	0.8899	0.9004
	IID	0.9193	0.9210	0.9185	0.9215
SVHN	P5C2	0.8185	0.8257	0.8274	0.8310
	P10C2	0.7892	0.8138	0.8148	0.8247
	P10C3	0.8770	0.8886	0.8807	0.8913
	P10C5	0.9120	0.9164	0.9086	0.9209
	IID	0.9274	0.9283	0.9259	0.9304
CIFAR10	P5C2	0.6768	0.6775	0.7226	0.7419
	P10C2	0.6727	0.6766	0.7267	0.7332
	P10C3	0.7782	0.7784	0.8051	0.8275
	P10C5	0.8822	0.8851	0.8766	0.8906
	IID	0.9188	0.9253	0.9135	0.9306
CIFAR100	P10C10	0.5431	0.5538	0.5615	0.5727
	P10C20	0.6481	0.6486	0.6512	0.6581
	P10C30	0.6951	0.6905	0.7021	0.7024
	P10C50	0.7128	0.7129	0.7150	0.7217
	IID	0.7228	0.7252	0.7259	0.7279



(a) $\beta = 1.$ (b) $\beta = 0.5.$ (c) $\beta = 0.1.$

Figure 6: Heatmaps of data distribution of CIFAR10 generated by Dir(β).

As shown in the Figure 5, we show the heat map of data distribution of ISIC2019 dataset (Codella et al. (2018); Tschandl et al. (2018); Combalia et al. (2019)). As can be seen, there are multiple statistical heterogeneity problems including partially class-disjoint data (PCDD).

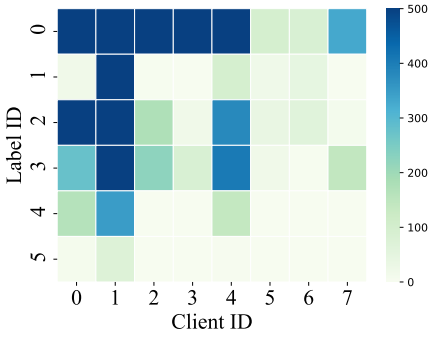


Figure 5: Heat map of the data distribution on ISIC2019 dataset.

C.2 Parameters

As for model-common parameters like optimizer, lr and batch-size are all aligned. We have verified $lr = 0.01$ is stable and almost the best for all methods in our settings. For method specific parameters like proximal term of FedProx, reshaping

Table 13: Performance of FedMR on two real-world datasets named HyperKvasir and ODIR under three PCDD situations.

Datasets	Split	FedAvg	FedProx	MOON	FedDyn	FedProc	FedMR	Δ
HyperKvasir	P10C2	69.67	68.43	69.33	70.02	69.54	70.79	+0.75
	P10C3	76.44	76.92	77.43	77.01	76.72	78.32	+0.89
	P10C5	89.48	89.23	89.13	89.34	89.27	89.54	+0.06
ODIR	P3C3	59.89	60.24	59.85	60.78	60.88	61.79	+0.91
	P4C3	56.53	56.42	56.89	57.12	55.78	57.67	+0.55
	P5C5	54.51	54.07	54.11	54.17	53.73	54.62	+0.21

Table 14: Results of FedMR on CIFAR10 when tuning μ_1 of intra-class loss under the same μ_2 of inter-class loss.

Split	$\mu_1 = 1$	$\mu_1 = 0.1$	$\mu_1 = 0.01$	$\mu_1 = 0.001$
P5C2	73.72	74.19	73.85	73.19
P10C2	72.99	73.32	73.00	73.12
P10C3	81.65	82.75	82.05	80.64
P10C5	88.76	89.06	88.96	88.64
P10C10	93.02	93.06	93.04	92.16

loss of FedMR, contrastive loss of MOON, dynamic regulazer of FedDyn and so forth are carefully searched and shown in the following.

C.2.1 Parameters in 4.2

We use grid search from 0.000001 to 1 (interval 10) to find best method-specific parameters of all method on different datasets as shown in Table 9. The total rounds are 100 for FMNIST, SVHN and CIFAR10 and 200 for CIFAR100.

