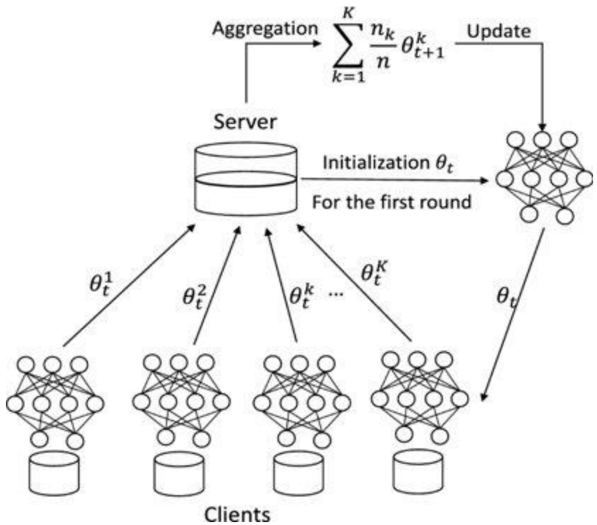
FL Client Selection

2022_Fall

x

Vanilla Federated Learning



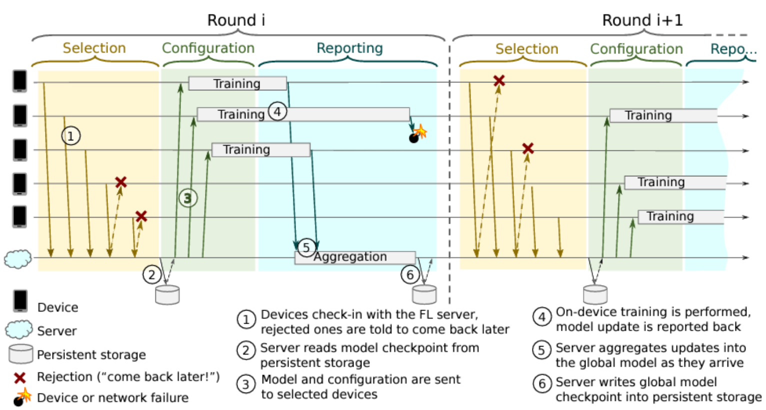


Figure 1: Federated Learning Protocol

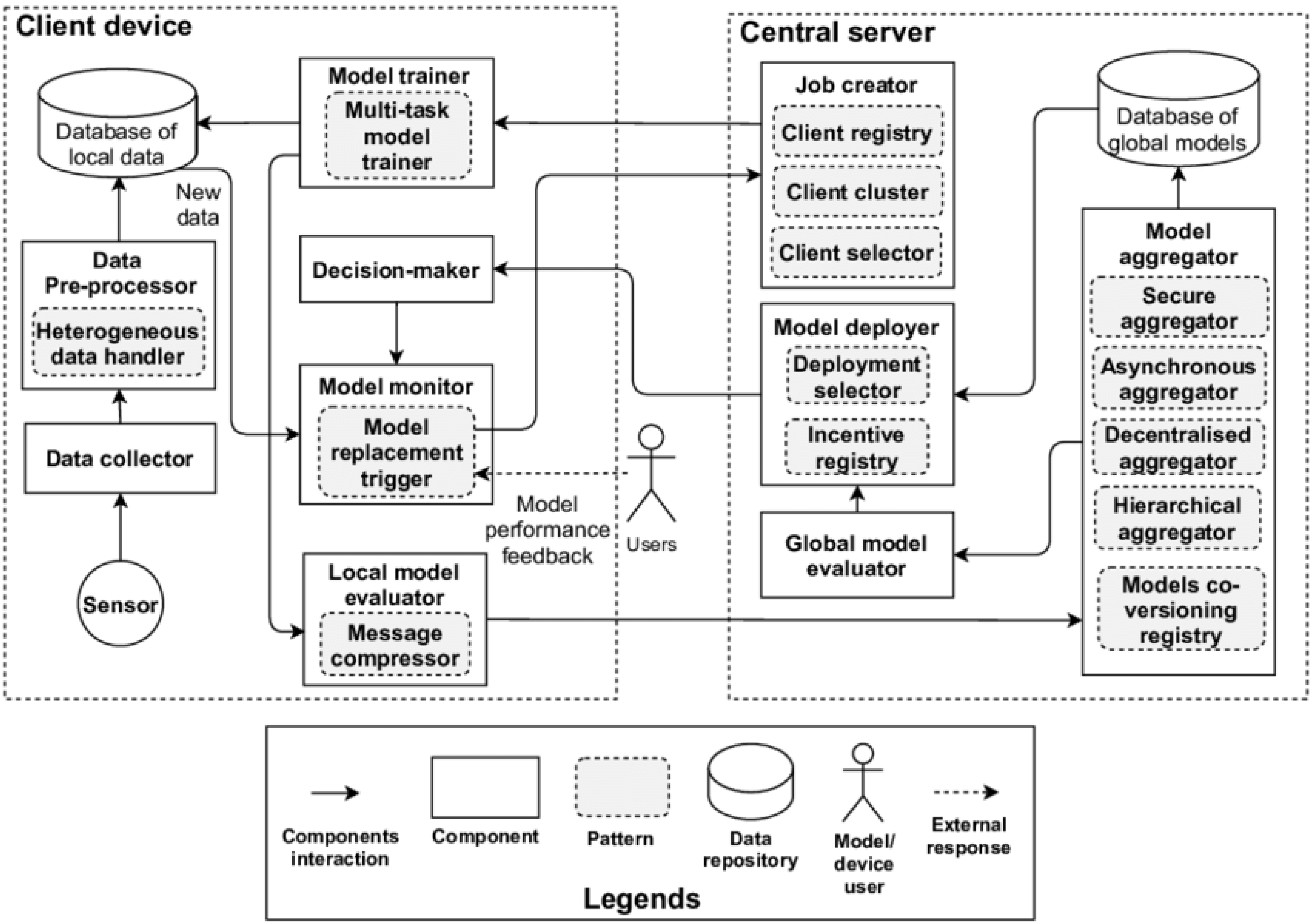
Algorithm 1 `FederatedAveraging`. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:
  initialize w_0
  for each round $t = 1, 2, \dots$ **do**
    $m \leftarrow \max(C \cdot K, 1)$
    $\boxed{S_t \leftarrow \text{(random set of } m \text{ clients)}}$
    for each client $k \in S_t$ **in parallel do**
      $w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$
    $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$

ClientUpdate(k, w): *// Run on client k*
  $\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size B)
  for each local epoch i from 1 to E **do**
    for batch $b \in \mathcal{B}$ **do**
      $w \leftarrow w - \eta \nabla \ell(w; b)$
  return w to server

FLRA: A Reference Architecture for Federated Learning Systems

FLRA: A Reference Architecture for Federated Learning Systems, https://arxiv.org/abs/2106.11570



FLRA: A Reference Architecture for Federated Learning Systems

FLRA: A Reference Architecture for Federated Learning Systems, https://arxiv.org/abs/2106.11570

The central servers interacts with a massive number of client devices that are both system heterogeneous and statistically heterogeneous. The magnitude of client devices number is also several times larger than that of the distributed machine learning systems [18,24]. To increase the model and system performance, client devices can be selected every round with predefined criteria (e.g., resource, data, or performance) via **client selector** component.

