Computation and Communication Efficient Federated Unlearning via On-server Gradient Conflict Mitigation and Expression

Supplementary Material

A. Related Works

A.1. Federated Unlearning

Recently, there have be en some FUL approaches, including 1) fast retraining, and 2) approximate unlearning.

Fast retraining: To design a fast training protocol for FUL, [7, 47] propose user data revocation strategies to ensure the efficient unlearning process among users in FL system. FedEraser [25] initiates the unlearning algorithm among the clients that request data removal. However, other clients' models still hold the contributions of the data to be removed, which will be later aggregated in a global model. To address this, their proposed solution uses the latest rounds of local model updates to approximate the current unlearned model update. Nonetheless, if the data samples to be removed have been used for training from the beginning, the model updates may still contain traces of these samples. [27] introduces a fast retraining approach by leveraging the diagonals of Fisher Information Matrix (FIM). This approach significantly improve the efficiency of the approximation of inverse Hessian. KNOT [38] proposes a clustered FL concept to improve the efficiency of retraining in FUL. Specifically, due to the clustered aggregation mechanism, the FUL concept using KNOT only has to retrain a partial set of clients, thus improving the retraining efficiency in FUL. NoT [20] introduces a weight negation operator that perturbs the model parameters away from their optimal values, thereby positioning the model more favorably for subsequent re-training.

Approximate Unlearning FATS [41] evaluates the involvement of clients in the previous training of the federated learning (FL) system. Consequently, the contributions of clients are considered accordingly in the unlearning stage. MetaFUL [46] proposes an efficient unlearning algorithm that leverages the impact of user data, which is measured during the learning stage. MoDe [55] applies slow-update techniques in federated unlearning to improve the smoothness of the unlearning process, thereby enhancing overall performance. However, the naive unlearning approach (i.e., excluding clients with revocation requests from the aggregation stage) does not ensure a reduced generalization gap between conventional machine unlearning and FUL. FedRemover [50] introduces a two-step approach for unlearning malicious clients, which includes: 1) attack detection, and 2) gradient direction analysis-based malicious client identification. This method significantly improves energy and computational efficiency in identifying clients for unlearning. FFMU [4] employs quantization on local models, demonstrating that it can efficiently unlearn kno wledge from the set of forget devices. The authors in [43] propose a TF-IDF Guided Pruning approach for efficient FUL. Specifically, the server uses TF-IDF, a statistical measure of channel impact on class discrimination, to determine which network channels should be pruned. However, this pruning-based approach not only removes information about the unlearning classes but also erases information from some remaining data, leading to a degradation in data discrimination.

To address this issue, [49] proposes an on-server unlearning process for FUL. However, this method requires on-server data, which negatively impacts the communication efficiency of the FL system. Conda [6] introduces an on-server unlearning method that identifies global model parameter sets most affected by forget clients. By dampening these parameters, Conda achieves effective unlearning while preserving the rest of the model parameters on the server side. FUSED [56] identifies the critical layers that contribute most to learning and performs updates selectively on these layers through a sparse update strategy. This approach significantly reduces the computational cost required during the unlearning phase.

The authors in [12] propose a mixup-based strategy to enrich the data used in unlearning. Ferrari [11] introduces a new metric for feature-level unlearning called feature sensitivity. FedME2 [48] presents an FUL architecture with two key components: 1) memory evaluation (MEval), and 2) erasure (MErase). MEval follows the rationale that the informational bottleneck significantly reduces redundant information from the data. By evaluating the final layers, it determines whether data is efficiently remembered within the model architecture. The authors design an evaluation loss function for the federated unlearning problem, thereby efficiently improving the performance of federated unlearning. FedOSD [32] enables client-side unlearning by incorporating an unlearning cross-entropy loss to remove undesired knowledge locally, and adopts gradient surgery to ensure gradient alignment among heterogeneous clients.

A.2. Prototype Learning

Prototype learning [37] has emerged as a powerful paradigm for classification tasks, where models learn representative embeddings, or prototypes, for each class. A

prototype typically refers to the mean feature vector of instances belonging to the same class [24]. Beyond serving as a classification tool, prototype learning can also be viewed as an implicit form of causal factor discovery, as each prototype may correspond to a distinct causal factor in the underlying CSM [18, 29].

In supervised classification, a query sample is classified by measuring its distance to the prototypes of each class, which has been shown to yield more robust and stable performance [35, 51], particularly in few-shot [5, 15, 26, 57] and zero-shot learning [9, 16] settings. Moreover, prototype-based approaches have gained increasing attention in semantic segmentation [58, 59], multimodal learning [8] and unsupervised representation learning [40].

In the context of FL, prototypes can serve as a compact and privacy-preserving form of abstract knowledge that enables efficient information sharing among clients. Recent studies have integrated prototype learning into FL frameworks to address data heterogeneity across clients [39, 53], mitigate domain shift [17, 45], and enhance personalization in local models [19, 39, 52].

