COALA: A Practical and Vision-Centric Federated Learning Platform

Weiming Zhuang [*1] **Jian Xu** [† *2] **Chen Chen** [1] **Jingtao Li** [1] **Lingjuan Lyu** [1]

Abstract

We present COALA, a vision-centric Federated Learning (FL) platform, and a suite of benchmarks for practical FL scenarios, which we categorize into three levels: task, data, and model. At the task level, COALA extends support from simple classification to 15 computer vision tasks, including object detection, segmentation, pose estimation, and more. It also facilitates federated multiple-task learning, allowing clients to tackle multiple tasks simultaneously. At the data level, COALA goes beyond supervised FL to benchmark both semi-supervised FL and unsupervised FL. It also benchmarks feature distribution shifts other than commonly considered label distribution shifts. In addition to dealing with static data, it supports federated continual learning for continuously changing data in real-world scenarios. At the model level, COALA benchmarks FL with split models and different models in different clients. COALA platform offers three degrees of customization for these practical FL scenarios, including configuration customization, components customization, and workflow customization. We conduct systematic benchmarking experiments for the practical FL scenarios and highlight potential opportunities for further advancements in FL.

1. Introduction

Federated learning (FL) is a distributed machine learning technique where a central server coordinates model training on decentralized clients (e.g., edge devices and institutions), which preserves data privacy by avoiding centralizing raw data from these clients (McMahan et al., 2017; Kairouz et al., 2019). It has received tremendous attention over the past few years in many application domains such as computer vision (CV) (He et al., 2021), natural language processing (NLP) (Wang et al., 2021; Cai et al., 2023), recommendation systems (Luo et al., 2022; Zhang et al., 2023a) and information retrieval (Yang et al., 2023). In particular, FL shows a great potential in computer vision applications as they are often associated with critical privacy issues due to sensitive information in images (e.g., facial recognition (Niu & Deng, 2022) and medical imaging (Nguyen et al., 2022)). Therefore, it is of significant interest and importance to exploit the potential of FL across broader CV tasks.

To facilitate the fast development of FL, many FL benchmarks and libraries have been developed, the majority of which focus on addressing the data heterogeneity and scalability challenges. Particularly, several benchmarks (Caldas et al., 2018; Hu et al., 2020; Lai et al., 2022; He et al., 2020) provide datasets to simulate data heterogeneity in FL, where the data distributions among clients are different. Besides, considerable efforts of FL frameworks or libraries (Lai et al., 2022; Garcia et al., 2022; Huba et al., 2022; Beutel et al., 2020; Bonawitz et al., 2019; Zhang et al., 2022) aim to support the simulation or real-world deployment of FL at scale. However, these benchmarks and libraries cannot well support the latest development of FL in CV applications in the following three aspects:

Task level. Most research works are limited to the simple image classification task (McMahan et al., 2017). Other more challenging but practical CV tasks such as object detection had been mostly overlooked. Although several benchmarks support several other CV tasks, they are either task-specific (Zhuang et al., 2020) or consider only one task in FL at a time (He et al., 2021; Hu et al., 2020). Recent work (Zhuang et al., 2023c) has demonstrated the potential of training multiple CV tasks together with a significantly improved trade-off between performance and efficiency.

Data level. The majority of works focus only on supervised learning, where all the training samples in clients contain labels. Recently, semi-supervised or unsupervised FL has emerged as an important topic as it is hard to obtain data labels on clients (Diao et al., 2022; Zhuang et al., 2022c). In addition, the existing libraries mostly consider skewed label distribution in data heterogeneity, while the feature shifts where the data in each client is from different domains are often overlooked. Moreover, static data distribution among

[†] Contributed during an internship at Sony AI [*]Equal contribution [1]Sony AI [2]Tsinghua University. Correspondence to: Lingjuan Lyu <lingjuan.lv@sony.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

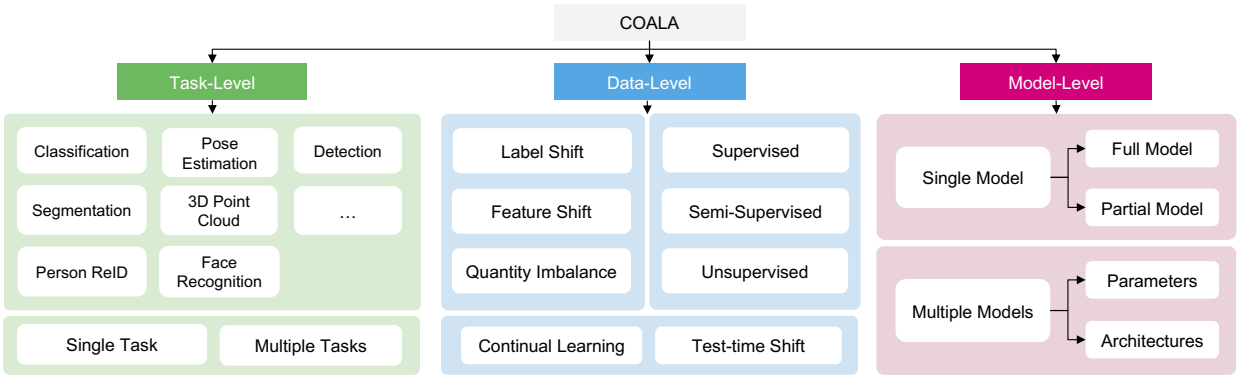


Figure 1. Illustration of three levels of practical FL scenarios supported by COALA. At the task level, we support diverse CV tasks and training of multiple tasks in FL. At the data level, we offer out-of-the-box benchmarks for different types of data heterogeneity, various degrees of data annotation availability, and dynamic changes in data. At the model level, we extend beyond single and full model FL training to split model training and multiple model training with different architectures or parameters on clients.

clients is usually assumed, which however may not hold in realistic scenarios, e.g., the distribution of images captured from street cameras in smart cities could evolve over time.

Model level. Most works predominantly consider only a single model with complete model architecture trained in clients. However, many works have demonstrated the feasibility of spitting a model and offloading a part of the model to the server to reduce computation on clients (Thapa et al., 2022; Li et al., 2023). Besides, recent studies consider more practical scenarios of training distinct models in different FL clients (Diao et al., 2021; Qin et al., 2023). Moreover, adopting foundation models in FL, especially by parameter-efficient fine tuning (PEFT), also draws increased attention (Zhuang et al., 2023a; Zhang et al., 2023d; Woisetschläger et al., 2024). Supporting and benchmarking these scenarios are vital for the further development of CV and FL in heterogeneous systems.

Contribution. In this work, we propose a new vision-centric FL platform, named COALA, together with a suite of FL benchmarks to bridge gaps in these three levels. COALA offers extensive benchmarks for diverse vision tasks and various new learning paradigms in FL, which, to the best of our knowledge, covers the most comprehensive and up-to-date research topics in current FL studies. At the task level, we extend to a broader spectrum of 15 CV tasks, including classification, object detection, segmentation, pose estimation, face recognition, and more (refer to Table 9). Additionally, we facilitate federated multiple-task learning, enabling clients to simultaneously train on more than one task. At the data level, COALA expands upon supervised FL and label distribution shift in data heterogeneity: it can support semi-supervised FL, unsupervised FL, and multi-domain FL with feature distribution shifts among local training data. Furthermore, COALA caters to federated continual learning, accounting for continuously changing data in clients in practical scenarios. At the model level, COALA supports traditional training of a single full model, computationally efficient split learning, and federated multiple-model training where clients can train multiple models with varying parameters and architectures.

COALA seamlessly supports new FL scenarios with three degrees of customization: configuration customization, components customization, and workflow customization (Section 4). It provides automated benchmarks and evaluations for a suite of practical FL scenarios in the task, data, and model levels (Section 3). Our systematic experiments showcase the capability of COALA to deliver comprehensive benchmarking across various FL scenarios. These results demonstrate the promising potential of COALA while also showing the opportunities for further enhancements in the emerging domains of CV and FL.

2. Related Work

Prior FL platforms. In the past few years, there have been a bunch of open-source platforms and tools being developed to facilitate algorithm evaluations and practical applications. For example, the FATE (WeBank, 2019), FederatedScope (Xie et al., 2023), FedML (He et al., 2020), OpenFed (Chen et al., 2023b), just name a few, are among the most famous frameworks released by industrial companies, while LEAF (Caldas et al., 2018), Flower (Beutel et al., 2020), FedScale (Lai et al., 2022), EasyFL (Zhuang et al., 2022a), FLUTE (Garcia et al., 2022), FedLab (Zeng et al., 2023) are mainly contributed by academia. Other frequently mentioned platforms in the literature include TensorFlow Federated (Tensorflow.org, 2019), PaddleFL (PaddlePaddle, 2020), PySyft (Ryffel et al., 2018) and FedVision (Liu et al., 2020). We also notice that some recent works, such as Felicitas (Zhang et al., 2022), PAPAYA (Huba et al., 2022), Flint (Wang et al., 2023a) and FS-Real (Chen et al., 2023a), pay more attention to the real-world constraints for more realistic device-cloud collaborative FL. It is worth noting that some frameworks

mainly focus on supporting flexible simulations for research purposes only, including FLUTE (Garcia et al., 2022) that aims at rapid prototyping of new algorithms at scale and FLGo (Wang et al., 2023b) that customizes FL tasks with shareable components and plugins. While each platform has its own specialties and advantages, there is still no platform that specializes and enables comprehensive vision tasks under practical FL settings (continual learning; multi-domain data, etc). By contrast, our proposed COALA is the first vision-centric FL platform that can be used for both distributed training simulation and realistic cross-device applications with high flexibility in customization.

