Gains: Fine-grained Federated Domain Adaptation in Open Set

Zhengyi Zhong[1,3,*], Wenzheng Jiang[1,*], Weidong Bao[1], Ji Wang[1,†], Qi Wang[2], Guanbo Wang[3], Yongheng Deng[3], Ju Ren[3]
[1]Laboratory for Big Data and Decision, National University of Defense Technology
[2]College of Science, National University of Defense Technology
[3]Department of Computer Science and Technology, Tsinghua University

Abstract

Conventional federated learning (FL) assumes a closed world with a fixed total number of clients. In contrast, new clients continuously join the FL process in real-world scenarios, introducing new knowledge. This raises two critical demands: detecting new knowledge, *i.e., knowledge discovery*, and integrating it into the global model, *i.e., knowledge adaptation*. Existing research focuses on coarse-grained knowledge discovery, and often sacrifices source domain performance and adaptation efficiency. To this end, we propose a fine-grained federated domain adaptation approach in open set (***Gains***). *Gains* splits the model into an encoder and a classifier, empirically revealing features extracted by the encoder are sensitive to domain shifts while classifier parameters are sensitive to class increments. Based on this, we develop fine-grained knowledge discovery and contribution-driven aggregation techniques to identify and incorporate new knowledge. Additionally, an anti-forgetting mechanism is designed to preserve source domain performance, ensuring balanced adaptation. Experimental results on multi-domain datasets across three typical data-shift scenarios demonstrate that *Gains* significantly outperforms other baselines in performance for both source-domain and target-domain clients. Code is available at: https://github.com/Zhong-Zhengyi/Gains.

1 Introduction

As a typical distributed intelligent model training paradigm, federated learning (FL) [33, 11, 36, 67] has garnered significant attention from researchers in recent years [15, 32, 39, 35, 45, 12, 66, 59, 19, 48]. Conventional FL is often studied in a setup with a fixed number of clients [33, 30], which limits its applicability in a more realistic scenario when new clients, *i.e.*, *target domain*, are allowed to join the learning process. To scale FL effectively in such scenarios, we have to deal with heterogeneous or evolving client data distributions, *e.g.*, IoT networks, cross-device applications. This prompts researchers to prioritize the study of two critical techniques in the field: **(i)** assessing whether the new client contributes previously unseen knowledge [37], referred to as *knowledge discovery*; **(ii)** devising strategies to integrate it into the global model for improving generalization under the updated domain setting [18, 7], which we call *knowledge adaptation*.

Existing challenges: Though the practical demands and corresponding techniques are well specified, bottlenecks still remain in achieving the deployment purpose (Fig. 1). Regarding *knowledge discovery*, it is rarely investigated in FL, and existing strategies hardly process complicated scenarios. Take the latest work, FOSDA [37], as an example; it facilitates the discovery of new classes, *i.e.*, *class*

[1]* Equal Contribution (zhongzhengyi20@nudt.edu.cn, jiangwenzheng@nudt.edu.cn)
[2]† Corresponding Author (wangji@nudt.edu.cn)

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

increment, in the presence of an open set. However, when faced with domain increment, which is more universal in life, FOSDA encounters the failure of dealing with new domain knowledge. Hence, a more ***fine-grained knowledge discovery*** approach is required to discriminate *class increment* or *domain increment*. As for *knowledge adaptation*, current methods primarily attempt to improve the performance of the newly trained model on target domains. Technically, they often suffer from performance degradation on the source domain while easily overlooking the efficiency of knowledge adaptation [18]. Consequently, we need to introduce a mechanism for ***rapid and balanced knowledge adaptation***, securing seamless integration of new knowledge while consolidating original capabilities.

Proposed solution: To this end, this paper presents a *fine-**G**rained federa**t**ed domai**n** adaptatio**n** method in open **s**et* (***Gains***), which aims at achieving fine-grained knowledge discovery and rapid adaptation without sacrificing the performance on the source domain. Specifically, we discover new knowledge and identify its type (*domain increment* or *class increment*) from the changes in model parameters and extracted features. Then, the federated aggregation process is optimized with the guidance of the quantified parameter and feature's contributions to the target domain, thereby accelerating the integration of new knowledge into the global model. Meanwhile, an ***a**nti-**f**orgetting **m**echanism* (AFM) is designed and used in the training process of the source-domain clients to circumvent source-domain performance degradation, achieving a balance between the target and source domains. To sum up, this work's contribution is three-fold:

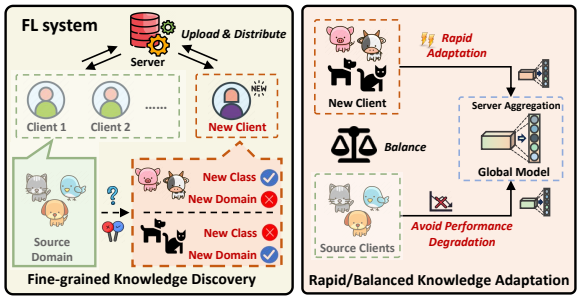


Figure 1: Challenge discription.

- ***Adaptation pipeline.*** We propose a novel training pipeline within FL that supports fine-grained discovery and discrimination of new knowledge from client updates and efficient integration of incremental knowledge into the global model.
- ***Practical solution.*** We present an efficient federated optimization method that enables contribution evaluation of diverse components during knowledge adaptation and suppresses performance decline on the source domain.
- ***Experimental validation.*** We conduct extensive experiments on typical multi-domain datasets under various levels of knowledge shifts. Empirically, *Gains* achieves superior performance on both target and source domain clients over other state-of-the-art methods.

2 Related work

Domain adaptation. Domain adaptation (DA) can be categorized based on the labeling status of the target domain into unsupervised DA, semi-supervised DA, and supervised DA [47]. They can also be divided based on whether the source domain data is involved into source-dependent DA and source-free DA [24]. The distribution shift is a lasting challenge [52], and typical DA approaches include adversarial learning-based methods and alignment-based methods. Adversarial learning-based methods introduce adversarial networks (such as GANs) to align the feature distributions between the source and target domains [8, 20, 4, 64, 13, 49]. Alignment-based methods achieve alignment between the source and target domains by minimizing the differences in feature or data distributions [23, 10, 54, 9]. Common alignment metrics include KL divergence [34], Maximum Mean Discrepancy (MMD) [25], and Wasserstein distance [34]. In addition, other methods such as self-training [38, 46] and meta-learning [21, 44, 50, 51] have also been applied in DA. Unlike most DA work that considers adapting the source model to the new domain and continual learning that considers catastrophic forgetting [29, 55, 56, 68, 65, 63], we focus on solving the problem of better adaptation to the new domain while avoiding performance degradation in the source domain.

Federated domain adaptation. The FDA methods primarily include domain alignment-based, data-based, learning-based, and aggregation optimization-based approaches [31]. Among them, domain alignment consists of feature [53, 7] and gradient alignment [58, 17]. Besides, mixed training approaches are also adopted. For instance, [60] uploads prototypes from different domains to the server for fine-tuning. In data adjustment methods, data augmentation [40, 3, 22] and data generation

[14, 27, 57] are commonly used. Chen et al. [3] generated data with other domain styles on a single client through style transfer between clients. In learning-based approaches, common strategies include adding alignment regularization terms [16], representation learning [1, 61], and transfer learning [2]. For example, Craighero et al. [7] proposed SemiFDA, which trains local feature extractors on clients to align them with the server. In aggregation optimization-based methods, the primary focus is on optimizing aggregation weights [62], gradients [43], and aggregation strategies [41, 6]. For example, FedHEAL [5] removes some less important updates from client models and determines aggregation weights based on the distance between the global model and each local model. AutoFedGP [18] calculates the distance between the source and target domains to derive a new automatic weighting scheme. The aforementioned FDA works are primarily based on the assumption of a closed environment. Currently, there is limited research on FDA in open environments. Even exists, *e.g.*, FOSDA [37], it is only applicable to class-incremental scenarios and does not consider the impact on the source domain.

3 Methodology

This section starts with a motivation example and outlines the pipeline of our developed federated domain adaptation scheme, *Gains*. Subsequently, we elaborate on fine-grained knowledge discrimination and contribution-driven knowledge adaptation as two key components in *Gains*.

