FedSCAl: Leveraging Server and Client Alignment for Unsupervised Federated Source-Free Domain Adaptation

Supplementary Material

Table of Contents

1 . Notations and Terminology **1**

2 . Algorithm and Computation Details **1**

2.1. Algorithm 1

2.2. Setup Use Cases 2

3 . Alternative peseudo-labeling methods **2**

4 . Results with LoA as BMD and UCon **3**

5 . T-SNE Visualization of representations . . . **4**

6 . Additional Experiments **4**

7 . Datasets and Hyper-parameter Settings . . . **5**

8 . Analysis of SCAl by varying the λ_l and λ_g . . **5**

9 . Computation and Communication Cost . . . **5**

10 On the use of Local and Global models **6**

1. Notations and Terminology

- D^s denotes the source domain.
- g denotes the feature extractor.
- h denotes the classifier.
- $\mathbb{P}_s(x, y)$ represents the source distribution.
- $\mathbb{P}_m(x, y)$ represents the m^{th} domain distribution.
- $\delta(\cdot)$ is the softmax operator and $\delta_{oh}(\cdot)$ returns the one hot output after the softmax operation.
- $\delta_{max}(\cdot)$ denotes the maximum valued element after applying the softmax operator, $\delta(\cdot)$.
- $\mathbf{w}$ denotes the model parameters.
- $\o$ denotes the adaptive threshold.
- $\mathbb{I}_{a>\tau}$ denotes indicator function, which outputs 1 if $a > \tau$ else 0.
- L_k^l denotes the client alignment loss for client k.
- L_k^g denotes the server alignment loss for client k.
- L_k^{SCAl} denotes the server and client alignment loss for client k.
- L_k^{LoA} denotes the LoA loss for client k.
- FedLoA is the LoA methods implemented in a federated fashion.
- LoA is the pseudo-labelling strategy implemented locally on the clients.
- FedSCAl is our proposed framework for solving classification tasks in the FFreeDA setup.
- $pAcc$ (FedSCAl) gives the pseudo-label accuracy when SCAl loss is used along with FedLoA methods.
- $pAcc$ (FedLoA) is the pseudo-label accuracy of FedLoA methods.
- Δ_{pAcc} is difference in the pseudo-label accuracy of $pAcc$ (FedSCAl) and $pAcc$ (FedLoA).

2. Algorithm and Computation Details

Algorithm 1: FedSCAl

Input: $\mathbf{w}_s$: pre-trained model on source domain
Output: $\mathbf{w}^T$: the final aggregated server model
Hyperparameters: λ_l, λ_g, T, E: local alignment loss weight, global alignment loss weight, communication rounds, local epochs

Initialize: server model $\mathbf{w}^0 \leftarrow \mathbf{w}_s$;
for $r \leftarrow 1$ **to** T **do**
  Sever samples Q clients with indices $p_1, p_2, \ldots, p_Q$;
  for $m \leftarrow 1$ **to** Q **do**
    server sends model $\mathbf{w}^{r-1}$ to client p_m;
    $w^r_{p_m}$ = `ClientUpdate` ($\mathbf{w}^{r-1}$, p_m) ;
  $\mathbf{w}^r$ = `ServerAggregate` ($\mathbf{w}^r_{p_1}$, ..., $\mathbf{w}^r_{p_Q}$) ,
Function `ClientUpdate` ($\mathbf{w}^{r-1}$,p_m) **:**
  set $\mathbf{w}_{k,0} = \mathbf{w}^{r-1}$;
  for $l \leftarrow 1$ **to** E **do**
    $\mathbf{w}_{k,l} = \mathbf{w}_{k,l-1} - \nabla l_k(\mathbf{w}_{k,l-1})$, where $l_k(\mathbf{w}_{k,l-1})$ is from Eq. 2 for FedSCAl ;
  set $\mathbf{w}^r_{p_m} = \mathbf{w}_{k,E}$;
  return $\mathbf{w}^r_{p_m}$;
Function `ServerAggregate` ($\mathbf{w}^r_{p_1}$, ..., $\mathbf{w}^r_{p_Q}$) **:**
  $\mathbf{w}^r = \frac{1}{Q}\sum_{i=1}^{Q} \mathbf{w}^r_{p_i}$

2.1. Algorithm

The total client loss $l_k(\mathbf{w})$ for client k is given in Eq. 2.

$$L_k^{SCAl} = \lambda_l L_k^l + \lambda_g L_k^g \quad (1)$$

$$l_k(\mathbf{w}) = L_k^{LoA} + L_k^{SCAl} \quad (2)$$

Here, L_k^{LoA} denotes the LoA loss for client k, and the alignment loss is given by L_k^{SCAl}. The parameters λ_l and λ_g represent the client and server alignment loss weights, respectively. A detailed discussion of the algorithm used to implement our SCAl loss with the pseudo-labeling strategy is provided in Algorithm 1.

