Federated Model Synchronization for Diagnostic Redefinition through a Novel Selective Parameter Unlearning

1 Data Distribution Analysis

To visualize the induced non-IID heterogeneity, Fig. 1 (PathMNIST) shows that clients receive skewed class proportions, with several classes underrepresented or absent at certain clients.

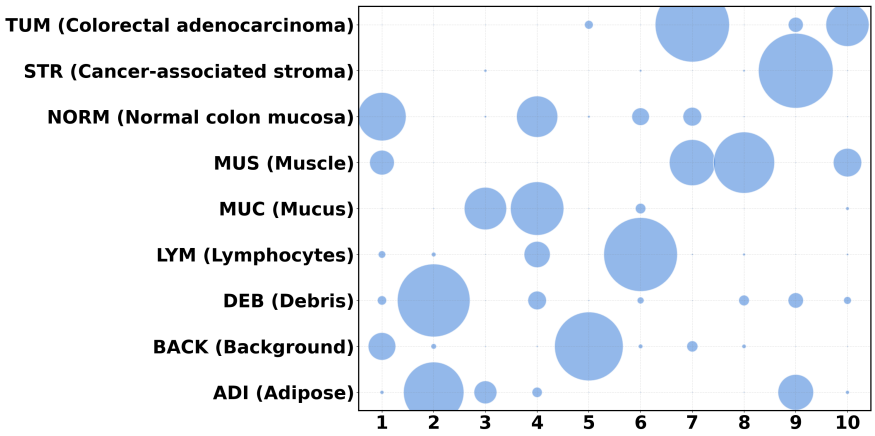

Figure 1: Non-IID data distribution across 10 clients achieved using Dirichlet distribution with $\alpha = 0.1$, showing heterogeneous class representation across clients for the PathMNIST dataset. The x-axis indicates client indices (1–10), while the y-axis denotes the nine tissue types. Bubble size reflects the number of samples per class at each client.

2 Additional Results on PathMNIST

Figure 2 compares the class-wise accuracy of our proposed approach compared to the other baselines. We observe that while the accuracy for the *Debris* class drops significantly, the performance for other tissue types remains stable or slightly improved. This is particularly interesting because even though tissue types share visual features, our method is able to selectively remove *Debris-specific* features without disturbing the model's ability to identify other tissues. This stability in identifying non-target tissues is crucial for medical applications where accurate diagnosis must be maintained.

Figure 2: Class-wise accuracy comparison on the PathMNIST dataset when Debris is used as the forget class. Our method demonstrates a significant drop in Debris recognition while preserving performance on other tissue types.

Method	Set	Adipose	Background	Debris	Lympho	Mucus	Muscle	Normal	Stroma	Tumor
Original	**FS**	0.79	0.82	0.85	0.74	0.75	0.80	0.77	0.78	0.83
	RS	0.88	0.87	0.87	0.87	0.87	0.86	0.87	0.87	0.87
Fine-tune	**FS**	0.76	0.79	0.83	0.71	0.73	0.77	0.74	0.76	0.81
	RS	0.88	0.87	0.88	0.88	0.88	0.87	0.88	0.87	0.88
Retrain	**FS**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00
	RS	0.89	0.87	0.90	0.90	0.87	0.87	0.88	0.89	0.89
TF-IDF	**FS**	0.08	0.12	0.06	0.11	0.08	0.13	0.11	0.09	0.03
	RS	0.87	0.86	0.87	0.87	0.86	0.85	0.86	0.86	0.88
DE-Un	**FS**	0.09	0.11	0.07	0.10	0.10	0.14	0.10	0.09	0.04
	RS	0.88	0.87	0.89	0.88	0.88	0.87	0.87	0.88	0.88
SalUn	**FS**	0.06	0.07	0.07	0.08	0.03	0.07	0.07	0.06	0.04
	RS	0.86	0.90	0.88	0.89	0.89	0.88	0.89	0.88	0.89
Ours w/o FT	**FS**	0.04	0.03	0.05	0.08	0.03	0.05	0.02	0.03	0.01
	RS	0.84	0.82	0.83	0.84	0.82	0.82	0.83	0.82	0.83
Ours	**FS**	0.04	0.04	0.01	0.05	0.04	0.05	0.04	0.04	0.03
	RS	0.90	0.89	0.90	0.90	0.88	0.89	0.90	0.90	0.90

Table 1: Class-wise unlearning performance (F1-Score) on the PathMNIST dataset. FS denotes the Forget Set (selected class to be forgotten) and RS denotes the Retain Set (remaining classes to be preserved).

