Text-Enhanced Data-free Approach for Federated Class-Incremental Learning

Minh-Tuan Tran[1], Trung Le[1], Xuan-May Le[2], Mehrtash Harandi[1], Dinh Phung[1]

[1]Monash University, [2]University of Melbourne

{tuan.tran7,trunglm,mehrtash.harandi,dinh.phung}@monash.edu

xuanmay.le@student.unimelb.edu.au

Abstract

Federated Class-Incremental Learning (FCIL) is an underexplored yet pivotal issue, involving the dynamic addition of new classes in the context of federated learning. In this field, Data-Free Knowledge Transfer (DFKT) plays a crucial role in addressing catastrophic forgetting and data privacy problems. However, prior approaches lack the crucial synergy between DFKT and the model training phases, causing DFKT to encounter difficulties in generating high-quality data from a non-anchored latent space of the old task model. In this paper, we introduce LANDER (Label Text Centered Data-Free Knowledge Transfer) to address this issue by utilizing label text embeddings (LTE) produced by pretrained language models. Specifically, during the model training phase, our approach treats LTE as anchor points and constrains the feature embeddings of corresponding training samples around them, enriching the surrounding area with more meaningful information. In the DFKT phase, by using these LTE anchors, LANDER can synthesize more meaningful samples, thereby effectively addressing the forgetting problem. Additionally, instead of tightly constraining embeddings toward the anchor, the Bounding Loss is introduced to encourage sample embeddings to remain flexible within a defined radius. This approach preserves the natural differences in sample embeddings and mitigates the embedding overlap caused by heterogeneous federated settings. Extensive experiments conducted on CIFAR100, Tiny-ImageNet, and ImageNet demonstrate that LANDER significantly outperforms previous methods and achieves state-of-the-art performance in FCIL. The code is available at https://github.com/tmtuan1307/lander.

1. Introduction

Federated Learning (FL) is a decentralized and privacy-preserving technique enabling collaboration among diverse entities, such as organizations or devices [23,31,39,65]. In FL, multiple users (clients) train a common (server) model in coordination with a server without sharing personal data. FL in has recently gained attention in various fields like healthcare [58], IoT [42], and autonomous driving [14]. While conventional FL studies assume static data classes and domains, the reality is that new classes can emerge, and data domains can change over time [3,7,34,64]. For example, [6] reveals shifting customer interests in an online store with seasons, and [60] discusses the need for healthcare models to adapt to detect new diseases. Handling continuously emerging data classes through entirely new models is impractical due to substantial computational resources. Alternatively, transfer learning from pre-existing models may be considered, but it faces the issue of catastrophic forgetting [24,25], degrading performance on previous classes.

To tackle catastrophic forgetting in FL, recent studies [9, 12, 36, 60] propose the concept of Federated Continual Learning (FCL) which incorporate Continual Learning (CL) principles [3, 11, 54, 64] into Federated Learning. In that, the most popular setting is Federated Class-Incremental Learning (FCIL) [5, 12, 60, 63, 66], providing the flexibility to add new classes at any time.

The key challenges in FCIL are to mitigate catastrophic forgetting and ensure data privacy. To tackle these challenges, the common FCIL process unfolds in two main phases: client/server (model) training and the Data-Free Knowledge Transfer (DFKT) phase. In this context, unlike classic knowledge distillation methods [18,43,44,46], DFKT [16,33,53,59] has emerged as a pivotal technique since it can transfer the knowledge from the last task model (as a teacher) to the current model (as a student) to mitigate the forgetting problem but without accessing raw training data, thus ensuring privacy. The core idea behind DFKT is to generate synthetic data with confident predictions from the teacher. In other words, these synthetic data points reside in the high-confidence region of the teacher's predictions. Subsequently, this synthetic data is utilized to train the student model, effectively addressing the issue of forgetting.

However, the utilization of DFKT in previous methods has frequently resulted in unsatisfactory outcomes

[5, 63, 66]. This can be attributed to the fact that existing methods lack anchors shared between the client and teacher/server models to constrain high-confidence regions of client and server models, making them more well-organized and thereby facilitating the generation of synthetic data. Consequently, due to the disorganized and non-anchored high-confidence regions of the teacher models, to cover all knowledge from the teacher, previous methods need to generate a large amount of synthetic data in these complex regions, including both high/low-quality images, limiting the effectiveness of DFKT in mitigating catastrophic forgetting. For instance, numerous existing cat images could confidently be classified as belonging to the dog class by any teacher model, as discussed in [8, 20, 37].

In this paper, we address the problem by introducing a novel method named LANDER (LAbel text ceNter Data-free knowledgE transfeR). LANDER leverages the label-text embedding (LTE) produced by pretrained language models to reorganize the latent embedding of training data, facilitating the synthesis of high-quality samples in DFKT. Specifically, our method treats the LTEs as the anchors and optimizes the feature embeddings of training samples around these LTEs area, ensuring this area contains more semantically meaningful information. Importantly, our method queries the LTEs from the language model only once. This LTEs is stored in memory for subsequent processing, and we do not involve the language model in the training process. During the DFKT phase, LANDER capitalizes on these advantages by generating samples in proximity to these LTEs, thereby creating the synthetic data with more valuable features. This approach mitigates catastrophic forgetting, as these more meaningful samples help the model retain knowledge from previous tasks. LANDER departs from conventional approaches by using LTE as input to the generator, shifting the source of randomness from the input level to the layer level, resulting in faster and more diverse sampling. Furthermore, we introduce the concept of the Bounding Loss to encourage sample embeddings to remain flexible within a defined radius, rather than attempting to make them as close as possible to the LTEs. This method retains inherent differences in embeddings and alleviates overlap arising from heterogeneous federated settings. Our contributions can be summarized as follows:

- We propose LANDER which leverage the power of pretrained language models in FCIL. It utilizes label text embeddings as anchors to enhance knowledge transfer from previous models to the current model.
- We propose preserving natural embedding differences with the Bounding Loss to address overlap issues in imbalanced federated settings.
- We enhance data privacy by introducing a learnable data stats for the data-free generator, eliminating the need for clients to disclose specific data information.

- Extensive experiments on CIFAR100, Tiny-ImageNet, and ImageNet show that LANDER outperforms previous methods, establishing itself as the state-of-the-art (SOTA) solution for FCIL.

2. Related Work

Data-free Knowledge Transfer. DFKT or data-free knowledge distillation is an approach that enables knowledge transfer from a teacher model to a student model without the need for training data. Recent techniques, such as those introduced in [10, 40], utilize a generative model to synthesize images, guiding the teacher's predictions. The student and generator undergo joint adversarial training, facilitating rapid exploration of synthetic distributions. Notably, the use of a pretrained text encoder in DFKT, as demonstrated in [53], has achieved SOTA results. DFKT is gaining popularity in CL [19, 33] and FL [35, 62] due to its ability to preserve knowledge without relying on memory while addressing privacy concerns.