Existing FL benchmarks. LEAF (Caldas et al., 2018) is the first FL benchmark that provides some datasets with statistical heterogeneity. However, the supported vision tasks and datasets are very limited, and only supervised learning is considered. FedReID (Zhuang et al., 2020) improved the federated person re-identification via benchmark analysis. An experimental study of representative FL algorithms for image classification on non-IID data silos was provided in (Li et al., 2022). OARF (Hu et al., 2022b) provided a benchmark suite that is diverse in data size, label distribution, feature distribution, and learning task complexity. FedScale (Lai et al., 2022) provided the natural partitions of real-world datasets with real client-data mapping to better simulate large-scale FL settings. The most related work is FedCV (He et al., 2021), which is the first work that evaluates representative vision tasks in FL settings, including classification, object detection, and segmentation. However, with the growing of FL in real-world applications, those benchmarks are not comprehensive enough to assess the effectiveness of FL algorithms on diverse vision tasks under various real scenarios. We also notice some benchmarks specialized on other tasks and applications, including audio (Zhang et al., 2023c), NLP (Lin et al., 2022), multi-modal learning (Feng et al., 2023), IoT (Alam et al., 2023), and model personalization (Chen et al., 2022), etc.

3. COALA Benchmark: Practical FL Scenarios

In this section, we first introduce the basic FL protocol, followed by the practical FL scenarios supported by our COALA benchmark from task level, data level, and model level, as summarized in Figure 1.

3.1. Basic FL Protocol

We consider a typical FL setup with m clients that collaboratively train a global model with parameters $\boldsymbol{w}$ under the coordination of a server, which could be formalized by:

$$\min_{\boldsymbol{w}} L(\boldsymbol{w}) = \sum_{i=1}^{m} \frac{|\mathcal{D}_i|}{|\mathcal{D}|} \underbrace{\mathbb{E}_{(x_i,y_i)\sim\mathcal{D}_i}\left[l(f(x_i;\boldsymbol{w}),y_i)\right]}_{:=L_i(\boldsymbol{w})}, \quad (1)$$

where $\mathcal{D}_i$ is the local training data in the client following an underlying distribution $\mathcal{P}_i(X,Y)$ on $\mathcal{X}_i \times \mathcal{Y}_i$, where $\mathcal{X}_i$ is the input space and $\mathcal{Y}_i$ is the label space. $\mathcal{D}$ denotes the collection of all training data among clients. $f(x_i;\boldsymbol{w})$ and $l(f(x_i;\boldsymbol{w}),y_i)$ denote the model output and loss function, respectively, given parameters $\boldsymbol{w}$ and a data point (x_i,y_i). The objective function $l(\cdot,\cdot)$ varies based on specific tasks, such as cross-entropy loss for classification and Mean Square Error (MSE) for regression. The global objective $L(\boldsymbol{w})$ can be regarded as a weighted average of local objectives. FedAvg (McMahan et al., 2017) is the de facto algorithm in FL, with advanced methods following this basic protocol while introducing new strategies in client training or server aggregation (Reddi et al., 2020a).

3.2. Task Level: Diverse CV Tasks and Multiple Tasks

COALA provides out-of-the-box support for 15 vision tasks, including classification, detection, segmentation, pose estimation, face recognition, person re-identification (ReID), 3D point cloud, and more. A comprehensive overview of the tasks, datasets, and models is presented in Table 9 in the Appendix. To the best of our knowledge, COALA is the most comprehensive FL platform for vision tasks and it is also highly customizable and easy to extend to new tasks. Beyond single-task training, COALA also facilitates the concurrent training of multiple tasks.

Federated Image Classification. Classification is the most common task in FL (McMahan et al., 2017; Li et al., 2020a), where the label space could be expressed by $\mathcal{Y} = \{1, .., C\}$ and C is the total number of categories. COALA offers seven datasets with varying scales and difficulty levels for classification. It enables the simulation of diverse non-IID data types, varying degrees of data annotation availability, and dynamic changes in data, detailed in Section 3.3.

Federated Object Detection. Object detection is a pivotal vision task in practical applications such as autonomous driving systems. However, federated object detection has received much less attention than classification (Liu et al., 2020; Kim et al., 2023). It follows the basic FL protocol, but with more complex data annotations as each image often contains multiple objects, each annotated with category labels and bounding box positions. As a result, it is much more challenging to simulate label distribution shifts for the detection task. COALA introduces two different non-IID simulations (detailed in Appendix C) on BDD100K dataset (Yu et al., 2020), which is a representative driving dataset for object detection. In addition, our platform supports both single-stage detectors like YOLO (Redmon et al., 2016) and two-stage detectors like Faster RCNN (Ren et al., 2015).

Federated Semantic Segmentation. Segmentation is another important vision task widely used in autonomous driving and medical image analysis. Training samples in seg-

mentation are annotated in a pixel-wise manner, making it more challenging to obtain sufficient labeled data in a single client. This difficulty raises the need for collaborative learning using FL. Meanwhile, the non-IID data also brings new challenges as different clients may have different objects and there may exist foreground-background inconsistency (Miao et al., 2023). For this task, we integrate the Pascal VOC (Everingham et al., 2010) with label shift for simulation and DeepLab models (Chen et al., 2018) for benchmarking.

Federated Pose Estimation. Pose estimation aims to detect the position and orientation of a person or an object (Toshev & Szegedy, 2014). To the best of our knowledge, no prior studies in FL investigate this task and provide empirical results. To bridge this gap, we integrate the popular dataset MPII Human Pose (Andriluka et al., 2014) with IID data partition to provide an initial exploration in FL.

Federated Face Recognition and Person ReID. The widespread use of face recognition (e.g., airport check-in and mobile Face ID) has sparked concerns about individual privacy. FL can provide privacy-aware training for face recognition models without access to private face images from clients (Meng et al., 2022; Niu & Deng, 2022; Liu et al., 2022; Zhuang et al., 2022b). Person re-identification (ReID) (Zhuang et al., 2020; 2021b; 2023b) aims at matching a person's identity across different cameras or locations. COALA integrates four datasets for face recognition and eight datasets for person ReID from independent sources to mimic the decentralized data with feature shift.

Federated Multiple-Task Learning. In addition to training a single vision task, federated multiple-task learning emerges as a new FL scenario where each client trains multiple tasks efficiently and simultaneously under resource constraints (Bhuyan et al., 2022). These tasks can be trained separately with multiple models or jointly by adopting multi-task learning (MTL) with a shared backbone encoder and multiple task-specific decoders (Zhuang et al., 2023c). We integrate the widely used Taskonomy dataset (Zamir et al., 2018) with each client containing data from one building to simulate the statistical heterogeneity and quantity imbalance to evaluate the potential of federated multiple-task learning.

3.3. Data Level: Realistic Data Patterns

COALA provides flexible support for different data patterns arising in FL, including the distribution shift (label shift, feature shift, test-time shift), quantity imbalance, continual learning, and different availability of data annotations (supervised, semi-supervised, unsupervised). We provide benchmarks and simulations for these realistic data patterns to evaluate FL algorithms.

Training Data from Multiple Domains. To tackle the data heterogeneity issue in FL, most studies use the single domain datasets with *label shift* (e.g., shared input space $\mathcal{X}_i = \mathcal{X}, \forall i \in [m]$) for evaluating the performance of either global model or local personalized models. However, in real-world applications, due to the diverse environments and independent data collection, data samples could originate from different domains, resulting in *feature shifts* (different feature distributions) among clients (i.e., $\mathcal{X}_i \neq \mathcal{X}_j, i, j \in [m]$). The most recent works study FL with feature distribution shifts (Zhuang & Lyu, 2023; 2024), using datasets such as Digits-Five, Office-Caltech, and DomainNet (Li et al., 2020b). COALA provides benchmarks using these datasets for the classification task. It also supports other tasks like person ReID with feature shifts. The BDD100K data collected from different weather, scene, and time-of-day attributes can also be considered as feature shifts.

Federated Semi-supervised Learning. As annotating all training data is time-consuming and expensive, it is usually impractical to assume that all the clients could have fully labeled training sets. Most clients are likely to have only a small portion of data labeled, with the remaining data being unlabeled; in some cases, all clients might be unlabeled while the central server has labeled data (Zhang et al., 2021; Liang et al., 2022; Diao et al., 2022). These scenarios are referred to as label-in-client and label-in-server situations in the literature (Jeong et al., 2021). COALA provides benchmarks for both scenarios. The goal of semi-supervised FL is to leverage the unlabeled data to train a better model without violating the privacy of local data.

Federated Unsupervised Learning. In an extreme case where all the clients only have unlabeled data, unsupervised learning can be adopted into the FL. COALA supports both federated self-supervised learning to learn generic visual representations (Zhuang et al., 2022c) and federated unsupervised learning for specific tasks like person ReID. Besides, we provide a benchmark using one popular self-supervised learning method called BYOL (Grill et al., 2020).

Federated Continual Learning. Conventional FL assumes data in clients remains static, yet in reality, data can be dynamically changing over time. Federated continual learning emerges as a solution to continuously update the model with the new data while preventing forgetting the old knowledge (Zhang et al., 2023b). COALA provides a benchmark for federated class-continual learning, enabling clients to continuously collect and learn new data classes. This introduces variability in the label space $\mathcal{Y}^t$ across different time slots t, with a typical setting where $\mathcal{Y}^t \cap \mathcal{Y}^{t'} = \varnothing$ for $t \neq t'$.

