Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Supplementary materials for

Ignatius IWAN, Bernardo Nugroho YAHYA, Seok-Lyong LEE, 2025. Federated model with contrastive learning and adaptive control variates for human activity recognition. *Front Inform Technol Electron Eng*, 26(6): 896-911. https://doi.org/10.1631/FITEE.2400797

1 Extension of the experiment section

This section covers the details of the experiment section such as performance measures, comparisons with pretraining methods, and the comparison of embedding visualization and communication exchange.

1.1 Performance measures

We used accuracy and F1-scores as the performance metrics. Let tp, tn, fp and fn denote the numbers of true positives, true negatives, false positives, and false negatives, respectively. Accuracy is defined as the percentage of tp divided by the number of all predictions for all classes. Let $n\mathrm{class}$ represent the total number of classes in the benchmark dataset and o the o-th class, then the accuracy equation can be written as Eq (S1):

$$\mathrm{Accuracy}=\frac{\sum_{o=1}^{n\mathrm{class}} tp_o}{\sum_{o=1}^{n\mathrm{class}}(\mathrm{tp}_o+\mathrm{tn}_o+\mathrm{fp}_o+\mathrm{fn}_o)}. \tag{S1}$$

To calculate the F1-score, precision and recall need to be calculated first. Precision measures the model performance to identify instances of a particular class o correctly. The equation for precision is Eq (S2):

$$\mathrm{Precision}=\frac{\sum_{o=1}^{n\mathrm{class}} \mathrm{tp}_o}{\sum_{o=1}^{n\mathrm{class}}(\mathrm{tp}_o+\mathrm{fp}_o)}. \tag{S2}$$

Meanwhile, recall measures the model performance to identify all instances of a particular class o. The equation for recall is Eq (S3):

$$\mathrm{Recall}=\frac{\sum_{o=1}^{n\mathrm{class}} \mathrm{tp}_o}{\sum_{o=1}^{n\mathrm{class}}(\mathrm{tp}_o+\mathrm{fn}_o)}. \tag{S3}$$

The F1-score metric serves as a good balance between precision and recall. It is less affected

by class imbalance than accuracy. Eq (S4) shows the formula for calculating the F1-score:

$$F1 - \text{score} = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}. \qquad \text{(S4)}$$

1.2 Comparison with pretraining methods

This section compares the performance of FedCoad with state-of-the-art HAR methods that use pretraining of the encoder model and which is then fine-tuned to the clients. As the baseline pretraining methods require labeled data for fine-tuning purposes, 20% of the test data is partitioned for baseline fine-tuning. Table S1 shows the results of FedCoad and pretraining methods.

Table S1 Comparison of the performance of pretraining methods

Method	Motionsense		WISDM		HHAR	
	Accuracy %	F1-score %	Accuracy %	F1-score %	Accuracy %	F1-score %
MetaHAR	81.00	75.60	83.91	75.83	83.46	82.25
ModCL	92.71	89.02	86.02	74.13	94.10	93.64
FedCoad (μ=1.0)	95.52	92.70	76.58	73.55	83.78	82.81
FedCoad (μ=5.0)	93.47	90.61	76.67	73.17	84.31	83.22
FedCoad (μ=10.0)	94.78	91.67	76.46	71.91	84.06	82.91

The FedCoad method significantly outperformed pretraining methods in the Motionsense dataset. In terms of accuracy, FedCoad (μ=1.0) achieved 95.52% and outperformed MetaHAR by 14.52 percentage points and the state-of-the-art ModCL by 2.81 percentage points. The results also showed the supremacy of FedCoad in terms of the F1-score. For example, FedCoad (μ=1.0) with an F1-score 92.52% significantly outperformed MetaHAR by 17.1 percentage points and ModCL by 3.68 percentage points. Therefore, FedCoad can outmatch pretraining methods without access to fine-tuning data in the Motionsense dataset.

In the WISDM dataset, the F1-score performance of FedCoad was on par with the pretraining methods. For example, the FedCoad (μ=1.0) f1-score of 73.55% was only 2.28 percentage points lower than that of Meta-HAR. However, the pretraining methods were more accurate than FedCoad. For instance, ModCL achieved 86.02% accuracy, and outperformed FedCoad (μ=5.0) by 9.35% percentage points.

In the HHAR dataset, the performance of FedCoad was similar to that of Meta-HAR. In terms of accuracy, FedCoad (μ=5.0) achieved 84.31% and even outperformed MetaHAR by 0.85 percentage points. Meanwhile, FedCoad (μ=5.0) achieved an F1-score of 83.22%, exceeding that of MetaHAR by 0.97 percentage points. However, the results show a significant performance gap between FedCoad and ModCL. For example, in terms of accuracy, ModCL achieved 94.10% which is 9.79 percentage points higher than that of FedCoad (μ=5.0). Unlike Motionsense and WISDM, HHAR clients had different smartphone devices (Samsung Galaxy S3 Mini, LG G, etc.) and showed strong device heterogeneity. In those conditions, fine-tuning or personalization seems necessary to achieve robust results.