B. Two-stage Optimization for learning stage of FOUL

The joint loss function is complex which makes the optimization problem NP-hard. Acknowledging this shortcoming, we decompose the FOUL training process into two steps, denoted as the *L2U Generator* and the *Meaningful Invariant Discriminator* (MID). The *L2U Generator* is responsible for training the featurizer to achieve good data reconstruction while maintaining the invariant and variant characteristics of z_K and z_V, respectively. The *MID* focuses on distilling essential knowledge into the invariant representation z_K. By splitting the complicated tasks into two simpler tasks with fewer loss components, the individual tasks become significantly more straightforward, thereby enhancing the overall training process. To train these two disentangled tasks jointly and efficiently, we use a cost-effective meta-learning technique called Reptile [31]. Reptile is proven to be computationally efficient while maintaining performance comparable to conventional meta-learning approaches [10]. In our Reptile-based L2U, the *L2U Generator* and *MID* are trained alternately in a continuous manner. At the start of each training round r, we begin with the *L2U Generator* to prioritize the training of meaningful representations that can be faithfully reconstructed. Specifically, the

Algorithm 1: Pseudo algorithm for L2U stage of FOUL.

Input: Initial model parameter θ_0, learning coefficients $\alpha_{\text{rec}}, \alpha_{\text{v}}, \alpha_{\text{iv}}, \alpha_{\text{gtc}}, \alpha_{\text{adv}} \in [0, 1)$ and learning rate $\eta_{\text{L2U}}, \eta_{\text{MID}}, \eta_{\text{adv}} \in \mathbb{R}^+$. Local iterations $E = I_{\text{L2U}} + I_{\text{MID}}$
for *training round* $r = 0, 1, \ldots, I$ **do**
  / Local Training */*
  for *client* $u \in \mathcal{U}$ **do**
    / L2U Generator */*
    for *iteration* $i = 0, 1, \ldots, I_{L2U} - 1$ **do**
      Compute loss for the representation disentanglement:
      $$\mathcal{L}_{\text{L2U}} = \alpha_{\text{rec}}\mathcal{L}_{\text{rec}} + \alpha_{\text{iv}}\mathcal{L}_{\text{iv}} + \alpha_{\text{v}}\mathcal{L}_{\text{v}}.$$
      Apply gradient descent according to Eq. (11).
    end for
    / Meaningful Invariant Discriminator */*
    for *iteration* $i = I_{L2U}, \ldots, I_{L2U} + I_{MID}$ **do**
      Compute loss for meaningful invariant representation:
      $$\mathcal{L}_{\text{MID}} = \alpha_{\text{iv}}\mathcal{L}_{\text{iv}} + \alpha_{\text{gtc}}\mathcal{L}_{\text{gtc}}.$$
      Apply gradient descent according to Eq. (12).
    end for
    Upload local model $\theta_u^{(r,E)}$ to the server.
  end for
  / Global Aggregation */*
  Compute global model $\theta_g^{(r)} = \sum_{u\in\mathcal{U}} \theta_u^{(r,E)}$.
end for

L2U model is updated within the *L2U Generator* as follows:

$$\begin{aligned}
\theta_E^{r,i+1} &= \theta_E^{r,i} - \eta_{\text{L2U}}\nabla_{\theta_E^{r,i}}\mathcal{L}_{\text{L2U}},\\
\theta_V^{r,i+1} &= \theta_V^{r,i} - \eta_{\text{L2U}}\nabla_{\theta_V^{r,i}}\mathcal{L}_{\text{L2U}},\\
\theta_K^{r,i+1} &= \theta_K^{r,i} - \eta_{\text{L2U}}\nabla_{\theta_K^{r,i}}\mathcal{L}_{\text{L2U}},\\
\theta_2^{r,i+1} &= \theta_2^{r,i} - \eta_{\text{L2U}}\nabla_{\theta_2^{r,i}}\mathcal{L}_{\text{L2U}}.
\end{aligned} \tag{11}$$

In the subsequent *MID*, we focus on the training of featurizers θ_E, θ_K and finetune the label classifier θ_{gtc} as follows:

$$\begin{aligned}
\theta_E^{r,i+1} &= \theta_E^{r,i} - \eta_{\text{MID}}\nabla_{\theta_V^{r,i}}\mathcal{L}_{\text{MID}}^{r,i}.\\
\xi^{r,i+1} &= \xi^{r,i} - \eta_{\text{MID}}\nabla_{\theta_{\text{gtc}}^{r,i}}\mathcal{L}_{\text{MID}}^{r,i}.
\end{aligned} \tag{12}$$

The details of the training process are shown in Algorithm 1.

C. Client-level Federated Unlearning Derivation

Theorem 1 (FOUL solution) Given $\Gamma = \{\gamma_u^{(r)} | u \in \mathcal{U}, \sum_{u\in\mathcal{U}} \gamma_u^{(r)} = 1\}$ is the set of learnable coefficients at each round r. Invariant gradient direction $g_{\text{UL}}^{(r)}$ is characterized as follows:

$$
\begin{aligned}
g_{\text{UL}}^{(r)} &= g_{\text{FL}}^{(r)} + \frac{\kappa \|g_{\text{FL}}^{(r)}\|}{\|g_{\Gamma_{\mathcal{R}}}^{(r)} - g_{\Gamma_{\mathcal{F}}}^{(r)}\|}(g_{\Gamma_{\mathcal{R}}}^{(r)} - g_{\Gamma_{\mathcal{F}}}^{(r)}), \\
\text{s.t.} \quad \Gamma_{\mathcal{R}}^*, \Gamma_{\mathcal{F}}^* &= \arg\min_{\Gamma_{\mathcal{R}}, \Gamma_{\mathcal{F}}} (g_{\Gamma_{\mathcal{R}}}^{(r)} - g_{\Gamma_{\mathcal{F}}}^{(r)}) \cdot g_{\text{FL}}^{(r)} + \sqrt{\kappa}\|g_{\text{FL}}^{(r)}\| \|g_{\Gamma_{\mathcal{R}}}^{(r)} - g_{\Gamma_{\mathcal{F}}}^{(r)}\|.
\end{aligned} \tag{13}
$$