Motivation. Without loss of generality, we use the LeNet model and MNIST dataset as an example, considering a scenario where the first three new clients' data is from the source domain, the fourth introduces 1–4 new classes, and the fifth brings new domain data (details are shown in Appendix. A). When new clients participate in training, the variations of the encoder, classifier, and extracted feature are measured by the distance (*e.g.*, Euclidean distance) before and after training in the target domain. From Fig. 2, we have the following findings: ***(i)*** the variation of the encoder (*i.e.*, $Diff^E$) does not show a clear fluctuation trend no matter in class or domain incremental scenarios; ***(ii)*** the changes in the classifier parameters (*i.e.*, $Diff^C$) are more pronounced in the class-incremental scenario; ***(iii)*** while both new classes and domain will bring obvious changes to the feature values (*i.e.*, $Diff^F$), it is more significant in the domain-incremental scenario. Therefore, it is reasonable to consider a combined evaluation of $Diff^C$ and $Diff^F$ to determine whether the new client introduces new knowledge and whether such knowledge is class- or domain-related.

Figure 2: Differences in the encoder (left), classifier (middle), and extracted feature values (right) when the new client carries different types of knowledge.

Framework. Inspired by the above empirical discoveries, we propose a fine-grained federated domain adaptation framework in open set, *Gains* (shown in Fig. 3). Specifically, it consists of two main components: knowledge discovery and knowledge adaptation. In the knowledge discovery stage, the target domain performs local training based on the source model and uploads the updated version back to the server. Then, the server uses public dataset to calculate the variations of $Diff^C$ and $Diff^F$, determining whether the new client introduces new knowledge and further discriminating its type in fine grains. Based on the results of this differentiation, in the knowledge adaptation stage, the contribution of different model components in each source model is calculated. After that, the server executes contribution-driven aggregation to accelerate the speed of target domain adaptation. Considering it may lead to an overemphasis on the target domain, potentially resulting in the performance degradation of the source domain, an anti-forgetting mechanism is included in the local training of the source client to balance the knowledge of the target and source domains.

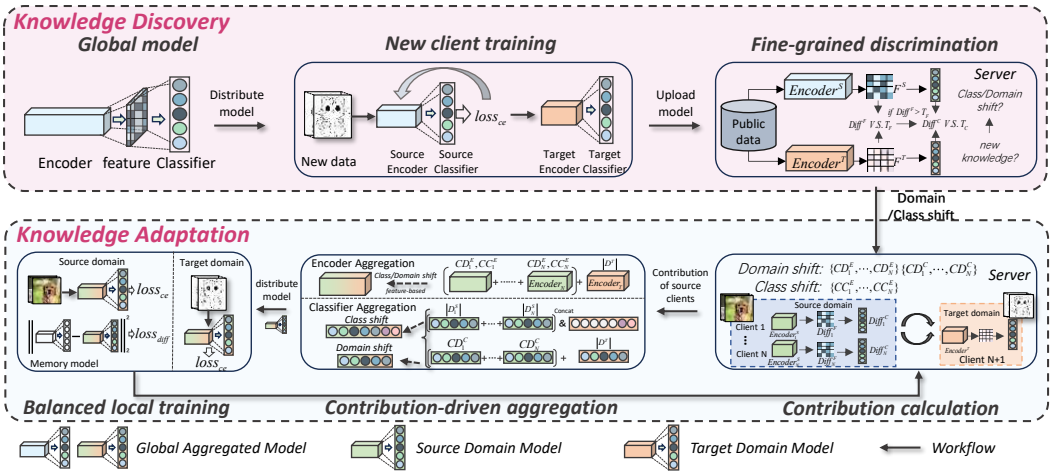


Figure 3: ***Gains* consists of two phases: knowledge discovery (upper) and knowledge adaptation (lower).** The former works on identifying the type of new knowledge, while the latter attempts to achieve rapid integration of new knowledge and strike a balance between new and old knowledge.

Notations. We denote the N client source domain dataset by $\mathcal{D}_n^S = \{(x_j^n, y_j^n)\big|_{j=0}^{|\mathcal{D}_n^S|}\}, n = 1, \cdots N$. The target domain dataset is $\mathcal{D}^T = \{(x_j, y_j)\big|_{j=0}^{|\mathcal{D}^T|}\}$. The server's public data is $\mathcal{D}^P = \{(x^p, y^p)\}$. The original pre-trained source domain model is $\mathcal{W}^S$, comprising an encoder E^S and a classifier C^S. Similarly, we write the target domain trained model as $\mathcal{W}^T$, which includes E^T and C^T. I is the total federated iteration and R is the local training epoch.

3.1 Fine-grained knowledge discrimination

When a new client enters, the server first distributes original source global model $\mathcal{W}^S$ to the target domain for local training Q times. The optimization process is as follows:

$$\mathcal{W}^T(q+1) = \mathcal{W}^T(q) - \eta \nabla \mathcal{L}(\mathcal{W}^T(q), \mathcal{D}^T), q = 0, \cdots, Q-1, \tag{1}$$

where η is learning rate, $\mathcal{L}(\mathcal{W}^T(q), \mathcal{D}^T)$ is the loss in the q-th local training epoch, and the final target model is $\mathcal{W}^T = \mathcal{W}^T(\hat{I})$. Once the target client finished the local training, $\mathcal{W}^T$ will be uploaded to the server. Then, $\mathcal{D}^P$ is input into E^S and E^T to obtain the feature values $F^S = E^S(x^p)$ and $F^T = E^T(x^p)$, respectively. According to the finding (iii) in the motivation, we first judge whether the new client brings new knowledge based on the variation of $Diff^F$. If $Diff^F$ is big enough, we believe that the data distribution from the target domain is different from that of the source domain, which means new knowledge is coming. Furthermore, according to finding (ii), we can determine whether this new knowledge is related to a new class by calculating $Diff^C$. Specifically, we use the Manhattan distance and the Euclidean distance to calculate $Diff^F$ and $Diff^C$, respectively.

We set T_F and T_C as thresholds for discovering new knowledge and determining the type of new knowledge, respectively. When $Diff^F > T_F$, we conclude that the target domain has introduced new knowledge. If $Diff^C > T_C$ simultaneously, it indicates that the new knowledge corresponds to a new class; otherwise, it is considered as new domain knowledge.

3.2 Contribution-driven knowledge adaptation

In the knowledge adaptation phase, two key issues need to be addressed: first, the rapid knowledge adaptation to the target domain; and second, the balance between new and old knowledge. To achieve the former one, we propose the contribution-driven aggregation strategy, which means assigning greater weights to clients with higher contributions. As for the latter balance problem, an anti-forgetting mechanism is presented.

Domain-incremental contribution-driven aggregation. In this paper, we believe that the more similar the source domain client is to the target domain, the more beneficial it is for the fusion of new

knowledge. Then the greater the contribution is. In the domain-incremental scenario, the encoder and the classifier adopt the feature-based and parameter-based contribution calculation methods, respectively. In the feature-based calculation, the encoder contribution $\mathcal{CD}_n^E(i)$ of the n-th source client to the target domain during the i-th iteration is calculated as follows:

$$\mathcal{CD}_n^E(i) = \frac{1}{(1+Diff_n^F(i)) \times \sum_{n=1}^{N} \left(1/(1+Diff_n^F(i))\right)} \times \frac{\sum_{n=1}^{N} \left|\mathcal{D}_n^S\right|}{\left|\mathcal{D}^T\right| + \sum_{n=1}^{N} \left|\mathcal{D}_n^S\right|}, \tag{2}$$

where $Diff_n^F(i)$ is measured by the distance between $F^T(i) = E^T(i)(x^p)$ and $F_n^S(i) = E_n^S(i)(x^p)$. $E^T(i)$ is the encoder uploaded by the target domain in i-th iteration, while $E_n^S(i)$ is from the n-th source client. Similarly, in parameter-based aggregation, the classifier contribution of n-th source client $\mathcal{CD}_n^C(i)$ is calculated as follows:

$$\mathcal{CD}_n^C(i) = \frac{1}{(1+Diff_n^C(i)) \times \sum_{n=1}^{N} \left(1/(1+Diff_n^C(i))\right)} \times \frac{\sum_{n=1}^{N} \left|\mathcal{D}_n^S\right|}{\left|\mathcal{D}^T\right| + \sum_{n=1}^{N} \left|\mathcal{D}_n^S\right|}, \tag{3}$$

where $Diff_n^C(i)$ is measured by the distance between $C^T(i)$ and $C_n^S(i)$. Ultimately, we obtain the contribution lists $\{\mathcal{CD}_1^E(i), \mathcal{CD}_2^E(i), \cdots, \mathcal{CD}_N^E(i)\}$ and $\{\mathcal{CD}_1^C(i), \mathcal{CD}_2^C(i), \cdots, \mathcal{CD}_N^C(i)\}$ of the source encoder and source classifier during the i-th iteration in the domain-incremental scenario. The aggregation processes are as follows:

$$E(i) = \sum_{n=1}^{N} \mathcal{CD}_n^E(i) \times E_n^S(i) + \frac{\left|\mathcal{D}^T\right|}{\left|\mathcal{D}^T\right| + \sum_{n=1}^{N} \left|\mathcal{D}_n^S\right|} \times E^T(i), \tag{4}$$

$$C(i) = \sum_{n=1}^{N} \mathcal{CD}_n^C(i) \times C_n^S(i) + \frac{\left|\mathcal{D}^T\right|}{\left|\mathcal{D}^T\right| + \sum_{n=1}^{N} \left|\mathcal{D}_n^S\right|} \times C^T(i). \tag{5}$$

The aforementioned aggregation process facilitate the rapid adaptation of knowledge by dynamically improving the contribution-based weights in each iteration.