2.2. Setup Use Cases

When federated learning (FL) is implemented across multiple surveillance camera sites, the requirement to manually tag each image becomes a significant administrative burden. Although servers managed by organizations often have access to labelled datasets, the sensitive biometric details—such as facial features—embedded in those images lead to privacy complications when using them directly for FL, thus necessitating the use of a pre-trained server model to initialize the federated training. Furthermore, disparities in conditions like lighting and background across various camera locations exemplify the challenges posed by differing domain characteristics.

3. Alternative peseudo-labeling methods

BMD [3] We leverage the Inter-class Balanced Prototype and Dynamic Pseudo Label findings from BMD. Class-biased strategies tend to aggregate biased data instances from easy-transfer classes, potentially generating noisy labels for hard data. To address this, the authors propose a global inter-class balanced sampling strategy, formulated as a multiple-instance learning problem. In this approach, each data instance in the target domain is represented by a feature vector and a classification result. The top-M most likely instances are aggregated to construct class-balanced feature prototypes, facilitating pseudo-label assignment as shown below

$$M_j = \underset{\substack{x_t \in X_t \\ |M_j|=M}}{arg\,max}\ \delta_k\left(f^t(x_t)\right) \quad (3)$$

$$c_j = \frac{1}{M}\sum_{i \in M_j} \hat{g}^t(x_t^i) \quad (4)$$

$$\hat{y}_t = \arg\min_j D_f(\hat{g}^t(x_t), c_j) \quad (5)$$

where $M = max\{1, \lfloor \frac{n_t}{r*K} \rfloor\}$, r is a hyperparameter denoting the top-M selection ratio, and J is the number of object classes in the target domain. To improve performance, a dynamic pseudo-labeling approach is deployed. At each epoch's start, feature prototypes and corresponding pseudo labels are globally updated. During iteration steps, feature prototypes are adjusted using an exponential moving average of cluster centroids. Dynamic pseudo labels are computed based on instance-feature prototype similarities, normalized across classes. A robust symmetric cross-entropy loss is adopted over standard cross-entropy. However, dynamic pseudo labels may overlook domain shifts, leading to less informative prototypes. To mitigate this, the static pseudo label-based self-training loss is combined with dynamic loss, balanced by hyper-parameters α and β. The dynamic pseudo labels $\hat{y_t}^d$ and the feature prototypes are updated as follows

$$\hat{y_t}^d = \frac{exp(\hat{g}^t(x_t), c_j)}{\sum_{k=1}^{K} exp(\hat{g}^t(x_t), c_j)} \quad (6)$$

$$p_j({x_t}^n) = \frac{exp(\hat{g}^t({x_t}^n), c_j)}{\sum_{j=1}^{J} exp(\hat{g}^t({x_t}^n), c_j)} \quad (7)$$

$$\hat{c_j} = \frac{\sum_{n=1}^{N} \hat{g}^t({x_t}^n) * p_j({x_t}^n)}{\sum_{n=1}^{N} p_j({x_t}^n)} \quad (8)$$

$$c_j = \lambda c_j + (1-\lambda)\hat{c_j} \quad (9)$$

$$L_{st} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} \mathbb{I}[k = \hat{y}_t] \log \delta_k(f^t(x_t^i)) \quad (10)$$

where $p_j^i(x_t)$ denotes the similarity of instance x_t with existing feature prototypes, $\hat{c_j}$ represents the feature prototype of class k calculated with the current training mini-batch, and λ is the momentum coefficient of EMA. Thus L^{bmd} becomes our L_k^{LoA} (Eq. 2)

$$\begin{aligned} L^{dyn} = &-\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{J} \hat{y}_{t,j}^d \log(\delta_j(f^t({x_t}^i))) - \\ &\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{J} \delta_j(f^t({x_t}^i)) \log(\hat{y}_{t,j}^d) \end{aligned} \quad (11)$$

$$L^{LoA} = \alpha * L^{st} + \beta * L^{dyn} \quad (12)$$

UCon-SFDA [6] The UCon-SFDA method extends the conventional contrastive loss for source-free domain adaptation by introducing uncertainty control mechanisms that target both negative and positive sample selection.