Continual Learning. Catastrophic forgetting, a significant challenge in machine learning [25], occurs when a model's performance on previously learned data decreases as it is trained on new examples. CL [61] addresses this issue by enabling models to acquire new knowledge while retaining existing knowledge. Strategies include regularization terms [1, 2, 45], experience replay [4, 50, 51], generative models [30, 52, 57], and isolating architectural parameters [13, 17, 38]. DFKT methods [19, 41, 59] prove promising for CL, especially in privacy-sensitive applications. CL encompasses various learning scenarios: task-incremental learning (TIL), domain-incremental learning (DIL), and class-incremental learning (CIL) [56]. TIL involves separate tasks, DIL maintains a consistent output space, and CIL gradually introduces new tasks and classes.

Federated Continual Learning. In real-world scenarios, local user data constantly evolves due to changing interests or data loss concerns. Federated continual learning (FCL) addresses the challenge of updating global models with evolving user data while retaining previous knowledge. Key contributions include [60], focusing on TIL with distinct task IDs, and [36], which employs knowledge distillation. Additionally, [12] diverges by assuming clients have sufficient memory for storing and sharing data. Other studies, like [22, 47, 55], explore FCL in diverse domains.

3. Federated Class-Incremental Learning

In our paper, our primary focus is on Federated Class-Incremental Learning, which applies class-incremental learning in a federated setting. FCIL framework comprises a global server $\mathcal{S}$ and multiple clients $(\mathcal{C}_1, \cdots, \mathcal{C}_m)$. In that, each client is trained to address a sequence of distinct and non-overlapping tasks without sharing their data with one
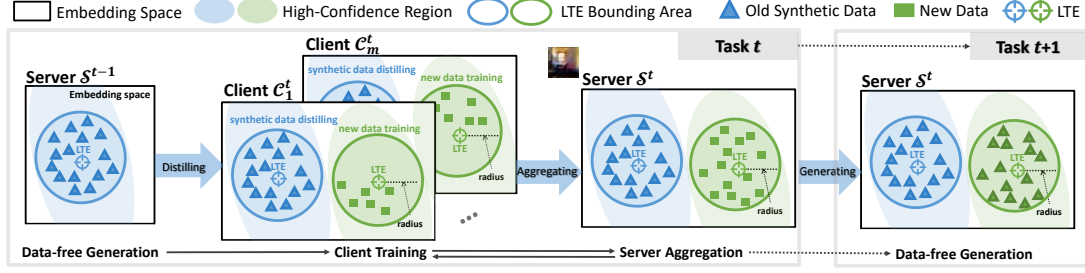
Figure 1. LANDER's motivation centers on using shared LTE as a key constraint for new task features and applying feature distillation from the previous server to anchor and organize the latent space of the client/server model. Subsequently, generating samples around the shared anchor LTE facilitates efficient data-free knowledge transfer from the previous to the current task.

another or with the central server to protect their data privacy. For task t, we denote $\mathcal{D}_k^t$ ($1 \leq k \leq m$) as the data set of task t of the client k, consisting of N_k^t pairs of samples and their labels $\{(\boldsymbol{x}_i^{kt}, \boldsymbol{y}_i^{kt})\}_{i=1}^{N_k^t}$. Labels $\boldsymbol{y}_i^t$ belong to non-overlapping subsets of classes $\mathcal{Y}_k^t \subset \mathcal{Y}$, where $\mathcal{Y}$ is the set of all possible classes. To provide clarity, we denote the server and clients model for task t as $\mathcal{S}^t$ and $(\mathcal{C}_1^t, \cdots, \mathcal{C}_m^t)$, respectively. The primary objective of FCIL is to maintain a global model performance in previous and current tasks. In our privacy-conscious scenario, the task sequence is presented in an undisclosed order, and each client $\mathcal{C}_k^t$ can exclusively access its local data $\mathcal{D}_k^t$ for task t during that task's training period, with no further access allowed thereafter. Note that the models are trained in a distributed setting, with each party having access to only a subset of the classes $\mathcal{Y}^t$ (*i.e.*, non-IID).

FCIL has recently gained significant attention for its challenging nature and its closer alignment with real-world scenarios, especially within the context of federated learning. It's worth noting that in most FL applications, task IDs are not readily available, and the preferred approach is to train a unified model that accommodates all observed data. Several methods have been proposed recently to address this problem. For example, FedCIL [66] suggests local training of the discriminator and generator to leverage generative replay, effectively compensating for the absence of old data and mitigating forgetting. On the other hand, MFCL [5] and TARGET [63] introduce a data-free approach in which the generative model is trained by the server. This approach reduces client training time and computational requirements while still eliminating the need for access to their private data.

It is clear that DFKT plays a pivotal role in mitigating the issue of catastrophic forgetting and ensuring privacy preservation in the majority of current FCIL methods. However, existing methods including TARGET [63], MFCL [5] and FedCIL [66] treat DFKT as standalone modules, without integrating it into the training phases. This isolated approach hampers their effectiveness in mitigating the model forget-

ting problem.

4. Our Proposed Method: LANDER

4.1. Motivations of LANDER

For data-free FCIL, we follow the framework at [5,63]. For task t, we firstly perform DFKT over the server model up to task $t - 1$ (i.e., $\mathcal{S}^{t-1}$) to generate synthetic data $\mathcal{M}^{t-1}$. Afterward, the server must engage in communication with the clients for c rounds, where each round comprises two phases: client-side training and server aggregation. In client-side training, given the client k, its model $\mathcal{C}_k^t$ is trained using the new task data $\mathcal{D}_k^t$ and the old task synthetic data $\mathcal{M}^{t-1}$. In server aggregation, the client models are then sent to the server side to aggregate the server model (i.e., $\mathcal{S}^t = \frac{1}{m} \sum_{k=1}^m \mathcal{C}_k^t$).

However, this naive mechanism lacks synergy between the clients and the server to constrain the complexity of high-confidence regions of the server model $\mathcal{S}^{t-1}$ in order to facilitate DFKT and generate more qualified synthetic old task images.

To organize the high-confidence regions of the client models and server model more effectively, we propose imposing constraints during the training of the client models. Specifically, given the client k, (i) for new data in $\mathcal{D}_k^t$, the feature vectors of $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}_k^t$ at the penultimate layer of the client model $\mathcal{C}_k^t$ must center around the LTE of the class $\boldsymbol{y}$, and (ii) the feature vectors of old-task synthetic data $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) \in \mathcal{M}^{t-1}$ of the client model $\mathcal{C}_k^t$ must distill those of the server model $\mathcal{S}^{t-1}$. This approach aims to organize the high-confident regions for classes of the client models more coherently around meaningful LTEs.