Class-incremental contribution-driven aggregation. Similarly, in the class-incremental scenario, for the encoder aggregation process, we adopt the same feature-based method to calculate the contribution. The Encoder contribution of the n-th source client in i-th iteration is $\mathcal{CC}_n^E(i)$. Then, the contribution list of the encoder in class-incremental scenarios is obtained $\{\mathcal{CC}_1^E(i), \mathcal{CC}_2^E(i), \cdots, \mathcal{CC}_N^E(i)\}$. The aggregation process is as follows:

$$E(i) = \sum_{n=1}^{N} \mathcal{CC}_n^E(i) \times E_n^S(i) + \frac{\left|\mathcal{D}^T\right|}{\left|\mathcal{D}^T\right| + \sum_{n=1}^{N} \left|\mathcal{D}_n^S\right|} \times E^T(i). \tag{6}$$

The aggregation of the classifier employs a channel-wise supplementation method. First, the classifiers from the source domain are aggregated based on the amount of data from each client, resulting in $C^S(i) = \sum_{n=1}^{\mathcal{N}} \frac{\left|D_n^S\right|}{\sum_{n=1}^{\mathcal{N}} |D_n^S|} \times C_n^S(i)$. Suppose there are K^S classes in the source domain and K^T new classes added in the target domain. Consequently, the classifier has $K^S + K^T$ channels. The parameters of the classifier aggregated from the source domain are denoted as $C^S(i) = [Channel_1^S, \cdots, Channel_{K^S}^S, Channel_{K^S+1}^S, \cdots, Channel_{K^S+K^T}^S]$, and the parameters of the target domain classifier are denoted as $C^T(i) = [Channel_1^T, \cdots, Channel_{K^s}^T, Channel_{K^s+1}^T, \cdots, Channel_{K^s+K^T}^T]$. In the final aggregated classifier, the channels corresponding to the source domain classes directly adopt the parameters from $C^S(i)$, while the channels for the target domain classes retain the parameters from $C^T(i)$. That is,

$$C(i) = \Big[\underbrace{Channel_1^S, \cdots, Channel_{K^S}^S}_{SourceDomain}, \underbrace{Channel_{K^S+1}^T, \cdots, Channel_{K^S+K^T}^T}_{TargetDomain} \Big]. \tag{7}$$

A theoretical convergence analysis of *Gains* is provided in the Appendix. F.

Anti-forgetting mechanism. The above aggregation may lead to a bias towards the target domain knowledge in the aggregated model, potentially causing a decline in performance on the source domain tasks. To mitigate this, we introduce an anti-forgetting mechanism for the source domain clients during each round of local training. Specifically, we control the distance between the current model $\mathcal{W}_n^S(i,r)$ and the memory model $\mathcal{W}_n^S(0,0)$ in the local training to prevent the local model

from excessively deviating from the historical model. Here, $\mathcal{W}_n^S(0,0)$ represents the local model in the source domain before the new client enters. $\mathcal{W}_n^S(i,r)$ is the n-th client model during the i-th global iteration and r-th local training epoch. The local loss function for the source clients is defined as follows:

$$\mathcal{L}(\mathcal{W}_n^S(i,r),\mathcal{D}_n^S) = -\frac{1}{|\mathcal{D}_{\mathrm{n}}^S|}\sum_{j=1}^{|\mathcal{D}_{\mathrm{n}}^S|}\sum_{c=1}^{K^S+K^T} y_{j,c}^n \log(\hat{y}_{j,c}^n) + \lambda \left\| \mathcal{W}_n^S(i,r) - \mathcal{W}_n^S(0,0) \right\|_2^2, \tag{8}$$

where λ is a balance coefficient. Through the above training process, we can achieve rapid federated domain adaptation while avoiding forgetting the source domain knowledge, thereby maintaining a balance between new and old knowledge.

3.3 Algorithm

As shown in Alg. 1, when a new client joins, the server distributes the source global model $\mathcal{W}^S$ to the target domain for local training, getting $\mathcal{W}^T$. Subsequently, the server decomposes $\mathcal{W}^S$ and $\mathcal{W}^T$ into an encoder and a classifier and derives the feature using the public dataset. Based on the differences in the feature extracted by E^S and E^T, as well as the parameter differences between C^S and C^T, the algorithm discriminates the type of new knowledge and confirms its type. Then, we calculate the contributions of the source clients to the target client in both encoders and classifiers. According to knowledge types and model components, specific aggregation strategies are used to accelerate knowledge adaptation. Furthermore, to prevent the aggregation process from overly favouring the target client, the anti-forgetting mechanism is incorporated into the local update process of the source clients. After all clients complete local training, they upload their models to the server for aggregation based on their contributions. This process repeats until convergence. If no new knowledge is detected at the outset, the original model is deployed directly on the newly joined clients for inference without any further training.

4 Experimental verification

This section first explores the threshold for knowledge discovery and validates *Gains* under three data shift scenarios. Then, to verify its scalability, we conduct experiments in more target domains and a sequential FDA scenario. Finally, ablation studies reveal the necessity of the AFM component.

4.1 Experiment setting

Our experiments are conducted on a single NVIDIA RTX 4090 GPU. We construct a federated learning framework that includes one server and 50 clients for validation. Following [7], we evaluate *Gains* in three scenarios of target data shifts: mild, medium, and strong shifts. Specifically, under the mild shift scenario, clients in both the source and target domains are drawn from the same sub-dataset but contain different classes. Under the medium shift scenario, all clients in the source domain are from one sub-dataset, while clients in the target domain are from another sub-dataset. Under the strong shift scenario, different clients in the source domain contain different sub-datasets, and clients in the target domain are from other sub-datasets. The main results are shown in Table 1.

Dataset. The datasets include the DigitFive (*i.e.*, DF) for the digit classification and the Amazon Review (*i.e.*, AR) for the product review. DF comprises five sub-datasets: MNIST, MNIST-M, SVHN, USPS, and SynthDigits. Each one contains 10 classes of digits from 0 to 9. The AR dataset records user reviews of products on the Amazon website and includes four subdatasets: Books, DVDs, Electronics, and Kitchen housewares. Each sub-dataset contains two classes.

Baselines. We include two categories of baselines. The first is to address the domain adaptation problem, including FOSDA [37], SemiFDA [7], AutoFedGP [18] and FedHEAL [5]. The second focuses on the heterogeneous problem, including FedAVG [33], FedProx [26], and FedProto [42].

Evaluations. (i) the accuracy of the target client (*T-Acc*); (ii) the average accuracy of the source clients (*S-Acc*); (iii) the global accuracy (*G-Acc*).