Table 1 reports the class-wise unlearning performance (F1-Score) on the PathMNIST dataset. For each experiment, one class is designated as the Forget Set (FS), while the remaining classes constitute the Retain Set (RS). As expected, all baseline methods maintain high performance on the RS, but differ significantly in how effectively they suppress information on the FS. Standard retraining nearly eliminates FS performance while preserving RS accuracy, but at the expense of computational cost. In contrast, existing unlearning

baselines (TF-IDF, DE-Un, SalUn) reduce FS scores to varying degrees but leave traces of residual knowledge. Our approach achieves the strongest forgetting effect—driving FS F1-scores close to zero—while still sustaining high RS performance, highlighting its ability to balance effective unlearning with retention fidelity.

Pruning Ratio	Set	Accuracy (%)
1%	FS (*Muscle+Stroma*)	3.15 ± 0.71
	RS	86.75 ± 1.22
2%	FS (*Muscle+Stroma*)	1.94 ± 1.18
	RS	82.32 ± 0.97

Table 2: Performance when forgetting two classes (*Muscle* and *Stroma*) simultaneously. We report accuracy on the forget set (FS) and retain set (RS) for two pruning ratios: 1% and 2%. All values are mean ± standard deviation over three runs.

2.1 Multi-Class Forgetting

To evaluate the effect of forgetting multiple classes on the overall behavior of the model, we extend our unlearning experiments to simultaneously remove two target classes—*Muscle* and *Stroma*. This scenario is more challenging than single-class unlearning, as it involves removing a larger portion of class-specific knowledge while attempting to preserve the rest of the model's functionality. We apply our pruning method with two different pruning ratios to see how the amount of zeroed parameters affects both the forgotten and retained classes.

The results show that the model successfully reduces performance on the combined forget set under both pruning configurations, confirming the method's ability to forget multiple classes in a single pass. However, we also observe that increasing the pruning ratio to achieve stronger forgetting leads to a more noticeable decline in retain-set accuracy. This decline reflects the increasing overlap between features of the forget and retain classes, particularly in medical imaging, where different conditions often share similar visual patterns.

Scenario	FS Acc. (%) ↓	RS Acc. (%) ↑
Original	84.6 ± 0.8	85.9 ± 1.0
Only-forget	1.2 ± 0.4	87.5 ± 0.8
No-forget	1.9 ± 0.8	88.1 ± 1.2

Table 3: Performance under heterogeneous client scenarios (PathMNIST, Debris as forget class). FS denotes Forget Set (↓ desirable), RS denotes Retain Set (↑ desirable). Values are mean ± std over three runs.

2.2 Client Scenarios

The proposed approach also supports heterogeneous participation of clients. To validate this, we conduct an experiment on the **PathMNIST** dataset (Debris as the forget class) where we distribute data among 10 clients as following:

1. **Only-forget client:** a client containing only forget-class samples.
2. **No-forget client:** a client containing only retain-class samples.

The remaining 9 clients are assigned data using the standard Dirichlet partitioning with concentration parameter $\alpha = 0.1$, simulating realistic heterogeneity across the federation.

Table 3 reports the results. The **Original** model before unlearning is included as a baseline reference. For the only-forget client, pruning removes class-specific parameters and fine-tuning is skipped, yet the aggregated model eliminates residual influence of the forgotten class. The no-forget client never observes the forget class locally, but after aggregation the global model still exhibits near-complete forgetting on the

test set. Both scenarios preserve retention accuracy comparable to the Original model. This provides an evidence that our proposed framework robustly accommodates heterogeneous client types.

