Federated Unlearning via On-server Gradient Conflict Mitigation and Expression

Minh-Duong Nguyen[1], Senura Hansaja Wanasekara[2], Le-Tuan Nguyen[1], Ken-Tye Yong[2], Quoc-Viet Pham[3], Nguyen H. Tran[2], Dung D. Le[1]

[1] VinUniversity, [2] University of Sydney, [3] Trinity College Dublin

E-mail: duong.nm2@vinuni.edu.vn
GitHub: Skydvn
Google Scholar: Minh-Duong Nguyen

June 3, 2026, Denver, Colorado

Outline

Overview

Key Contributions

Proposed Method

Experimental Evaluations

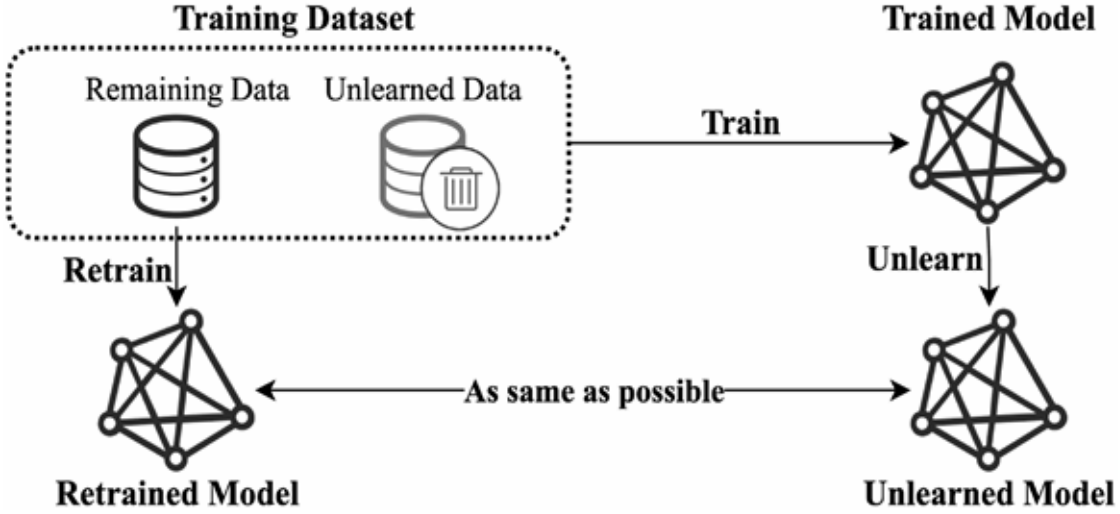
Overview

Federated Unlearning

Federated Unlearning consists of two stages: 1) Train stage and 2) Unlearn stage.

1) ***Train stage:*** Collaborative learn a global model using the full dataset, including both retain and forget data.

→ This is where **standard FL algorithms** operate.



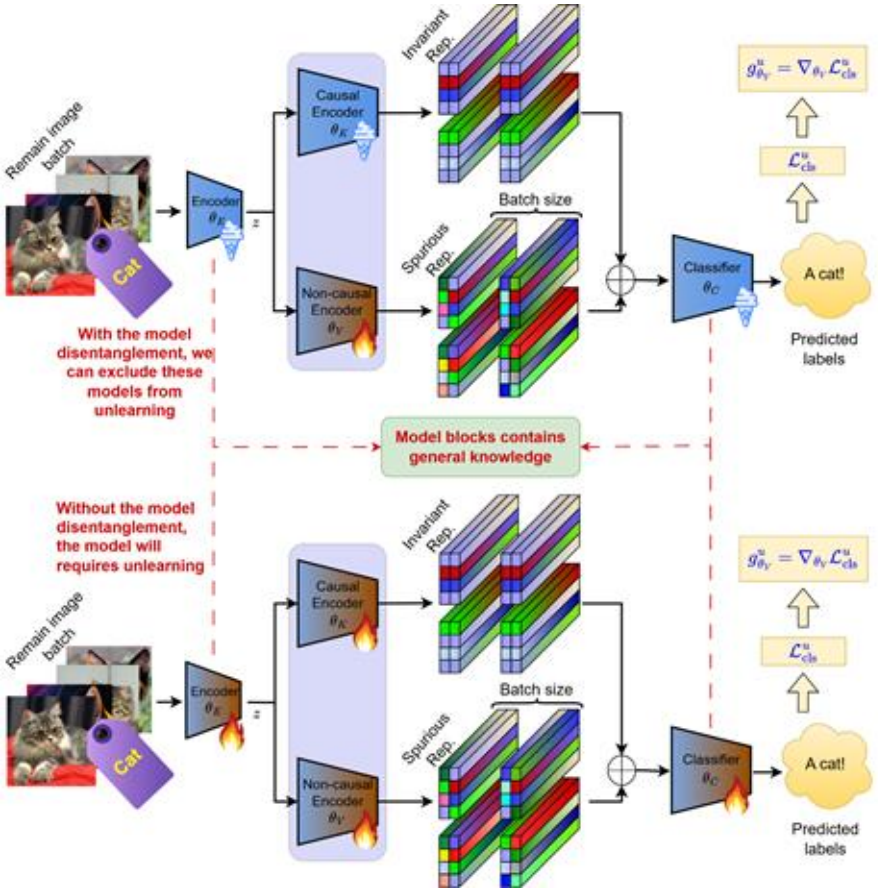
2) Unlearn stage: Match the performance of a model trained only on retain data, while removing knowledge of forget data.