4.2 New knowledge discovery

The key to discovering new knowledge lies in setting an appropriate threshold, *i.e.*, T_F and T_C. In Fig. 2, we observe that when new clients introduce unseen class or domain knowledge, the $Diff^F$

Algorithm 1: *Gains*

Input: Number of source clients N; original source global model $\mathcal{W}^S$ and client model $\{\mathcal{W}_1^S(0,0), \mathcal{W}_2^S(0,0), \cdots, \mathcal{W}_N^S(0,0)\}$; number of iteration I; number of local training R; public data $\mathcal{D}^P = \{(x^p, y^p)\}$

Output: Global model $\mathcal{W}$

```
Distribute original source model W^S to target client
W^T ← Target client performs local updating based on W^S
Target client Uploads W^T to the server
//Knowledge Discovery
Split the W^S into encoder E^S and classifier C^S, split the W^T into E^T and C^T
F^S ← E^S(x^p), F^T ← E^T(x^p)
Calculating Diff^C and Diff^F
if Diff^F > T_F then
    Target client brings new knowledge
    if Diff^C > T_C then
        Class Increment=True
    else
        Domain Increment=True
    //Knowledge Adaptation
    for iteration i = 0, ··· , I do
        if Domain Increment=True then
            Calculating encoder contributions {CD_1^E, CD_2^E, ··· , CD_N^E} based on Eq. (2)
            Calculating classifier contributions {CD_1^C, CD_2^C, ··· , CD_N^C} based on Eq. (3)
            Aggregating all clients' parameters using Eq.(4) and Eq. (5)
        if Class Increment=True then
            Calculating encoder contributions {CC_1^E, CC_2^E, ··· , CC_N^E} based on Eq. (2)
            Aggregating all clients' parameters using Eq.(6) and Eq. (7)
        Server distributes the aggregated model to all clients
        for client n = 1, ··· , N do
            Locally update model R rounds using Eq.(8)
            Upload W_n^S(i, R) to the server
        Target client locally update model R rounds and upload to the server
else
    Apply the original model to newly joined clients for inference tasks without training
```

increases significantly, with all values exceeding 1000. Furthermore, in the case of class increment, the $Diff^C$ undergoes substantial changes. Even when only a new class is added to the target client, the parameter change of the classifier is still greater than 0.25, which is significantly higher than that of domain increment clients. Therefore, for the DigitFive dataset, we consider setting the threshold T_F to 1000 and the threshold T_C to 0.25. For the Amazon Review dataset, given the limited number of classes, we only conduct validation in the domain increment scenario. Taking DVDs as the source domain data and Kitchen Hardware as the target domain data as an example, when the new client does not introduce new data, the $Diff^F$ fluctuates between 50 and 150. However, when the new nodes bring in new domain data, the change value increases to 534.76. Therefore, we consider setting the threshold T_F for the Amazon Review dataset to 400.

4.3 Knowledge adaptation

Mild data shift. Under the mild data shift scenario, we experiment using the MNIST data from the DigitFive dataset, assuming that the target domain contains data labeled as {1, 5}, while the source domain consists of {0, 2, 3, 4, 6, 7, 8, 9}. *Gains* achieves 99.34% new client accuracy (*T-Acc*) while maintaining 93.21% source client accuracy (*S-Acc*) and 94.44% global accuracy (*G-Acc*). This demonstrates *Gains*'s effectiveness in class-incremental scenarios. The feature-based contribution calculation and channel-wise classifier aggregation allow seamless integration of new

Table 1: Main results. The bold font represents the optimal result.

Scenario	Metric	Federated Domain Adaptation					Heter-FL		
		Ours	FOSDA [TNNLS'24]	SemiFDA [ICDM'24]	AutoFedGP [ICLR'24]	FedHEAL [CVPR'24]	FedAVG [AISTATS'17]	FedProx [MLSys'20]	FedProto [AAAI'22]
DigitFive									
Mild	*T-Acc*	**99.34**	0.00	0.00	68.11	22.60	55.73	72.35	77.61
	S-Acc	93.21	12.72	13.53	0.00	99.29	0.36	**99.53**	0.16
	G-Acc	**94.44**	10.18	10.83	13.62	83.95	11.44	94.09	62.12
Medium	*T-Acc*	**97.91**	11.29	7.91	9.78	93.68	90.79	94.88	45.66
	S-Acc	**90.09**	19.46	19.44	6.22	88.71	76.20	86.50	33.56
	G-Acc	**91.65**	17.82	17.14	6.93	89.70	79.12	88.18	43.23
Strong	*T-Acc*	**98.98**	11.29	31.14	10.37	96.98	85.80	85.29	31.28
	S-Acc	**93.18**	13.60	14.21	11.60	83.32	43.90	43.32	62.23
	G-Acc	**94.34**	13.13	17.60	11.35	86.05	52.28	51.72	37.47
Amazon Review									
Medium	*T-Acc*	**84.60**	49.55	50.45	50.50	50.56	66.74	74.55	50.11
	S-Acc	**82.81**	49.55	49.33	50.50	50.56	67.19	74.44	50.11
	G-Acc	**83.09**	49.82	49.82	50.58	50.48	67.38	74.12	50.01
Strong	*T-Acc*	**80.54**	50.48	55.41	50.03	83.34	51.20	53.73	50.10
	S-Acc	**84.95**	50.27	59.25	50.02	86.54	51.36	53.95	50.11
	G-Acc	**83.85**	50.33	58.29	50.02	85.74	51.32	53.89	50.10

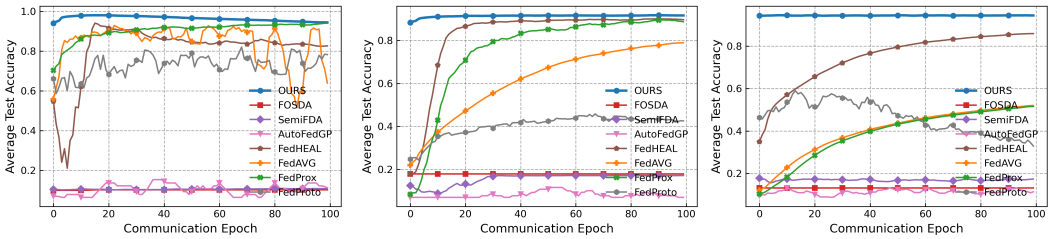
classes. Meanwhile, the anti-forgetting mechanism further ensures stable source performance by constraining parameter drift during local updates.

Medium data shift. For a more complex scenario, medium data shift, we conduct validation using DigitFive and Amazon Review datasets. As for DigitFive, the source domain data is derived from SVHN, while the target domain's data is from MNIST. For Amazon Review, the corresponding data are DVDs and Books, respectively. In Table 1, *Gains* achieves 97.91% *T-Acc* and 90.09% *S-Acc* in DigitFive, outperforming all baselines. Notably, FedHEAL achieves competitive T-Acc (93.68%) but exhibits unstable source performance (*S-Acc*=88.71%). A similar phenomenon can be observed in the Amazon Review dataset. This validates the effect of *Gains* in domain-incremental scenarios: leveraging feature gap in the encoder and parameter variation in the classifier to dynamically prioritize source clients with higher contributions.

Strong data shift. In extreme cases, each client in source and target domains may come from different domains, which refer to as strong data shift. For DigitFive, we assume that the target domain client data is from MNIST, and the source domain consists of four clients, each holding MNIST-M, SVHN, USPS, and SynthDigits datasets, respectively. For the Amazon Review, the target domain is Books, and source-domain clients are from the DVDs, Electronics, and Kitchen Housewares datasets. In Table 1, *Gains* achieves 98.98% *T-Acc* and 93.18% *S-Acc* in DigitFive, demonstrating robustness to extreme heterogeneity. Similarly, *Gains* achieves 80.54% *T-Acc*, 84.95% *S-Acc* and 83.85% *G-Acc* in Amonzon Review, showing significant advantages over other methods.

Adaptation speed. The above content illustrates that the *Gains* method can improve learning performance in the source and target domains. To further demonstrate its advantage in domain adaptation speed, we visualize the training process of DigitFive under different methods in Fig. 4, where the vertical axis represents the global accuracy and the horizontal axis represents the number of epochs. It can be seen that our method not only achieves the highest accuracy but also has the fastest convergence speed, enabling it to reach better results more quickly. This is because we optimize the aggregation process of the encoder and classifier based on their respective contributions, which allows for more efficient adaptation of new knowledge on the basis of the source domain model. The convergence process diagram for the Amazon Review dataset is provided in the Appendix. C.

Generalization verification. In Table 1, we only validate some cases under the mild (Mi), medium (Me), and strong (St) data shift scenarios. To further verify the generalization ability of *Gains*, we change the source/target domain datasets and test the DF and AR datasets under above three scenarios, and the results are shown in Table 2. Here, {1,5} indicates that the target domain data labels are 1 and 5. "SV-MT" represents the scenario where the source domain is SVHN and the target domain is MNIST under the Me data shift. MTM, BK, DD, and KC are the abbreviations for the MNISTM, Book, DVDs, and Kitchen datasets, respectively. As shown in Table 2, under the same multi-domain dataset, our method still maintains a comparable level when the target domain is different, indicating strong generalization capabilities of *Gains*. Please refer to Appendix. D for more validations on the generalization.



(a) mild data shift. (b) medium data shift. (c) strong data shift.

Figure 4: Training process of DigitFive under different data shift scenarios.