Job creation	Mandatory	Job creator	Initialises training job and global model
	Optional	Client registry	Improves system's **maintainability** and **reliability** by maintaining client's information
		Client cluster	Tackles **statistical heterogeneity** & **system heterogeneity** by grouping clients with similar data distribution or resources before aggregation
		Client selector	Improves **model & system's performance** by selecting high performance client devices
Data collection & preprocessing	Mandatory	Data collector	Collects raw data through sensors or smart devices deployed
		Data preprocessor	Preprocesses raw data
	Optional	Heterogeneous Data Handler	Tackles **statistical heterogeneity** through data augmentation methods

FLRA: A Reference Architecture for Federated Learning Systems

FLRA: A Reference Architecture for Federated Learning Systems, https://arxiv.org/abs/2106.11570

Local model training.
Once the client receives the job from the central server, the model trainer component performs model training based on configured hyperparameters (number of epochs, learning rate, etc.). In the standard federated learning training process proposed by McMahan in [28], only model parameters (i.e., weight/gradient) are mentioned to be sent from the central server, whereas in this reference architecture, the models include not only the model parameters but also the hyperparameters.

Model evaluation.
The **local model evaluator** component measures the performance of the local model and uploads the model to the model aggregator on the central server if the performance requirement is met. In distributed machine learning systems, the performance evaluation on client devices is not conducted locally, and only the aggregated server model is evaluated. However, for federated learning systems, local model performance evaluation is required for system operations such as client selection, model co-versioning, contributions calculation, incentive provision, client clustering, etc.

Model training	Mandatory	Model trainer	Trains local model
		Local model evaluator	Evaluates local model performance after each local training round
		Model aggregator	Aggregates local models to produce new global model

FLRA: A Reference Architecture for Federated Learning Systems

FLRA: A Reference Architecture for Federated Learning Systems, https://arxiv.org/abs/2106.11570

this technique is particularly relevant when faced with nonIID data which can produce personalised model that may outperform the best possible shared global model [18]

The conventional design of a federated learning system that relies on a central server to orchestrate the learning process might lead to a single point of failure. **A decentralise aggregator** performs model exchanges and aggregation in decentralised manner to improve system reliability. The known uses of decentralised aggregator include BrainTorrent [31] and FedPGA [15]. Blockchain can be employed as a decentralised solution for federated learning systems.

Optional	Multi-task model trainer	Improves **model performance** (personalisation) by adopting multi-task training methods
	Message compressor	Improves **communication efficiency** through message size reduction to reduce bandwidth consumption
	Secure aggregator	Improves **data privacy** & **system security** through different secure multiparty computation protocols
	Asynchronous aggregator	Improves **system performance** by reducing aggregation pending time of late client updates
	Decentralised aggregator	Improves system **reliability** through the removal of single-point-of-failure
	Hierarchical aggregator	Improves **system performance** & tackle **statistical heterogeneity** & **system heterogeneity** by aggregating models from similar clients before global aggregation
	Model co-versioning registry	Improves system's **accountability** by recording the local models associated to each global models to track clients' performances

FLRA: A Reference Architecture for Federated Learning Systems

FLRA: A Reference Architecture for Federated Learning Systems, https://arxiv.org/abs/2106.11570

The **incentive registry** component maintains all the client devices' incentives based on their contributions and agreed rates to motivate clients to contribute to the training. Blockchain has been leveraged in FLChain [3] and DeepChain [36] to build a incentive registry.

Model deployment	Mandatory	Model deployer	Deploys completely-trained-models
		Decision maker	Decides model deployment
	Optional	Deployment selector	Improves **model performance** (personalisation) through suitable model users selection according to data or applications
		Incentive registry	Increases clients' **motivatability**
Model monitoring	Mandatory	Model monitor	Monitors model's data inference performance
	Optional	Model replacement trigger	Maintains **system & model performance** by replacing outdated models due to performance degrades

Federated Learning Design and Functional Models

Federated Learning Design and FunctionalModels: Survey, https://www.researchsquare.com/article/rs-2101865/latest.pdf