Proof. We consider $x = g_{\text{UL}}^{(r)}$ and consider the maximization problem with x as optimization variable. Denote $\phi = \kappa^2 \|g_{\text{FL}}^{(r)}\|^2$. Note that $\min_u \langle g_u^{(r)}, g_{\text{FL}}^{(r)} \rangle = \min_\gamma \langle \sum_u \gamma_u h_u^{(r)}, h_g^{(r)} \rangle$. The Lagrangian of the objective is

$$
\max_x \min_{\lambda,\gamma} \Big(\sum_{u\in\mathcal{U}_{\mathcal{R}}} \gamma_u h_u^{(r)} \Big)^\top x - \beta \Big(\sum_{v\in\mathcal{U}_{\mathcal{F}}} \gamma_v h_v^{(r)} \Big)^\top x - \frac{\lambda}{2}\|g_{\text{FL}}^{(r)} - x\|^2 + \frac{\lambda}{2}\phi, \text{ s.t. } \lambda \geq 0. \tag{14}
$$

Since the problem is a convex programming and and Slater's condition is satisfied for $\kappa > 0$ (meanwhile, if $\kappa = 0$, it can be easily verified that all results hold trivially), the strong duality holds. Consequently, the order of the $\min$ and $\max$ operations can be interchanged. For instance,

$$
\min_{\lambda,\gamma} \max_x \underbrace{\Big(\sum_{u\in\mathcal{U}_{\mathcal{R}}} \gamma_u h_u^{(r)} \Big)^\top x - \beta \Big(\sum_{v\in\mathcal{U}_{\mathcal{F}}} \gamma_v h_v^{(r)} \Big)^\top x - \frac{\lambda}{2}\|g_{\text{FL}}^{(r)} - x\|^2 + \frac{\lambda}{2}\phi}_{A_1}, \text{ s.t. } \lambda \geq 0. \tag{15}
$$

Taking A_1 into consideration. If we consider λ, γ as constant, x is the variable, x achieves the optimal solution when $\partial A_1 / \partial x = 0$. Thus, we have the followings:

$$
\frac{\partial A_1}{\partial x} = \sum_{u\in\mathcal{U}_{\mathcal{R}}} \gamma_u h_u^{(r)} - \beta \sum_{v\in\mathcal{U}_{\mathcal{F}}} \gamma_v h_v^{(r)} - \lambda(x - g_{\text{FL}}^{(r)}) = 0. \tag{16}
$$

Another speaking, we have

$$
x = g_{\text{FL}}^{(r)} + \Big(\sum_{u\in\mathcal{U}_{\mathcal{R}}} \gamma_u h_u^{(r)} - \beta \sum_{v\in\mathcal{U}_{\mathcal{F}}} \gamma_v h_v^{(r)} \Big) / \lambda. \tag{17}
$$

Therefore, we have the followings:

$$
\begin{aligned}
A_1 = &\Big(\sum_{u\in\mathcal{U}_{\mathcal{R}}} \gamma_u h_u^{(r)} \Big)^\top \Big(g_{\text{FL}}^{(r)} + \Big(\sum_{u\in\mathcal{U}_{\mathcal{R}}} \gamma_u h_u^{(r)} - \beta \sum_{v\in\mathcal{U}_{\mathcal{F}}} \gamma_v h_v^{(r)} \Big) / \lambda \Big) \\
&- \beta \Big(\sum_{v\in\mathcal{U}_{\mathcal{F}}} \gamma_v h_u^{(r)} \Big)^\top \Big(g_{\text{FL}}^{(r)} + \Big(\sum_{u\in\mathcal{U}_{\mathcal{R}}} \gamma_u h_u^{(r)} - \beta \sum_{v\in\mathcal{U}_{\mathcal{F}}} \gamma_v h_v^{(r)} \Big) / \lambda \Big) - \frac{1}{2\lambda} \Big\| \sum_{u\in\mathcal{U}_{\mathcal{R}}} \gamma_u h_u^{(r)} - \beta \sum_{v\in\mathcal{U}_{\mathcal{F}}} \gamma_v h_v^{(r)} \Big\|^2 + \frac{\lambda}{2}\phi.
\end{aligned} \tag{18}
$$