→ This requires a different strategy from standard FL because the objective and data composition change.

Key Contributions

1	Propose **FOUL** (Federated On-server UnLearning), a two-stage framework for client-wise FUL.
2	**Learning-to-Unlearn (L2U):** disentangle features into causal / domain-invariant and non-causal / domain-specific parts.
3	**On-server unlearning:** update only the non-causal branch via gradient matching, using gradients from both retain and forget clients.
4	Introduce **Domain-wise FUL benchmark** setting based on domain-generalization datasets.
5	Introduce **Time-to-Forget (T2F)**, a metric measuring how quickly the model reaches effective forgetting.

Proposed Method



What is need-to-be-unlearn block?

1. During training, we disentangle FL into two types of blocks

- **Causal blocks:** capture general, task-relevant knowledge shared across clients.
- **Non-causal blocks:** capture client-specific or domain-specific knowledge.

2. For client unlearning, only the corresponding non-causal blocks need to be updated.

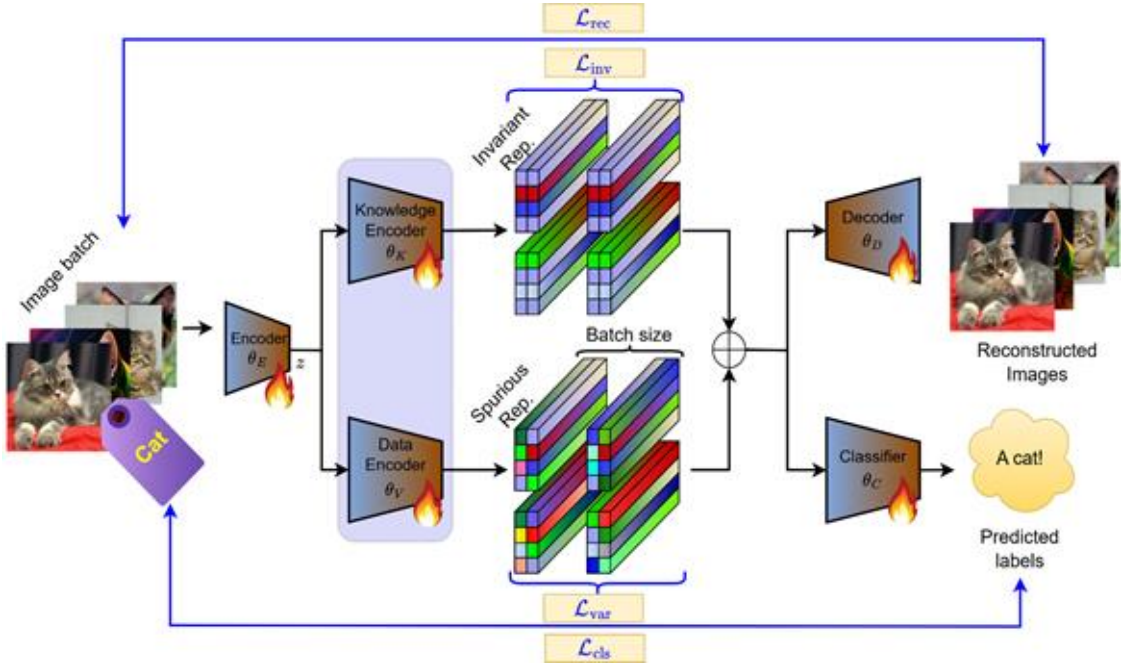
➔ Unlearn client-specific knowledge without retraining the entire model.