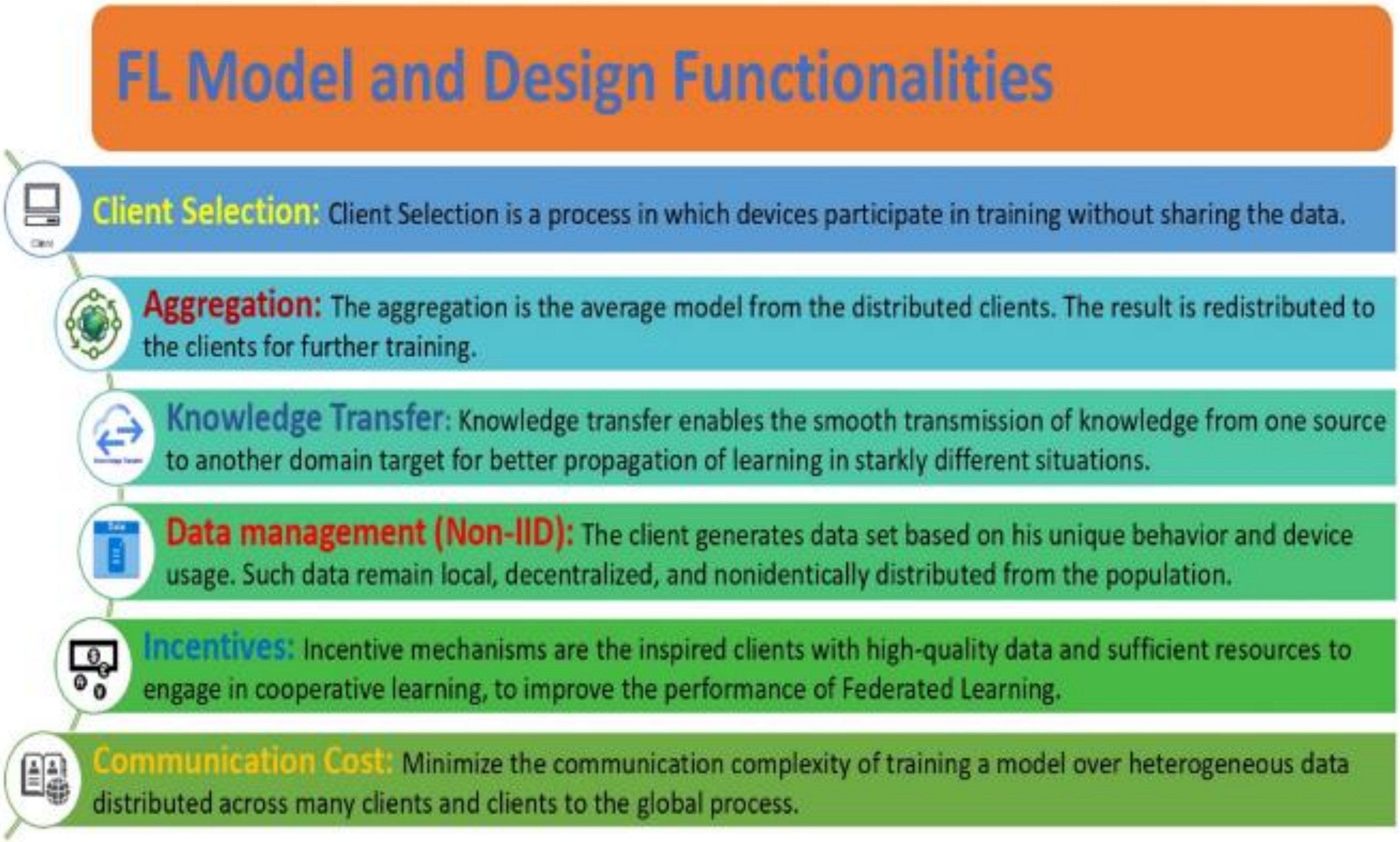


Fig. 1 Functionalities and Design Requirements of Federated Learning

Federated Learning Design and Functional Models

Federated Learning Design and FunctionalModels: Survey,
https://www.researchsquare.com/article/rs-2101865/latest.pdf