C.2.2 Parameters in 4.3

We use grid search from 0.000001 to 1 (interval 10) to find best method-specific parameters parameters of FedProx, MOON, FedMR and so forth on different datasets as shown in Table 10. The total rounds are 100 for CIFAR10 (P10C3), 200 for CIFAR10 (P50C3) and 400 for CIFAR10 (P100C3) and CIFAR100.

C.3 More results on real-world datasets

We additionally test our FedMR on two more datasets, named HyperKvasir (Borgli et al. (2020)) and ODIR (Li et al. (2021a)) under three partitions. As shown in Table 13, our method achieves the best average improvement of 1.02 and 1.05 relative to FedAvg and of 0.57 and 0.56 relative to best baseline on the HyperKvasir and ODIR respectively.

C.4 More about intra-class loss

Since there are two additional loss in our method, it might raise heavy tuning problem. As shown in Table 14, we record the results when tuning μ_1 under the same μ_2 on CIFAR10 and empirically observe that the performance is good and stable for μ_1 from a large range, which means we only need to carefully tune μ_2. What's more, we also demonstrate the value of intra-class loss to verify the effect of deccorelation during the federated training. As shown in the Figure 7, we could see that intra-class loss converges and maintains a relatively low value easily, supporting our assumption in the Theorem 1 that the dimensions are decorrelated well by intra-class loss.

Table 15: Performance on SVHN under partitions generated by dirichlet distribution.

Dataset	Method	Full Participation(10 clients)			Partial Participation(50 clients)		
	#Partition	IID	$\beta = 0.5$	$\beta = 0.1$	IID	$\beta = 0.5$	$\beta = 0.2$
SVHN	FedAvg	92.74	91.24	75.24	91.29	89.29	84.70
	Best Baseline	93.50	92.46	76.26	91.67	91.27	87.78
	FedMR(Ours)	93.04	92.50	78.19	91.77	91.31	89.37

C.5 Performance on General Setting

To verify the effectiveness, we conduct the experiments on SHAKESPEARE (Shakespeare et al. (1989)) and SVHN, whose data distributions follow the non-PCDD setting. Specially, SHAKESPEARE is also commonly used as real-world data heterogeneity challenge in federated learning. We select 50 clients of the SHAKESPEARE into federated training and our method outperforms all methods and achieves improvement of 3.86% to FedAvg and 1.10% to the best baseline. As for SVHN, we split it by Dirichlet distribution as many previous FL works (Ye et al. (2023b); Li et al. (2020b); Fan et al. (2022)). According to the results in the Table 15, we can find FedMR still remains applicable and achieves comparable performance.

C.6 Lite Version

Here we introduce our light version for FedMR. Since our reshaping loss will introduce additional computation times, we randomly select part of samples in the mini-batch to compute inter-class loss for a computation friendly version. In Table 11, we randomly select 0, 10, 50 and 128 samples in the mini-batch on the four dataset (P10C10 for CIFAR100 and P5C2 for the others) to calculate the inter-class loss and record the accuracy. As can be seen, the computation time is reduced significantly and simultaneously maintains the competing performance.

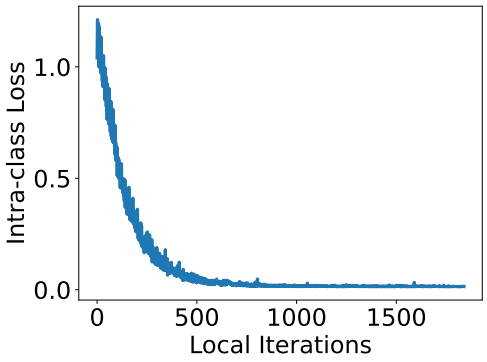


Figure 7: Intra-class loss value of our FedMR during the federated training.

C.7 Ablation

In this part, we provide detailed results of ablation on FMNIST, SVHN, CIFAR10 and CIFAR100. As shown in the Table 12, the intra-class loss generally plays a more important role in the performance improvement of FedMR on four datasets under PCDD. Their combination complements each other and thus shows a best improvement than any of the single loss, confirming our design from the joint perspective to prevent the collapse under PCDD.