Sequential FDA. In the previous experiments, we primarily focus on the scenario where only a single new client joins in FL. In this part, we take DigitFive as an example to verify the performance when continuous new clients arrival (*i.e.*, sequential FDA). In the class-incremental scenario, we assume that the source domain classes are {0,1,2,3}, and subsequently, three clients carrying {4,5}, {6,7}, and {8,9} join the FL process. In the domain-incremental scenario, the source domain is the SVHN, and the target domains include MNIST, MNIST-M, and SynthDigits, respectively. Table 3 shows the results after incorporating different target domain data into the training process. It can be observed that *Gains* still exhibits strong robustness in sequential FDA.

Ablation study. This part examine the role of the Anti-forgetting Mechanism in *Gains* using DigitFive dataset. As shown in Table 4, the absence of the AFM indeed causes significant performance degradation for the source clients across all scenarios, illuminating the effectiveness of this component. Moreover, the performance drop is most pronounced in the class-incremental scenario (*i.e.*, mild data shift). This is consistent with our observations in the motivation, as the changes to the model parameters are most significant during class increment. Without AFM, in the mild data shift scenario, the client model deviates most severely from its original parameters, resulting in the greatest performance decline.

Table 2: Generalization verification.

		{1,5}	{6,9}	{0,1,5}
Mi-DF	TA	99.34	94.42	99.59
	SA	93.21	96.03	87.16
	GA	94.44	95.71	89.64
		SV-MT	MT-MTM	SYN-MTM
Me-DF	TA	97.91	94.46	90.49
	SA	90.09	99.56	98.57
	GA	91.65	98.54	96.95
		MT	SV	MTM
St-DF	TA	98.98	91.67	93.94
	SA	93.18	97.58	96.20
	GA	94.34	96.40	95.75
		DD-BK	BK-DD	ET-KC
Me-AR	TA	84.60	82.01	86.59
	SA	82.81	86.85	89.93
	GA	83.09	85.88	89.26
		BK	DD	KC
St-AR	TA	80.54	78.22	85.38
	SA	84.95	88.90	87.73
	GA	83.85	86.23	87.14

Table 3: The performance of sequential FDA.

		{4,5}	{6,7}	{8,9}
Mi	TA	99.88	91.35	96.89
	SA	93.53	99.43	99.35
	GA	96.82	98.08	99.00
		MNIST	MNISTM	SYN
Me	TA	95.27	83.53	93.53
	SA	87.91	90.05	89.66
	GA	89.38	88.96	90.21

Table 4: Ablation study of AFM.

	Mild	Medium	Strong
AFM	99.05	90.09	94.77
w/o AFM	9.24	84.35	92.46

4.4 Computing complexity analysis.

Compared with traditional federated learning, *Gains* mainly increases the computational load during the server-side contribution calculation. Its complexity is $O(N \cdot P \cdot d)$, where N is the number of source domain clients, P is the size of the public dataset, and d is the number of model parameters. Inevitably, extra computational costs occur during the above process. However, by calculating the weights based on contribution, more efficient aggregation can be achieved, thereby significantly reducing the number of federated iterations and reducing the overall training time. Taking the DigitFive dataset in the mild shift scenario as an example, the consumed computing resources and the number of iterations are as shown in Table 5.

4.5 Sensitivity analysis of the thresholds

Although the thresholds are manually set, the model exhibits strong robustness to threshold variations. As can be seen from Figure 2, the changes in $Diff^F$ and $Diff^C$ are very significant, which means

that the thresholds can take values over a wide range, with $Diff^F$ ranging from 700 to 3400 and $Diff^C$ from 0.05 to 0.27. We also conduct experiment tests on various thresholds using the mild shift scenario in the DigitFive dataset as an example. Assuming the source domain data is MNIST-M and the target domain data is MNIST, with $T_F \in 800, 1000, 1200$ and $T_C \in 0.20, 0.25, 0.27$. The experimental results obtained are shown in Table and Table...From the above two tables, it can be seen that the model performance remains stable when parameters fluctuate within reasonable ranges (performance variation < 1%).

5 Conclusion and discussion

Conclusion. This paper presents a novel fine-grained federated domain adaptation framework in open set (*Gains*) that addresses the challenges of fine-grained knowledge discovery and rapid and balanced knowledge adaptation. By splitting the model into an encoder and a classifier, *Gains* effectively identifies the type new knowledge based on the variations in extracted features and model parameters, enabling more precise knowledge adaptation. The proposed contribution-driven aggregation strategy accelerates the integration of new knowledge into the global model, while the anti-forgetting mechanism ensures the preservation of source domain performance. Extensive experiments on multiple datasets demonstrate that *Gains* can achieve balanced adaptation and rapid convergence under various data shift scenarios.

Table 5: Convergence Comparison of Different Methods.

Method	Converge Round	Time
Gains	**5**	**807.45**
FedHEAL	40	1368.4
FedAVG	20	1977.20
FedProx	40	6880.80
FedProto	32	9519.68

Table 6: Accuracy Results for Different T_F Values.

T_F	T-Acc	S-Acc	G-Acc
800	99.62	92.01	93.15
1000	99.34	93.21	94.44
1200	99.24	93.06	93.91

Table 7: Accuracy values for different T_C settings.

T_C	T-Acc	S-Acc	G-Acc
0.20	99.75	92.34	93.29
0.25	99.34	93.21	94.44
0.27	99.86	92.71	93.01

Discussion. This paper proposes a fine-grained domain adaptation framework in FL. Although the pipeline achieves satisfactory results, some limitations still exist. First, in the knowledge discovery phase, it still relies on manually set thresholds, and achieving automatic knowledge discovery remains a significant challenge. Second, in the knowledge identification phase, we consider domain increment and class increment. However, for more complex scenarios, such as task increment or scenarios involving both class increment and domain increment, further exploration is needed. In addition, it's worth noting *Gains* is significantly different from traditional federated continual learning. First, the settings are different. FCL primarily focuses on scenarios where existing clients encounter new data, while FDA focuses on cases where new clients join and bring unseen data. Second, the objectives are different. FCL primarily addresses the catastrophic forgetting caused by new data in existing clients. In contrast, *Gains* focuses on rapidly adapting to the new domain while preventing performance degradation of the source domain clients, achieving efficient and balanced domain adaptation.

Acknowledgment. This work is supported by National Natural Science Foundation of China under Grant 62273352 and 72501290.

References

[1] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[2] Junbin Chen, Jipu Li, Ruyi Huang, Ke Yue, Zhuyun Chen, and Weihua Li. Federated transfer learning for bearing fault diagnosis with discrepancy-based weighted federated averaging. *IEEE Transactions on Instrumentation and Measurement*, 71:1–11, 2022.

[3] Junming Chen, Meirui Jiang, Qi Dou, and Qifeng Chen. Federated domain generalization for image recognition via cross-client style transfer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 361–370, 2023.

[4] Lin Chen, Huaian Chen, Zhixiang Wei, Xin Jin, Xiao Tan, Yi Jin, and Enhong Chen. Reusing the task-specific classifier as a discriminator: Discriminator-free adversarial domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7181–7190, 2022.

[5] Yuhang Chen, Wenke Huang, and Mang Ye. Fair federated learning under domain skew with local consistency and domain diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12077–12086, 2024.

[6] Seokhyun Chung and Raed Al Kontar. Federated condition monitoring signal prediction with improved generalization. *IEEE Transactions on Reliability*, 73(1):438–450, 2024.

[7] Michele Craighero, Giorgio Rossi, Beatrice Rossi, Diego Carrera, Diego Stucchi, Pasqualina Fragneto, and Giacomo Boracchi. Semifda: Domain adaptation in semi-supervised federated learning. In *2024 IEEE International Conference on Data Mining (ICDM)*, pages 687–692. IEEE, 2024.

[8] Jun Dan, Mushui Liu, Chunfeng Xie, Jiawang Yu, Haoran Xie, Ruokun Li, and Shunjie Dong. Similar norm more transferable: Rethinking feature norms discrepancy in adversarial domain adaptation. *Knowledge-Based Systems*, 296:111908, 2024.

[9] Jun Dan, Weiming Liu, Mushui Liu, Chunfeng Xie, Shunjie Dong, Guofang Ma, Yanchao Tan, and Jiazheng Xing. Hogda: Boosting semi-supervised graph domain adaptation via high-order structure-guided adaptive feature alignment. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 11109–11118, 2024.

[10] Jun Dan, Weiming Liu, Xie Xie, Hua Yu, Shunjie Dong, and Yanchao Tan. Tfgda: Exploring topology and feature alignment in semi-supervised graph domain adaptation through robust clustering. *Advances in Neural Information Processing Systems*, 37:50230–50255, 2024.