Federated Learning Design and Functional Models: Survey 7

Table 2 Summary of Client Selection methods

S.No	Client Selection Models	Goal	Enviornment and Dataset	Evaluation Parameters
1	Fed CS-Client Selection [34]	Client Selection based on client resource conditions.	MEC, ML Tasks, MNIST, CIFAR-10	Accuracy CIFAR-10 (0.54%), MNIST (74%).
2	MAB-based Client Selection [35]	Client Selection based on the rich throughput and computation cost.	MEC, ML Tasks, CIFAR-10	Learning Rate and Communication round accuracy.
3	Power-of-Choice based Client Selection [41]	Guarantee of FL training with a biased client selection method.	DNN, FMNIST	Communication Round Accuracy (71.2% to 76.5%), Convergence speed, Global Loss.
4	Arbitrary client sampling probabilities [45]	Reduced convergence time.	MNIST and EMNIST	Convergence time, Communication cost and Accuracy.
5	FedAECS [46]	Energy consumption management and balancing the trade-off between accuracy and cost in Edge client learning.	MATLAB-Simulation	Energy Consumption and Accuracy.
6	FedGP [47]	Correlation-based client selection strategy.	FMNIST and CIFAR-10	Communication Overhead.
7	FedCor [37]	Correlation-based client selection strategy.	FMNIST and CIFAR-10	Communication Cost.
8	Multi-Arm Bandit (MAB) [42]	Dynamic selection of clients for improved overall accuracy based on the base learning.	MNIST	Communication Cost.
9	PyramidFL [36]	Achieving a higher final model performance (i.e., time-to-accuracy).	Multiple dataset is used.	Accuracy, Clock Time.
10	FedCorr [39]	Identification of noisy clients and separate measurement of all clients to find incorrect labels based on sample losses.	CIFAR-10/100	Communication Test Accuracy.
11	DRFL [40]	Biased Client Selection to encourage fairness and adjust the wait dynamically.	FMNIST	Fairness.
12	Distributed Client Selection [43]	Client Devices for minimizing overall cost.	CIFAR-10, FMNIST, and MNIST	Communication cost, Trade-Off.
13	Fuzzy logic –Client selection [48]	Client selection based on the quantity of samples, throughput, computational capabilities, and sample freshness.	CIFAR-10	Communication Round Accuracy.
14	FedPNS [49]	Selection of nodes that propel faster model convergence.	CIFAR-10, CIFAR-100	Accuracy over Communication Round.
15	Oort [51]	Use of guided participant for selection and performance based on the time to accuracy and for quick training of clients.	Google Speech, OpenImage and etc.	Communication Rounds and Accuracy.
16	E3CS [52]	Boosting of convergence speed.	EMNIST-Letter and CIFAR-10	Communication Rounds and Accuracy.
17	FedMCCS [53]	Multi-Criteria Approach for client selection based on memory, timing, energy, and client resources.	NSL-KDD	Communication Rounds and Accuracy.

Client Selection

Empirical Measurement of Client Contribution for Federated Learning with Data Size Diversification, https://ieeexplore.ieee.org/document/9906094

Client contribution evaluation is crucial in federated learning(FL) to effectively select influential clients. Contrary to data valuation in centralized settings, client contribution evaluation in FL faces a lack of data accessibility and consequently challenges stable quantification of the impact of data heterogeneity. To address this instability of client contribution evaluation, we introduce an empirical method, Federated Client Contribution Evaluation through Accuracy Approximation(FedCCEA), which exploits data size as a tool for client contribution evaluation.