2.3 Selective Parameter Pruning

We compare pruning at different granularities on PathMNIST with *Debris* as the forget class to analyze the effectiveness of our approach. We include: (i) *Random-Param*, pruning a random 1% of parameters; (ii) *Select-Layer*, where an entire layer with the highest aggregate importance score is zeroed out; (iii) *Channel-level* baselines (TF-IDF, DE-Un); and (iv) *Ours*.

Method	FS Acc (%)	RS Acc (%)	NCR
Random-Param	23.3	71.1	1
Select-Layer (entire layer)	0.0	43.4	1
TF-IDF (channel)	2.8	86.0	5
DE-Un (channel)	3.1	87.6	4
Ours	**1.1**	**88.8**	**1**

Table 4: Ablation on pruning granularity for PathMNIST (Debris forget class). FS: forget-set accuracy ↓, RS: retain-set accuracy ↑, NCR: communication rounds.

Removing an entire layer achieves forgetting but causes severe degradation in retain-class accuracy, highlighting the drawback of such coarse pruning. Random pruning fails to erase the forget class. Channel-level methods perform better but remain inferior to our approach, which achieves near-complete forgetting with strong retention.

3 Additional Results on COVID-19 Radiography Dataset

The class-wise performance analysis is shown in Figure 3. We observe a substantial drop in accuracy for the target class, while the accuracy on the retained classes remains preserved. Most importantly, the model maintains its high accuracy in detecting *COVID-19*, which is crucially important for clinical applications. This demonstrates that our method can selectively remove class-specific features while keeping other important diagnostic capabilities intact. Different respiratory conditions in chest X-rays often present with similar visual patterns, making this a particularly challenging unlearning scenario.

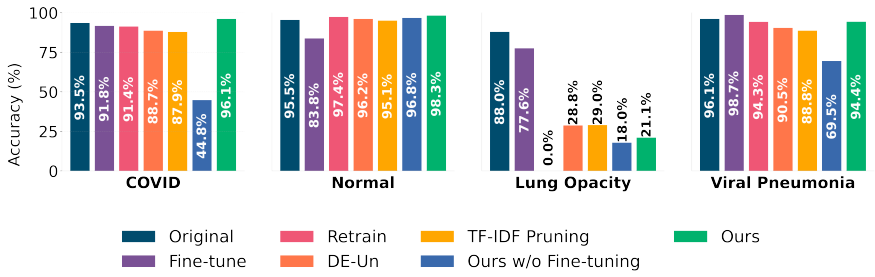

Figure 3: Class-wise accuracy for COVID-19 Radiography Dataset when Lung Opacity is the forget class. Our approach selectively reduces accuracy on the target class while preserving strong performance on COVID-19, Normal, and Viral Pneumonia classes.

Table 5 shows the class-wise F1-scores for unlearning on the COVID-19 Radiography dataset. Retraining removes all knowledge of the forget set ($F1 \approx 0$) and keeps the retain set performance high, but it is very costly. Fine-tuning forgets only partially, leaving traces behind. Baseline methods (TF-IDF, DE-Un, SalUn) reduce forget set scores to some extent but not fully. Our method goes further by lowering forget set performance even below random guessing, while still keeping retain set accuracy almost unchanged, showing both effective and efficient forgetting.

Method	Set	COVID-19	Lung Opacity	Normal	Viral Pneumonia
Original	**FS**	0.91	0.90	0.92	0.89
	RS	0.96	0.96	0.95	0.96
Fine-tune	**FS**	0.83	0.81	0.84	0.81
	RS	0.95	0.92	0.94	0.95
Retrain	**FS**	0.00	0.00	0.00	0.00
	RS	0.94	0.95	0.93	0.95
TF-IDF	**FS**	0.38	0.32	0.36	0.38
	RS	0.95	0.94	0.94	0.97
DE-Un	**FS**	0.41	0.32	0.35	0.39
	RS	0.94	0.95	0.92	0.95
SalUn	**FS**	0.28	0.26	0.29	0.38
	RS	0.96	0.96	0.95	0.96
Ours w/o FT	**FS**	0.28	0.21	0.27	0.31
	RS	0.91	0.91	0.90	0.93
Ours	**FS**	0.31	0.25	0.29	0.34
	RS	0.97	0.97	0.96	0.97

Table 5: Class-wise unlearning performance (**F1-score**) on the COVID-19 Radiography dataset. FS indicates the Forget Set (target class for removal) and RS represents the Retain Set (remaining classes to preserve).