Moreover, on the server side, we aggregate the client models to obtain the server model. Hence, the well-organized high-confident regions for classes of the client models are inherited by the server model. Using LTE as the anchor, this inherited property certainly facilitates the server model to be data-freely distilled more effectively for generating more qualified images for old tasks, thereby al-
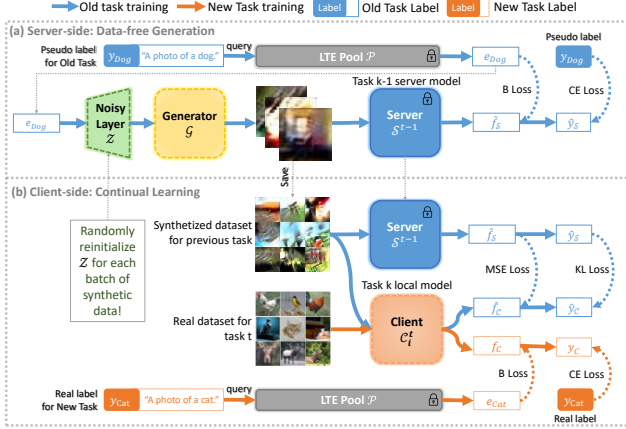
Figure 2. General Architecture of LANDER: (a) We utilize the previous server model (trained on task $t-1$) to synthesize the data. (b) Subsequently, we use this data to train the k-th task. We use the LTE as the anchor to constrain the features in both the client and generator, enhancing performance.

leviating catastrophic forgetting.

Figure 1 illustrates the motivations of our proposed LANDER. In particular, the well-organized high-confident regions for classes of the client models are inherited by the server model through the aggregation operation, further facilitating the DFKT phase for generating high-quality images of the old tasks.

In the next section, we first discuss how to effectively organize the high-confidence regions of the client/server model (Section 4.2). Then, we delve into the details of the two main components of our method: Client-Side Training (Section 4.3) and Server-Side Data Generation (Section 4.4). The overall architecture of our method is illustrated in Figure 2, while Algorithm 1 provides a comprehensive overview of the entire training process for LANDER.

4.2. How to Effectively Organize the High-confidence Regions?

We consider a typical client model $\mathcal{C}$ (i.e., $\mathcal{C}_k$ for client index k) and a new task data $\mathcal{D}$ (i.e., $\mathcal{D}_k^t$ for some task t and client index k). Given a new data/label pair $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}$, we extract the feature $\boldsymbol{f}_\mathcal{C}$ at the penultimate layer and prediction $\boldsymbol{y}_\mathcal{C}$ for $\boldsymbol{x}$ of the client model $\mathcal{C}$ (i.e., $\boldsymbol{y}_\mathcal{C}, \boldsymbol{f}_\mathcal{C} = \mathcal{C}(\boldsymbol{x})$). We now discuss how to establish constraints on $\boldsymbol{f}_\mathcal{C}$ to effectively organize the high-confidence regions of $\mathcal{C}$ on $\mathcal{D}$.

In previous works, the model $\mathcal{C}$ is trained solely by minimizing the cross-entropy loss (CE) between the prediction $\boldsymbol{y}_\mathcal{C}$ and the actual label $\boldsymbol{y}$ (ie, $CE(\boldsymbol{y}_\mathcal{C}, \boldsymbol{y})$). While this approach places the training data $\boldsymbol{x}$ in the high-confidence regions of $\mathcal{C}$, it does not impose any constraints on the feature $\boldsymbol{f}_\mathcal{C}$ of $\boldsymbol{x}$, leading to unsatisfactory organization and scattering throughout the regions.

This raises the need for common anchors shared between the client and server to both constrain the feature $\boldsymbol{f}$ and

---

Algorithm 1: LANDER

Input: E: local epochs, I: generation rounds, g generator training steps, T: number of tasks, c: number of communication rounds, synthetic data $\mathcal{M}^0 = \{\}$, LTE pool $\mathcal{P} = \{\}$.

1 **foreach** *each task* $t = 1, \cdots, T$ **do**
2   Store all embeddings $\boldsymbol{e_y} = \mathcal{E}(Y_{\boldsymbol{y}}^t)$ into $\mathcal{P}$;
3   **if** $t \neq 1$ **then** $\mathcal{M}^{t-1} = $ **DataGeneration**$(S^{t-1}, \mathcal{P})$;
4   **for** c *rounds* **do**
5   **foreach** *each client* $i = 1, \cdots, m$ **do**
6        $\mathcal{C}_k^t = \mathcal{S}^t$;
7        $\mathcal{C}_k^t = $ **ClientUpdate**$(\mathcal{C}_k^t, \mathcal{S}^{t-1}, \mathcal{M}^{t-1}, \mathcal{P})$;
8      $\mathcal{S}^t = \sum_{k=1}^m \frac{n_k}{n} \mathcal{C}_k^t$;

9 **ClientUpdate**$(\mathcal{C}_k^t, \mathcal{S}^{t-1}, \mathcal{M}^{t-1}, \mathcal{P})$
10   **for** E *epochs* **do**
11   **foreach** *batch* $((\boldsymbol{x}, \boldsymbol{y}), \hat{\boldsymbol{x}}) \sim \mathcal{D}_k^t \cup \mathcal{M}^{t-1}$ **do**
12   Query $\boldsymbol{e_y} \sim \mathcal{P}$;
13   **if** $t = 1$ **then**
14   Update $\mathcal{C}_k^t$ by minimizing Eq. 4
15   **else** Update $\mathcal{C}_k^t$ by minimizing Eq. 6 ;
16   **return** $\mathcal{C}_k^t$

17 **DataGeneration**$(\mathcal{S}^{t-1}, \mathcal{P})$
18   Initialize $\mathcal{G}, \mathcal{Q}, \mu, \sigma, \mathcal{M}^{t-1} = \{\}$;
19   **for** I *rounds* **do**
20   Initializes noisy layer $\mathcal{Z}$ and pseudo label $\hat{\boldsymbol{y}}$;
21   Query $\boldsymbol{e_{\hat{y}}} \sim \mathcal{P}$;
22   **for** g *steps* **do**
23        $\hat{\boldsymbol{x}} = (\mathcal{G}(\mathcal{Z}(\boldsymbol{e_{\hat{y}}})) - \mu)/\sigma$;
24   Update $\mathcal{G}, \mathcal{Z}, \mu, \sigma$ by minimizing Eq. 16;
25      $\mathcal{M}^{t-1} = \mathcal{M}^{t-1} \cup \hat{\boldsymbol{x}}$;
26   **for** *batch* $\hat{\boldsymbol{x}} \sim \mathcal{M}^{t-1}$ **do**
27   Update $\mathcal{Q}$ by minimizing Eq. 12;
28   **return** $\mathcal{M}^{t-1}$

---

facilitate the generation of synthetic data. To address this requirement, we propose considering the label-text embedding of $\boldsymbol{y}$ as anchors and optimizing the feature embeddings of corresponding training samples around them. In the field of text embedding, a common observation is that text with similar meanings tends to exhibit closer embedding proximity to one another [28]. Therefore, LTE has the ability to encapsulate useful interclass information, making it an ideal choice for this problem.