클라이언트 기여 평가는 영향력 있는 클라이언트를 효과적으로 선택하기 위해 연합 학습(FL)에서 중요하다. 중앙 집중식 설정의 데이터 평가와 달리, FL의 클라이언트 기여 평가는 데이터 접근성의 부족에 직면하여 결과적으로 데이터 이질성의 영향을 안정적으로 정량화하는 데 어려움을 겪는다. 이러한 클라이언트 기여 평가의 불안정성을 해결하기 위해 데이터 크기를 클라이언트 기여 평가 도구로 활용하는 경험적 방법인 FedCCEA(FedCCEA)를 소개한다.

After several FL simulations, FedCCEA approximates the test accuracy using the sampled data size and extracts the client contribution from the trained accuracy approximator. In addition, FedCCEA grants data size diversification , which reduces the massive variation in accuracy resulting from game-theoretic strategies. Several experiments have shown that FedCCEA strengthens the robustness to diverse heterogeneous data environments and the practicality of partial participation.

Client Selection

Empirical Measurement of Client Contribution for Federated Learning with Data Size Diversification, https://ieeexplore.ieee.org/document/9906094

데이터 중심 접근 방식에서 FL은 클라이언트 수준에서 기여도를 측정하여 데이터 품질을 고려합니다. 클라이언트 기여는 일반적으로 연합 모델 성능에 대한 각 클라이언트에 대한 데이터 세트의 영향으로 정의됩니다. 고객 기여도 측정은 연합 모델 개선의 두 가지 특정 측면에 적용된다.

(i) 클라이언트 선택
딥 러닝 모델의 맥락과 관련하여, 모든 데이터가 동일한 가치를 가지는 것은 아니다[5]. 따라서 고품질 및 저품질 데이터를 보존하고 폐기하는 것은 고성능 딥 러닝 모델을 훈련하기 위한 전제 조건이다[6]. 마찬가지로, 모든 클라이언트가 연합 설정에 동일하게 기여하는 것은 아닙니다 [7]–[10]. 영향력 있는 고객을 선택하고 불필요한 고객을 제거하기 위해 이러한 고객을 면밀히 모니터링하고 각 고객의 기여도를 측정해야 합니다.

(ii) 인센티브 할당
경제적으로, 고객 기여는 이익을 극대화하면서 인센티브를 공정하게 배분하는 데 적합한 표준입니다 [11]–[15]. 고객 기여와 함께 적절한 인센티브 배분은 각 클라이언트의 고품질 데이터 양이 모델 정확도에 영향을 미치는 FL에 높은 기여자가 적극적으로 참여하도록 동기를 부여할 수 있다. 이러한 인센티브 메커니즘은 중앙 서버(또는 조정자)에 의한 고성능 연합 모델을 통해 비즈니스 시스템에서 수익과 비용을 효율적으로 관리하는 데 도움이 될 수 있습니다.

Then the question is, “how do we evaluate the client contribution in the FL setting?" Unfortunately, a different view from data valuation of centralized learning is required.
그렇다면 질문은 "FL 설정에서 고객 기여도를 어떻게 평가하느냐"는 것이다. 불행하게도, 중앙 집중식 학습의 데이터 평가와는 다른 관점이 필요하다

Client Selection

Empirical Measurement of Client Contribution for Federated Learning with Data Size Diversification, https://ieeexplore.ieee.org/document/9906094

Moreover, the impact of data distribution, noise, and data quantity is not as clear as in centralized settings because they strongly rely on combinations with other clients. As shown in Fig. 1 ....
또한 데이터 배포, 노이즈 및 데이터 양의 영향은 중앙 집중식 설정처럼 명확하지 않습니다. 다른 클라이언트와의 조합에 크게 의존하기 때문입니다. 그림 1에 나타난 바와 같이...