$$L_{CL} = L_{CL}^{+} \ + \ \lambda_{CL}^{-} L_{CL}^{-} \quad (13)$$

Here,

$$L_{CL}^{+} = -\frac{1}{N_T}\sum_{i=1}^{N}\sum_{x_i^+ \in C_i} S_\theta(x_i^+; x_i), \quad (14)$$

$$L_{CL}^{-} = \frac{1}{N}\sum_{i=1}^{N}\sum_{x_i^- \in B \backslash \{x_i\}} S_\theta(x_i^-; x_i), \quad (15)$$

Table 1. Comparison of FedLoA and FedSCAl on **Office-Home** dataset, when BMD is used as underlying LoA method.

Method	Initial Sever Model is pre-trained on `Art`				Initial Sever Model is pre-trained on `Clipart`				Initial Sever Model is pre-trained on `Product`				Initial Sever Model is pre-trained on `Real`			
	Clipart	**Product**	**Real**	**Avg.**	**Art**	**Product**	**Real**	**Avg.**	**Art**	**Clipart**	**Real**	**Avg.**	**Art**	**Clipart**	**Product**	**Avg.**
FedLoA	58.74	86.06	87.77	77.52	80.2	85.05	86.7	83.98	78	57.59	87.45	74.35	78.42	57.93	87.8	74.72
FedSCAl (Ours)	**63.84**	**88.39**	**89.19**	**80.47**	**81.62**	**88.88**	**88.91**	**86.47**	**80.1**	**62.98**	**88.37**	**77.15**	**79.51**	**60.81**	**89.19**	**76.5**

Table 2. Results on **Domain-Net** Dataset, when the underlying LoA method used is **BMD**. We report the accuracy (%) for each target domain and the average of all the domains for each pre-trained model. It is evident that adding our SCAl loss boosts the performance across all the domains.

Method	(a) Initial Sever Model is pre-trained on `Clipart`						(b) Initial Sever Model is pre-trained on `Infograph`					
	Infograph	**Painting**	**Quickdraw**	**Real**	**Sketch**	**Avg.**	**Clipart**	**Painting**	**Quickdraw**	**Real**	**Sketch**	**Avg.**
FedLoA	56.96	92.48	73.24	96.28	87.28	81.24	81.63	91.10	66.93	95.82	82.83	83.66
FedSCAl (Ours)	**58.19**	**94.89**	**76.68**	**96.60**	**91.94**	**83.66**	**86.38**	**94.62**	**76.44**	**96.78**	**89.37**	**88.72**
Method	(c) Initial Sever Model is pre-trained on `Painting`						(d) Initial Sever Model is pre-trained on `Quickdraw`					
	Clipart	**Infograph**	**Quickdraw**	**Real**	**Sketch**	**Avg.**	**Clipart**	**Infograph**	**Painting**	**Real**	**Sketch**	**Avg.**
FedLoA	80.24	52.09	66.97	95.68	80.09	75.01	61.51	38.00	58.35	62.10	60.72	56.13
FedSCAl (Ours)	**87.03**	**56.59**	**77.82**	**96.60**	**88.83**	**81.38**	**71.57**	**44.40**	**74.65**	**75.04**	**72.21**	**67.57**
Method	(e) Initial Sever Model is pre-trained on `Real`						(f) Initial Sever Model is pre-trained on `Sketch`					
	Clipart	**Painting**	**Quickdraw**	**Infograph**	**Sketch**	**Avg.**	**Clipart**	**Painting**	**Quickdraw**	**Real**	**Infograph**	**Avg.**
FedLoA	71.47	87.12	52.30	47.79	76.82	67.10	82.67	90.44	75.48	96.39	53.61	79.72
FedSCAl (Ours)	**80.14**	**91.51**	**68.82**	**52.97**	**84.33**	**75.55**	**88.25**	**94.53**	**79.05**	**96.65**	**57.20**	**83.14**

and $S_\theta(u;v) = \langle f(u;\theta), f(v;\theta)\rangle$ denotes the dot-product similarity between feature embeddings. To mitigate false negatives, a dispersion control term is added:

$$L_{DC} = -\frac{1}{N}\sum_{i=1}^{N} d_\theta(\mathrm{AUG}(x_i), x_i), \tag{16}$$

where $\mathrm{AUG}(\cdot)$ denotes a stochastic data augmentation and $d_\theta(u,v) = \langle f(u;\theta), \log f(v;\theta)\rangle$ The negative component of the uncertainty-controlled loss becomes

$$L^-_{UCon} = \lambda^-_{CL} L^-_{CL} + \lambda_{DC} L_{DC} \tag{17}$$

with λ_{DC} as the dispersion weight. On the positive side, an uncertain set $\mathcal{U}$ is defined by the confidence ratio criterion $p_{(1)}/p_{(2)} \leq \tau$, where $p = f(x;\theta)$ and $p_{(k)}$ is its k-th largest entry. For each $x_i \in \mathcal{U}$, a partial-label set $\mathcal{Y}^{PL}_{x_i}$ of size K_{PL} is formed from its top-K_{PL} historical predictions. The relaxed positive loss then reads

$$L^+_{UCon} = L^+_{CL} + \lambda_{PL}\frac{1}{N}\sum_{i=1}^{N}\sum_{y\in\mathcal{Y}^{PL}_{x_i}} \mathbf{1}[x_i \in \mathcal{U}]\, \ell_{\mathrm{CE}}(y,\, f(x_i;\theta)) \tag{18}$$

where $\ell_{\mathrm{CE}}(\cdot,\cdot)$ is the cross-entropy loss and λ_{PL} the partial-label weight. Finally, the overall adaptation objective is

$$L^{LoA} = L^+_{UCon} + L^-_{UCon}, \tag{19}$$

balancing dispersion and partial-label contributions.

4. Results with LoA as BMD and UCon

The two tables 1 and 4 present a comparison between FedLoA and the proposed FedSCAl method on the Office-Home and DomainNet datasets, using BMD as the underlying Local Adaptation (LoA) method. In both tables, different initial server model pre-training domains are considered, and performance is evaluated on the remaining target domains. Across all settings, FedSCAl consistently outperforms FedLoA, demonstrating better generalization to unseen domains. For instance, on Office-Home, when pre-trained on `Art`, the average accuracy improves from 77.52% (FedLoA) to 80.47% (FedSCAl), and similar gains are observed across other source domains. Likewise, on the more challenging DomainNet dataset, FedSCAl shows strong improvements in every pre-training scenario for example, when initialized on `Painting`, the average accuracy increases from 75.01% to 81.38%, and from 79.72% to 83.14% when initialized on Sketch. These results confirm that incorporating the SCAl loss enhances adaptation and improves performance across diverse and unseen target domains. Similarly the table 3 compares the performance of FedLoA and the proposed FedSCAl method on the Office-Home dataset, using UCon as the underlying Local Adaptation (LoA) method. Results are reported across four different initial server pre-training domains: `Art`, `Clipart`, `Product`, and `Real`, with accuracy evaluated on the remaining target domains. Across all configurations,

Table 3. Comparison of FedLoA and FedSCAl on **Office-Home** dataset, when UCon is used as underlying LoA method.

Method	Initial Sever Model is pre-trained on **Art**				Initial Sever Model is pre-trained on **Clipart**				Initial Sever Model is pre-trained on **Product**				Initial Sever Model is pre-trained on **Real**			
	Clipart	**Product**	**Real**	**Avg.**	**Art**	**Product**	**Real**	**Avg.**	**Art**	**Clipart**	**Real**	**Avg.**	**Art**	**Clipart**	**Product**	**Avg.**
FedLoA	53.44	80.23	82.35	72.01	74.4	80.51	80.76	78.56	70.09	50.16	81.52	67.26	75.92	55	84.65	71.86
FedSCAl (Ours)	**57.76**	**81.34**	**82.71**	**73.94**	**77.17**	**82.99**	**85.16**	**81.77**	**75.9**	**55.46**	**83.96**	**71.77**	**77.6**	**59.34**	**86.82**	**74.59**

Table 4. Results on **Domain-Net** Dataset, when the underlying LoA method used is **UCon**. We report the accuracy (%) for each target domain and the average of all the domains for each pre-trained model. Clearly, adding our SCAL loss boosts the performance.