Inspired by [53], we initially prompt the label text $Y_{\boldsymbol{y}}$ for $\boldsymbol{y}$ by adding the text `"A photo of a {class_name}"` where `{class_name}` represents the label of the class. For example, if the label is `"dog"` the prompt will be `"A photo of a dog."` After that, the LTE is calculated using the pretrained LM as follows:

$$\boldsymbol{e_y} = \mathcal{E}(Y_{\boldsymbol{y}}), \quad \forall \boldsymbol{y} \in \mathcal{Y} . \tag{1}$$

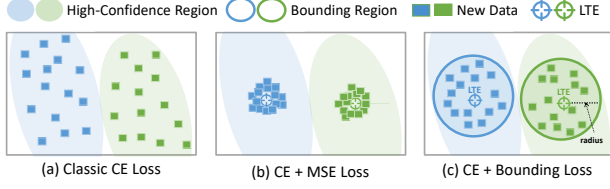Importantly, the embedding $\boldsymbol{e_y}$ is generated once and then

Figure 3. Server latent space when using only CE loss; CE with MSE Loss; and CE with B Loss to constrain the feature embedding. By using our B Loss, the latent features are organized around but still remain flexible within a defined radius r of the LTE center, mitigating the embedding overlap problem.

stored in the LTE pool $\mathcal{P}$, remaining fixed throughout the entire training process without any further fine-tuning of the pretrained language model $\mathcal{E}$. So, all that is required here is the text of the label. Then, we can generate the embedding of all labels e_y globally and send it to the client just once.

To consider the LTE as the anchor, a naive approach is to minimize the Mean Squared Error (MSE) loss between the alignment $\mathcal{W}(\cdot)$ of the feature $f_{\mathcal{C}}$ and the label embedding e_y.

$$\text{MSE}(f_{\mathcal{C}}, e_y) = \left\| e_y - \mathcal{W}(f_{\mathcal{C}}) \right\|^2 . \tag{2}$$

Bounding Loss. However, in our experiments conducted in Section 5.3, we found that striving to make the feature $f_{\mathcal{C}}$ as close as possible to the LTE e_y can exacerbate the data imbalance problem in federated learning. This issue arises from the fact that if a client has too few data samples for a particular class, it becomes easy to make the features of these data almost identical to the LTE. This results in embedding overlap between data from different clients, causing the embeddings to be too tight and lacking natural differences for similar classes. To address this problem, we introduce the concept of Bounding Loss (B Loss) (Eq. 3) to mitigate imbalance issues within heterogeneous federated settings. This approach encourages the embeddings of samples to remain flexible within a defined radius r, rather than pushing them to be as close as possible to the anchor LTE. Figure 3 illustrates the comparison of our B Loss and other methods. Using B Loss, the LTE is intended to be an anchor point, and each data point should naturally maintain some distance from it due to their inherent differences. This mitigates the overlapping problem and improves performance in heterogeneous federated settings.

$$\text{B}(f_{\mathcal{C}}, e_y) = \max\left(0, \left\| e_y - \mathcal{W}(f_{\mathcal{C}}) \right\|^2 - r\right) , \tag{3}$$

where a linear projector $\mathcal{W}(\cdot)$ is created to align feature dimensions effectively at a modest cost.

4.3. Client-Side: LT-Centered Training

On the client side of task t, the server sends the synthesized data from the previous task, denoted as $\mathcal{M}^{t-1}$, and the previous server model $\mathcal{S}^{t-1}$ to a specific client k. Then, we train the local model $\mathcal{C}_k^t$ for task k using both $\mathcal{M}^{t-1}$ and their real training data $\mathcal{D}_k^t$ simultaneously.

For the new task data, we utilize the CE loss to optimize the client model with the real training data and establish constraints on $f_{\mathcal{C}}$ using Bounding Loss at Eq. (3) to effectively organize the high-confidence regions of $\mathcal{C}_k^t$ on $\mathcal{D}_k^t$.

$$\mathcal{L}_{cur} = \text{CE}(y_{\mathcal{C}}, y) + \lambda_{ltc}\text{B}(f_{\mathcal{C}}, e_y) , \tag{4}$$

where $y_{\mathcal{C}}, f_{\mathcal{C}}$ are from $\mathcal{C}_k^t(x)$, and $(x, y) \in \mathcal{D}_k^t$.

For the synthetic data, we perform the knowledge distillation on $\mathcal{M}^{t-1}$ to help the model remember the old tasks.

$$\mathcal{L}_{pre} = \text{KL}(\hat{y}_{\mathcal{C}}, \hat{y}_{\mathcal{S}}) + \text{MSE}(\hat{f}_{\mathcal{C}}, \hat{f}_{\mathcal{S}}) , \tag{5}$$

where the old synthetic data/label pair $(\hat{x}, \hat{y}) \in \mathcal{M}^{t-1}$, $\hat{y}_{\mathcal{C}}, \hat{f}_{\mathcal{C}}$ are from $\mathcal{C}_k^t(\hat{x})$ and $\hat{y}_{\mathcal{S}}, \hat{f}_{\mathcal{S}}$ are from $\mathcal{S}^{t-1}(\hat{x})$. The KL (Kullback-Leibler divergence) term is utilized to distill the logits of the previous model into the current model. Furthermore, we propose using the MSE term to ensure that the feature embedding of the current model remains consistent with the previous one. This consistency helps organize the high-confidence regions of $\mathcal{C}_k^t$.

Combining the above losses, we obtain the loss of the client $\mathcal{C}_k^t$ as follows:

$$\mathcal{L}_{\mathcal{C}} = \alpha_{cur}^t\mathcal{L}_{cur} + \alpha_{pre}^t\mathcal{L}_{pre} . \tag{6}$$

Inspired by [19], to address the difficulty of preserving previous knowledge grows as the ratio of previous classes to new classes gets larger, the scale factor α_{cur}^t and α_{pre}^t are adaptively set as follows:

$$\alpha_{cur}^t = \frac{1 + 1/\kappa}{\delta}\alpha_{cur}; \alpha_{pre}^t = \kappa\delta\alpha_{pre} , \tag{7}$$

where $\kappa = \log_2(\frac{|\mathcal{Y}^t|}{2} + 1)$, $\delta = \sqrt{\frac{|\mathcal{Y}^{1:t-1}|}{|\mathcal{Y}^t|}}$, $|\mathcal{Y}^t|$ is the number of classes in task t, α_{cur} and α_{pre} are the base factors.