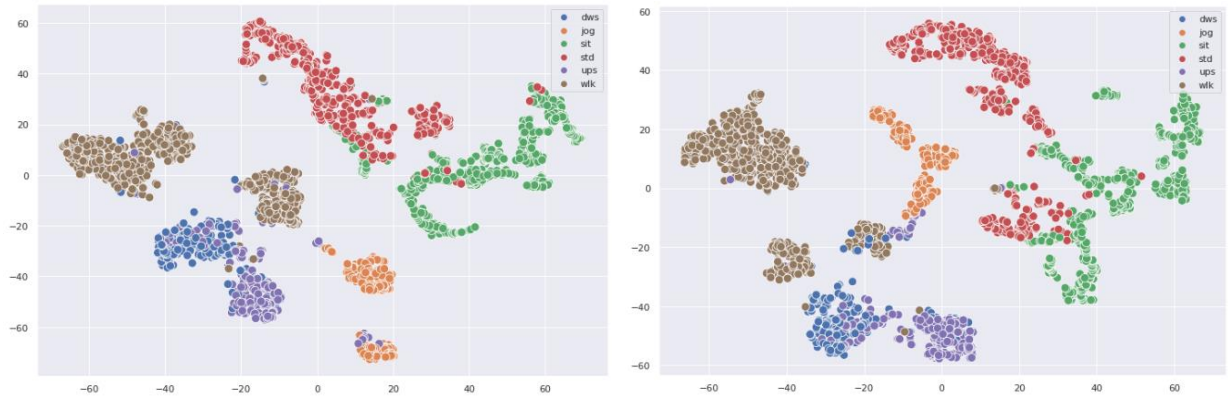
In summary, the performance of FedCoad was comparable to that of pretraining methods in heterogeneous environments such as MotionSense and WISDM. However, FedCoad had a significant performance gap in environments with strong device heterogeneity, such as the HHAR

dataset. Therefore, developing personalization approaches to handle such conditions should be a focus of future work.

1.3 Visualization of extracted embeddings

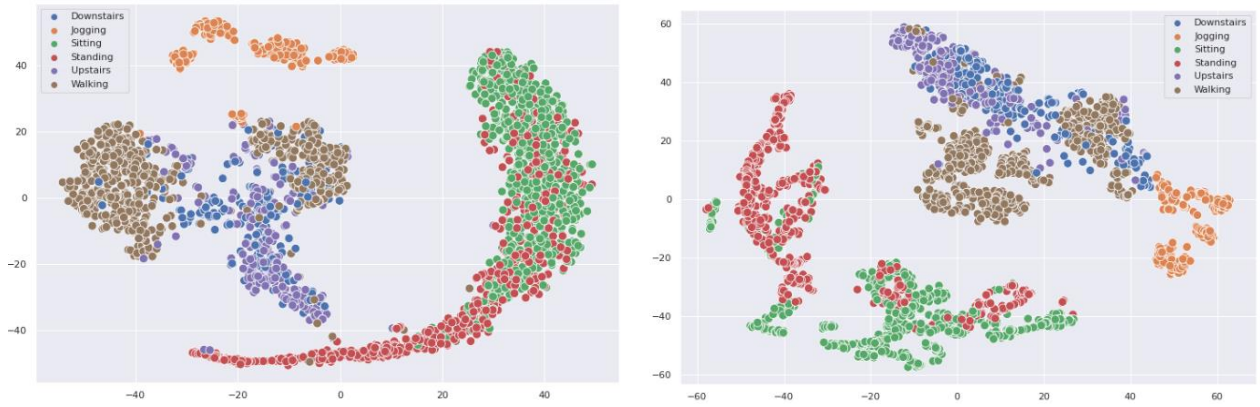
In this section we compare the difference between features learned in the centralized setting and FedCoad in the FL setting. T-SNE of the encoder features are shown to gain insight into the learned features. The encoder in the centralized setting learns robust features, and representations of all classes are separate (Fig. S1a). Previously, the FedAvg representation has many overlapping representations for different activities (Fig. S1b). For example, the representation of 'sitting' has a vast distance between its clusters. On the other hand, the FedCoad representation (Fig. S1b) has a small distance between representations in the same cluster and a vast inter-cluster distance between distinct activities similar to the centralized approach. Thus, it is evident that FedCoad can help global model convergence, which resembles a centralized approach result.