Substituting $g_{\Gamma_{\mathcal{R}}}^{(r)} = \sum_{u\in\mathcal{U}_{\mathcal{R}}} \gamma_u h_u^{(r)}$ and $g_{\Gamma_{\mathcal{F}}}^{(r)} = \sum_{v\in\mathcal{U}_{\mathcal{R}}} \gamma_v h_v^{(r)}$. Consider the optimization problem of Eq. (18), we have:

$$
\begin{aligned}
A_1 &= g_{\Gamma_{\mathcal{R}}}^{(r)\top} \Big(g_{\text{FL}}^{(r)} + (g_{\Gamma_{\mathcal{R}}}^{(r)} - \beta g_{\Gamma_{\mathcal{F}}}^{(r)})/\lambda \Big) - \beta g_{\Gamma_{\mathcal{F}}}^{(r)\top} \Big(g_{\text{FL}}^{(r)} + (g_{\Gamma_{\mathcal{R}}}^{(r)} - \beta g_{\Gamma_{\mathcal{F}}}^{(r)})/\lambda \Big) - \frac{1}{2\lambda}\|g_{\Gamma_{\mathcal{R}}}^{(r)} - \beta g_{\Gamma_{\mathcal{F}}}^{(r)}\|^2 + \frac{\lambda}{2}\phi \\
&= (g_{\Gamma_{\mathcal{R}}}^{(r)} - \beta g_{\Gamma_{\mathcal{F}}}^{(r)})^\top g_{\text{FL}}^{(r)} + \frac{1}{\lambda}(g_{\Gamma_{\mathcal{R}}}^{(r)} - \beta g_{\Gamma_{\mathcal{F}}}^{(r)})^\top (g_{\Gamma_{\mathcal{R}}}^{(r)} - \beta g_{\Gamma_{\mathcal{F}}}^{(r)}) - \frac{1}{2\lambda}\|g_{\Gamma_{\mathcal{R}}}^{(r)} - \beta g_{\Gamma_{\mathcal{F}}}^{(r)}\|^2 + \frac{\lambda}{2}\phi \\
&= (g_{\Gamma_{\mathcal{R}}}^{(r)} - \beta g_{\Gamma_{\mathcal{F}}}^{(r)})^\top g_{\text{FL}}^{(r)} + \frac{1}{2\lambda}\|g_{\Gamma_{\mathcal{R}}}^{(r)} - \beta g_{\Gamma_{\mathcal{F}}}^{(r)}\|^2 + \frac{\lambda}{2}\phi.
\end{aligned} \tag{19}
$$

Therefore, we have Eq. 15 is equivalent to

$$
\min_{\lambda,\gamma} \underbrace{(g_{\Gamma_{\mathcal{R}}}^{(r)} - \beta g_{\Gamma_{\mathcal{F}}}^{(r)})^\top g_{\text{FL}}^{(r)} + \frac{1}{2\lambda}\|g_{\Gamma_{\mathcal{R}}}^{(r)} - \beta g_{\Gamma_{\mathcal{F}}}^{(r)}\|^2 + \frac{\lambda}{2}\phi}_{A_2}. \tag{20}
$$

Next, we consider λ as variable to find the optimal value. Subsequently, optimization problem Eq. (20) is equivalent to the following relationship:

$$\frac{\partial}{\partial \lambda} A_2 = -\frac{1}{2\lambda^2} \|g_{\Gamma_{\mathcal{R}}}^{(r)} - \beta g_{\Gamma_{\mathcal{F}}}^{(r)}\|^2 + \frac{1}{2}\phi = 0. \tag{21}$$

Therefore, the equation achieves the optimality as $\lambda = \|g_{\Gamma_{\mathcal{R}}}^{(r)} - g_{\Gamma_{\mathcal{F}}}^{(r)}\|/\phi^{1/2}$. Combining with Eq. (20) and Eq. (17), we have the followings:

$$\begin{aligned} g_{\text{UL}}^{(r)} &= g_{\text{FL}}^{(r)} + \frac{\kappa \|g_{\text{FL}}^{(r)}\|}{\|g_{\Gamma_{\mathcal{R}}}^{(r)} - \beta g_{\Gamma_{\mathcal{F}}}^{(r)}\|} (g_{\Gamma_{\mathcal{R}}}^{(r)} - \beta g_{\Gamma_{\mathcal{F}}}^{(r)}), \\ \text{s.t.} \quad \Gamma_{\mathcal{R}}^*, \Gamma_{\mathcal{F}}^* &= \arg\min_{\Gamma_{\mathcal{R}}, \Gamma_{\mathcal{F}}} (g_{\Gamma_{\mathcal{R}}}^{(r)} - \beta g_{\Gamma_{\mathcal{F}}}^{(r)}) \cdot g_{\text{FL}}^{(r)} + \sqrt{\kappa} \|g_{\text{FL}}^{(r)}\| \|g_{\Gamma_{\mathcal{R}}}^{(r)} - \beta g_{\Gamma_{\mathcal{F}}}^{(r)}\|. \end{aligned} \tag{22}$$

This solves the problem.

D. Pseudo Algorithm for unlearning stage of FOUL

Algorithm 2: Unlearning Stage via On-server Matching Gradient

Input: set of retain users $\mathcal{U}_{\mathcal{R}}$, forget users $\mathcal{U}_{\mathcal{F}}, \mathcal{U} = \{\mathcal{U}_{\mathcal{R}}, \mathcal{U}_{\mathcal{F}}\}$, number of communication rounds R, local learning rate η, global learning rate η_g, searching space hyper-parameter κ.

