Traceable Federated Continual Learning

Supplementary Material

7. Theoretical Analysis

In this section, we provide theoretical analyses to support our framework. Specifically, we respectively analyze the effectiveness of the key modules in TagFed.

Privacy analysis of information transfer. In TagFed, the information transfer is a key step to enable local task learning and central knowledge aggregation. Although TagFed does not upload the raw data directly, it exchanges the feature maps obfuscated with designed noise as the message. Therefore, we attempt to analyze the privacy property of our framework. Concretely, we utilize the following Theorem to measure privacy.

Theorem 7.1. *When an unbiased attacker tries to reconstruct the training data from the trained model* $\hat{\mathbf{x}} = \mathrm{Att}(\mathbf{z}')$*, the average mean-square error (MSE) of* $\hat{x}$ *is bounded by:*

$$\mathbb{E}\left[\|\hat{\mathbf{x}} - \mathbf{x}\|_2^2\right] \geq d/\operatorname{Tr}\left(\mathcal{I}_{z'}(\mathbf{x})\right) \tag{3}$$

where d *is the dimension of* $\boldsymbol{x}$ *and* Tr *is the trace of a matrix.*

Here, $\operatorname{Tr}\left(\mathcal{I}_{z'}(\mathbf{x})\right)/d$ is called *diagonal Fisher information leakage*, or dFIL. dFIL is a strong indicator of privacy when it comes to input reconstruction attacks for split inference. For an unbiased attacker, dFIL indicates the reconstruction feasibility. Details can be found in [10].

In our work, TagFed allows clients to choose different variances of noise according to the client requirement. Specifically, each client can provide a privacy and utility requirement, which will be used to calculate the concrete noise to ensure the trade-off. Here we calculated the noise based on an empirical value (dFIL = 1) and introduced it into our training process to obtain values in the main text.

Effectiveness analysis of group-wise knowledge aggregation. In this part, we mainly explore whether the group-wise scheme contributes to knowledge aggregation. To begin with, we make the following commonly used assumptions.

Assumption 7.2. The objective functions $F_1, F_2, ..., F_N$ in each device are all L-smooth: for all $\mathbf{v}$ and $\mathbf{w}$, $F_k(\mathbf{v}) \leq F_k(\mathbf{w}) + (\mathbf{v}-\mathbf{w})^T \nabla F_k(\mathbf{w}) + \frac{L}{2}\|\mathbf{v}-\mathbf{w}\|_2^2$.

Assumption 7.3. The objective functions $F_1, F_2, ..., F_N$ in each device are all μ-strongly convex: for all $\mathbf{v}$ and $\mathbf{w}$, $F_k(\mathbf{v}) \geq F_k(\mathbf{w}) + (\mathbf{v}-\mathbf{w})^T \nabla F_k(\mathbf{w}) + \frac{\mu}{2}\|\mathbf{v}-\mathbf{w}\|_2^2$.

According to a related work [47], the *weight divergence* among the uploaded information can affect the aggregation performance, which can be formally defined as the following Theorem.

Theorem 7.4. *Suppose Assumption 1 and 2 hold the federation synchronization is conducted every H steps. The weight divergence after the m-th synchronization can be bounded by*

$$\begin{aligned} &\|\boldsymbol{\Delta w_m}\| \leq \\ &\sum_{k=1}^{N} \frac{sample^{(k)}}{\sum_{k=1}^{N} sample^{(k)}} (z^k)^T \|\boldsymbol{\Delta w_{m-1}}\| \\ &+ \eta \sum_{k=1}^{N} \frac{sample^{(k)}}{\sum_{k=1}^{N} sample^{(k)}} \left\|p^k - p^{global}\right\| \sum_{j=1}^{H-1} \\ &\left(z^k\right)^j \max_{i=1} \left\|\nabla_{\boldsymbol{w}} \mathbb{E}_{\boldsymbol{x}|y=i} \log f_i(\boldsymbol{x}, \boldsymbol{w})\right\| \end{aligned} \tag{4}$$

where $z^k = 1 + \eta \sum_{i=1}^{C} p^{(k)}(y=i)\lambda_{\boldsymbol{x}|y=i}$ *and* C *is the number of category.* p *represents the data distribution.*

Based on the inequality, we can observe that the distribution heterogeneity among uploaded information has a large impact on the final divergence degree. Our group-wise scheme enables aggregation on the features of the same task, which significantly mitigates the distribution heterogeneity and benefits the final performance.

8. Baseline Description

We briefly summarize the baselines as follows:

- *PackNet+FedAvg (PF) [32, 34]:* This baseline is a combination of CL methods and FL methods. Concretely, we first pack multiple tasks into a model by iterative pruning and then average the parameters of the client models as federation.
- *GLFC [8]:* GLFC is a state-of-the-art approach focusing on addressing the catastrophic forgetting problem in FCL. The key idea is to implement a class-aware gradient compensation loss and a class-semantic relation distillation loss to balance the forgetting of old classes.
- *FedWeIT [44]:* This method employs weighted inter-client transfer to address the issue of interference from irrelevant clients.
- *FedKNOW [31]:* This method ensures the prevention of catastrophic forgetting and mitigation of negative knowledge transfer by effectively combining signature tasks identified from the past local tasks and other clients' current tasks.