[11] Tao Fan, Hanlin Gu, Xuemei Cao, Chee Seng Chan, Qian Chen, Yiqiang Chen, Yihui Feng, Yang Gu, Jiaxiang Geng, Bing Luo, et al. Ten challenging problems in federated foundation models. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

[12] Lele Fu, Sheng Huang, Yuecheng Li, Chuan Chen, Chuanfu Zhang, and Zibin Zheng. Learn the global prompt in the low-rank tensor space for heterogeneous federated learning. *Neural Networks*, 187:107319, 2025.

[13] Xiang Gu, Xi Yu, Yan Yang, Jian Sun, and Zongben Xu. Adversarial reweighting with α-power maximization for domain adaptation. *International Journal of Computer Vision*, 132(10):4768–4791, 2024.

[14] Wei Guo, Yijin Wang, Xin Chen, and Pingyu Jiang. Federated transfer learning for auxiliary classifier generative adversarial networks: framework and industrial application. *Journal of intelligent manufacturing*, 35(4):1439–1454, 2024.

[15] Wei Huang, Dexian Wang, Xiaocao Ouyang, Jihong Wan, Jia Liu, and Tianrui Li. Multimodal federated learning: Concept, methods, applications and future directions. *Information Fusion*, 112:102576, 2024.

[16] Wenke Huang, Mang Ye, Zekun Shi, He Li, and Bo Du. Rethinking federated learning with domain shift: A prototype view. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16312–16322. IEEE, 2023.

[17] Enyi Jiang, Yibo Jacky Zhang, and Sanmi Koyejo. Principled federated domain adaptation: Gradient projection and auto-weighting. *arXiv preprint arXiv:2302.05049*, 2023.

[18] Enyi Jiang, Yibo Jacky Zhang, and Sanmi Koyejo. Principled federated domain adaptation: Gradient projection and auto-weighting. In *ICLR*, 2024.

[19] Wenzheng Jiang, Ji Wang, Xiongtao Zhang, Weidong Bao, Cheston Tan, and Flint Xiaofeng Fan. Fedhpd: Heterogeneous federated reinforcement learning via policy distillation. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '25, page 2568–2570, Detroit, MI, USA, 2025.

[20] Mengmeng Jing, Jingjing Li, Ke Lu, Lei Zhu, and Heng Tao Shen. Visually source-free domain adaptation via adversarial style matching. *IEEE Transactions on Image Processing*, 33:1032–1044, 2024.

[21] Arsham Gholamzadeh Khoee, Yinan Yu, and Robert Feldt. Domain generalization through meta-learning: A survey. *Artificial Intelligence Review*, 57(10):285, 2024.

[22] Dominik Lewy, Jacek Mańdziuk, Maria Ganzha, and Marcin Paprzycki. Statmix: Data augmentation method that relies on image statistics in federated learning. In *International Conference on Neural Information Processing*, pages 574–585. Springer, 2022.

[23] Jianchen Li, Jiqing Han, Fan Qian, Tieran Zheng, Yongjun He, and Guibin Zheng. Distance metric-based open-set domain adaptation for speaker verification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[24] Jingjing Li, Zhiqi Yu, Zhekai Du, Lei Zhu, and Heng Tao Shen. A comprehensive survey on source-free domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[25] Jingjing Li, Jidong Zhao, and Ke Lu. Joint feature selection and structure preservation for domain adaptation. In *IjCAI*, pages 1697–1703, 2016.

[26] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

[27] Wei Li, Jinlin Chen, Zhenyu Wang, Zhidong Shen, Chao Ma, and Xiaohui Cui. Ifl-gan: Improved federated learning generative adversarial network with maximum mean discrepancy model aggregation. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12):10502–10515, 2022.

[28] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

[29] Yichen Li, Qunwei Li, Haozhao Wang, Ruixuan Li, Wenliang Zhong, and Guannan Zhang. Towards efficient replay in federated incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12820–12829, 2024.

[30] Yichen Li, Wenchao Xu, Haozhao Wang, Yining Qi, Jingcai Guo, and Ruixuan Li. Personalized federated domain-incremental learning based on adaptive knowledge matching. In *European Conference on Computer Vision*, pages 127–144. Springer, 2024.

[31] Ying Li, Xingwei Wang, Rongfei Zeng, Praveen Kumar Donta, Ilir Murturi, Min Huang, and Schahram Dustdar. Federated domain generalization: A survey. *arXiv preprint arXiv:2306.01334*, 2023.

[32] Bingyan Liu, Nuoyan Lv, Yuanchun Guo, and Yawen Li. Recent advances on federated learning: A systematic survey. *Neurocomputing*, page 128019, 2024.

[33] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[34] Yingwei Pan, Ting Yao, Yehao Li, Yu Wang, Chong-Wah Ngo, and Tao Mei. Transferrable prototypical networks for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2239–2247, 2019.

[35] Zhuang Qi, Lei Meng, Zhaochuan Li, Han Hu, and Xiangxu Meng. Cross-silo feature space alignment for federated learning on clients with imbalanced data. In *The 39th Annual AAAI Conference on Artificial Intelligence (AAAI-25)*, pages 19986–19994, 2025.

[36] Zhuang Qi, Sijin Zhou, Lei Meng, Han Hu, Han Yu, and Xiangxu Meng. Federated deconfounding and debiasing learning for out-of-distribution generalization. *arXiv preprint arXiv:2505.04979*, 2025.

[37] Zixuan Qin, Liu Yang, Fei Gao, Qinghua Hu, and Chenyang Shen. Uncertainty-aware aggregation for federated open set domain adaptation. *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS*, 2024.

[38] Arun Reddy, William Paul, Corban Rivera, Ketul Shah, Celso M de Melo, and Rama Chellappa. Unsupervised video domain adaptation with masked pre-training and collaborative self-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18919–18929, 2024.

[39] Xuankun Rong, Jianshu Zhang, Kun He, and Mang Ye. Can: Leveraging clients as navigators for generative replay in federated continual learning. In *Forty-second International Conference on Machine Learning*, 2025.

[40] Donald Shenaj, Eros Fanì, Marco Toldo, Debora Caldarola, Antonio Tavera, Umberto Michieli, Marco Ciccone, Pietro Zanuttigh, and Barbara Caputo. Learning across domains and devices: Style-driven source-free domain adaptation in clustered federated learning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 444–454, 2023.

[41] Donald Shenaj, Eros Fanì, Marco Toldo, Debora Caldarola, Antonio Tavera, Umberto Michieli, Marco Ciccone, Pietro Zanuttigh, and Barbara Caputo. Learning across domains and devices: Style-driven source-free domain adaptation in clustered federated learning. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 444–454, 2023.

[42] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 8432–8440, 2022.

[43] Chris Xing Tian, Haoliang Li, Yufei Wang, and Shiqi Wang. Privacy-preserving constrained domain generalization via gradient alignment, 2023.

[44] Anna Vettoruzzo, Mohamed-Rafik Bouguelia, and Thorsteinn Rögnvaldsson. Meta-learning for efficient unsupervised domain adaptation. *Neurocomputing*, 574:127264, 2024.

[45] Guancheng Wan, Xiaoran Shang, Guibin Zhang, Jinhe Bi, Yuxin Wu, Liangtao Zheng, Xin Lin, Yue Liu, Yanbiao Ma, Wenke Huang, and Bo Du. HYPERION: Fine-grained hypersphere alignment for robust federated graph learning. In *NeurIPS*, 2025.

[46] Bin Wang, Fei Deng, Shuang Wang, Wen Luo, Zhixuan Zhang, and Peifan Jiang. Siamseg: Self-training with contrastive learning for unsupervised domain adaptation semantic segmentation in remote sensing. *arXiv preprint arXiv:2410.13471*, 2024.

[47] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[48] Pan Wang, Zhengyi Zhong, and Ji Wang. Efficient federated learning via aggregation of base models. *PLoS One*, 20(8):e0327883, 2025.

[49] Qi Wang, Yiqin Lv, Yixiu Mao, Yun Qu, Yi Xu, and Xiangyang Ji. Robust fast adaptation from adversarially explicit task distribution generation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 1481–1491, 2025.

[50] Qi Wang and Herke Van Hoof. Doubly stochastic variational inference for neural processes with hierarchical latent variables. In *International Conference on Machine Learning*, pages 10018–10028. PMLR, 2020.

[51] Qi Wang and Herke Van Hoof. Learning expressive meta-representations with mixture of expert neural processes. *Advances in neural information processing systems*, 35:26242–26255, 2022.