Output: $\theta_{V,g}^{(R)}$

Clients Update:

for *client* $u \in \mathcal{U}$ **do**

  Receive global non-causal encoder $\theta_{V,u}^{(r)} = \theta_{V,g}^{(r)}$;

  Merge global non-causal encoder $\theta_{V,u}^{(r)}$ into local model $\theta_u^{(r)}$;

  for *local epoch* $e \in E$ **do**

    Sample mini-batch ζ from local data $\mathcal{D}_u$;

    Calculate gradient $\nabla_{\theta_{V,u}^{(r,e)}} \mathcal{E}(\theta_u^{(r,e)}, \zeta)$;

    Update client's non-causal encoder: $\theta_{V,u}^{(r,e+1)} = \theta_{V,u}^{(r,e)} - \eta \nabla_{\theta_{V,u}^{(r,e)}} \mathcal{E}(\theta_u^{(r,e)}, \zeta)$;

  end for

  Upload client's non-causal encoder $\theta_{V,u}^{(r,E)}$ to server.

end for

Server Optimization:

for *round* $r = 0, \ldots, R$ **do**

  Clients Updates;

  Calculate $g_u^{(r)} = \theta_{V,u}^{(r,E)} - \theta_{V,u}^{(r)}, \quad \mathbf{g}^{(r)} = \{g_u^{(r)} | u \in \mathcal{U}\}$;

  Calculate $g_{FL}^{(r)}$ (e.g., $g_{FL}^{(r)} = \frac{1}{U} \sum_{u=1}^{U} g_u^{(r)}$ as the FedAvg update);

  Calculate $g_{\Gamma_{\mathcal{R}}}^{(r)} = \sum_{u \in \mathcal{U}_{\mathcal{R}}} \gamma_u g_u^{(r)}$ as the retain user set update);

  Calculate $g_{\Gamma_{\mathcal{F}}}^{(r)} = \sum_{v \in \mathcal{U}_{\mathcal{F}}} \gamma_v g_v^{(r)}$ as the forget user set update);

  Solve for $\Gamma^* = \{\Gamma_{\mathcal{R}}^*, \Gamma_{\mathcal{F}}^*\}$, where $\Gamma_{\mathcal{R}} = \{\gamma_u | u \in \mathcal{U}_{\mathcal{R}}\}, \Gamma_{\mathcal{F}} = \{\gamma_v | v \in \mathcal{U}_{\mathcal{F}}\}$:

$$\Gamma_{\mathcal{R}}^*, \Gamma_{\mathcal{F}}^* = \arg \min_{\Gamma_{\mathcal{R}}, \Gamma_{\mathcal{F}}} (g_{\Gamma_{\mathcal{R}}}^{(r)} - g_{\Gamma_{\mathcal{F}}}^{(r)}) \cdot g_{\text{FL}}^{(r)} + \sqrt{\kappa} \|g_{\text{FL}}^{(r)}\| \|g_{\Gamma_{\mathcal{R}}}^{(r)} - g_{\Gamma_{\mathcal{F}}}^{(r)}\|.$$

  Update unlearn gradient: $g_{\text{FOUL}}^{(r)} = g_{\text{FL}}^{(r)} + \frac{\kappa \|g_{\text{FL}}^{(r)}\|}{\|g_{\Gamma_{\mathcal{R}}}^{(r)} - g_{\Gamma_{\mathcal{F}}}^{(r)}\|} (g_{\Gamma_{\mathcal{R}}}^{(r)} - g_{\Gamma_{\mathcal{F}}}^{(r)})$.

  Update the model: $\theta_{V,g}^{(r)} = \theta_{V,g}^{(r-1)} - \eta_g g_{\text{FOUL}}^{(r)}$.

end for

E. Detailed Results

Table 3. Comparison of methods on ResNet-18 with VLCS and OfficeHome datasets. The table reports Forget Accuracy (FA), Remaining Accuracy (RA), Testing Accuracy (TA), and Membership Inference Attack (MIA), with values in parentheses showing the difference from the Retrain baseline. For the FOUL method, (1) corresponds to the L2U phase, and (2) refers to the on-server gradient matching unlearning phase. Note that we **bold** results achieving the closest TA accuracy to Retrain.