In a dataset such as WISDM (Fig. S2a-S2b), the features that the encoder learned in the centralized setting and FedCoad are scattered in many small clusters for specific activity and lack an appropriate distance between clusters of each distinct activity representation. For example, some 'walking' activity representations are closer to 'downstairs' and 'upstairs' activities. It is possible that the encoder part of the model has difficulty in learning unified features due to the imbalance in class distribution and most clients lack the complete set of training data, which is challenging. Despite that, FedCoad in the federated setting achieved a similar representation to that of the centralized setting.



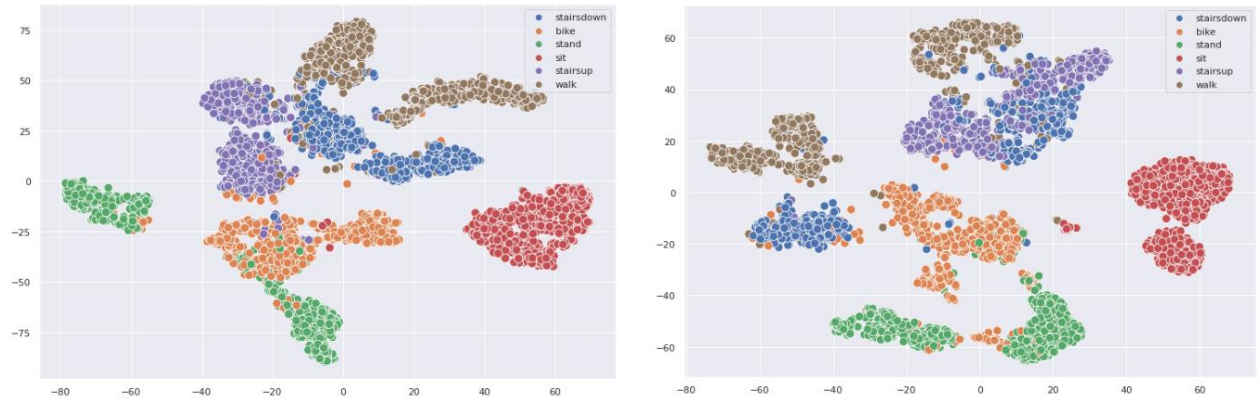
(a) (b)

Fig. S1 T-SNE of the centralized setting (a) and FedCoad (b) in the MotionSense dataset



(a) (b)

Fig. S2 T-SNE of the centralized setting (a) and FedCoad (b) in the WISDM dataset



(a) (b)

Fig. S3 T-SNE of the centralized setting (a) and FedCoad (b) in the HHAR dataset

In HHAR, there are fewer clients, but more data are available for each client. Therefore, the clusters are dense and filled with lots of representations (Fig. S3a–S3b). From the results, the encoder trained with FedCoad and a centralized setting managed to learn separable features for all classes.

1.4 Communication exchange comparison

In this section we evaluate the communication costs associated with exchanging parameters in the FL training and take into account model parameters and additional parameters introduced by each method. Table S2 shows the communication cost between one client and the server in Motionsense. For SCAFFOLD and FedCoad, both methods require updating the control variates, thus incurring additional costs. From the results, ModCL had the highest communication cost among the methods. Since the ModCL method requires a specific layer for each modality (accelerometer, gyroscope, etc.), the model parameter size is higher than in other methods. On the other hand, MetaHAR achieved the lowest communication cost with 0.33 MB. During pretraining,

MetaHAR exchanges only the encoder parameters which means there are fewer parameters to send.

When compared with FedAvg and MetaHAR, FedCoad incurred additional costs of up to 1.36 MB and 1.71 MB. Nevertheless, FedCoad outperformed FedAvg and had reasonable performance compared with MetaHAR, even with the absence of fine-tuned data in the previous sections. Compared to ModCL, FedCoad significantly reduced the communication cost by up to 90.4% (19.18 MB). Therefore, the communication cost of FedCoad was considerably lower than that of the state-of-the-art method, ModCL, and is beneficial for real-life implementation.

Table S2 Communication exchange cost for one round in MotionSense

Methods	Model parameters (MB)	Additional parameters (MB)	Total (MB)
FedAvg	0.68	-	0.68
FedProx	0.68	-	0.68
SCAFFOLD	0.68	1.36	2.04
FedAvgM	0.68	-	0.68
MOON	0.68	-	0.68
MetaHAR	0.33	-	0.33
ModCL	21.22	-	21.22
FedCoad	0.68	1.36	2.04

2 Ablation studies

In this section we describe the influence of components in the FedCoad framework. We also show the effects of the hyperparameters such as temperature (τ) and client availability on FedCoad performance results.