Test-time Distribution Shift. Addressing test-time distribution shifts is an emerging challenge in FL, where the data distribution could differ in training and testing. Such shifts could be label shifts, feature shifts, or covariate shifts (Tan et al., 2023). COALA provides benchmarks for test-time distribution shift in FL. The challenge is how to utilize the diverse data in FL to learn either a shift-resilient model (Tan

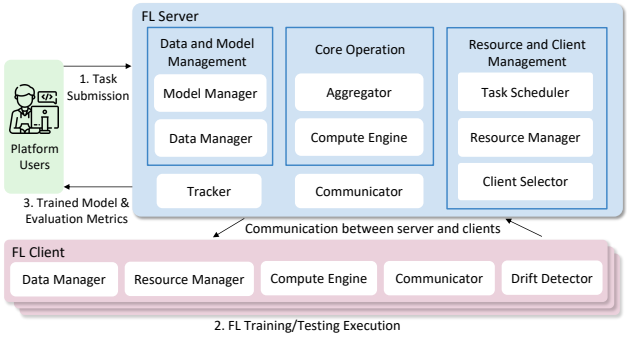


Figure 2. Illustration of COALA platform that enables automated benchmarking for practical FL scenarios.

et al., 2023) or better adaptation strategies (Bao et al., 2023).

3.4. Model Level: Model Configuration

Rather than training a single full model as in typical FL, COALA provides comprehensive support for different model configurations. This includes splitting the full model or training different models for different clients.

Single Model. The majority of FL studies consider training a single model, where the model architecture is the same among clients and the participating clients start training from the same model parameters. We term it as training the *full model*. In contrast, some FL works only transmit and communicate a *partial model* among clients and the server. For example, several studies only communicate the backbone while keeping the classifier locally in clients (Zhuang et al., 2020; Dong et al., 2022). COALA provides benchmarks for both the full model and partial model. To further support the latest development of foundation models, we further support federated parameter-efficient fine-tuning (PEFT) (Hu et al., 2022a) of foundation models.

Multiple Models. In addition to a single model, COALA also supports benchmarking for the configuration of multiple models. We define multiple models as either the model architectures in clients are different or the model parameters with which the clients start training are different. These settings reflect the practical FL scenarios: different model architectures could address system heterogeneity in FL where clients could have varied resources (Diao et al., 2021); different model parameters in clustered FL and personalized FL for varied data distribution in clients (Li & Wang, 2019).

4. COALA Platform: System Design

Existing FL platforms cannot adequately support diverse and practical FL scenarios as discussed in Section 3. To bridge this gap, we present COALA, a new FL platform designed for the latest advancements in the field. COALA automates FL training and evaluation and is scalable and highly customizable. Figure 2 depicts the overall architecture of COALA. We discuss the system workflow and three degrees of customizations it supports in this section and provide details of the system components in Appendix F.

4.1. System Workflow

COALA platform streamlines and automates the FL workflow, providing a seamless process from training task initiation to output delivery.

Task Submission and Initialization: Users commence the FL process by submitting configurations and customized components to COALA platform. Then, the task scheduler works with the resource manager to allocate resources for the server to start a new FL training task.

FL Training and Testing Execution: At the start of training each task, the data manager and model manager load the dataset and model for training, respectively. The server then selects FL clients based on the resource availability of the clients. Then, the system executes the standard FL life cycle: 1) the server distributes the task (configurations and models) to the clients via communicator; 2) the client executes the training/testing and then uploads the results and trained model parameters to the server; 3) the aggregator in the server aggregates these models and obtains a new model for the next round of training.

Output Delivery: The server tracker collects and consolidates the evaluation metrics from the server and clients. The evaluation metrics contain both system metrics (e.g., computation time, communication time, memory consumption, etc.) and algorithmic metrics (e.g., accuracy and loss). These evaluation metrics together with the trained model are delivered to the users at the end of training.

4.2. Three Degrees of Customization

The COALA platform is highly customizable to support a wide range of practical FL scenarios. Users can customize their FL applications and algorithms in three different degrees: configuration degree, component degree, and workflow degree. We summarize the procedure of our customization in Appendix D.

Configuration Customization: The first and easiest customization is through configurations. COALA platform loads a configuration file to set up the initial FL settings. Users can easily modify configurations to use out-of-the-box datasets, models, algorithms, and FL settings (e.g., number of participating clients and number of local epochs). At the data level, users can customize the dataset for training and the way to partition the dataset (e.g., label shifts and domain shifts). At the model level, COALA allows them to customize the model for training with the full model. An example of the configuration is provided in Appendix D.

Component Customization: COALA extends to component customization, supporting users to customize system components and load them to the platform as plugins. This includes customization of data, model, server and client executions, which are not provided by the platform. Leveraging our modular design, users can inherit existing implementations and easily register customized components into the platform via APIs. List in Appendix D offers an illustrative example of customizing client implementation, including training execution, testing execution, and content uploads. This component-level customization facilitates many FL scenarios in Section 3, such as training diverse models at the model level; Users can personalize server distribution and customize aggregation methods like adopting knowledge distillation (Lin et al., 2020) or clustering (Ghosh et al., 2020). COALA then executes the standard FL workflow with these plugins.

Workflow Customization: In addition, COALA allows users to customize FL workflow (i.e., execution logic in both the FL server and the FL client), other than standard FL workflow. COALA platform wraps the server execution logic inside a function in the server. Users can customize the server workflow by inheriting and extending to a new server execution. At the same time, they can still customize components or reuse the out-of-the-box components in the platform. For instance, in the semi-supervised FL scenario, users can design a customized FL workflow where the server initially trains with publicly available data, distributes to clients for unsupervised training, fine-tunes the aggregated model with additional public data post-aggregation, and subsequently distributes the refined model to clients for the next round of training. Such customization empowers users to tailor FL workflows to their unique requirements, opening more possibilities for new application and algorithm development.

5. Benchmark Experiments

In this section, we present representative benchmark results to show how COALA facilitates effective implementation and benchmarking of practical FL scenarios. We conduct the benchmark mostly using FedAvg and more details of experimental settings are provided in Appendix B.

5.1. Task Level Benchmark

Image Classification: We benchmark classification with the standard CIFAR-10/100 datasets and partition it into 100 clients with label distribution shifts using the Dirichlet distribution with the concentration parameter $\alpha = 0.5$ (Dir(0.5)). Table 1 shows that label distribution shift affects the model performance, which aligns with other FL studies.

Object Detection: We provide the benchmark for federated object detection using the BDD100K dataset for the autonomous driving application. We simulate data heterogeneity in two ways: 1) a Dirichlet distribution-based partition strategy on weather attributes (*Dir*); 2) a hierarchical strategy by repeating Dirichlet distribution-based partition on all three attributes (weather, scene, time-of-day) (*H-Dir*). Both result in feature shifts and data quantity imbalance among clients, as visualized in Appendix C. The model we use is the YOLO-V5[1] model. Table 2 illustrates the effective training of object detection models in FL. The *H-Dir* strategy causes greater divergence in data distribution than the *Dir* strategy, thus resulting in lower performance. It is also interesting to see that the feature shift along with data quantity imbalance may not cause a significant performance drop, and could even achieve slightly better performance. This could be because the randomly selected clients contain more data in *Dir* simulation than the IID setting, as data quantity is highly skewed (we randomly select 4 out of 10 clients and 12 out of 100 clients each for these experiments). Deeper investigations could present potential opportunities for future research.

Table 1. Benchmark of federated image classification task.

Datasets	# Client	Heterogeneity	Acc. (%)
CIFAR10	100	IID	90.90
CIFAR10	100	Dir(0.5)	84.08
CIFAR100	100	IID	63.52
CIFAR100	100	Dir(0.5)	56.80
Digits-5	5	Feature shift	94.14
Office-Caltech	4	Feature shift	84.09
DomainNet	6	Feature shift	66.47

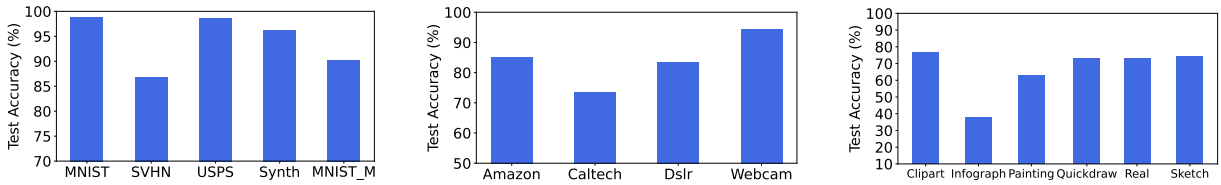
Table 2. Benchmark of federated object detection on BDD100K.

# client	Attributes	Precision	Recall	mAP@0.5
10	IID	61.06	32.68	34.45
10	Dir(0.5)	59.41	32.71	33.43
10	H-Dir(0.5)	58.91	29.89	31.36
100	IID	58.84	29.24	30.71
100	Dir(0.5)	58.64	30.61	32.07
100	H-Dir(0.5)	58.75	29.82	31.42

Semantic Segmentation: We train the semantic segmentation model DeepLabV3 on the PASCAL VOC with pretrained MobileNet-V2 and ResNet-50 as backbones, respectively. We simulate 10 FL clients and randomly select 4 in each training round. We use the first category of semantic objects in each image as the label for the label distribution shifts simulation. Figure 4 indicates that the label shift problem severely degrades the performance using both the MobileNet-V2 and ResNet-50. These benchmark results suggest that non-IID is also a significant issue in segmentation, which could be opportunities for robust algorithms and solutions to address this challenge.

Pose Estimation: COALA provides benchmark results for pose estimation with the MPII dataset on three different

[1]https://github.com/ultralytics/yolov5



(a) Digits-Five (b) Office-Caltech (c) DomainNet

Figure 3. Domain-wise test accuracy of the global model. The same model does not perform equivalently well on different domains.

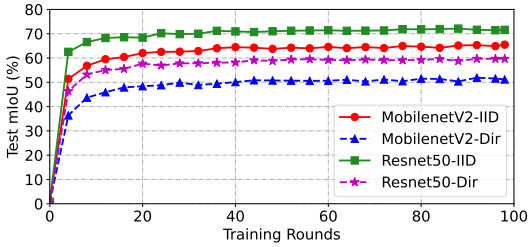


Figure 4. Benchmark of federated semantic segmentation on VOC.