Group	Methods	VLCS				OfficeHome			
		FA (↓)	RA (↑)	TA (↑)	MIA (↓)	FA (↓)	RA (↑)	TA (↑)	MIA (↓)
Retrain	Retrain	63.14	73.88	70.16	52.96	45.90	66.32	59.08	47.64
	FATS	66.51 (+3.37)	72.09 (−1.79)	68.62 (−1.54)	58.68 (+5.72)	48.73 (+2.83)	64.05 (−2.27)	**58.13** (−0.95)	57.46 (+9.82)
	NoT	65.52 (+2.38)	71.04 (−2.84)	67.83 (−2.33)	61.93 (+8.97)	47.57 (+1.67)	61.51 (−4.81)	56.15 (−2.93)	59.79 (+12.15)
Appro. Unlearn.	FedCDP	69.78 (+6.64)	69.95 (−3.93)	68.63 (−1.53)	74.56 (+21.60)	50.07 (+4.17)	61.83 (−4.49)	57.73 (−1.35)	78.06 (+30.42)
	FedRecovery	69.33 (+6.19)	68.79 (−5.09)	67.61 (−2.55)	77.34 (+24.38)	47.65 (+1.75)	60.81 (−5.51)	56.35 (−2.73)	80.41 (+32.77)
	FedOSD	65.74 (+2.60)	71.85 (−2.03)	67.98 (−2.18)	61.52 (+8.56)	47.81 (+1.91)	64.15 (−2.17)	56.96 (−2.12)	57.21 (+9.57)
	FFMU	66.25 (+3.11)	69.75 (−4.13)	66.55 (−3.61)	63.06 (+10.10)	49.72 (+3.82)	62.66 (−3.66)	56.05 (−3.03)	62.79 (+15.15)
	FUSED	68.08 (+4.94)	70.91 (−2.97)	**69.81** (−0.35)	60.84 (+7.88)	49.41 (+3.51)	62.55 (−3.77)	57.55 (−1.53)	58.72 (+11.08)
	MoDE	64.93 (+1.79)	70.72 (−3.16)	68.19 (−1.97)	62.28 (+9.32)	48.10 (+2.20)	61.39 (−4.93)	56.44 (−2.64)	60.15 (+12.51)
FOUL	(1)	61.72 (−1.42)	76.78 (+2.90)	68.97 (−1.19)	56.85 (+3.89)	43.69 (−2.21)	67.03 (+0.71)	61.96 (+2.88)	56.53 (+8.89)
	(1) + (2)	62.91 (−0.23)	77.15 (+3.27)	69.10 (−1.06)	55.83 (+2.87)	44.17 (−1.73)	65.78 (−0.54)	60.77 (+1.69)	57.48 (+9.84)

F. Evaluations on convergence of learning phase

As the joint loss function comprises multiple components, it is theoretically challenging to optimize. Therefore, we conduct ablation experiments on VLCS dataset by varying the coefficients of each component in the joint loss function, as illustrated in Figures 8. In each experiment, we fix all other coefficients to 1 while varying only one coefficient at a time. As shown in the figures, the model converges within fewer than 100 training rounds, which is comparable to the convergence behavior of vanilla FL. Moreover, the low invariant and variance losses indicate that the learned disentangled representations closely align with the theoretical expectations, where the causal and non-causal representations exhibit invariant and variant properties, respectively.

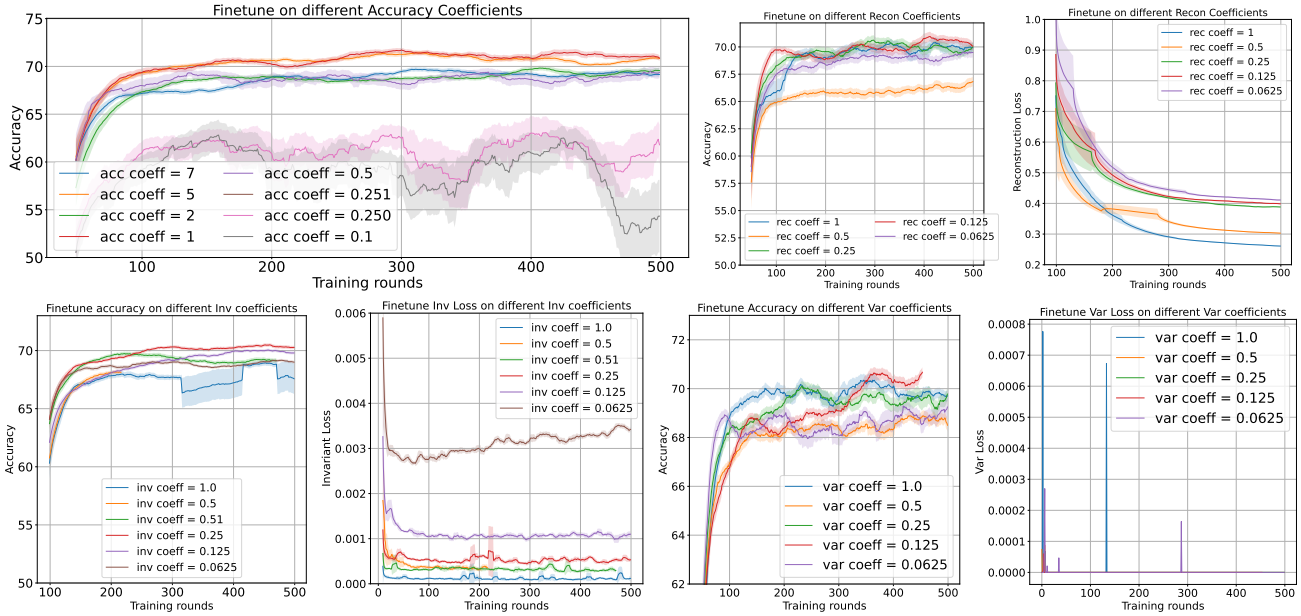


Figure 8. Learning phase of FOUL under different learning coefficient settings.