9. Additional Empirical Results

Due to the page limitation of the main text, here we show our additional empirical results to further demonstrate the superiority of TagFed.

Algorithm 1 Pipeline of TagFed

Operation:

1: Distribute a shared model M_{ini} to each client
2: **for** *i=1 to* R **do**
3: Conduct these steps in **Client side** and **Server side** sequentially
4: **end for**
5: Obtain the final improved model M_{final}

Client side:

1: For each client, train M_{ini} with its own task sequence $S_t = \{s_1, s_2, ..., s_n\}$ based on our *traceable task learning*
2: Add noise to the feature maps of the hidden layers, upload them as well as the logits to the server
3: Wait for the predicted values sent back from the server
4: Conduct training based on Eq. 2

Server side:

1: Collect the uploaded feature maps $\{FM_1^i, FM_2^i, ..., FM_N^i\}(i = 1, 2, , , n)$ from n clients
2: Feed them into different server models based on the uploaded client logits
3: Conduct training based on Eq. 1

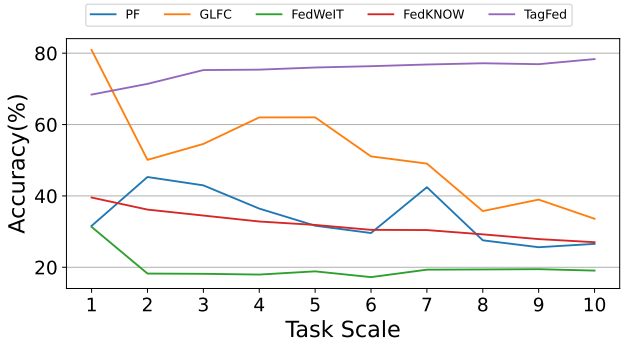

Figure 8. Performance of different task scales on CIFAR-100.

9.1. Performance on the different task scales

In the main text, we only test the performance on a fixed task scale. In this part, we attempt to explore the performance on different task scales. Figure 8 - Figure 10 summarize the results. From these tables, we can draw the following conclusions: (1) *PackNet+FedAvg* cannot achieve good performance no matter how many tasks are in the sequence. This demonstrates that traditional federated aggregation methods that modify the weights are not suitable to our scenario; (2) The performance of typical FCL baselines,

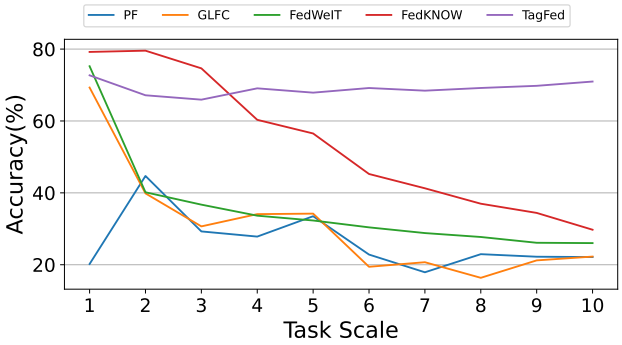

Figure 9. Performance of different task scales on ImageNetSubset.

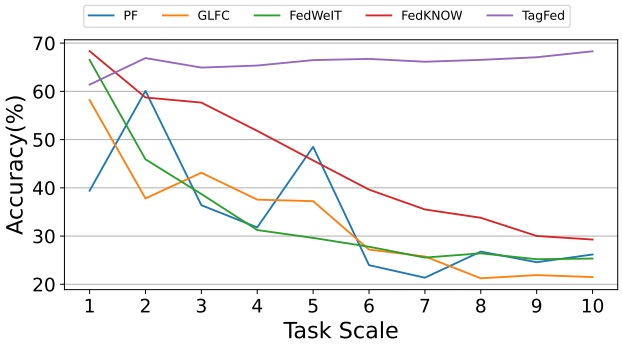

Figure 10. Performance of different task scales on TinyImageNet.

FedWeIT, *FedKNOW*, and *GLFC* will degrade significantly as we increase the number of tasks. This suggests that it cannot cope with massive repetitive tasks; (3) Our TagFed maintains a stable and high accuracy for all task scales. In addition, we find that usually the more tasks we have, the superior performance we can achieve.

9.2. Task-specific performance on other datasets

Besides the performance of TinyImageNet in the main text, here we record the task-specific performance on ImageNetSubset and CIFAR-10, in order to test the generalization of our framework. Figure 11 and Figure 12 show the detailed accuracy performance. We can clearly see that TagFed can achieve consistent improvement on both datasets. Notably, The performance gap is higher in the middle tasks than others. We believe this is because the repeatability interference in these locations may have a higher impact on the baselines, leading to poor performance.

9.3. Ablation study on other datasets

In the main text, we conduct the ablation study on CIFAR-100. Here we record the performance of other datasets, in order to see whether each module in TagFed is still effective. As shown in Table 4 and Table 5, both TTL and GKA play an important role in the final performance.