4.4. Global-Side: LT-Centered Data Generation

Data-free generation, a forefront tool in FCIL, models global data distribution without compromising client privacy. Techniques like [12, 39, 63] utilize a pretrained server model S^{t-1} and random noise $z \sim \mathcal{N}(0, I)$ via a generator $\mathcal{G}$ to craft synthetic images. However, a drawback is the use of random noise lacking meaningful information, resulting in low-quality samples and limitations in addressing the forgetting problem. Our approach, inspired by [53], leverages meaningful LTE as input to capture valuable interclass information, generating high-quality images swiftly. We introduce a noisy layer at the layer level, incorporating random noise to prevent overreliance on unchanging label information. Random reinitialization of the noisy layer with each iteration enhances diversity in synthesized images, effectively mitigating the risk of overemphasizing label information.

First, we randomly sample the pseudo label $\hat{y}$ from a categorical distribution, and then we consider the LTEs of $\hat{y}$ query from $\mathcal{P}$ (i.e. $e_{\hat{y}} \sim \mathcal{P}$) as the input for the noisy layer $\mathcal{Z}$. Subsequently, the output of $\mathcal{Z}(e_{\hat{y}})$ is fed into the generator $\mathcal{G}$ to produce a batch of synthetic images $\hat{x}$.

$$\hat{x} = \mathcal{G}(\mathcal{Z}(e_{\hat{y}})) , \qquad (8)$$

where $\mathcal{Z}$ is designed as the combination of a `BatchNorm` layer and a single `Linear` layer, follows to [53].

Learnable Data Stats. In DFKT, reusing the training data stats as the normalization values for synthetic data is a common technique to generate high-quality images [15,53,63]. In these approaches, synthetic data $\hat{x}$ is normalized using the mean and standard deviation calculated from the entire training dataset. However, in a real-world federated setting, computing these stats can be challenging since the data are decentralized, and providing this information may raise privacy concerns. To address this issue, we propose using learnable data stats (LDS) with learnable mean μ and standard deviation σ, which are then trained together with $\mathcal{Z}$ and $\mathcal{G}$. From that, the synthetic data is generated as follows:

$$\hat{x} = (\mathcal{G}(\mathcal{Z}(e_{\hat{y}})) - \mu)/\sigma . \qquad (9)$$

We also conducted experiments to demonstrate the benefits of using LDS, as discussed in Section 5.3. Without requiring the client's data stats, LDS still achieves comparable performance to the use of training data stats and significantly outperforms methods using a random stats or those without normalization technique.

Further, to effectively perform knowledge transfer, these synthetic data need to provide three essential properties including: *Similarity, Stability* and *Diversity*.

Similarity. The synthetic data $\hat{x}$ needs to be similar to the real training data. However, due to lack of access to the client data, we achieve this by minimizing the logits of S^{t-1} and the pseudo label $\hat{y}$ through the following cross-entropy (CE) loss.

$$\mathcal{L}_{\mathcal{G}}^{oh} = \text{CE}(\hat{y}_{\mathcal{S}^{t-1}}, \hat{y}) , \qquad (10)$$

where we denote $\hat{y}_{\mathcal{S}^{t-1}}$ as the prediction of $\mathcal{S}^{t-1}$ on $\hat{x}$.

Stability. To enhance the generator's stability, we use the common batch normalization regularization [15,59] to align the mean and variance of features at the batch normalization layer with their running counterparts.

$$\mathcal{L}_{\mathcal{G}}^{bn} = \sum_{l} \left(\|\mu_l(\hat{x}) - \mu_l\| + \|\sigma_l^2(\hat{x}) - \sigma_l^2\| \right) , \qquad (11)$$

where $\mu_l(\hat{x})$ and $\sigma_l^2(\hat{x})$ are the mean and variance of the l-th `BatchNorm` layer of $\mathcal{G}$, and μ_l and σ_l^2 are the mean and variance of the l-th `BatchNorm` layer of $\mathcal{S}^{t-1}$.

Diversity. To avoid the generation of similar images, LANDER incorporates an additional discriminator (student) network $\mathcal{Q}$. Specifically, for the synthetic images $\hat{x}$, $\mathcal{Q}$ is

trained to minimize the difference between its predictions and those of the server (teacher) model.

$$\mathcal{L}_{\mathcal{Q}} = \text{KL}(\hat{y}_{\mathcal{Q}}, \hat{y}_{\mathcal{S}^{t-1}}) + \text{MSE}(\hat{f}_{\mathcal{Q}}, \hat{f}_{\mathcal{S}^{t-1}}) , \qquad (12)$$

where $\hat{y}_{\mathcal{Q}}, \hat{f}_{\mathcal{Q}}$ are from $\mathcal{Q}(\hat{x})$.

On the other hand, by minimizing the negative KL Loss in Eq. 14, the generator is designed to produce images that the student network has not learned before.

$$\mathcal{L}_{\mathcal{G}}^{adv} = -\omega \text{KL}(\hat{y}_{\mathcal{Q}}, \hat{y}_{\mathcal{S}^{t-1}}) , \qquad (13)$$

$$\omega = \mathbb{1}(\arg\max(\hat{y}_{\mathcal{S}^{t-1}}) \neq \arg\max(\hat{y}_{\mathcal{Q}})) , \qquad (14)$$

where $\mathbb{1}(P)$ yields 1 if P is true and 0 if P is false. This optimization process enables the generation of a diverse set of images that cover the entire high-confidence space of the previous server model.

Furthermore, to utilize the capabilities of the anchored latent space as discussed in Section 4.3, we propose to use the Bounding Loss (Eq. 3) to ensure that the synthetic images remain within the LTEs area. This ensures that these data contain more meaningful and helpful information for transferring knowledge from the previous server model.

$$\mathcal{L}_{\mathcal{G}}^{ltc} = \text{B}(\hat{f}_{\mathcal{S}^{t-1}}, e_{\hat{y}}) . \qquad (15)$$

In summary, the final objective for the generator $\mathcal{G}$, noisy layer $\mathcal{Z}$, learnable mean μ and standard deviation σ is provided as followed:

$$\mathcal{L}_{\mathcal{G}, \mathcal{Z}, \mu, \sigma} = \mathcal{L}_{\mathcal{G}}^{adv} + \lambda_{bn}\mathcal{L}_{\mathcal{G}}^{bn} + \lambda_{oh}\mathcal{L}_{\mathcal{G}}^{oh} + \lambda_{ltc}\mathcal{L}_{\mathcal{G}}^{ltc} . \qquad (16)$$

Specifically, we set $\lambda_{bn} = 1.0$, $\lambda_{oh} = 0.5$, and $\lambda_{ltc} = 5$ in all experiments.

5. Experiments

5.1. Experimental Setting

We conduct experiments on CIFAR-100 [26], Tiny-ImageNet [29], and ImageNet [27] to evaluate our proposed approach. Following to [5,52,63], we partition the dataset classes into tasks, mimicking class continual learning with 5 and 10 tasks. ResNet18 [21] serves as the backbone for all experiments.