G. Assessing the domain generalization capabilities of FOUL

We evaluate the performance of FOUL in extracting and predicting invariant representations to verify the robustness of causality-invariant features against domain generalization gaps. To this end, we employ the DomainBed library. Specifically, we train three distinct FOUL models on the Colored-MNIST dataset [14], each using a different domain for training, and subsequently evaluate them on all remaining domains. The results are summarized in Table 4.

Table 4 highlights two key findings. First, the invariant representations z_K capture substantial causal information, enabling accurate prediction of the ground-truth labels in the source domain, which is consistent with the underlying CSM. Second, these invariant representations remain consistent across different data domains. Consequently, by preserving the knowledge of the semantic featurizer during the unlearning phase, FOUL retains the domain-invariant knowledge, ensuring that unlearning on the forget set minimally impacts performance on the retain set.

Table 4. Assessment of FOUL across different domains. The evaluation is conducted on the Colored-MNIST dataset, where each domain corresponds to images with varying degrees of correlation between color and label. Two aspects are assessed: (1) Accuracy: evaluating the model's ability to predict labels in each domain using only the invariant representations; and (2) Invariance: measuring the disparity between the invariant representations z_K^i generated from a target domain i and those z_K^j obtained from the training domain j, to quantify the consistency of invariant features across domains.

Algorithm	**+90%**	**+80%**	**-90%**
	Domain 1 (+90%)		
Accuracy	89.5 $\pm$ 0.1	82.3 $\pm$ 0.5	73.2 $\pm$ 0.3
Invariant	0.0003 $\pm$ 10e-5	0.0002 $\pm$ 10e-5	0.0002 $\pm$ 10e-5
	Domain 2 (+80%)		
Accuracy	73.9 $\pm$ 0.1	82.3 $\pm$ 0.5	73.2 $\pm$ 0.3
Invariant	0.0003 $\pm$ 10e-5	0.0002 $\pm$ 10e-5	0.0002 $\pm$ 10e-5
	Domain 3 (-90%)		
Accuracy	89.5 $\pm$ 0.1	80.7 $\pm$ 0.5	73.2 $\pm$ 0.3
Invariant	0.0003 $\pm$ 10e-5	0.0002 $\pm$ 10e-5	0.0002 $\pm$ 10e-5

H. Assessment of Knowledge Extraction Capability

To determine the extent of invariant knowledge extraction from the data and the appropriate semantic knowledge size to represent the data from each label, we keep the size of the embedding layer constant (i.e., $C_{\text{IB}} = 32$), while simultaneously varying the number of invariant channels and setting the variant channel size to $C_{\text{v}} = C_{\text{IB}} - C_{\text{iv}}$. This approach is adopted to ensure that the performance of data reconstruction remains unaffected by the informational bottleneck. The results evaluated on VLCS dataset are shown in Fig. 9. As can be seen, the L2U stage performs optimally when the number of invariant knowledge channels are set to $C \geq 16$. Additionally, the data reconstruction remains consistent as the invariant knowledge size is increased. This indicates that higher data reconstruction efficiency can be achieved by using a larger invariant knowledge size (e.g., $C_{\text{iv}} \geq 24$, $C_{\text{Var}} \leq 8$). This also implies that a high compression ratio of $(3 \times 32 \times 32) : (8 \times 64) = 6 : 1$ can be attained without quality loss in the variant data, which needs to be transmitted over the physical channel.