[52] Qi Cheems Wang, Zehao Xiao, Yixiu Mao, Yun Qu, Jiayi Shen, Yiqin Lv, and Xiangyang Ji. Model predictive task sampling for efficient and robust adaptation. *arXiv preprint arXiv:2501.11039*, 2025.

[53] Seunghan Yang, Seokeon Choi, Hyunsin Park, Sungha Choi, Simyung Chang, and Sungrack Yun. Feature diversification and adaptation for federated domain generalization. In *European Conference on Computer Vision*, pages 52–70. Springer, 2024.

[54] Yuxiang Yang, Lu Wen, Pinxian Zeng, Binyu Yan, and Yan Wang. Dane: A dual-level alignment network with ensemble learning for multi-source domain adaptation. *IEEE Transactions on Instrumentation and Measurement*, 2024.

[55] Hao Yu, Xin Yang, Xin Gao, Yihui Feng, Hao Wang, Yan Kang, and Tianrui Li. Overcoming spatial-temporal catastrophic forgetting for federated class-incremental learning. MM '24, page 5280–5288, New York, NY, USA, 2024. Association for Computing Machinery.

[56] Hao Yu, Xin Yang, Xin Gao, Yan Kang, Hao Wang, Junbo Zhang, and Tianrui Li. Personalized federated continual learning via multi-granularity prompt. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 4023–4034, New York, NY, USA, 2024. Association for Computing Machinery.

[57] Junkun Yuan, Xu Ma, Defang Chen, Fei Wu, Lanfen Lin, and Kun Kuang. Collaborative semantic aggregation and calibration for federated domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12528–12541, 2023.

[58] Ling-Li Zeng, Zhipeng Fan, Jianpo Su, Min Gan, Limin Peng, Hui Shen, and Dewen Hu. Gradient matching federated domain adaptation for brain image classification. *IEEE transactions on neural networks and learning systems*, 2022.

[59] Fuyao Zhang, Xinyu Yan, Tiantong Wu, Wenjie Li, Tianxiang Chen, Yang Cao, Ran Yan, Longtao Huang, Wei Yang Bryan Lim, and Qiang Yang. Oblivionis: A lightweight learning and unlearning framework for federated large language models, 2025.

[60] Jingyuan Zhang, Yiyang Duan, Shuaicheng Niu, YANG CAO, and Wei Yang Bryan Lim. Enhancing federated domain adaptation with multi-domain prototype-based federated fine-tuning. In *The Thirteenth International Conference on Learning Representations*, 2025.

[61] Jiusi Zhang, Jilun Tian, Pengfei Yan, Shimeng Wu, Hao Luo, and Shen Yin. Multi-hop graph pooling adversarial network for cross-domain remaining useful life prediction: A distributed federated learning perspective. *Reliability Engineering & System Safety*, 244:109950, 2024.

[62] Ruipeng Zhang, Qinwei Xu, Jiangchao Yao, Ya Zhang, Qi Tian, and Yanfeng Wang. Federated domain generalization with generalization adjustment. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3954–3963, 2023.

[63] Xueyi Zhang, Chengwei Zhang, Tao Wang, Jun Tang, Songyang Lao, and Haizhou Li. Slow-fast time parameter aggregation network for class-incremental lip reading. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 747–756, 2023.

[64] Juepeng Zheng, Yibin Wen, Mengxuan Chen, Shuai Yuan, Weijia Li, Yi Zhao, Wenzhao Wu, Lixian Zhang, Runmin Dong, and Haohuan Fu. Open-set domain adaptation for scene classification using multi-adversarial learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 208:245–260, 2024.

[65] Zhengyi Zhong, Weidong Bao, Ji Wang, Jianguo Chen, Lingjuan Lyu, and Wei Yang Bryan Lim. Sacfl: Self-adaptive federated continual learning for resource-constrained end devices. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

[66] Zhengyi Zhong, Weidong Bao, Ji Wang, Shuai Zhang, Jingxuan Zhou, Lingjuan Lyu, and Wei Yang Bryan Lim. Unlearning through knowledge overwriting: Reversible federated unlearning via selective sparse adapter. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 30661–30670, 2025.

[67] Zhengyi Zhong, Weidong Bao, Ji Wang, Xiaomin Zhu, and Xiongtao Zhang. Flee: A hierarchical federated learning framework for distributed deep neural network over cloud, edge, and end device. *ACM Trans. Intell. Syst. Technol.*, 13(5), October 2022.

[68] Heming Zou, Yunliang Zang, and Xiangyang Ji. Structural features of the fly olfactory circuit mitigate the stability-plasticity dilemma in continual learning. *arXiv preprint arXiv:2502.01427*, 2025.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The contributions and scope of this paper are claimed in the abstract and introduction. Detailed information can be found in the experimental results in section 4.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We have included the theoretical analysis of *Gains* in the Appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide all the experiment details, including the code link, in the experiment section and appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The data used in this paper is public data, and we have provided the download link in the appendix. We also provide an anonymous GitHub code link in section 4. All the details of the experiments are shown in section 4 and the appendix.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We describe the data split details in section 4.3: mild data shift, medium data shift and shift. More details like hyperparameters and optimizer are shown in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Following the standard experimental setup, we repeat each experiment over 3 random seeds and report the mean of the results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the computing resources in the experiment section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We reviewed and followed the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We provide the potential broader impacts in the Appendix. E.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original papers that produced the code package and datasets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The provided Python code cannot be used without the authors' permission.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Motivation experiment settings

In the motivation experiments, we treat the last layer of LeNet as the classifier and the remaining layers as the encoder. In the class-incremental scenario, the source domain data contains classes {3, 4, 6, 7, 8, 9}. After source-domain training is completed, we sequentially introduce new clients, where the first three are from the source domain, the fourth contains new classes, and the fifth is from a new domain. Under the class-incremental setting, we consider the target client data classes to be {5}, {1, 5}, {0, 1, 5}, and {0, 1, 2, 5}, corresponding to the addition of 1, 2, 3, and 4 classes, respectively. In the domain-incremental scenario, the target domain is the SVHN dataset.

B Experimental details

During the experiment, the model used for the DigitFive [1] dataset is a CNN model, while the model used for the Amazon Review dataset [2] is an LSTM. The corresponding hyperparameters for the two datasets are as follows:

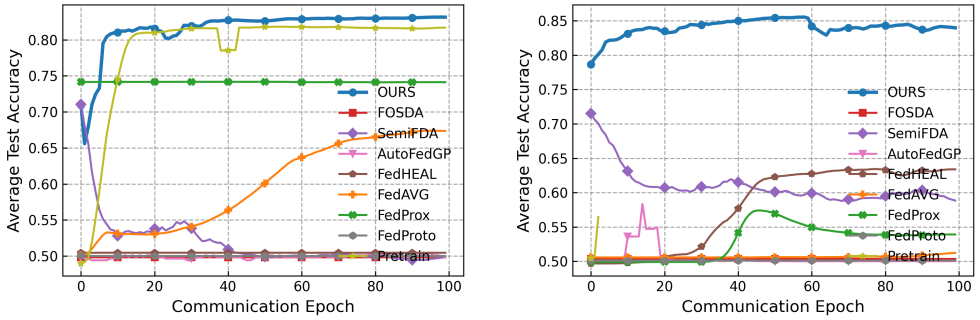
Table 8: Hyperparameter setting.

	Learning Rate	Optimizer	Batch Size
DigitFive	0.005	SGD	128
Amazon Review	0.5	SGD	64

The public dataset used by the server for new knowledge discovery is collected from open sources and typically includes various types of globally known data. Under the scenarios of mild data shift and medium data shift, after determining the data classes contained in the source domain clients, we split the data using the Dirichlet distribution with a hyperparameter of 0.1.

C Adaptation speed of Amazon Review

Fig. 5 presents the training process of *Gains* and other baselines on the Amazon Review dataset under the medium data shift and strong data shift scenarios. As shown in the figure, *Gains* achieves convergence in global performance with only a small number of epochs. This further indicates that *Gains* can accelerate the target domain adaptation process and more rapidly integrate target domain knowledge into the global model.



(a) Amazon Review, medium data shift.

(b) Amazon Review, strong data shift.

Figure 5: Training process of Amazon Review under different data shift scenarios.

[1] https://ai.bu.edu/M3SDA

[2] https://nijianmo.github.io/amazon/index.html

D More validations on generalization

In the main manuscript, we validated the effectiveness of *Gains* on part of the cases under three data shift scenarios. In this part, we will verify all the cases under medium data shift and strong data shift, further supporting the generalization capability of *Gains*. Table 9 and Table 11 show the results under different data shift scenarios for the DigitFive dataset, while Table 10 and Table 12 present the results for the Amazon Review dataset under similar conditions. It can be observed that for the DigitFive dataset, the *T-Acc*, *S-Acc*, and *G-Acc* all exceed 90% across all scenarios. Similarly, for the Amazon Review dataset, the *T-Acc*, *S-Acc*, and *G-Acc* are mostly above 80%.