Table 4. Effect of the proposed modules of different task numbers on ImageNetSubset. *Ours-w/oModule* denotes the performance of our framework without using the Module.

Task Scale	2	4	6	8	10	Avg
Ours-w/oTTL	52.15%	40.00%	34.48%	30.60%	30.72%	37.59%
Ours-w/oGKA	62.85%	64.10%	62.42%	62.83%	64.12%	63.26%
Ours	**67.15%**	**69.08%**	**69.17%**	**69.18%**	**70.97%**	**69.11%**

Table 5. Effect of the proposed modules of different task numbers on TinyImageNet. *Ours-w/oModule* denotes the performance of our framework without using the Module.

Task Scale	2	4	6	8	10	Avg
Ours-w/oTTL	57.10%	42.40%	36.30%	35.60%	36.54%	41.59%
Ours-w/oGKA	59.50%	58.90%	61.07%	60.48%	61.06%	60.20%
Ours	**66.90%**	**65.35%**	**66.73%**	**66.53%**	**68.31%**	**66.76%**

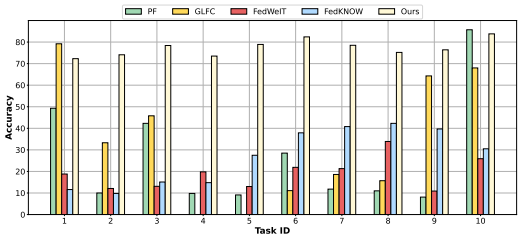

Figure 11. The task-specific performance on the CIFAR-100 dataset.

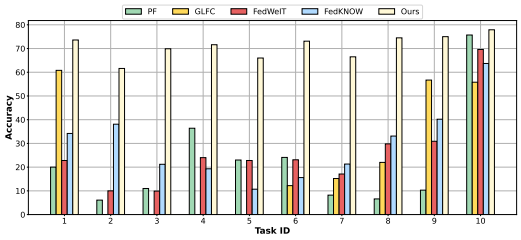

Figure 12. The task-specific performance on the ImageNetSubset dataset.

9.4. Non-iid condition and repeatability degree on other datasets

Figure 13 and Figure 14 show the results on ImageNetSubset and TinyImageNet, focusing on the non-iid condition and the repeatability degree. From these figures, we can find a similar conclusion to the results in the main text: TagFed can handle the non-iid condition well and will not be affected by the task repeatability degree.

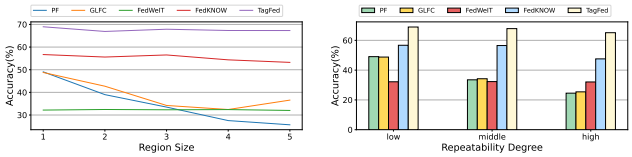

Figure 13. Results on the ImageNetSubset. Left: Performance on the non-iid condition. Right: Effect of the Repeatability Degree.

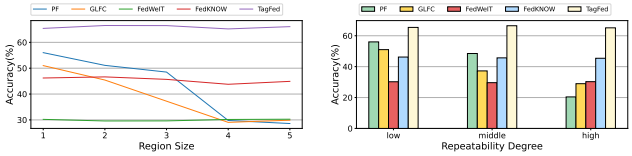

Figure 14. Results on the TinyImageNet. Left: Performance on the non-iid condition. Right: Effect of the Repeatability Degree.

9.5. Noise-introduced performance on other datasets

In this subsection, we evaluate the noise-introduced performance on CIFAR-100 and ImageNetSubset and demonstrate the results in Table 6 and Table 7. From these tables, we can clearly observe that for any dataset, TagFed can exceed other baselines under the privacy-preserving situation.

10. Algorithm

Algorithm 1 shows the whole pipeline of our proposed TagFed. For the client side, we mainly conduct our traceable task learning. For the server side, we focus on the group-wise knowledge aggregation. Note that the operations will be implemented sequentially until training convergence.

Table 6. Results of the noise-introduced situation on CIFAR-100. Here *PF* denotes *PackNet+FedAvg*.

Task scale	**2**	**3**	**5**
PF	45.30%	42.96%	31.66%
GLFC	50.10%	54.57%	62.02%
FedWeIT	18.24%	18.17%	18.86%
FedKNOW	36.17%	34.50%	31.87%
Ours(1/dFIL = 1)	70.50%	73.86%	73.60%
Ours(w/o noise)	71.40%	75.27%	75.99%

Table 7. Results of the noise-introduced situation on ImageNet-Subset. Here *PF* denotes *PackNet+FedAvg*.

Task scale	**2**	**3**	**5**
PF	44.70%	29.27%	33.50%
GLFC	39.85%	30.67%	34.22%
FedWeIT	40.16%	36.72%	32.31%
FedKNOW	79.56%	74.64%	56.52%
Ours(1/dFIL = 1)	66.70%	63.90%	64.10%
Ours(w/o noise)	67.15%	65.93%	67.88%