2.1 Component ablation studies

The objective of the ablation study was to investigate the contribution of the model contrastive learning and control variates to the performance of the FedCoad method in terms of accuracy. The experiment environment followed the setting from Section 4.2. In the first part, Fedcoad removes the model contrastive learning and relies on cross entropy and control variates for training. In contrast, in the second part, FedCoad removes the control variates and uses only model contrastive loss for learning.

Table S3 Accuracy of FedCoad components in ablation studies

Methods	MotionSense	WISDM	HHAR
FedCoad W/o model contrastive learning	91.25 %	74.97 %	81.15%
FedCoad W/o control variates	91.57 %	72.97 %	77.60 %
FedCoad	93.47 %	76.67 %	84.31 %

Table S3 shows the ablation experiment results of the FedCoad method. The results show that the FedCoad method that combines model contrastive learning and control variates achieved the highest accuracy across the three benchmark datasets. Removing either one of the components resulted in lower accuracy. For example, in HHAR, relying solely on control variates degraded the accuracy performance by 3.16 percentage points compared to the FedCoad. Meanwhile,

relying on model contrastive learning alone lowered the performance by 6.71 percentage points compared to FedCoad. This shows that both model contrastive learning and control contribute positively to the performance of the FedCoad method.

2.2 Effect of temperature

In the study of Chen et al. (2020), temperature (τ) denoted a hyperparameter that controls or gives a penalty to the negative pair. In this case, the negative pair refers to the similarity of representation from the previous local model and the current local model. As the value of τ decreases (Wang & Liu, 2020), it gives a higher penalty to the negative pair.

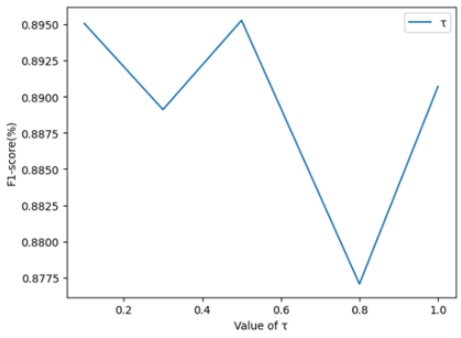


Fig. S4 Effect of temperature τ

During the experiment, τ was varied from $\{0.1, 0.3, 0.5, 0.8, 1.0\}$ while μ was set to 0.1 and the learning rate was set to 0.001. As the value of τ increases, the performance tends to decrease. From the experiment, $\tau = 0.1$ or $\tau = 0.5$ performed the best compared to other τ values. Therefore, it is important to set a smaller value for τ to give more penalty to a similar representation between the previous model and the current model.

2.3 Effect of client availability

In this section we examine the effect of client availability, which is the number of clients that can attend the training at each round. Unlike the centralized setting, multiple clients are working together to train a global model for the FL setting. In real-life scenarios, it is hard for all clients to be available at the same time. In most cases, there is only a certain percentage of clients who can connect at the designated time.

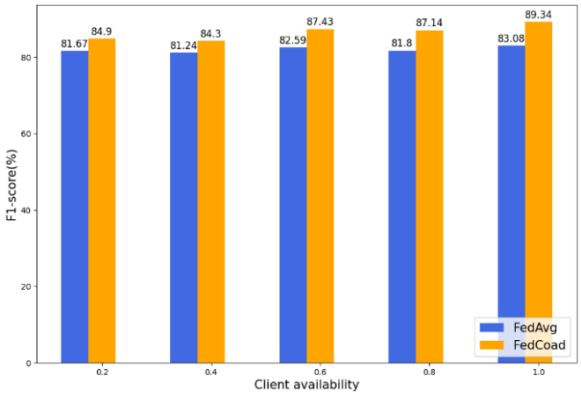


Fig. S5 Effect of client availability on FedAvg and FedCoad performance

The results from the comparison of FedCoad and FedAvg are shown in Fig. S5. The percentage refers to the fraction of clients from the total clients in the original MotionSense dataset that are available every round. As the percentage of clients increases, only small improvements are observed for FedAvg. For example, the performance of FedAvg when only 20% of clients are available (F1-score 81.67%) increases by only 1.41% when all clients are available. Unlike FedAvg, FedCoad performance significantly increases when more clients are available. For example, the performance of FedAvg when only 20% of clients are available (F1-score 84.9%) increases by 2.53% when there are 60% available and by 4.44% when all clients are available. Therefore, compared to FedAvg, FedCoad performance can scale as more clients become available.

References

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. 37th International Conference on Machine Learning, ICML 2020, PartF16814(Figure 1), 1575-1585.

Wang, F., & Liu, H. (2021). Understanding the behaviour of contrastive loss. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 2495-2504).