Proposed Method

Train stage

We apply four losses to design a joint loss for model disentanglement during the training stage:

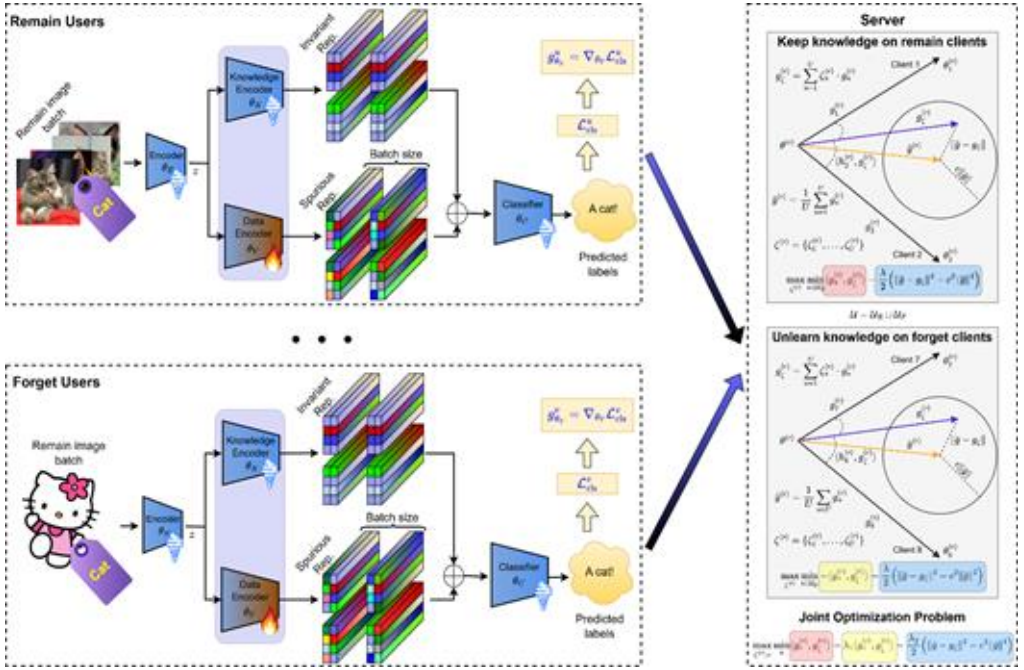
- Reconstruction loss ($\mathcal{L}_{rec}$) guarantees the meanings of the disentangled representations
- Causal loss ($\mathcal{L}_{inv}$) and non-causal loss ($\mathcal{L}_{var}$) guarantees the corresponding causality properties of causal and non-causal encoders.
- Classification loss ($\mathcal{L}_{cls}$) is the main task loss, guarantees the task of the FL.