The evaluation employs traditional CL metrics, including average accuracy and a forgetting score [24,25]. Following to [63], our approach is compared against four baseline types in FCIL: 1) Finetune, which learns each task on each client sequentially; 2) FedWeIT [60], a widely used regularization-based method; 3) FedEWC [25] and FedLwF [32], the application of common data-free CL techniques in the federated scenario; 4) TARGET [63], a current SOTA FCIL method. For details on task configuration, hyperparameters, additional results, and visualizations, please refer to the **Supplemental Material**.

Table 1. The average accuracy (%) and forgetting values over 3 trials for all learned tasks on CIFAR-100 are presented for different task numbers (5, 10) under both IID and non-IID settings. NIID(β) indicate the Dirichlet parameter is set to β, 'Acc' denotes average accuracy, and '$\mathcal{F}$' signifies the forgetting measure [24,25]. The results of Finetune, FedEWC, FedWeIT, FedLwF and TARGET are from [63] and the best results are highlighted in bold. It's important to note that we exclude the results of MFCL [5] and Fed-CIL [66] due to their reporting in a significantly different setting, and their source codes are unavailable.

|	Acc($\uparrow$)								$\mathcal{F}(\downarrow)$							
Data partition	IID		NIID (1)		NIID (0.5)		NIID (0.1)		IID		NIID (1)		NIID (0.5)		NIID (0.1)	
Tasks	T=5	T=10	T=5	T=10	T=5	T=10	T=5	T=10	T=5	T=10	T=5	T=10	T=5	T=10	T=5	T=10
Finetune	16.12	7.83	16.33	8.45	15.49	7.64	-	-	78.12	75.89	77.59	74.89	74.95	71.52	-	-
FedEWC [25]	16.51	8.01	16.06	8.84	16.86	8.04	-	-	71.12	65.06	68.02	62.14	62.40	65.23	-	-
FedWeIT [60]	28.45	20.39	28.56	19.68	24.57	15.45	-	-	52.12	43.18	49.84	45.82	45.96	48.54	-	-
FedLwF [32]	30.61	23.27	30.94	21.16	27.59	17.98	-	-	45.32	37.71	42.71	41.03	41.25	45.23	-	-
TARGET [63]	36.31	24.76	34.89	22.85	33.33	20.71	28.32	19.25	32.23	35.45	34.48	38.25	39.23	42.23	38.23	45.23
LANDER (Ours)	**52.60**	**40.21**	**51.78**	**37.21**	**48.23**	**33.35**	**43.42**	**29.29**	**18.03**	**25.56**	**18.92**	**28.92**	**30.61**	**32.86**	**15.20**	**28.69**
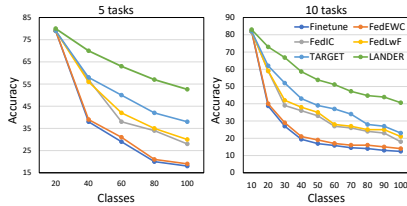
Figure 4. Average accuracy on incremental tasks.

5.2. Main Results

Experiments on CIFAR100. Following [63], we conduct experiments on 5 and 10 tasks in both IID and non-IID scenarios. Table 1 shows the final average accuracy of the server model and the forgetting measure for each experiment. Notably: 1) LANDER outperforms prior methods by two digits in both accuracy and forgetting score in all settings; 2) Increasing tasks and lowering Dirichlet parameters substantially reduce accuracy for compared methods; 3) Even in highly skewed settings, LANDER consistently outperforms others, highlighting its superior performance. Figure 4 illustrates our model's superior performance in all incremental tasks. The graph depicts average accuracy on current and previous tasks, emphasizing the model's effectiveness in enabling local clients to learn new classes in a streaming manner while mitigating forgetting.

Experiments on Large-scale Datasets. To evaluate LANDER's effectiveness, we conducted additional assessments on the more challenging Tiny-ImageNet and ImageNet datasets. For the Tiny-ImageNet dataset (Table 2), we present final average accuracy and forgetting measures for all tasks in both IID and non-IID settings with 5 tasks. The results show our method consistently achieves an approximately 3% higher average accuracy than FedLwF and TARGET. Additionally, our method exhibits significantly lower forgetting measures than FedLwF in both settings, highlighting its effectiveness in mitigating catastrophic forgetting in the presence of extreme data distributions.

For ImageNet (Table 3), as most FCIL methods do not report results on this dataset, our comparison primarily in-

Table 2. The Average Accuracy (%) and Forgetting for all learned tasks on Tiny-ImageNet for 5 tasks.

	Data partition	IID	NIID(1)	NIID(0.5)	NIID(0.1)	NIID(0.05)
Acc($\uparrow$)	FedLwF [32]	24.32	22.56	21.76	18.78	18.59
	TARGET [63]	26.25	24.12	23.95	21.15	20.95
	LANDER (Ours)	**30.29**	**28.21**	**27.98**	**25.27**	**25.02**
$\mathcal{F}(\downarrow)$	FedLwF [32]	34.57	33.94	37.23	31.43	31.19
	TARGET [63]	23.43	25.12	24.58	20.54	20.83
	LANDER (Ours)	**21.65**	**23.09**	**23.03**	**17.93**	**18.14**

Table 3. The Average Accuracy (%) on ImageNet for 5 tasks.

	Method	Task 1	Task 2	Task 3	Task 4	Task 5
IID	TARGET [63]	77.16	55.32	45.67	36.19	31.83
	LANDER (Ours)	**77.32**	**65.42**	**56.34**	**48.82**	**43.24**
NIID(1)	TARGET [63]	76.78	54.74	42.67	31.19	29.83
	LANDER (Ours)	**76.91**	**63.35**	**54.25**	**45.35**	**41.75**

volves the current SOTA FCIL method, TARGET [63]. For a fair comparison, we re-conducted TARGET's experiments to align with our settings. The results clearly demonstrate LANDER outperforms other methods in accuracy, showcasing its efficacy on a large-scale dataset.

5.3. Ablations Studies

Effectiveness of LT-Centered Generation. In Figure 5a, we analyze the impact of removing LTE constraints in data-free generation (woLTG) and the combination of a noisy layer and LTE as input (woNL) on our methods. The results indicate that these two components play important roles in our LANDER.

Effectiveness of Bounding Loss. In Figure 5b, we assess the impact of different radius r in the Bounding Loss within a heterogeneous setting. The results demonstrate that: 1) our method achieves the best results with a radius of r=0.015, which is approximately half of the minimum class-wise L2 distance in CIFAR100; 2) with a higher Dirichlet parameter, the method without using Bounding Loss (r=0) exhibits significantly lower accuracy compared to the one using r=0.015, highlighting the benefits of our Bounding Loss in addressing the imbalance problem in federated learning.