Method	(a) Initial Sever Model is pre-trained on **Clipart**						(b) Initial Sever Model is pre-trained on **Infograph**					
	Infograph	**Painting**	**Quickdraw**	**Real**	**Sketch**	**Avg.**	**Clipart**	**Painting**	**Quickdraw**	**Real**	**Sketch**	**Avg.**
FedLoA	56.94	93.31	67.76	96.33	87.06	80.28	80.63	91.86	66.03	95.86	81.51	83.18
FedSCAl (Ours)	**60.07**	**94.42**	**74.14**	**96.51**	**90.22**	**83.08**	**85.63**	**93.56**	**76**	**96.58**	**86.24**	**87.6**

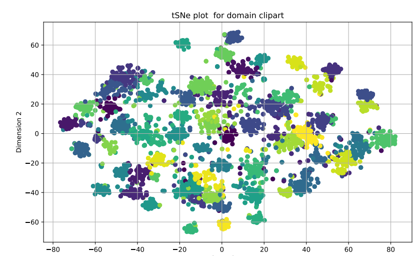
Method	(c) Initial Sever Model is pre-trained on **Painting**						(d) Initial Sever Model is pre-trained on **Quickdraw**					
	Clipart	**Infograph**	**Quickdraw**	**Real**	**Sketch**	**Avg.**	**Clipart**	**Infograph**	**Painting**	**Real**	**Sketch**	**Avg.**
FedLoA	81.02	53.85	61.36	96.12	83.27	75.12	47.61	37.89	53.75	57.07	48.14	48.89
FedSCAl (Ours)	**85.23**	**58.58**	**68.63**	**96.32**	**86.24**	**79**	**68.4**	**50.86**	**70.68**	**71.45**	**65.46**	**68.37**

Method	(e) Initial Sever Model is pre-trained on **Real**						(f) Initial Sever Model is pre-trained on **Sketch**					
	Clipart	**Painting**	**Quickdraw**	**Infograph**	**Sketch**	**Avg.**	**Clipart**	**Painting**	**Quickdraw**	**Real**	**Infograph**	**Avg.**
FedLoA	81.53	93.08	64.63	56.5	85.49	76.25	84.25	93.92	70.98	96.33	57.98	80.69
FedSCAl (Ours)	**85.48**	**94.21**	**73.85**	**59.86**	**88.86**	**80.45**	**88.77**	**94.78**	**76.04**	**96.5**	**60.21**	**83.26**

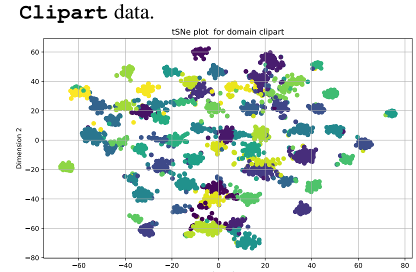
FedSCAl consistently outperforms FedLoA. For example, when the initial model is pre-trained on `Clipart`, the average accuracy improves from 78.56% (FedLoA) to 81.77% (FedSCAl), and similarly, for the `Real` pre-training setting, the average increases from 71.86% to 74.59%. These results show that integrating the SCAl loss enhances the adaptation quality and leads to improved generalization under UCon-based local training.

5. T-SNE Visualization of representations

In Fig. 1, we visualize the t-SNE embeddings of the learned representations on a client from the `Clipart` domain, where the initial server model is pre-trained on the `Art` domain. Under FedLoA, the representations exhibit overlapping and poorly formed class clusters, indicating suboptimal adaptation to the target domain. In contrast, FedSCAl leads to relatively well-separated and compact clusters, demonstrating improved inter-class separability. This highlights that FedSCAl is more effective in aligning the source-pretrained server model with the client's target distribution, particularly in challenging cross-domain scenarios. The improved clustering structure validates the effectiveness of SCAl loss in enhancing federated adaptation and feature discriminability.



(a) t-SNE plot for FedLoA when the server model is pre-trained on **Art** and the client has **Clipart** data.



(b) t-SNE plot for FedSCAl when the server model is pre-trained on **Art** and the client has **Clipart** data.

Figure 1. **(a)** t-SNE plot of representations learned by FedLoA, **(b)** t-SNE plot for representations learned after adding our proposed loss (FedSCAl). It can be seen that adding our SCAl loss leads to better clustering of the learned representations.