Proposed Method

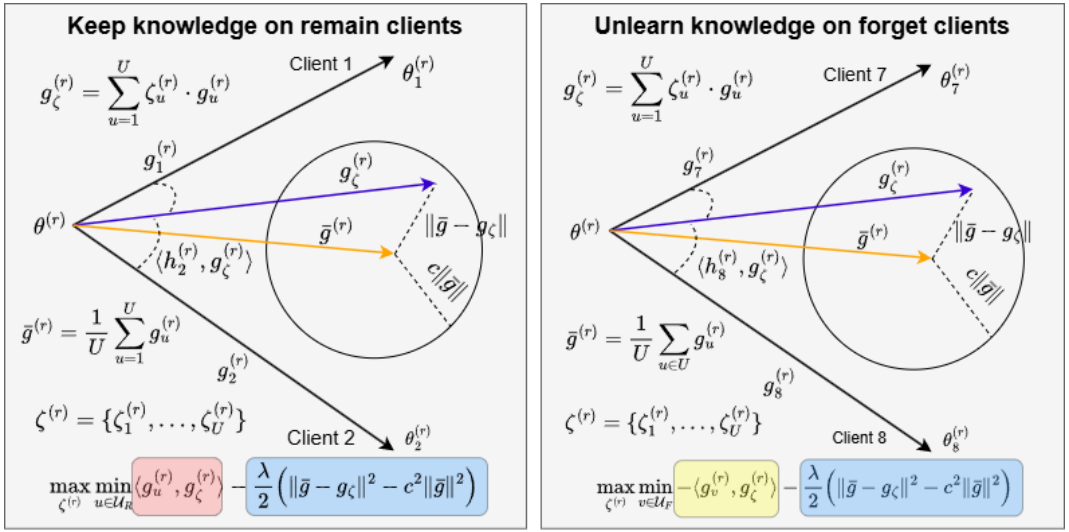
Unlearn stage

1. We train on the client side normally to get the local gradients.
2. We leverages the updated local gradients to train on the server side.
3. We propose a gradient inner product based optimization to find a solution that **converge** on remain clients, and **diverge** the forget clients.
4. We only apply the updated gradients on the **non-causal encoders**.



Proposed Method

Gradient Expression and Mitigation



1. Retained-client gradients are pulled into a shared acute-angle region to preserve convergence
2. Forget-client gradients are pushed into an obtuse-angle region relative to retained clients, preventing forgotten data from positively influencing the global update.

Experimental Evaluations

Group	Methods	PACS				TerraIncognita			
		FA (↓)	RA (↑)	TA (↑)	MIA (↓)	FA (↓)	RA (↑)	TA (↑)	MIA (↓)
Retrain	Retrain	70.51	82.84	77.45	50.02	30.64	42.41	38.94	50.71
	FATS	74.45 (+3.94)	80.91 (−1.93)	75.98 (−1.47)	55.72 (+5.70)	33.07 (+2.43)	40.96 (−1.45)	37.34 (−1.60)	60.81 (+10.10)
	NoT	73.24 (+2.73)	79.25 (−3.59)	75.28 (−2.17)	59.13 (+9.11)	32.26 (+1.62)	38.32 (−4.09)	36.08 (−2.86)	62.90 (+12.19)
Appro. Unlearn.	FedCDP	77.37 (+6.86)	78.13 (−4.71)	76.41 (−1.04)	72.26 (+22.24)	34.56 (+3.92)	38.79 (−3.62)	36.75 (−2.19)	81.91 (+31.20)
	FedRecovery	76.48 (+5.97)	76.97 (−5.87)	74.81 (−2.64)	75.24 (+25.22)	32.11 (+1.47)	37.49 (−4.92)	35.59 (−3.35)	84.03 (+33.32)
	FedOSD	72.89 (+2.38)	80.15 (−2.69)	75.49 (−1.96)	58.84 (+8.82)	32.63 (+1.99)	40.77 (−1.64)	36.76 (−2.18)	60.54 (+9.83)
	FFMU	73.31 (+2.80)	78.27 (−4.57)	74.14 (−3.31)	60.62 (+10.60)	33.76 (+3.12)	39.39 (−3.02)	36.04 (−2.90)	65.85 (+15.14)
	FUSED	75.94 (+5.43)	79.34 (−3.50)	76.86 (−0.59)	58.72 (+8.70)	33.64 (+3.00)	38.73 (−3.68)	36.74 (−2.20)	62.03 (+11.32)
	MoDE	72.53 (+2.02)	79.01 (−3.83)	75.79 (−1.66)	59.79 (+9.77)	32.17 (+1.53)	38.04 (−4.37)	35.74 (−3.20)	63.47 (+12.76)
FOUL	(1)	**69.53** (−0.98)	**93.11** (+14.55)	**77.14** (−0.31)	53.82 (+3.80)	**27.97** (−2.67)	**43.81** (+1.40)	38.16 (−0.78)	**56.40** (+5.69)
	(1) + (2)	70.97 (+0.46)	92.33 (+14.49)	76.43 (−1.02)	**51.93** (+1.91)	29.92 (−0.72)	42.13 (−0.28)	**39.16** (+0.22)	57.11 (+6.40)

Table 1: Comparison of methods on ResNet-18 with PACS and TerraIncognita datasets.