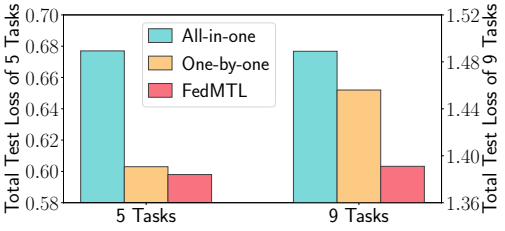


Figure 5. Federated multiple task learning with 5 tasks and 9 tasks.

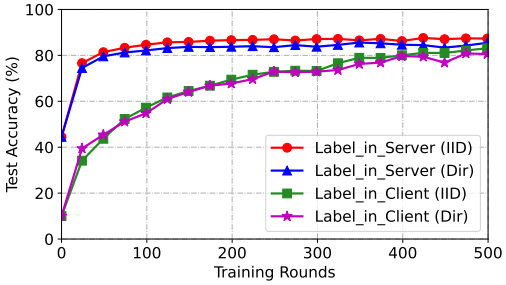


Figure 6. Benchmark results of semi-supervised FL.

Table 3. Benchmark of federated pose estimation on MPII dataset.

Backbone	ResNet-34	ResNet-50	ResNet-101
Head	96.08	96.39	96.17
Shoulder	94.56	94.48	94.65
Elbow	86.84	87.66	87.63
Wrist	80.83	82.15	82.75
Hip	87.46	86.95	87.66
Knee	81.58	82.03	82.39
Ankle	77.49	77.07	77.33
Mean	87.04	87.30	87.57

sizes of ResNet backbones. Since the human has multiple joints to be estimated, we focus on the Mean performance for the Percentage of Correct Keypoints (PCK@0.5). Table 3 demonstrates that the pose estimation model achieves competitive performance in FL. However, it is interesting to find that a larger backbone does not have a significant impact on performance.

Federated Multiple-Task Learning: We evaluate federated multiple-task learning via two sets of experiments with 5 and 9 different vision tasks, respectively. Two simple solutions to deal with multiple tasks are the *all-in-one* method that uses a shared encoder for all tasks and the *one-by-one* method that trains tasks sequentially. We benchmark them with FedMTL method (Zhuang et al., 2023c), which trains with all tasks combined first and then divides the tasks into groups for further training according to the task affinity. Figure 5 shows that FedMTL not only can leverage the task-agnostic knowledge but also encourage collaboration between tasks with higher affinity to improve overall performance. We believe this sets a good foundation for future works to further improve upon in this nascent field.

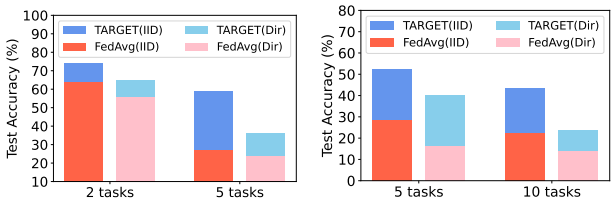
5.2. Data Level Benchmark

Feature Shift: We provide benchmark results for feature shift with multi-domain datasets, including Digits-5, Office-Caltech, and DomainNet. Each client exclusively contains samples from a specific domain. We follow the prior works (Li et al., 2020b) and simulate the number of clients to be equal to the number of domains of each dataset. Table 1 shows that the FedAvg algorithm can already achieve a certain level of performance even in the presence of feature shifts, but there exist some performance gaps for different domains, as shown in Figure 3. The possible factors for this phenomenon could include the varied sample sizes, the underlying difficulty in discriminating objects in each domain, and the fairness of the learning algorithm. This prompts further research into developing fairer global models or more suitable personalized models.

Semi-supervised Learning: We benchmark both the label-in-server and the label-in-client scenarios for federated semi-supervised learning. The baseline methods are SemiFL (Diao et al., 2022) and FedAvg with FixMatch. We use the CIFAR-10 dataset for evaluation and the amount of labeled samples for each class is set as 400 for the label-in-server scenarios. For the label-in-client scenarios, we choose 10 labeled samples per class for each client. The

Table 4. Benchmark results of federated self-supervised learning based on BYOL method.

Dataset	Method	# client	IID	Dir(0.5)
CIFAR10	FedBOYL	4	83.84	82.63
	FedBOYL	20	75.31	64.39
CIFAR100	FedBOYL	4	48.65	47.19
	FedBOYL	20	39.48	38.69



(a) CIFAR10 (b) CIFAR100

Figure 7. Results of class-incremental federated learning.

remaining unlabeled samples among clients are partitioned either by random or Dirichlet-based allocation. Figure 6 shows that the label-in-server setting achieves better performance than the label-in-client setting as the global model fine-tuned by the server-side data is more unbiased and the pseudo-labels could be more reliable.

Self-supervised Learning: We use the CIFAR-10/100 datasets to benchmark federated self-supervised learning with BOYL algorithm (FedBOYL) (Zhuang et al., 2021a). For the performance evaluation, we adopt the kNN evaluation on the outputs generated by the backbones. Table 4 demonstrates that self-supervised representation learning is promising even in the FL settings.

Continual learning: We benchmark federated continual learning with different numbers of incremental tasks on CIFAR-10/100 datasets, where each task contains a distinct set of classes. Both IID and non-IID label distribution are considered. We benchmark both FedAvg and TARGET (Zhang et al., 2023b), which leverages both real data of the current task and synthetic data generated by investigating historical global model. The performance metric is the average accuracy of all tasks. Figure 7 shows that synthetic data is effective in mitigating catastrophic forgetting issues.

Test-time Adaptation: We consider the covariate shift during the test stage. Following (Tan et al., 2023), we simulate the covariate shift by adding corruptions to the raw test images. We compare the vanilla FedAvg without any adaptation and FedICON (Tan et al., 2023) that applies the contrastive loss during feature representation learning and also conducts unsupervised model adaptation during the test stage. It can be seen from Table 5 that in the multi-domain settings, applying contrastive loss for feature learning and local personalized classifier head for decision making can lead to performance improvement even without any data shift. FedICON can generally perform better than vanilla FedAvg in the presence of data distribution shifts.

Table 5. Benchmark results (%) of test-time adaptation after FL.

Dataset	Shift Type	FedAvg	FedICON
CIFAR10 (IID)	Original	92.45	93.01
CIFAR10 (IID)	Covariate	59.41	64.07
CIFAR10 (Dir)	Original	89.22	92.38
CIFAR10 (Dir)	Covariate	53.43	76.62
Digits-Five	Original	94.16	95.21
Digits-Five	Covariate	89.63	95.27
Office-Caltech	Original	84.09	90.48
Office-Caltech	Covariate	65.02	76.27

Table 6. Benchmark results (%) of federated split learning.

Dataset	Method	# client	IID	Dir(0.5)
CIFAR10	MocoSFL	100	79.23	79.45
	MocoSFL	1000	69.75	69.40
CIFAR100	MocoSFL	100	43.29	43.30
	MocoSFL	1000	35.34	37.39

Table 7. Test accuracy (%) of clustered and personalized FL.

Datasets	FedAvg	FedAvg-FT	IFCA	FedRep
CIFAR10	90.17	91.16	90.51	85.20
Digits-5	94.74	94.97	95.60	94.14

Table 8. Accuracy (%) and percentage of trainable parameters (% Params.) of fine-tuning ViT-B in FL.

Dataset	Fed-Linear		Fed-LoRA	
	Acc.	% Params.	Acc.	% Params.
CIFAR10 (IID)	95.36	0.0090	97.89	0.0519
CIFAR10 (Dir)	95.26	0.0090	97.89	0.0519
CIFAR100 (IID)	81.30	0.0895	90.57	0.1324
CIFAR100 (Dir)	80.50	0.0895	89.61	0.1324
DomainNet	88.45	0.0090	89.17	0.0519

5.3. Model Level Benchmark

Federated Split Learning: To evaluate federated split learning, we employ the self-supervised MocoSFL (Li et al., 2023) on CIFAR-10/100 datasets. We provide benchmark results with two relatively large-scale setups, 100 and 1000 clients. Table 6 shows that split training is capable of achieving high performance regardless of the data heterogeneity.

Multiple Models with Clustered/Personalized FL: We benchmark multiple models using the popular IFCA (Ghosh et al., 2020) and FedRep (Collins et al., 2021), as well as FedAvg. We simulate 30 clients on two datasets: Digits-5 dataset (with feature shifts) and CIFAR-10 dataset (with label shifts). The number of clusters is set to 5 by default. Table 7 demonstrates that multiple models can improve local performance than a single global model in the presence of data heterogeneity. In particular, IFCA may be more suitable for feature shift while local fine-tuning of the global model,

e.g., FedAvg-FT, is promising in both settings.

Federated Parameter-Efficient Fine-Tuning: Our platform supports federated fine-tuning of foundation models. We compare a parameter-efficient fine-tuning (PEFT) method called LoRA (Hu et al., 2022a) (Fed-LoRA) with simply training a linear classifier head (Fed-Linear) on ViT-B/16 model (Dosovitskiy et al., 2021) in FL. We set the rank of LoRA as r=1 and evaluate both fine-tuning methods with learning rates 5e-3 and 50 communication rounds. Table 8 shows that ViT is robust to data heterogeneity, which demonstrates the great potential of leveraging foundation models with PEFT techniques in FL.

6. Conclusion

In this paper, we present COALA, a practical and vision-centric FL platform. It is highly customizable and automates benchmarking of a wide range of practical FL scenarios from multiple levels such as task, data, and model. For the task level, COALA supports federated learning of 15 vision tasks and federated multiple-task learning. For the data level, COALA covers heterogeneous feature and label distributions, continually changing data distribution, and learning with different degrees of annotation availability. For the model level, COALA facilities federated split learning and federated learning of models with different architectures or parameters. The comprehensive benchmarking experiments under a wide range of FL scenarios validate the great potential of our proposed COALA. We hope that COALA sets a good foundation and will be useful to accelerate further advancements and landing of FL in various industry scenarios.