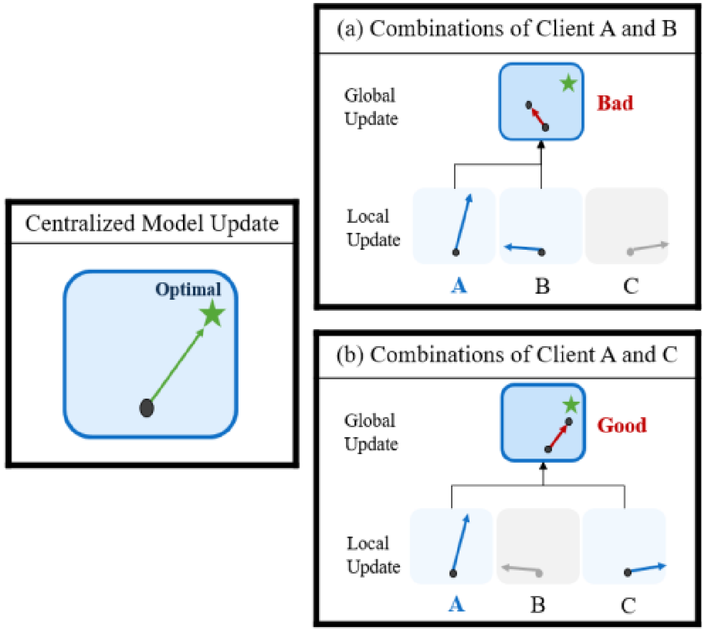


FIGURE 1. Examples of the combinatorial impact of client A in a single round with different combinations.

Client Selection

Empirical Measurement of Client Contribution for Federated Learning with Data Size Diversification, https://ieeexplore.ieee.org/document/9906094

From early explorations, Shapley Value [8], [21], a game-theoretic evaluation method, predicts the overall combinatorial impact of clients on performance by averaging the marginal test accuracy with all the possible client subsets including and excluding a client as shown in Fig. 2.
Although it is a theoretically well-structured evaluation method, the client contribution measurement by Shapley Value faces challenges with extreme accuracy fluctuations of some combinations in heterogeneous data environments. These drastic combinatorial effects result in unstable client contribution

On the contrary, FedCCEA enables several simulations with data size diversification to analyze the impact of client A on model performance, while considering various data size sets.

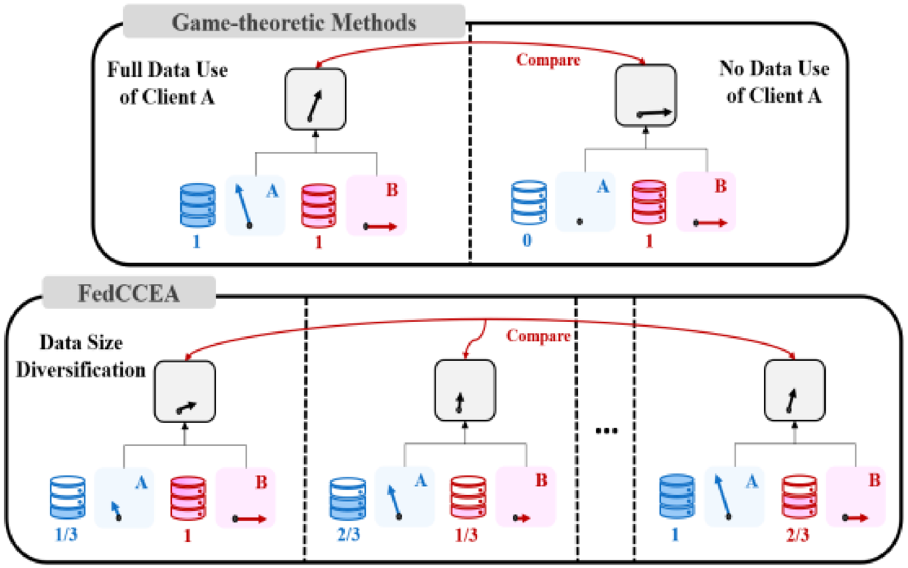


FIGURE 2. Data use cases of each contribution evaluation method while measuring contribution of client A. Regarding the game-theoretic methods, *full or no* data of client A are used to compare the global updates, including and excluding client A. On the contrary, FedCCEA enables several simulations with *data size diversification* to analyze the impact of client A on model performance, while considering various data size sets.