6. Additional Experiments

As mentioned in the main paper, due to space constraints we provide the Office-31 results in Table 5. It reports the results on the Office-31 dataset, where the initial server model is pre-trained on one of the three domains (Amazon, DSLR, or Webcam), and clients are drawn from the remaining two

Table 5. Results on **Office-31** Dataset, while the initial server pre-trained model is trained on one of the three domains (Amazon, Webcam, and DSLR), and the clients are distributed with the remaining two domains such that each client contains a subset of exactly one Domain.

Method	(a) Initial Server Model: **Amazon**			(b) Initial Server Model: **DSLR**			(c) Initial Server Model: **Webcam**		
	DSLR	**Webcam**	**Avg.**	**Amazon**	**Webcam**	**Avg.**	**Amazon**	**DSLR**	**Avg.**
LoA	92.97	94.21	93.59	75.61	93.96	84.79	78.83	98.93	88.61
FedLoA	96.65	95.64	96.15	81.11	95.68	88.39	79.19	97.79	88.56
FedProx	95.14	94.58	94.86	79.03	94.58	86.81	77.35	97.56	87.46
FedMOON	95.72	95.05	95.39	80.15	95.21	87.68	78.79	97.61	88.20
FedWCA	95.65	90.9	93.28	75.66	97.78	86.72	75.88	99.73	87.81
LADD	84.54	85.03	84.78	63.9	94.72	79.31	70.6	99.4	84.99
Dual Adapt	88.86	87.42	88.14	69.08	97.02	83.05	73.81	99.35	86.58
FedSCAl (Ours)	96.65	96.27	**96.46**	81.69	96.81	**89.25**	82.87	98.66	**90.76**

domains, with each client containing data from only a single domain. The performance is measured on the held-out client domains and reported individually as well as averaged. The proposed FedSCAl method consistently outperforms all baselines across all settings. For example, when the server is initialized on Amazon, FedSCAl achieves the highest average accuracy of 96.46%, surpassing methods like FedLoA (96.15%), FedProx (94.86%), and FedMOON (95.39%). Similarly, in the DSLR-pretrained setting, FedSCAl reaches an average of 89.25%, showing gains over FedLoA (88.39%) and other competing methods. In the Webcam-pretrained case, FedSCAl achieves the highest average accuracy of 90.76%, demonstrating its robust cross-domain adaptation ability under varied initialization conditions.

It can be seen that the LoA method performs inferior to FedLoA and FedSCAl even though there is no client-drift in this case, the key issue being the training on limited single client data. In the case of FedLoA, the performance is not the best as client-drift limits its performance. FedSCAl performs the best as it minimizes the client-drift across the domains. It can be seen that FedSCAl outperforms the FedLoA baseline by 2.2% when the server model is pre-trained on Webcam data, and it beats FedLoA by 0.86% when the server model is pre-trained on DSLR data.

7. Datasets and Hyper-parameter Settings

Office-31 [4] dataset contains three domains: Amazon, DSLR, and Webcam. While we consider one domain as a source, the other 2 domains are distributed among 6 clients (3 each for a particular domain). The participation rate is set to 0.5. For Office-Home [5] and Office-31 we set λ_l and λ_g to 1. For Domain-Net [1] we set λ_l to 3 and λ_g to 0.3. We follow the protocol of [2] for adapting the Domain-Net dataset to the federated setup. We use an SGD optimizer for adaptation and a learning rate of 0.03 for the Office-Home, Office-31, and Domain-Net datasets. A batch size of 64 is used for training at each client for all datasets. Local training epochs for each client are kept at 5 for all datasets. As FedProx and FedMOON are designed for supervised settings, we implement them in the FFreeDA setting with the LoA method as described in Section 3.1 of the main paper.

8. Analysis of SCAl by varying the λ_l and λ_g

We perform an ablation study on the regularization weights λ_l and λ_g, where the initial server model is pre-trained on the `Quickdraw` domain, and the clients are distributed with data from the remaining domains. The plots in the Figure 2 show the average accuracy across all clients as we vary one regularizer while keeping the other fixed. We observe that while there is some fluctuation in performance, the overall accuracy remains relatively stable across a range of λ_l and λ_g.