Impact Statement

This work aims to largely advance the research and applications of federated learning in the area of computer vision. The developed platform is possible to help integrate cutting-edge research progress in FL while accelerating the landing of large-scale FL in the industry.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

Alam, S., Zhang, T., Feng, T., Shen, H., Cao, Z., Zhao, D., Ko, J., Somasundaram, K., Narayanan, S. S., Avestimehr, S., et al. Fedaiot: A federated learning benchmark for artificial intelligence of things. *arXiv preprint arXiv:2310.00109*, 2023.

Andriluka, M., Pishchulin, L., Gehler, P., and Schiele, B. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

Bao, W., Wei, T., Wang, H., and He, J. Adaptive test-time personalization for federated learning. *arXiv preprint arXiv:2310.18816*, 2023.

Beutel, D. J., Topal, T., Mathur, A., Qiu, X., Parcollet, T., and Lane, N. D. Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390*, 2020.

Bhuyan, N., Moharir, S., and Joshi, G. Multi-model federated learning with provable guarantees. In *EAI International Conference on Performance Evaluation Methodologies and Tools*, pp. 207–222. Springer, 2022.

Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečný, J., Mazzocchi, S., McMahan, B., Van Overveldt, T., Petrou, D., Ramage, D., and Roselander, J. Towards federated learning at scale: System design. In Talwalkar, A., Smith, V., and Zaharia, M. (eds.), *Proceedings of Machine Learning and Systems*. 2019.

Cai, D., Wu, Y., Wang, S., Lin, F. X., and Xu, M. Efficient federated learning for modern NLP. pp. 37:1–37:16. ACM, 2023.

Caldas, S., Duddu, S. M. K., Wu, P., Li, T., Konečnỳ, J., McMahan, H. B., Smith, V., and Talwalkar, A. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

Chen, D., Gao, D., Kuang, W., Li, Y., and Ding, B. pFL-bench: A comprehensive benchmark for personalized federated learning. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=2ptbv_JjYKA.

Chen, D., Gao, D., Xie, Y., Pan, X., Li, Z., Li, Y., Ding, B., and Zhou, J. FS-REAL: towards real-world cross-device federated learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, pp. 3829–3841. ACM, 2023a.

Chen, D., Tan, V. J., Lu, Z., Wu, E., and Hu, J. Openfed: A comprehensive and versatile open-source federated learning framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5017–5025, 2023b.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. Encoder-decoder with atrous separable convolution

for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018.

Collins, L., Hassani, H., Mokhtari, A., and Shakkottai, S. Exploiting shared representations for personalized federated learning. In *International conference on machine learning*, pp. 2089–2099. PMLR, 2021.

Diao, E., Ding, J., and Tarokh, V. Heterofl: Computation and communication efficient federated learning for heterogeneous clients. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

Diao, E., Ding, J., and Tarokh, V. Semifl: Semi-supervised federated learning for unlabeled clients with alternate training. In *NeurIPS*, 2022.

Dong, X., Zhang, S. Q., Li, A., and Kung, H. Spherefed: Hyperspherical federated learning. In *European Conference on Computer Vision*, pp. 165–184. Springer, 2022.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

du Terrail, J. O., Ayed, S.-S., Cyffers, E., Grimberg, F., He, C., Loeb, R., Mangold, P., Marchand, T., Marfoq, O., Mushtaq, E., Muzellec, B., Philippenko, C., Silva, S., Teleńczuk, M., Albarqouni, S., Avestimehr, S., Bellet, A., Dieuleveut, A., Jaggi, M., Karimireddy, S. P., Lorenzi, M., Neglia, G., Tommasi, M., and Andreux, M. FLamby: Datasets and benchmarks for cross-silo federated learning in realistic healthcare settings. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=GgM5DiAb6A2.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88: 303–338, 2010.

Feng, T., Bose, D., Zhang, T., Hebbar, R., Ramakrishna, A., Gupta, R., Zhang, M., Avestimehr, S., and Narayanan, S. Fedmultimodal: A benchmark for multimodal federated learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, pp. 4035–4045. ACM, 2023.

Garcia, M. H., Manoel, A., Diaz, D. M., Mireshghallah, F., Sim, R., and Dimitriadis, D. Flute: A scalable, extensible framework for high-performance federated learning simulations. *arXiv preprint arXiv:2203.13789*, 2022.

Ghosh, A., Chung, J., Yin, D., and Ramchandran, K. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33: 19586–19597, 2020.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

He, C., Li, S., So, J., Zhang, M., Wang, H., Wang, X., Vepakomma, P., Singh, A., Qiu, H., Shen, L., et al. Fedml: A research library and benchmark for federated machine learning. *arXiv preprint arXiv:2007.13518*, 2020.

He, C., Shah, A. D., Tang, Z., Fan, D., Sivashunmugam, A. N., Bhogaraju, K., Shimpi, M., Shen, L., Chu, X., Soltanolkotabi, M., and Avestimehr, S. Fedcv: A federated learning framework for diverse computer vision tasks. *CoRR*, abs/2111.11066, 2021. URL https://arxiv.org/abs/2111.11066.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022a.

Hu, S., Li, Y., Liu, X., Li, Q., Wu, Z., and He, B. The oarf benchmark suite: Characterization and implications for federated learning systems. *arXiv preprint arXiv:2006.07856*, 2020.

Hu, S., Li, Y., Liu, X., Li, Q., Wu, Z., and He, B. The oarf benchmark suite: Characterization and implications for federated learning systems. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(4):1–32, 2022b.

Huba, D., Nguyen, J., Malik, K., Zhu, R., Rabbat, M., Yousefpour, A., Wu, C.-J., Zhan, H., Ustinov, P., Srinivas, H., et al. Papaya: Practical, private, and scalable federated learning. *Proceedings of Machine Learning and Systems*, 4:814–832, 2022.

Jeong, W., Yoon, J., Yang, E., and Hwang, S. J. Federated semi-supervised learning with inter-client consistency & disjoint learning. In *9th International Conference on Learning Representations, ICLR*. OpenReview.net, 2021.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

Kim, T., Lin, E., Lee, J., Lau, C., and Mugunthan, V. Navigating data heterogeneity in federated learning: A semi-supervised approach for object detection. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=2D7ou48q0E.

Lai, F., Dai, Y., Singapuram, S. S. V., Liu, J., Zhu, X., Madhyastha, H. V., and Chowdhury, M. Fedscale: Benchmarking model and system performance of federated learning at scale. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, 2022.

Li, D. and Wang, J. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

Li, J., Lyu, L., Iso, D., Chakrabarti, C., and Spranger, M. Mocosfl: enabling cross-client collaborative self-supervised learning. In *The Eleventh International Conference on Learning Representations, ICLR*. OpenReview.net, 2023.

Li, Q., Diao, Y., Chen, Q., and He, B. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 965–978. IEEE, 2022.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems 2020*, pp. 429–450. 2020a.

Li, X., JIANG, M., Zhang, X., Kamp, M., and Dou, Q. Fedbn: Federated learning on non-iid features via local batch normalization. In *International Conference on Learning Representations*, 2020b.

Liang, X., Lin, Y., Fu, H., Zhu, L., and Li, X. Rscfed: Random sampling consensus federated semi-supervised learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 2022.

Lin, B. Y., He, C., Ze, Z., Wang, H., Hua, Y., Dupuy, C., Gupta, R., Soltanolkotabi, M., Ren, X., and Avestimehr, S. Fednlp: Benchmarking federated learning methods for natural language processing tasks. In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, 2022.

Lin, T., Kong, L., Stich, S. U., and Jaggi, M. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33: 2351–2363, 2020.

Liu, C., Wang, C., Chien, S., and Lai, S. Fedfr: Joint optimization federated framework for generic and personalized face recognition. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI*, 2022.

Liu, Y., Huang, A., Luo, Y., Huang, H., Liu, Y., Chen, Y., Feng, L., Chen, T., Yu, H., and Yang, Q. Fedvision: An online visual object detection platform powered by federated learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 13172–13179, 2020.

Luo, S., Xiao, Y., and Song, L. Personalized federated recommendation via joint representation learning, user clustering, and model adaptation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, pp. 4289–4293. ACM, 2022.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pp. 1273–1282. PMLR, 2017.

Meng, Q., Zhou, F., Ren, H., Feng, T., Liu, G., and Lin, Y. Improving federated learning face recognition via privacy-agnostic clusters. In *The Tenth International Conference on Learning Representations, ICLR*. OpenReview.net, 2022.

Miao, J., Yang, Z., Fan, L., and Yang, Y. Fedseg: Class-heterogeneous federated learning for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8042–8052, 2023.

Nguyen, D. C., Pham, Q.-V., Pathirana, P. N., Ding, M., Seneviratne, A., Lin, Z., Dobre, O., and Hwang, W.-J. Federated learning for smart healthcare: A survey. *ACM Computing Surveys (CSUR)*, 55(3):1–37, 2022.

Niu, Y. and Deng, W. Federated learning for face recognition with gradient correction. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI*, 2022.

PaddlePaddle. Paddlefl vanilla implementation, 2020. URL https://github.com/PaddlePaddle/PaddleFL/tree/master/python/paddle_fl/paddle_fl/examples/femnist_demo.

Qin, Z., Deng, S., Zhao, M., and Yan, X. Fedapen: Personalized cross-silo federated learning with adaptability to statistical heterogeneity. In *Proceedings of the 29th ACM*

SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 1954–1964, 2023.

Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečnỳ, J., Kumar, S., and McMahan, H. B. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020a.

Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečnỳ, J., Kumar, S., and McMahan, H. B. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020b.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

Ryffel, T., Trask, A., Dahl, M., Wagner, B., Mancuso, J., Rueckert, D., and Passerat-Palmbach, J. A generic framework for privacy preserving deep learning. *arXiv preprint arXiv:1811.04017*, 2018.

Tan, Y., Chen, C., Zhuang, W., Dong, X., Lyu, L., and Long, G. Taming heterogeneity to deal with test-time shift in federated learning. In *International Workshop on Federated Learning for Distributed Data Mining*, 2023.

Tensorflow.org. Tensorflow federated, 2019. URL https://github.com/tensorflow/federated.

Thapa, C., Arachchige, P. C. M., Camtepe, S., and Sun, L. Splitfed: When federated learning meets split learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8485–8493, 2022.

Toshev, A. and Szegedy, C. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1653–1660, 2014.

Wang, C., Deng, J., Meng, X., Wang, Y., Li, J., Lin, S., Han, S., Miao, F., Rajasekaran, S., and Ding, C. A secure and efficient federated learning framework for NLP. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pp. 7676–7682. Association for Computational Linguistics, 2021.

Wang, E., Chen, B., Chowdhury, M., Kannan, A., and Liang, F. Flint: A platform for federated learning integration. *Proceedings of Machine Learning and Systems*, 5, 2023a.

Wang, Z., Fan, X., Peng, Z., Li, X., Yang, Z., Feng, M., Yang, Z., Liu, X., and Wang, C. Flgo: A fully customizable federated learning platform. *arXiv preprint arXiv:2306.12079*, 2023b.

WeBank. Federated ai technology enabler (fate), 2019. URL https://github.com/FederatedAI/FATE.

Woisetschläger, H., Isenko, A., Wang, S., Mayer, R., and Jacobsen, H. A survey on efficient federated learning methods for foundation model training. *CoRR*, abs/2401.04472, 2024.

Xie, Y., Wang, Z., Gao, D., Chen, D., Yao, L., Kuang, W., Li, Y., Ding, B., and Zhou, J. Federatedscope: A flexible federated learning platform for heterogeneity. *Proc. VLDB Endow.*, 16(5):1059–1072, 2023.

Yang, M., Xu, J., Ding, W., and Liu, Y. Fedhap: Federated hashing with global prototypes for cross-silo retrieval. *IEEE Transactions on Parallel and Distributed Systems*, 2023.

Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., and Darrell, T. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2636–2645, 2020.

Zamir, A. R., Sax, A., Shen, W., Guibas, L. J., Malik, J., and Savarese, S. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3712–3722, 2018.

Zeng, D., Liang, S., Hu, X., Wang, H., and Xu, Z. Fedlab: A flexible federated learning framework. *J. Mach. Learn. Res.*, 24:100:1–100:7, 2023.

Zhang, C., Long, G., Zhou, T., Yan, P., Zhang, Z., Zhang, C., and Yang, B. Dual personalization on federated recommendation. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, 2023a.

Zhang, J., Chen, C., Zhuang, W., and Lyu, L. Target: Federated class-continual learning via exemplar-free distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4782–4793, 2023b.

Zhang, Q., Wu, T., Zhou, P., Zhou, S., Yang, Y., and Jin, X. Felicitas: Federated learning in distributed cross device collaborative frameworks. In Zhang, A. and Rangwala, H. (eds.), *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pp. 4502–4509. ACM, 2022.

Zhang, T., Feng, T., Alam, S., Lee, S., Zhang, M., Narayanan, S. S., and Avestimehr, S. Fedaudio: A federated learning benchmark for audio tasks. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023c.

Zhang, Z., Yang, Y., Yao, Z., Yan, Y., Gonzalez, J. E., Ramchandran, K., and Mahoney, M. W. Improving semi-supervised federated learning by reducing the gradient diversity of models. In *2021 IEEE International Conference on Big Data (Big Data)*, pp. 1214–1225. IEEE, 2021.

Zhang, Z., Yang, Y., Dai, Y., Wang, Q., Yu, Y., Qu, L., and Xu, Z. Fedpetuning: When federated learning meets the parameter-efficient tuning methods of pre-trained language models. In Rogers, A., Boyd-Graber, J. L., and Okazaki, N. (eds.), *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 9963–9977. Association for Computational Linguistics, 2023d.

Zhuang, W. and Lyu, L. Is normalization indispensable for multi-domain federated learning? *arXiv preprint arXiv:2306.05879*, 2023.

Zhuang, W. and Lyu, L. Fedwon: Triumphing multi-domain federated learning without normalization. In *The Twelfth International Conference on Learning Representations, ICLR*, 2024. URL https://openreview.net/forum?id=hAYHmV1gM8.

Zhuang, W., Wen, Y., Zhang, X., Gan, X., Yin, D., Zhou, D., Zhang, S., and Yi, S. Performance optimization of federated person re-identification via benchmark analysis. In *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 955–963, 2020.

Zhuang, W., Gan, X., Wen, Y., Zhang, S., and Yi, S. Collaborative unsupervised visual representation learning from decentralized data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4912–4921, 2021a.

Zhuang, W., Wen, Y., and Zhang, S. Joint optimization in edge-cloud continuum for federated unsupervised person re-identification. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 433–441, 2021b.

Zhuang, W., Gan, X., Wen, Y., and Zhang, S. Easyfl: A low-code federated learning platform for dummies. *IEEE Internet Things J.*, 9(15):13740–13754, 2022a.

Zhuang, W., Gan, X., Zhang, X., Wen, Y., Zhang, S., and Yi, S. Federated unsupervised domain adaptation for face recognition. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE, 2022b.

Zhuang, W., Wen, Y., and Zhang, S. Divergence-aware federated self-supervised learning. In *The Tenth International Conference on Learning Representations, ICLR*. OpenReview.net, 2022c.

Zhuang, W., Chen, C., and Lyu, L. When foundation model meets federated learning: Motivations, challenges, and future directions. *CoRR*, abs/2306.15546, 2023a. URL https://doi.org/10.48550/arXiv.2306.15546.

Zhuang, W., Gan, X., Wen, Y., and Zhang, S. Optimizing performance of federated person re-identification: Benchmarking and analysis. *ACM Transactions on Multimedia Computing, Communications and Applications*, 19(1s): 1–18, 2023b.

Zhuang, W., Wen, Y., Lyu, L., and Zhang, S. Mas: Towards resource-efficient federated multiple-task learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23414–23424, 2023c.

A. Introduction of Representative Datasets

Digits-Five (Li et al., 2020b) is an integrated dataset consisting of five digits recognition sub-sets, including MNIST, MNIST-M, Synthetic Digits, SVHN, and USPS, each of which has a special style of digits from 0 to 9.

Office-Caltech-10 (Li et al., 2020b) consists of 10-category data samples from four different sources, including the Amazon merchant website, the Caltech-101 dataset, a high-resolution DSLR camera, and a webcam. Due to distinct feature distributions, each data source can represent a specific domain. There is no pre-split of training and test parts for office-caltech, so we randomly split the samples in each domain, of which 70% are selected as training samples and the remainder as test samples.

DomainNet (Li et al., 2020b) is a popular multi-domain dataset containing images from six distinct domains, including Clipart, Infograph, Painting, Quickdraw, Real, and Sketch. It has 345 object categories and we choose the top ten most common classes for our experiments as in (Li et al., 2020b).

BDD100K (Yu et al., 2020) is a large-scale, diverse driving video database designed for research on autonomous vehicles and computer vision tasks. In this platform, we mainly make use the images for object detection. It covers a wide range of driving scenarios including weather conditions and urban environments. A variety of object categories, including pedestrians, cyclists, vehicles, etc, are annotated for evaluation.

Pascal VOC (Everingham et al., 2010) (Visual Object Classes) dataset is a widely used benchmark in computer vision for object detection and image segmentation tasks. It covers common categories such as people, animals, vehicles, and household items.

MPII (Andriluka et al., 2014). MPII Human Pose dataset is a popular benchmark for articulated human pose estimation. The dataset includes around 25K images containing over 40K people with annotated body joints. The images were systematically collected using an established taxonomy of every day human activities.

Table 9. Currently tasks, datasets and scenarios supported by our platform. 'Sup' means 'Supervised'. **For each task, the dataset list can be easily extended by integrating new datasets**.

Task	Dataset	Split	Annotation
Image Classification	CIFAR10/100, FEMNIST, Landmarks	IID, Dir, # of classes	Sup, Semi-Sup, Self-sup
	Digits-Five, Office-Caltech, DomainNet	Multi-Domain	Sup
Object Detection	BDD100K	IID, Dir, H-Dir	Sup, Semi-Sup
Semantic Segmentation	Pasacal VOC	IID, Dir, # of classes	Sup
Pose Estimation	MPII	IID	Sup
Person Re-identification	MSMT17, Market-1501, CUHK01, CUHK03-NP	Multi-Domain	Sup
	PRID2011, VIPeR, 3DPeS, iLIDS-VID		
Face Recognition	BUPT-Balancedface, MS-Celeb-1M, WebFace, RFW	Multi-Domain	Sup
3D-Point Cloud	ModelNet40	IID	Sup
Multiple Tasks *	Taskonomy	IID, Non-IID	Sup

* Multiple tasks training includes semantic segmentation, depth estimation, surface normal, keypoint, edge texture, edge occlusion, reshaping, principle curvature, auto-encoder

B. Details of Experimental Settings

Data Split. In our benchmark experiments, we use the default training-test data split supported in the platform as described in Table 10. Basically, we make use of all the available training data, except the Digits5 and DomainNet, where we make some sampling to select only an identical amount of training data across different domains as in (Li et al., 2020b).