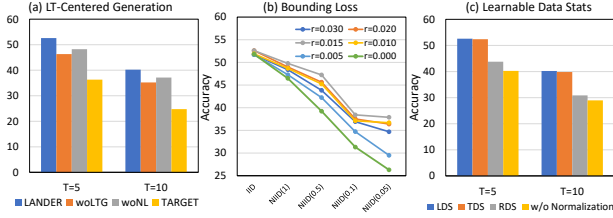**Effectiveness of Learnable Data Stats.** In Figure 5c, we

Figure 5. (a) Accuracies in 5 and 10 tasks of our method with and without LT-centered generation. (b) Incremental Accuracy on CIFAR-100 for different values of r in Bounding Loss. (c) Accuracies for different kinds of data normalization values.

compare the normalization using our LDS with that using training data stats (TDS) [63, 66], random data stats (RDS), and without using normalization. The results show: 1) Without normalization or with random data stats, our work's performance decreases significantly; 2) Our LDS performs comparably or slightly better than the TDS for data normalization, demonstrating its benefits in our work.

5.4. Further Analysis

Comparison with Different Prompting Engineering Templates. We assess the influence of various prompting engineering techniques for generating label text. We introduce three approaches to prompt label text: P1: "a class of a {class_name}", P2: "a photo of a {class_name}", P3: "a photo of a {class_index}". Table 4 highlights that P2 outperform the best in the comparison. Furthermore, even when using only the label index, P3 maintains a performance advantage over the best baseline. This indicates the application of using label indices in datasets with less meaningful labels, highlighting the practical effectiveness of LANDER.

Table 4. Accuracies of different prompt engineering methods.

	IID				NIID(0.5)			
Text Encoder	SOTA	P1	P2	P3	SOTA	P1	P2	P3
Accuracy	36.31	52.45	**52.60**	50.12	33.33	48.09	**48.23**	45.39

Comparison with Different Text Encoder. We evaluate our LANDER across three common text encoders: Doc2Vec [28], SBERT [49], and CLIP [48]. Table 5 reveals that LANDER performs well across diverse language models, leveraging their capacity to capture label-text relations. When coupled with the label-index prompt engineering method discussed earlier, our approach adapts effectively to different domains, even without meaningful label-text. Furthermore, utilizing foundational models like CLIP enhances our model's performance marginally, this indicate the benefit of multimodal models to our works. Finally, we select CLIP as the text encoder for this paper.

Table 5. Accuracies of our LANDER with different LM.

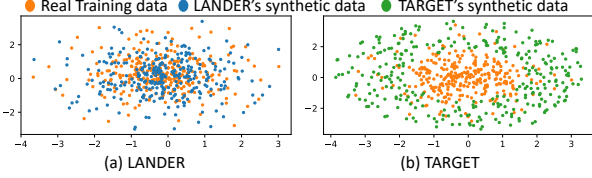|	IID				NIID(0.5)			
Text Encoder	SOTA	Doc2Vec	SBERT	CLIP	SOTA	Doc2Vec	SBERT	CLIP
Accuracy	36.31	52.45	52.53	**52.60**	33.33	48.21	48.18	**48.23**

Figure 6. Visualizing t-SNE on synthetic and real data in a randomly chosen class in CIFAR-100. Our LANDER generates data in latent space that closely resembles real data.
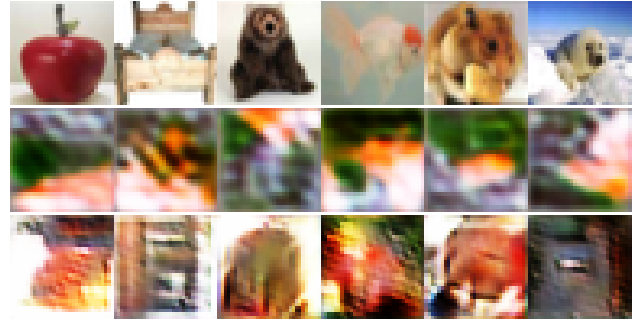

Figure 7. Real data (top) vs synthetic data generated by TARGET (middle) and our LANDER (bottom) in CIFAR-100. Each column shows samples from the same class.

Visualization. The t-SNE visualization in Figure 6 illustrates synthetic data generated by our LANDER and TARGET. It is evident that, using the anchor LTE, LANDER generates samples with embeddings more similar to real data compared to TARGET [63]. This observation underscores the reason behind the improvement in our approach. **Visualization on Synthetic Data**. Figure 7 compares real and synthetic data from LANDER and TARGET. Our synthetic samples intentionally differ from specific training examples, preserving privacy. Unlike TARGET's meaningless images, our samples capture essential class knowledge and effectively represent the entire class. Consequently, the inclusion of synthetic samples significantly mitigates the issue of catastrophic forgetting.

6. Conclusion

In this paper, we propose LANDER to mitigate the forgetting issue in federated learning. Specifically, we treat LTEs as anchor points of data feature embeddings during model training, enriching surrounding feature embeddings. In the DFKT phase, LANDER leverages thse LTE anchors to synthesize more meaningful samples, effectively addressing forgetting. Extensive experimental results demonstrate that our method achieves SOTA performance in FCIL.

Acknowledgements

References

[1] Hongjoon Ahn, Sungmin Cha, Donggyu Lee, and Taesup Moon. Uncertainty-based continual learning with adaptive regularization. *Advances in neural information processing systems*, 32, 2019. 2

[2] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*, pages 139–154, 2018. 2

[3] Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11254–11263, 2019. 1

[4] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *Advances in neural information processing systems*, 32, 2019. 2

[5] Sara Babakniya, Zalan Fabian, Chaoyang He, Mahdi Soltanolkotabi, and Salman Avestimehr. A data-free approach to mitigate catastrophic forgetting in federated class incremental learning for vision tasks. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 1, 2, 3, 6, 7

[6] Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8218–8227, 2021. 1

[7] Eden Belouadah and Adrian Popescu. Il2m: Class incremental learning with dual memory. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 583–592, 2019. 1

[8] Tuan Anh Bui, Trung Le, Quan Tran, He Zhao, and Dinh Phung. A unified wasserstein distributional robustness framework for adversarial training. *arXiv preprint arXiv:2202.13437*, 2022. 2

[9] Thang D Bui, Cuong V Nguyen, Siddharth Swaroop, and Richard E Turner. Partitioned variational inference: A unified framework encompassing federated and continual learning. *arXiv preprint arXiv:1811.11206*, 2018. 1

[10] Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3514–3522, 2019. 2

[11] Khanh Doan, Quyen Tran, Tuan Nguyen, Dinh Phung, and Trung Le. Class-prototype conditional diffusion model for continual learning with generative replay. *arXiv preprint arXiv:2312.06710*, 2023. 1

[12] Jiahua Dong, Lixu Wang, Zhen Fang, Gan Sun, Shichao Xu, Xiao Wang, and Qi Zhu. Federated class-incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10164–10173, 2022. 1, 2, 5

[13] Sayna Ebrahimi, Franziska Meier, Roberto Calandra, Trevor Darrell, and Marcus Rohrbach. Adversarial continual learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 386–402. Springer, 2020. 2

[14] Ahmet M Elbir, Burak Soner, Sinem Çöleri, Deniz Gündüz, and Mehdi Bennis. Federated learning in vehicular networks. In *2022 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, pages 72–77. IEEE, 2022. 1