Method	Params (M) (M)	Comm. (MB) (MB)	Comp. (FLOPs) (FLOPs)
Retrain	11.3	42.73	$5.81e^{16}$
FATS	11.3	42.73	$5.81e^{16}$
NoT	11.3	34.72	$3.38e^{16}$
FedCDP	11.3	42.73	$5.82e^{16}$
FedRecovery	11.3	42.73	$5.96e^{16}$
FedOSD	11.3	42.73	$5.85e^{16}$
FFMU	11.3	42.73	$5.85e^{16}$
FUSED	11.3	**0.98**	$2.81e^{16}$
MoDE	11.3	42.73	$3.72e^{16}$
FOUL (1) + (2)	11.3	16.02	$\mathbf{2.35e^{16}}$

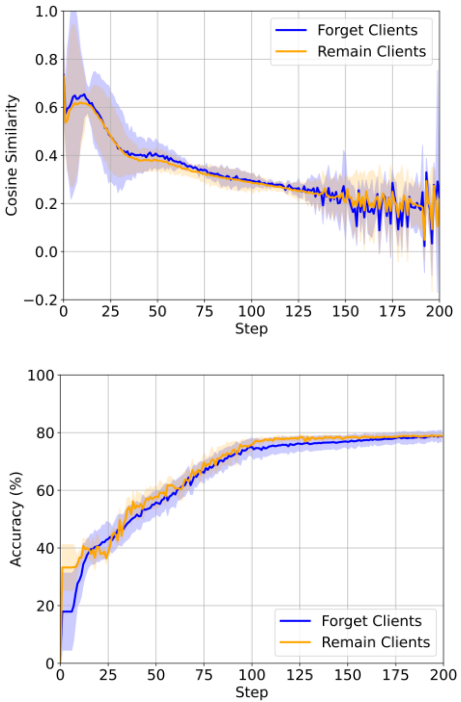
Table 2: Comparison of computational and communication cost

- FOUL consistently outperforms all baselines across different architectures and datasets.
- FOUL achieves a smaller average performance gap while maintaining competitive system costs.

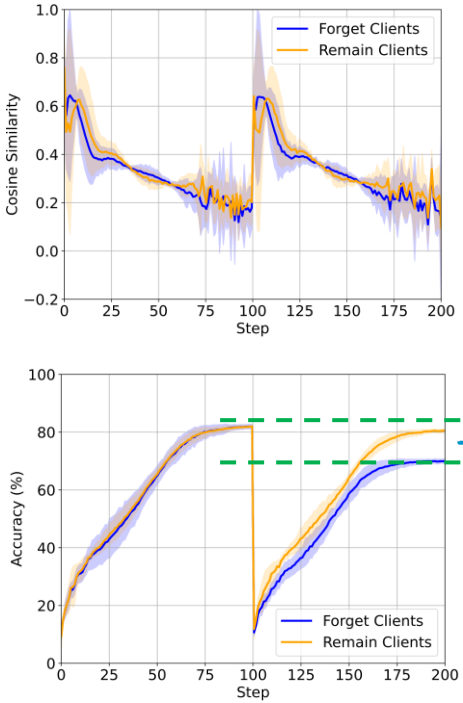
Experimental Evaluations

Time-to-Forget

Retrain with pre-trained model

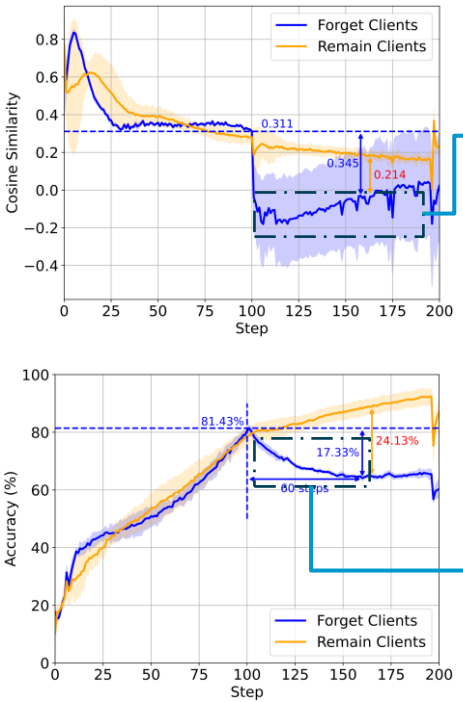


Retrain with model reset



The unlearn efficiency is low

FOUL



The gradient angle is reduced selectively.

Only clients in forget set are unlearned

Thank you