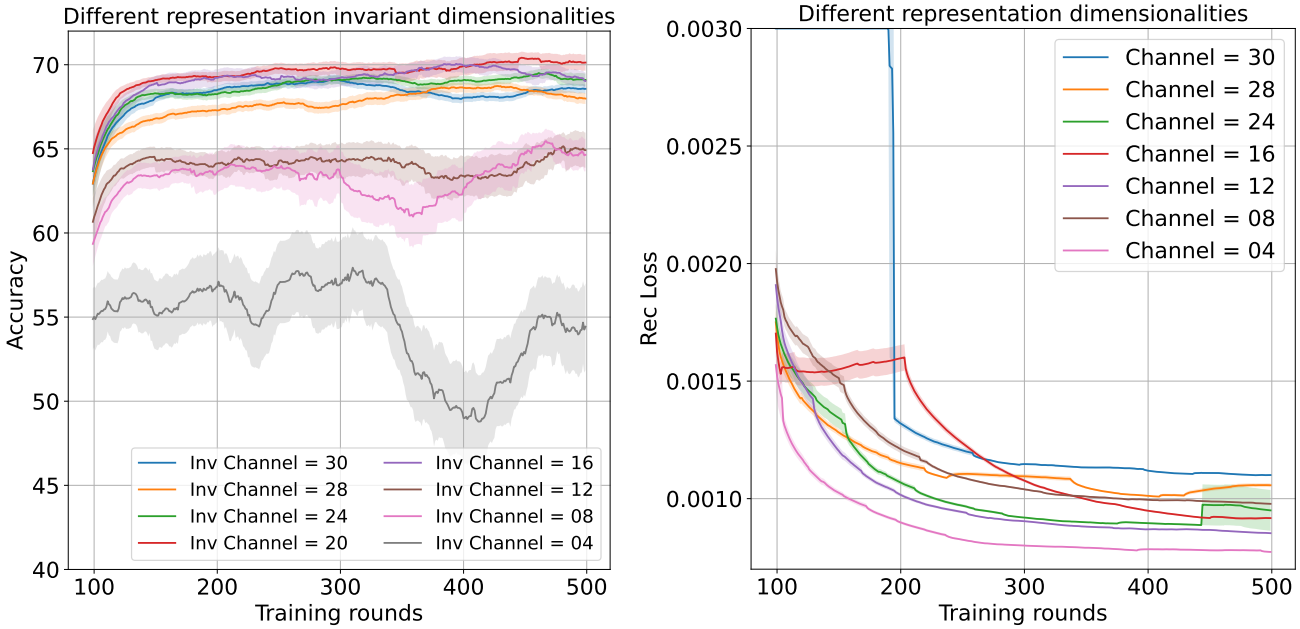
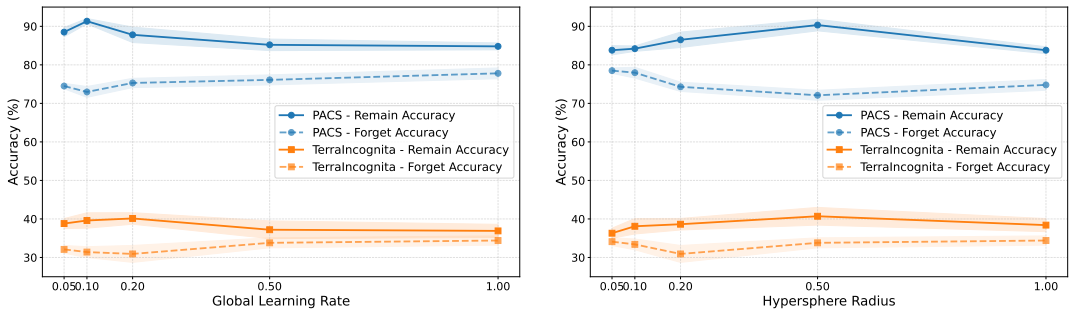


Figure 9. Illustrations of the efficacy of invariant knowledge featurizer.

I. Ablation test on unlearning stage



(a) (b)

Figure 10. Effects of different global learning rates and hypersphere radii on FOUL performance. For both PACS and Terra Incognita, the model achieves the best remaining accuracy (RA) and forgetting accuracy (FA) when $\eta = 0.2$ and $\kappa = 0.5$.

Learning rate. In Fig. 10a, we present the FOUL results obtained with varying learning rates η, while fixing $\kappa = 0.5$. Each result is averaged over three independent runs. For both the PACS and TerraIncognita datasets, the gradient matching task performs best when $\eta = 0.2$, achieving the best RA and FA.

Sensitivity of gradient matching. We evaluate the sensitivity of the gradient matching process to different values of κ on two datasets. Each experiment is conducted three times, and the results are averaged. As shown in Fig. 10b, FOUL is optimal when the hypersphere radius is set to 0.5 on PACS and between 0.2 and 0.5 on TerraIncognita, where the gap between RA and FA is the largest.

Disentanglement Efficiency. To consider the disentanglement efficiency towards the unlearning phase, we deploy the ablation test according to different causal/non-causal representations dimensionality. We measure the RA and FA according to each cases and evaluate. The results is given in Figure 11.

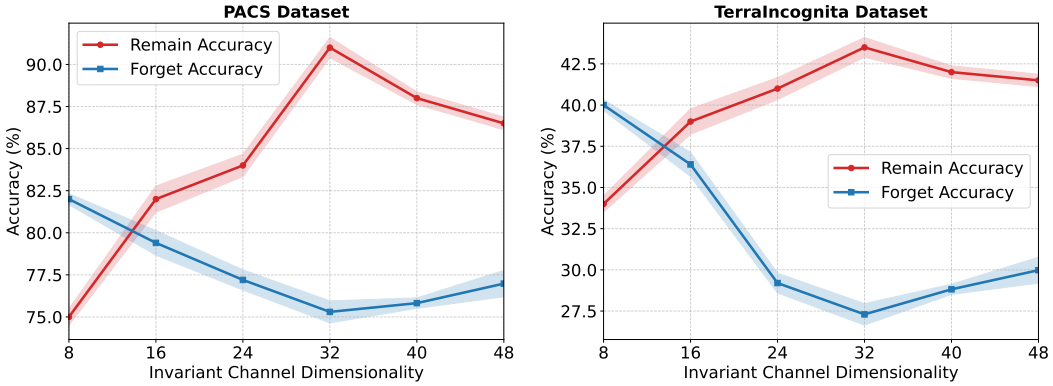


Figure 11. Ablation test on different causal dimensionality of the model disentanglement.

J. Discussion

FOUL offers an efficient unlearning mechanism by directly aggregating information from both forget and retain clients on the server. Moreover, restricting unlearning to a sub-network significantly reduces computational cost. However, FOUL also has limitations. In particular, the gradient-alignment optimization performs best when the number of clients is moderate (e.g., fewer than 100). When the client population becomes extremely large, the optimization may converge to suboptimal solutions. Addressing scalability and robustness in such large-client regimes is an important direction for future research.