4 Computation Cost Analysis

4.1 Efficiency and Runtime Overhead

We compare wall-clock runtime per client on an NVIDIA RTX A6000 (batch size 64, high-resolution PathMNIST with 224×224×3 images). Table 6 reports the time to complete a single round and the number of communication rounds (NCR) each method requires. For retraining, one standard FL round takes ∼13 minutes and full unlearning needs 100 rounds. Our method completes unlearning in a single round (∼28 minutes), which *includes* both the Shapley-based parameter scoring for the forget set and one fine-tuning pass. For the channel-level baseline, the time for the first round is (∼34 minutes) which includes channel selection and fine-tuning, and (∼13 minutes) per round for next three rounds. The fine-tuning baseline runs one round (∼13 minutes) but does not achieve effective forgetting. Overall, our framework achieves effective unlearning in a single round.

Method	**Runtime / Round**	**NCR**
Retrain	∼13 minutes	100
Fine-tune	∼13 minutes	1
Channel-level pruning	∼34 minutes	4
Ours	∼28 minutes	1

Table 6: Runtime comparison on PathMNIST (high-resolution 224×224×3, Debris as forget class). Runtime is per round; NCR is number of communication rounds.

4.2 Overhead of Parameter Selection

Our Shapley-value approximation computes the importance of parameter θ_j via Monte-Carlo sampling:

$$\hat{\phi}_j \approx \frac{1}{R} \sum_{r=1}^{R} \|\nabla_{\theta_j} L_r \cdot \theta_j\|_1,$$

where $\nabla_{\theta_j} L_r$ is the gradient of the loss on a forget-set mini-batch in iteration r. Each iteration requires one forward+backward pass, and gradients for all parameters are obtained simultaneously.

Complexity. For a client with forget set size $|D_f|$, batch size B, and R repeats:

$$C_{\text{shap}} = \frac{|D_f|}{B} \cdot R \cdot T_{FB},$$

where T_{FB} is the cost of one forward+backward pass. A standard training round with local dataset size $|D|$ and E_{local} epochs costs:

$$C = \frac{|D|}{B} \cdot E_{\text{local}} \cdot T_{FB}.$$

Hence,

$$\frac{C_{\text{shap}}}{C} = \frac{|D_f| \cdot R}{|D| \cdot E_{\text{local}}}.$$

Example. The PathMNIST dataset contains 107,180 samples across 9 classes. With 10 participating clients, each client receives approximately $|D| \approx 10{,}000$ samples. If one class is designated as the forget set, then on average each client contributes about $|D_f| \approx 1{,}000$ forget-class samples.

For local training we use $E_{\text{local}} = 5$ epochs. The Shapley-value scoring is repeated $R = 10$ times, so the relative overhead is:

$$\frac{C_{\text{shap}}}{C} = \frac{|D_f| \cdot R}{|D| \cdot E_{\text{local}}} = \frac{1000 \cdot 10}{10000 \cdot 5} = \frac{10000}{50000} = 0.2.$$

Thus, the parameter-scoring overhead is only 20% of a single training round. In wall-clock terms (NVIDIA RTX A6000, batch size 64, high-resolution 224×224×3 PathMNIST), this corresponds to ∼28 minutes for our complete unlearning step (scoring + fine-tuning), compared to ∼13 minutes for one standard FL training round and over 10 hours total for retraining with 100 rounds.

5 Statistical Significance Analysis

In this experiment, we conduct statistical hypothesis testing to evaluate whether the observed differences in performance are statistically significant. We employed Welch's two-sample t-test (independent samples with unequal variance), which is appropriate for three runs.

Baseline Selection

The choice of baseline is critical for meaningful evaluation:

- **Forget Set (FS):** The baseline is the **Retrain–FS** configuration, where the model is retrained from scratch after removing the forget set. This represents the "ideal" unlearning scenario, and thus all FS results were compared against it.
- **Retain Set (RS):** The baseline is the **Original–RS** configuration, i.e., the performance of the original model before unlearning. This captures the best-case retention performance, and all RS results were compared against it.