Model Selection. We provide some predefined model architectures for each set of experiments. For example, simple CNN for Digits-5, ResNet-18 for CIFAR-10/100, AlexNet for Office-Caltech and DomainNet, YOLO-V5 series for Object Detection in BDD100K, etc. For most experiments, we train the global model from scratch as in conventional FL studies.

For semantic segmentation and pose estimation, we utilize the pre-trained backbones.

Training Hype-parameters. For local training, SGD is selected as the default local optimizer with mini-batch size 32, learning rate 0.01, momentum 0.9, weight decay 0.0005 and local epoch $E = 5$ unless otherwise mentioned. The number of communication rounds is set to 100 for three multi-domain datasets and 200 for other datasets. In particular, we extend the communication rounds in semi-supervised training to 500 as it requires more epochs for convergence. For parameter-efficient fine-tuning of foundation models, we set the local epoch $E = 1$ as ViT is more computational intensive and the number of communication rounds as 50 as it can converge much faster than training from scratch of other CNN-based models. For all experiments, we report the best test metrics achieved for performance evaluation.

System Scale. For different tasks, the numbers of clients and participating rate are adjusted according to the total training data size. For Digits5, Office-Caltech and DomainNet, by default we allocate data from each specific domain to one client for simplicity. For CIFAR-10/100, we consider 100 clients with 10% participation rate in supervised/semi-supervised learning and 4/20 clients with full participation for self-supervised learning as contrastive learning methods usually need large batch size. All experiments are run on a AWS cloud server equipped with four V100 GPUs.

Table 10. Dataset names, training and test sample sizes and used models for benchmark evaluation.

Datasets	Training Size	Test Size	Models	Metric
CIFAR10/100	50,000	10,000	ResNet-18	Accuracy
Digits-5	37,190	14,376	CNN	Accuracy
Office-Caltech	1,771	776	AlexNet	Accuracy
DomainNet	6,000	4,573	AlexNet	Accuracy
BDD100K	60,000	30,000	YOLOv5n	mAP@0.5
Pascal VOC	1,464	1,449	DeepLabV3	mIoU
MPII	22,246	2,958	PoseResNet	PCK@0.5

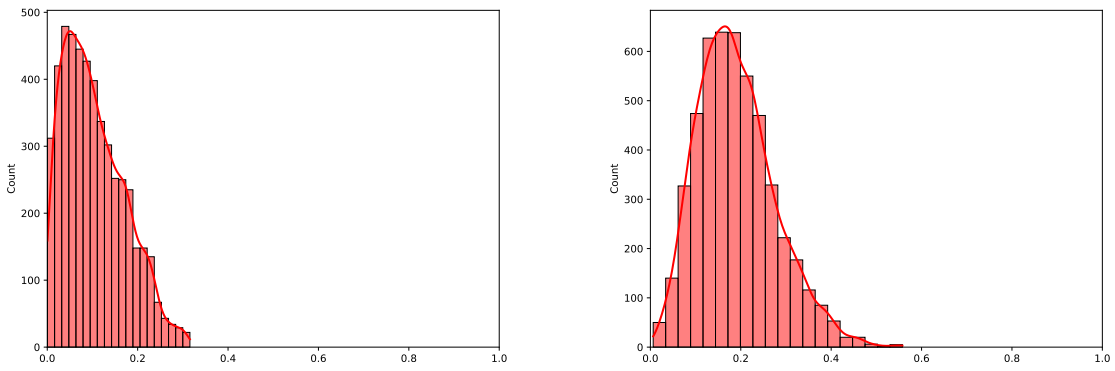
C. Attribute Distributions of BDD100K under Different Splits

Since it is difficult to measure the label distribution skew for object detection datasets as each image will contain multiple objects belonging to different kinds of categories, we turn to measure the feature distribution shift across clients. BDD100K is a dataset that provides rich information about the collected images, including three attributes, i.e., weather, scene, time-of-day. It contains diverse scene types including city streets, residential areas, and highways, and diverse weather conditions at different times of the day. Therefore, those attributes can be utilized to simulate feature distribution shift by manipulating the percentage of images belonging to each specific attribute, just similar to the label-based dataset split.

It is worth noting that each image will have all those three attributes, which means the distributions of those attributes are not independent. The first way is to choose one attribute as the main attribute (e.g., weather) for data split, where label-based split methods can be directly applied, such as IID allocation and Dirichlet-based allocation (Dir). The second way is to utilize all three attributes in a hierarchical split manner, which we call hierarchical Dirichlet-based allocation (H-Dir). More precisely, we first select an attribute as the main attribute and for each category in this attribute, then we apply the Dirichlet-based allocation to get the sample proportions across clients. Next, for each client and for the samples of each specific category in the main attribute, we repeat the Dirichlet-based allocation based on the second and third attributes. For each attribute triplet (weather, scene, time-of-day), we make the random sampling to allocate images to each client. Considering the category imbalance issue, we will use a Dirichlet concentration parameter proportional to its prior for each category in each attribute. Moreover, the calculated data amount under each triplet of attributes for each client may exceed the total amount of images belonging to that attribute triplet, where we just set as the amount of available images. By default, we use the weather as the main attribute.

In the following, we make some visualizations for training data distributions. Figure 8 presents the statistics of pairwise Jensen–Shannon divergence that measures the difference in attribute distributions across different clients. From which we can find that the hierarchical Dirichlet-based allocation will indeed result in more divergent feature distributions across clients. We also visualize the amount of samples allocated to each client that belong to different categories in each attribute in Figure 9. It can be found that compared with the Dirichlet-based allocation on only one attribute, the hierarchical Dirichlet-based allocation considering all three attributes can improve the imbalance level across clients for each attribute

and each category.



(a) Dir(0.5) (b) H-Dir(0.5)

Figure 8. Pairwise Jensen–Shannon divergence of attribute distributions across clients. (a) Divergence under Dir(0.5) split; (b) Divergence under H-Dir(0.5) split. The hierarchical Dirichlet-based split will result in more divergent data distributions across clients.

D. Basic Steps for Customizing FL Applications

With the help of COALA, it is easy to construct a new FL task with customized datasets and models. Here, we provide some general steps for customizing new FL applications. Note that not all steps are always required. For example, if the users want to only customize the model while reusing the datasets provided in COALA, they can skip the first step of customizing the dataset. The main steps are listed as follows.

Step 1: Customize Dataset.

The preparation of training data is the first step for FL applications as different vision tasks may contain various kinds of data and annotation formats. Different data transformations and split strategies should also be considered. In the platform, we provide some basic implementations for incorporating Tensor Data, Image Data, and Torch style dataloader. For frequently used datasets in research, we directly provide the functions to load them without extra effort. In addition, we provide many examples that users can refer to for customizing the datasets easier and faster. Users can register their customized dataset into the COALA with a simple `register_dataset` API.

```
import coala
# train_data, test_data are customized datasets provided the user
coala.register_dataset(train_data, test_data)
```

Step 2: Customize Model.

We have provided many models on COALA, while users can also customize models for their specific applications or algorithms. They can register the models through `register_model` API:

```
import coala
from model import new_model
# new_model is customized by the user
coala.register_model(new_model)
```

Step 3: Customize Client and Server Executions.

COALA allows users to customize client and server executions by customizing components or functions in the client and server by writing customized classes. They can just customize the parts they want and reuse the other parts of the system. For example, if users want to change the loss function, they can simply have a CustomizedClient that implements `load_loss_fn` with the new loss function for training. The following is the skeleton of a small subset of components for

Client customization.

```
import coala
from coala.client import BaseClient

class CustomizedClient(BaseClient):
   def __init__(*args, **kwargs):
      # initialization

   # ... many more functions ...

   def load_loss_fn(self, conf):
      # customized loss function implementation

   def train(self, conf):
      # customized client training process implementation

   def test(self, conf):
      # customized client evaluation process implementation

   def construct_upload_request(self):
      # customized content to upload to the server implementation

   # ... many more functions ...

coala.register_client(CustomizedClient)
```

```
import coala
from coala.server import BaseServer

class CustomizedServer(BaseServer):
   def __init__(*args, **kwargs):
      # initialization

   # ... many more functions ...

   def aggregation(self):
      # customized aggregation strategy implementation

   # ... many more functions ...

coala.register_server(CustomizedServer)
```

Step 4: Customize Configurations and Start Training.

The configuration includes the FL training system setup, such as the number of clients, local training epochs, the data splitting method, and training hype parameters, such as local optimizer, learning rate, and local epochs. Users can simply customize the configurations without the prior steps of customizations. These configuration details can be included in a "yaml" file or directly defined as a dictionary in a python file, which will be merged into the default configurations.

```
config = {
     "data": {
        "dataset": "domainnet",
        "split_type": "iid",
        "num_of_clients": 6,
     },
     "server": {
        "rounds": 100,
        "clients_per_round": 6,
     },
     "client": {
        "local_epoch": 5,
        "optimizer": {
```

```
            "type": "SGD",
            "lr": 0.01,
            "momentum": 0.9,
            "weight_decay": 0.0005,
        },
    },
    "model": "alexnet",
    "test_mode": "test_in_client",
}
```

E. Comparison with Existing FL Libraries, Benchmarks, Platforms

We have summarized and compared our proposed COALA with representative prior benchmarks on task-level in Table 11, data level in Table 12, and model-level in Table 13. Our COALA has more comprehensive coverage on task level, data level, and model level compared with these prior works. The following are the compared prior works: LEAF (Caldas et al., 2018), FedCV (He et al., 2021) based on FedML (He et al., 2020), FedScale (Lai et al., 2022), OARF (Hu et al., 2022b), FedReID (Zhuang et al., 2020), FLamby (du Terrail et al., 2022), (Xie et al., 2023) with pFL-Bench (Chen et al., 2022), Felicitas (Zhang et al., 2022), and FLGO (Wang et al., 2023b) [2].