Table 9: DigitFive, medium data shift.

Source domain	Target domain	*T-Acc*	*S-Acc*	*G-Acc*
MNIST-M	USPS	93.39	96.48	95.86
MNIST	USPS	92.10	99.50	98.02
SynthDigits	USPS	98.55	98.30	98.35
SVHN	USPS	90.16	90.78	90.65
USPS	MNIST-M	93.42	99.25	98.08
MNIST	MNIST-M	94.46	99.56	98.54
SynthDigits	MNIST-M	90.49	98.57	96.95
SVHN	MNIST-M	90.97	92.38	91.30
USPS	MNIST	98.99	99.41	99.33
MNIST-M	MNIST	99.07	97.89	98.12
SynthDigits	MNIST	98.89	98.59	98.65
SVHN	MNIST	97.91	90.09	91.65
USPS	SynthDigits	94.16	99.30	98.27
MNIST-M	SynthDigits	92.76	97.38	96.45
MNIST	SynthDigits	92.23	99.53	98.07
SVHN	SynthDigits	92.69	91.10	91.42
USPS	SVHN	92.19	99.20	95.79
MNIST-M	SVHN	92.52	96.94	94.06
MNIST	SVHN	93.25	99.30	95.68
SynthDigits	SVHN	91.69	98.79	97.37

Table 10: Amazon Review, medium data shift.

Source domain	Target domain	*T-Acc*	*S-Acc*	*G-Acc*
Books	Kitchen	82.22	86.43	85.59
DVDs	Kitchen	83.16	86.36	85.72
Electronics	Kitchen	86.59	89.93	89.26
Kitchen	Books	77.54	88.97	86.68
DVDs	Books	80.13	83.83	83.09
Electronics	Books	76.37	88.36	85.97
Kitchen	DVDs	77.36	89.94	87.42
Books	DVDs	82.01	86.85	85.88
Electronics	DVDs	77.50	88.65	86.42
Kitchen	Electronics	85.55	89.67	88.85
Books	Electronics	77.66	87.95	85.89
DVDs	Electronics	82.75	87.40	86.47

Table 11: DigitFive, strong data shift.

Source domain	Target domain	*T-Acc*	*S-Acc*	*G-Acc*
MNIST-M, MNIST, SynthDigits, SVHN	USPS	98.49	95.56	96.14
USPS, MNIST, SynthDigits, SVHN	MNIST-M	93.94	96.20	95.75
USPS, MNIST-M, SynthDigits, SVHN	MNIST	98.98	93.18	94.34
USPS, MNIST-M, MNIST, SVHN	SynthDigits	97.02	95.96	96.17
USPS, MNIST-M, MNIST, SynthDigits	SVHN	91.67	97.58	96.40

Table 12: Amazon Review, strong data shift.

Source domain	Target domain	*T-Acc*	*S-Acc*	*G-Acc*
Books, DVDs, Electronics	Kitchen	85.38	87.73	87.14
Kitchen, DVDs, Electronics	Books	80.54	84.95	83.85
Kitchen, Books, Electronics	DVDs	78.22	88.90	86.23
Kitchen, Books, DVDs	Electronics	86.32	85.61	85.79

E Broader impact

This paper is the first to propose a fine-grained knowledge discovery and integration pipeline in the FDA. It can significantly enhance the autonomous evolution capabilities of distributed nodes in open environments without human intervention. Additionally, we have open-sourced our code for reference in future work.

F Theoretical analysis

In this subsection, we will analyze the convergence of *Gains* using domain-increment as an example. The following assumptions are made:

Assumption 1 (Smoothness and Strong Convexity)**.** *The local loss function convex and M-smooth. Then, we have:*

- ***M-smoothness****:* $\forall \mathcal{W}_n^S(i,e+1), \mathcal{W}_n^S(i,e)$,
$$\mathcal{L}(\mathcal{W}_n^S(i,r+1)) - \mathcal{L}(\mathcal{W}_n^S(i,r)) - \left\langle \nabla\mathcal{L}(\mathcal{W}_n^S(i,r)), \mathcal{W}_n^S(i,r+1) - \mathcal{W}_n^S(i,r) \right\rangle \leq \frac{M}{2} \left\| \mathcal{W}_n^S(i,r) - \mathcal{W}_n^S(i,r+1) \right\|_2^2$$

.

Assumption 2 (Smoothness and Strong Convexity)**.** *As the number of iterations increases, the contributions of each source domain client to the target domain gradually become stable.*

The other assumptions are the same as those in Reference [28]. We first analyze the convergence of the Encoder. During each round of global update, the global parameters are:

$$\mathcal{W}(i+1) = \mathcal{W}(i) - \eta \left(\sum_{n=1}^{\mathcal{N}} \mathcal{CD}_n^E(i) \nabla\mathcal{L}\left(\mathcal{W}_n^S(i,R)\right) + \beta(i) \nabla\mathcal{L}\left(\mathcal{W}^T(i,R)\right) \right).$$

Under the smoothness assumption, if the local loss functions of the clients are convex and M-smooth, then the global loss function is also convex and M-smooth, yielding the following result:

$$\mathcal{L}(\mathcal{W}(i+1)) - \mathcal{L}(\mathcal{W}(i)) - \langle \nabla\mathcal{L}(\mathcal{W}(i)), \mathcal{W}(i+1) - \mathcal{W}(i) \rangle \leq \frac{M}{2} \left\| \mathcal{W}(i) - \mathcal{W}(i+1) \right\|_2^2.$$

Let $\mathcal{W}(i) = -\eta \left(\sum_{n=1}^{\mathcal{N}} \mathcal{CD}_n^E(i) \nabla\mathcal{L}\left(\mathcal{W}_n^S(i,R)\right) + \beta(i) \nabla\mathcal{L}\left(\mathcal{W}^T(i,R)\right) \right) = -\eta \nabla\mathcal{L}(\mathcal{W}(i))$ where $\beta(i) = \frac{|\mathcal{D}^T|}{|\mathcal{D}^T| + \sum_{n=1}^{\mathcal{N}} |\mathcal{D}_n^S|}$, we can get:

$$\mathcal{L}(\mathcal{W}(i+1)) - \mathcal{L}(\mathcal{W}(i)) + \eta \left\langle \nabla\mathcal{L}(\mathcal{W}(i)), \nabla\mathcal{L}(\mathcal{W}(i)) \right\rangle \leq \frac{M\eta^2}{2} \left\| \nabla\mathcal{L}(\mathcal{W}(i)) \right\|_2^2.$$

For simplicity,

$$\mathcal{L}(\mathcal{W}(i+1)) - \mathcal{L}(\mathcal{W}(i)) \leq \frac{M\eta^2}{2} \left\| \nabla\mathcal{L}(\mathcal{W}(i)) \right\|_2^2 - \eta \left\| \nabla\mathcal{L}(\mathcal{W}(i)) \right\|_2^2.$$

From the above equation, it can be derived that to ensure the total loss value decreases with each iteration, $\eta - \frac{M\eta^2}{2} > 0$ must be satisfied. Therefore, after I times of iterations, we get:

$$\sum_{i=0}^{I} \left\| \nabla\mathcal{L}(\mathcal{W}(i)) \right\|_2^2 \leq \frac{\mathcal{L}(\mathcal{W}(0)) - \mathcal{L}(\mathcal{W}(I))}{\left(\frac{M\eta^2}{2} - \eta \right)}.$$

Since $\mathcal{L}(\mathcal{W}(I)) > 0$, the following conclusion is obtained:

$$\frac{1}{I}\sum_{i=0}^{I}\|\nabla\mathcal{L}(\mathcal{W}(i))\|_2^2 \leq \frac{2\mathcal{L}(\mathcal{W}(0))}{I\left(M\eta^2 - 2\eta\right)}.$$

Then when $I \to \infty$,

$$\lim_{I\to\infty}\frac{1}{I}\sum_{i=0}^{I}\|\nabla\mathcal{L}(\mathcal{W}(i))\|_2^2 = 0.$$

This indicates that as the number of iterations increases, the global gradient norm tends towards zero, thereby ensuring the convergence of the algorithm. The convergence analysis of the Classifier is similar to that of the Encoder and will not be reiterated here.