[15] Gongfan Fang, Kanya Mo, Xinchao Wang, Jie Song, Shitao Bei, Haofei Zhang, and Mingli Song. Up to 100x faster data-free knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6597–6604, 2022. 6

[16] Gongfan Fang, Jie Song, Xinchao Wang, Chengchao Shen, Xingen Wang, and Mingli Song. Contrastive model inversion for data-free knowledge distillation. *arXiv preprint arXiv:2105.08584*, 2021. 1

[17] Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A Rusu, Alexander Pritzel, and Daan Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*, 2017. 2

[18] Michael Fu, Van Nguyen, Chakkrit Kla Tantithamthavorn, Trung Le, and Dinh Phung. Vulexplainer: A transformer-based hierarchical distillation for explaining vulnerability types. *IEEE Transactions on Software Engineering*, 2023. 1

[19] Qiankun Gao, Chen Zhao, Bernard Ghanem, and Jian Zhang. R-dfcil: Relation-guided representation learning for data-free class incremental learning. In *European Conference on Computer Vision*, pages 423–439. Springer, 2022. 2, 5

[20] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 2

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[22] Ziyue Jiang, Yi Ren, Ming Lei, and Zhou Zhao. Fedspeech: Federated text-to-speech with continual learning. *arXiv preprint arXiv:2110.07216*, 2021. 2

[23] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021. 1

[24] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. 1, 6, 7

[25] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 1, 2, 6, 7

[26] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 6

[28] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR, 2014. 4, 8

[29] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 6

[30] Timothée Lesort, Hugo Caselles-Dupré, Michael Garcia-Ortiz, Andrei Stoian, and David Filliat. Generative models from the perspective of continual learning. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019. 2

[31] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020. 1

[32] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 6, 7

[33] Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. Data-free knowledge distillation for deep neural networks. *arXiv preprint arXiv:1710.07535*, 2017. 1, 2

[34] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017. 1

[35] Kangyang Luo, Shuai Wang, Yexuan Fu, Xiang Li, Yunshi Lan, and Ming Gao. DFRD: Data-free robustness distillation for heterogeneous federated learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2

[36] Yuhang Ma, Zhongle Xie, Jue Wang, Ke Chen, and Lidan Shou. Continual federated learning based on knowledge distillation. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, volume 3, 2022. 1, 2

[37] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 2

[38] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018. 2

[39] 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. 1, 5

[40] Paul Micaelli and Amos J Storkey. Zero-shot knowledge transfer via adversarial belief matching. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[41] Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks. 2015. 2

[42] Dinh C Nguyen, Ming Ding, Pubudu N Pathirana, Aruna Seneviratne, Jun Li, and H Vincent Poor. Federated learning for internet of things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 23(3):1622–1658, 2021. 1

[43] Tuan Nguyen, Van Nguyen, Trung Le, He Zhao, Quan Hung Tran, and Dinh Phung. Cycle class consistency with distributional optimal transport and knowledge distillation for unsupervised domain adaptation. In *Uncertainty in Artificial Intelligence*, pages 1519–1529. PMLR, 2022. 1

[44] Thanh Nguyen-Duc, Trung Le, He Zhao, Jianfei Cai, and Dinh Phung. Adversarial local distribution regularization for knowledge distillation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4681–4690, 2023. 1

[45] Pingbo Pan, Siddharth Swaroop, Alexander Immer, Runa Eschenhagen, Richard Turner, and Mohammad Emtiyaz E Khan. Continual deep learning by functional regularisation of memorable past. *Advances in Neural Information Processing Systems*, 33:4453–4464, 2020. 2

[46] Cuong Pham, Van-Anh Nguyen, Trung Le, Dinh Phung, Gustavo Carneiro, and Thanh-Toan Do. Frequency attention for knowledge distillation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2277–2286, 2024. 1

[47] Aman Priyanshu, Mudit Sinha, and Shreyans Mehta. Continual distributed learning for crisis management. *arXiv preprint arXiv:2104.12876*, 2021. 2

[48] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 8

[49] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019. 8

[50] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[51] Dongsub Shim, Zheda Mai, Jihwan Jeong, Scott Sanner, Hyunwoo Kim, and Jongseong Jang. Online class-incremental continual learning with adversarial shapley value. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9630–9638, 2021. 2

[52] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017. 2, 6

[53] Minh-Tuan Tran, Trung Le, Xuan-May Le, Mehrtash Harandi, Quan Hung Tran, and Dinh Phung. Nayer: Noisy layer data generation for efficient and effective data-free knowledge distillation. *arXiv preprint arXiv:2310.00258*, 2023. 1, 2, 4, 5, 6

[54] Quyen Tran, Lam Tran, Khoat Than, Toan Tran, Dinh Phung, and Trung Le. Koppa: Improving prompt-based continual learning with key-query orthogonal projection and prototype-based one-versus-all. *arXiv preprint arXiv:2311.15414*, 2023. 1

[55] Anastasiia Usmanova, François Portet, Philippe Lalanda, and German Vega. A distillation-based approach integrating continual learning and federated learning for pervasive services. *arXiv preprint arXiv:2109.04197*, 2021. 2

[56] Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019. 2

[57] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, Zhengyou Zhang, and Yun Fu. Incremental classifier learning with generative adversarial networks. *arXiv preprint arXiv:1802.00853*, 2018. 2

[58] Jie Xu, Benjamin S Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. Federated learning for healthcare informatics. *Journal of Healthcare Informatics Research*, 5:1–19, 2021. 1

[59] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deep-inversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8715–8724, 2020. 1, 2, 6

[60] Jaehong Yoon, Wonyong Jeong, Giwoong Lee, Eunho Yang, and Sung Ju Hwang. Federated continual learning with weighted inter-client transfer. In *International Conference on Machine Learning*, pages 12073–12086. PMLR, 2021. 1, 2, 6, 7

[61] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017. 2

[62] Jie Zhang, Chen Chen, Bo Li, Lingjuan Lyu, Shuang Wu, Shouhong Ding, Chunhua Shen, and Chao Wu. Dense: Data-free one-shot federated learning. *Advances in Neural Information Processing Systems*, 35:21414–21428, 2022. 2

[63] Jie Zhang, Chen Chen, Weiming Zhuang, and Lingjuan Lyu. Target: Federated class-continual learning via exemplar-free distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4782–4793, 2023. 1, 2, 3, 5, 6, 7, 8

[64] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13208–13217, 2020. 1

[65] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018. 1

[66] Martin Zong, Zengyu Qiu, Xinzhu Ma, Kunlin Yang, Chunya Liu, Jun Hou, Shuai Yi, and Wanli Ouyang. Better teacher better student: Dynamic prior knowledge for knowledge distillation. In *The Eleventh International Conference on Learning Representations*, 2023. 1, 2, 3, 7, 8