Table 11. Task-level comparison of our proposed COALA and prior FL libraries, benchmarks, and platforms.

Benchmarks	Classification	Object Detection	Segmentation	Person ReID	Face Recognition	Pose Estimation, 3D Point Cloud, etc.
LEAF	✓	×	×	×	✓	×
FedML (FedCV)	✓	✓	✓	×	×	×
FedScale	✓	✓	×	×	×	×
OARF	✓	×	×	×	✓	×
FedReID	×	×	×	✓	×	×
FLamby	✓	×	✓	×	×	×
FederatedScope	✓	×	×	×	×	×
Felicitas	✓	×	×	×	×	×
FLGO	✓	✓	✓	×	×	×
COALA (Ours)	✓	✓	✓	✓	✓	✓

Table 12. Data-level comparison of our proposed COALA and prior FL libraries, benchmarks, and platforms.

Benchmarks	Supervised FL	Semi-supervised FL	Unsupervised FL	Label Shift	Domain Shift	Continual Shift, Test-time Shift
LEAF	✓	×	×	✓	×	×
FedML (FedCV)	✓	×	×	✓	✓	×
FedScale	✓	×	×	✓	✓	×
OARF	✓	×	×	✓	✓	×
FedReID	✓	×	✓	×	✓	×
FLamby	✓	×	×	✓	✓	×
FederatedScope	✓	×	×	✓	×	×
Felicitas	✓	✓	×	✓	×	×
FLGO	✓	×	×	✓	✓	×
COALA (Ours)	✓	✓	✓	✓	✓	✓

F. System Component Overview

The system architecture comprises both the FL server and FL client, each housing a suite of components crucial for supporting a spectrum of realistic FL scenarios.

Task Scheduler: The task scheduler plays a pivotal role in enabling the execution of multiple tasks by scheduling them based

[2]The reference the implementation from their paper and open-source codes (if available) dated the paper submission date (01/02/2024)

Table 13. Model-level comparison of our proposed COALA and prior FL libraries, benchmarks, and platforms.

Benchmarks	Single Model	Federated Split Learning	Personalized Models	FedPEFT for FM
LEAF	✓	×	×	×
FedML (FedCV)	✓	✓	✓	×
FedScale	✓	×	×	×
OARF	✓	×	✓	×
FedReID	✓	×	✓	×
FLamby	✓	×	✓	×
FederatedScope	✓	×	✓	×
Felicitas	✓	×	✓	×
FLGO	✓	×	✓	×
COALA (Ours)	✓	✓	✓	✓

on the availability of resources. Tasks are queued for execution, and the resource manager collaborates to allocate resources efficiently.

Resource Manager: Operating in both the client and server environments, the resource manager keeps meticulous records of client and server resources, including energy consumption, computation capacity, memory, and network connectivity. It relays the availability of clients to the server and handles the allocation of computation resources for training and testing tasks. The server resource manager coordinates with the task scheduler to manage server resources, ensuring optimized execution.

Client Selector: This component implements algorithms for the selection of clients based on their availability. Users have the flexibility to customize client selection algorithms.

Aggregator: At the core of the server, the aggregator executes FL server processes, supporting a variety of aggregation methods such as FedAvg (McMahan et al., 2017) by default, FedProx (Li et al., 2020a), and FedYogi (Reddi et al., 2020b). Users can use these methods directly or customize new aggregation methods.

Compute Engine: The compute engine in the client supports model training and evaluation, with primary compatibility for PyTorch and additional support for frameworks like TensorFlow. In the server, the compute engine is optimized for realistic FL settings, enabling further training with available public data. This is particularly beneficial for semi-supervised FL algorithms, leveraging server data for initial model training and subsequent fine-tuning with aggregated models from clients.

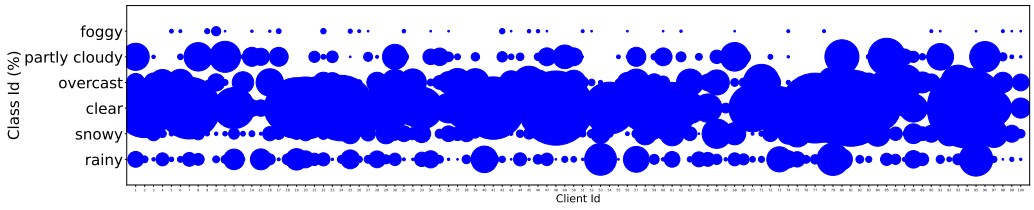
Data Manager and Model Manager: The data manager oversees data loading and partitioning for both simulation and real-world training, while the model manager facilitates model loading and customization. The COALA platform offers a variety of pre-packaged datasets and models (refer to Table 9), and users can effortlessly extend and implement their datasets and models using these components.

Communicator: Responsible for managing remote communication between the server and clients, the communicator employs gRPC as the default protocol framework. It utilizes the industry-standard Protobuf for data serialization, with additional support for MQTT in cases where FL clients are deployed to Internet of Things (IoT) devices.

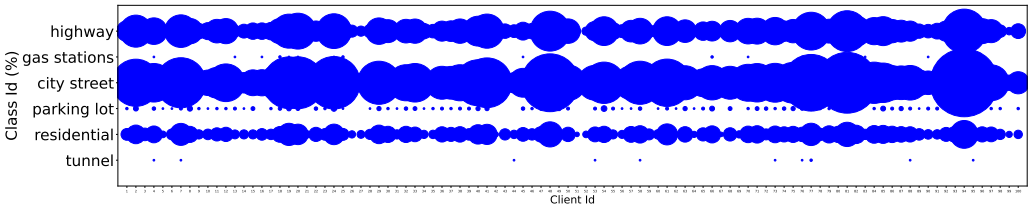
Tracker: This module collates evaluation metrics from both server and clients, capturing both system and algorithmic metrics. Users enjoy the flexibility to customize their tracking service, enabling metric storage in various formats such as disk files or through integration with external libraries or services like wandb [3] or tensorboard (Abadi et al., 2016).

Drift Detector: Designed for practical scenarios involving continual learning, the drift detector continuously monitors data distribution in clients. Upon detecting drifts, it intelligently caches data to client storage and promptly notifies the server. The server, in turn, collects information from clients and initiates federated continual learning on the affected clients, ensuring adaptability to evolving data patterns.

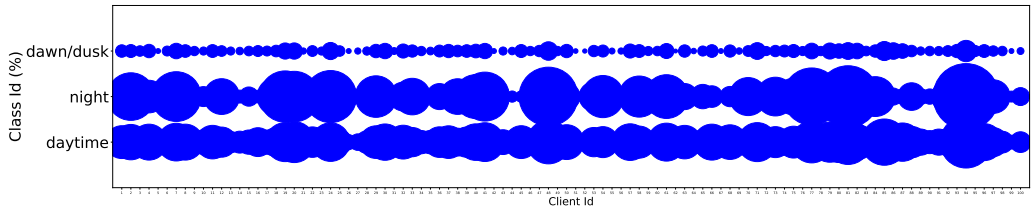
[3] https://wandb.ai



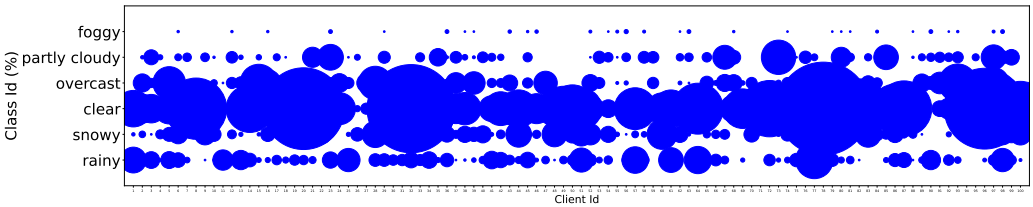
(a) Weather Dir(0.5)



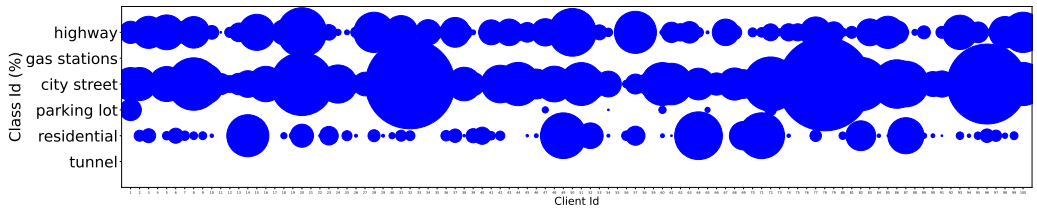
(b) Scene Dir(0.5)



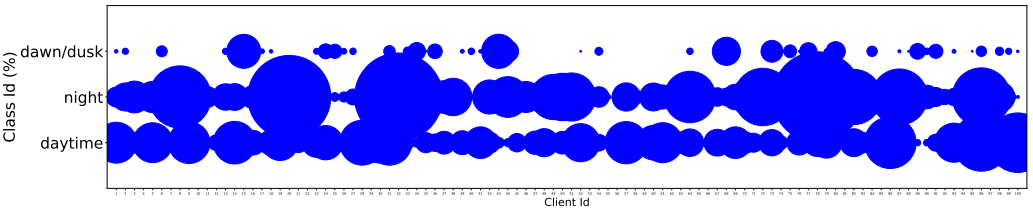
(c) Time-of-day Dir(0.5)



(d) Weather H-Dir(0.5)



(e) Scene H-Dir(0.5)



(f) Time-of-day H-Dir(0.5)

Figure 9. Attribute distributions across 100 clients under Dir (0.5) and H-Dir(0.5) data allocations. (a)-(c) are Dirichlet-based allocations only considering the weather attribute; (d)-(f) are Hierarchical Dirichlet-based allocations considering all three attributes.