PathMNIST Dataset

Table 7 presents raw p-values from Welch's t-test for statistical significance analysis on the PathMNIST dataset across all nine pathological tissue classes. The statistical evaluation follows a structured protocol where FS (forget set) methods are compared against the Retrain FS baseline to assess forgetting effectiveness, while RS (retain set) methods are tested against the Original RS baseline to evaluate knowledge retention. As expected, Retrain FS serves as the baseline reference (no values reported), while Original RS naturally yields $p = 1.0$ across all classes since it represents self-comparison. Lower p-values indicate stronger statistical significance in performance differences, with values below 0.05 typically considered statistically significant.

Method	Set	Adipose	Background	Debris	Lympho	Mucus	Muscle	Normal	Stroma	Tumor
Original	FS	0.0002	0.0002	0.0002	0.0003	0.0003	0.0003	0.0004	0.0003	0.0003
	RS	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Fine-tune	FS	0.0003	0.0003	0.0003	0.0005	0.0003	0.0004	0.0004	0.0004	0.0003
	RS	0.4755	0.6439	0.2149	0.3351	0.4128	0.6224	0.6522	0.3424	0.3316
Retrain	FS	–	–	–	–	–	–	–	–	–
	RS	0.3451	0.7797	0.0683	0.0828	0.9553	0.9507	0.4337	0.1158	0.1375
TF-IDF	FS	0.0570	0.0294	0.1028	0.0471	0.0805	0.0260	0.0297	0.0483	0.1424
	RS	0.1339	0.3428	0.8407	0.4564	0.1780	0.3391	0.5558	0.5181	0.4847
DE-Un	FS	0.0256	0.0179	0.0467	0.0308	0.0435	0.0215	0.0194	0.0358	0.0871
	RS	0.6419	0.6657	0.1399	0.2757	0.1114	0.2092	0.5140	0.2649	0.1876
SalUn	FS	0.0146	0.0097	0.0218	0.0191	0.0230	0.0129	0.0109	0.0165	0.0282
	RS	0.1913	0.2101	0.1486	0.2282	0.2387	0.1662	0.2851	0.1909	0.2331
Ours w/o FT	FS	0.0125	0.0086	0.0194	0.0152	0.0184	0.0098	0.0084	0.0127	0.0215
	RS	0.0123	0.0092	0.0175	0.0134	0.0158	0.0102	0.0099	0.0128	0.0144
Ours	FS	0.0089	0.0067	0.0103	0.0098	0.0107	0.0073	0.0064	0.0091	0.0122
	RS	0.8698	0.1947	0.0462	0.2457	0.1559	0.1875	0.3642	0.2821	0.2375

Table 7: Raw p-values from Welch's t-test on the PathMNIST dataset. FS rows are compared against the Retrain–FS baseline; RS rows against the Original–RS baseline.

COVID-19 Radiography Dataset

Table 8 reports raw p-values from Welch's t-test for the COVID-19 Radiography dataset following the same statistical evaluation protocol. FS methods are compared against the Retrain FS baseline, while RS methods are tested against the Original RS baseline. As expected, Retrain FS serves as the baseline itself (no values reported), while Original RS naturally yields $p = 1.0$ since it represents a self-comparison.

Method	**Set**	**COVID-19**	**Lung Opacity**	**Normal**	**Viral Pneumonia**
Original	FS	0.0005	0.0000	0.0000	0.0003
	RS	1.0000	1.0000	1.0000	1.0000
Fine-tune	FS	0.0002	0.0000	0.0001	0.0000
	RS	0.8063	0.0334	0.4331	0.2791
Retrain	FS	–	–	–	–
	RS	0.1584	0.2746	0.0812	0.2371
TF-IDF	FS	0.0001	0.0009	0.0001	0.0006
	RS	0.1677	0.1528	0.0293	0.4226
DE-Un	FS	0.0002	0.0016	0.0001	0.0013
	RS	0.0500	0.0958	0.5015	0.1518
SalUn	FS	0.0001	0.0029	0.0007	0.0049
	RS	0.2114	0.7851	0.1205	0.2122
Ours w/o FT	FS	0.0003	0.0016	0.0013	0.0010
	RS	0.0032	0.0164	0.0039	0.0030
Ours	FS	0.0014	0.0005	0.0054	0.0001
	RS	0.2602	0.1842	0.8294	0.4411

Table 8: Raw p-values from Welch's t-test on the COVID-19 Radiography dataset. FS rows are compared against the Retrain–FS baseline; RS rows against the Original–RS baseline.