9. Computation and Communication Cost

If c denotes the Forward Pass Computation Cost (FPCC), and e is the number of local epochs, then the proposed regularizer will have an effective FPCC of ce+ce+c. This effectively incurs only twice the FPCC for client training in each communication round when the local epochs e is larger than 1. The backward computation due to our FedSCAl framework remains almost the same. We save computation by storing pseudo-labels generated by the first forward pass through the global model parameters for the weakly augmented samples. Our method communicates only the model parameters and incurs no additional communication cost.

The Table 6 reports the end-to-end training time (in seconds) for different federated learning baselines for single communication round. Among all methods, FedSCAL, our proposed approach, shows a moderate increase in computation time compared to others. This is expected, as FedSCAL incorporates both local and global consistency mechanisms. Despite this slight overhead, FedSCAL delivers

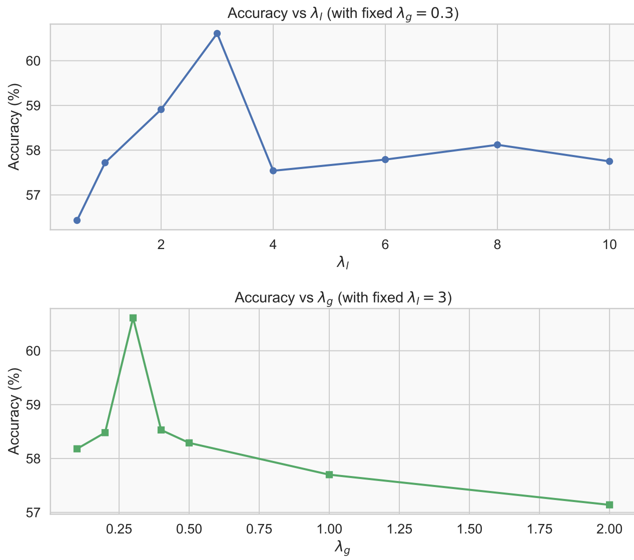


Figure 2. Ablation of SCAl Regularization Weights: Average accuracy across clients while varying λ_l and λ_g.

Table 6. Training time comparison of different methods.

Method	Time (sec)
FedLoA	156.93
LADD	114.90
FedProx	158.28
FedWCA	163.83
FedSCAL	170.08

significantly better performance, demonstrating that the additional computation is minimal relative to its overall gains.

10. On the use of Local and Global models

As demonstrated in our experiments, both the local and global consistency losses are essential to the effectiveness of the SCAL mechanism. We now formalize why this is the case. Let $p^i_{l,k}$ denote the probability that the *local* model of client k incorrectly predicts the weakly augmented sample i. Similarly, let p^i_g denote the error probability of the *global* model on the same sample. The consistency loss for sample i receives incorrect supervision only when both models are wrong, which occurs with probability $p^i_{l,k}\, p^i_g$. Thus, the probability of receiving correct supervision is

$$1 - p^i_{l,k}\, p^i_g.$$

Notice that

$$1 - p^i_{l,k}\, p^i_g > 1 - p^i_{l,k} \qquad \text{and} \qquad 1 - p^i_{l,k}\, p^i_g > 1 - p^i_g,$$

since $p^i_{l,k}\, p^i_g < p^i_{l,k}$ and $p^i_{l,k}\, p^i_g < p^i_g$. In other words, combining both local and global predictions strictly increases the probability of obtaining correct supervision compared to relying on either model alone. This explains why the joint use of local and global consistency losses yields more reliable pseudo-labels and contributes significantly to the success of SCAL.

References

[1] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073, 2012. 5

[2] Xiaoxiao Li, Meirui JIANG, Xiaofei Zhang, Michael Kamp, and Qi Dou. FedBN: Federated learning on non-IID features via local batch normalization. In *International Conference on Learning Representations*, 2021. 5

[3] Sanqing Qu, Guang Chen, Jing Zhang, Zhijun Li, Wei He, and Dacheng Tao. Bmd: A general class-balanced multicentric dynamic prototype strategy for source-free domain adaptation. In *European Conference on Computer Vision*, pages 165–182. Springer, 2022. 2

[4] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Computer Vision – ECCV 2010*, pages 213–226, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. 5

[5] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017. 5

[6] Gezheng Xu, Hui Guo, Li Yi, Charles Ling, Boyu Wang, and Grace Yi. Revisiting source-free domain adaptation: a new perspective via uncertainty control. In *The Thirteenth International Conference on Learning Representations*, 2025. 2