Key Findings

These statistical results reinforce the main findings:

- Forget sets are consistently and significantly degraded, validating the effectiveness of unlearning across datasets.
- Retain sets remain stable, with most methods showing no significant deviation from the baseline, demonstrating that retention is preserved.

- Our proposed method achieves strong targeted forgetting while largely maintaining retention, thereby striking the desired balance between unlearning and utility.

6 Detailed Algorithm

This algorithm details the full procedure used to perform federated unlearning. The method operates entirely on client devices and exchanges only updated model parameters with the server, never raw data. Given a pre-trained global model θ, a set of client datasets $\{\mathcal{D}_i\}_{i=1}^{N}$, a target forget class set $\mathcal{C}_f$, pruning ratio t, number of repeats R, local epochs E, and the number of sampled parameters k, the goal is to produce a global unlearned model θ_{global}.

Algorithm 1 Proposed Method

Require: Pre-trained federated model θ, Client datasets $\{\mathcal{D}_i\}_{i=1}^{N}$, Forget class $\mathcal{C}_f$, Pruning ratio t, Number of repeats R, Number of epochs E, Number of parameters to be sampled k

Ensure: Unlearned model θ_{global}

for *each client* $i \in \{1, \ldots, N\}$ **do**
  $\mathcal{D}_i^r \leftarrow \{(\mathbf{x_i}, y_i) \in \mathcal{D}_i : y_i \notin \mathcal{C}_f\}$
  $\mathcal{D}_i^f \leftarrow \{(\mathbf{x_i}, y_i) \in \mathcal{D}_i : y_i \in \mathcal{C}_f\}$
  $\theta_i^* \leftarrow \theta_i$
  if $|\mathcal{D}_i^f| > 0$ **then**
    $\phi_i \leftarrow \mathbf{0}_{|\theta_i|}$, $n \leftarrow |\theta_i|$
    for $(\mathbf{x}, y)$ *in* $\mathcal{D}_i^f$, $r = 1$ *to* R **do**
      indices $\leftarrow$ RandomSelect$(n, n-k)$
      $\theta_{\text{pruned}} \leftarrow \theta_i$, θ_{pruned}[indices] $\leftarrow 0$
      $\mathcal{L} \leftarrow \mathcal{L}(y, f_{\theta_{\text{pruned}}}(\mathbf{x}))$, *Compute* $\nabla_{\theta_{pruned}} \mathcal{L}$
      start_idx, end_idx $\leftarrow 0, 0$
      for *param in* θ_{pruned} **do**
        start_idx, end_idx $\leftarrow$ end_idx, end_idx + $|param|$
        if *param.gradient* $\neq$ None **then**
          $w \leftarrow \theta_{pruned}$[start_idx : end_idx]
          ϕ_i[start_idx : end_idx] $\leftarrow \phi_i$[start_idx : end_idx] + $\|$*param.gradient.flatten()* $\cdot\, w\|$
        end if
      end for
    end for
    $P_{\text{critical}} \leftarrow \text{Top}(t)(\phi_i, \theta_i)$, $\theta_i^* \leftarrow \theta_i$, $\theta_i^*[P_{\text{critical}}] \leftarrow 0$
  end if
  if $|\mathcal{D}_i^r| > 0$ **then**
    Fine-tune θ_i^* *on* $\mathcal{D}_i^r$ *for* 5 *epochs*
  end if
end for
$\theta_{\text{global}} \leftarrow \frac{1}{N} \sum_{i=1}^{N} \theta_i^*$
return θ_{global}