Client Selection

Empirical Measurement of Client Contribution for Federated Learning with Data Size Diversification, https://ieeexplore.ieee.org/document/9906094

II. RELATED WORKS

A. DATA VALUATION

Data Valuation, a phrase similar to Client Contribution Evaluation, has been widely studied recently to improve centralized machine learning models and to explain black-box predictions.

B. CLIENT CONTRIBUTION EVALUATION FOR FL

In addition to a model-centric approach that focuses on FL optimization [36]–[39], the client contribution evaluation in our study is a data-centric solution to the client-drift problem of FedAvg [2]. The server provides more credit to major clients and fewer credit to minor clients.

TABLE 1. Summary of Client Contribution Evaluation Methods for Federated Learning

Information Used	Keyword	Literature	Description
Local Weights/ Local Gradients	Leave-one-out	[14], [25]	Measure the marginal performance difference of a specific client's participation.
	Shapley Value	[8], [21], [26], [27]	Measure the weighted mean of the marginal performance difference of all possible subsets with a specific client's participation.
	Weight Difference	[28]	Use client contribution based on the directional difference of local weights/gradients for incentive allocation.
	DRL Models	[29], [30]	Empirically predict each client contribution using REINFORCE or DQN models with local weights/gradients.
Local Data Size	Data Quantity	[12]	Simply define a local data size as a total value of each local dataset for incentive allocation.
	FedCCEA	Ours	Empirically predict an averaged impact of each local dataset using deep learning models with diverse cases of data size.

Client Selection

Empirical Measurement of Client Contribution for Federated Learning with Data Size Diversification, https://ieeexplore.ieee.org/document/9906094

On the other hand, the information of local data size has rarely been exploited for client contribution evaluation because a large amount of data does not clearly lead to a higher contribution in federated learning when data heterogeneity exists.
Previously, the local data size was defined as a client contribution for simple construction of a DRL-based incentive mechanism [12] with strong assumptions. However, in addition to the local data size, quantification of the impact of data heterogeneity(e.g. data corruption and non-IID) is required to correctly measure the client contribution in any data environment.

그러나 로컬 데이터 크기 외에도 모든 데이터 환경에서 클라이언트 기여도를 올바르게 측정하기 위해서는 데이터 이질성(예: 데이터 손상 및 비 IID)의 영향을 정량화해야 한다.

TABLE 1. Summary of Client Contribution Evaluation Methods for Federated Learning

Information Used	Keyword	Literature	Description
Local Weights/ Local Gradients	Leave-one-out	[14], [25]	Measure the marginal performance difference of a specific client's participation.
	Shapley Value	[8], [21], [26], [27]	Measure the weighted mean of the marginal performance difference of all possible subsets with a specific client's participation.
	Weight Difference	[28]	Use client contribution based on the directional difference of local weights/gradients for incentive allocation.
	DRL Models	[29], [30]	Empirically predict each client contribution using REINFORCE or DQN models with local weights/gradients.
Local Data Size	Data Quantity	[12]	Simply define a local data size as a total value of each local dataset for incentive allocation.
	FedCCEA	Ours	Empirically predict an averaged impact of each local dataset using deep learning models with diverse cases of data size.

Client Selection

Empirical Measurement of Client Contribution for Federated Learning with Data Size Diversification, https://ieeexplore.ieee.org/document/9906094

IV. EXPERIMENTS
In this section, we want to answer the following questions:

1) How does accuracy variation occur in the Shapley Value evaluation and how does FedCCEA address this problem?
2) Is FedCCEA evaluation accurate even in the strong nonIID and noisy environments?
3) Is FedCCEA evaluation accurate even with partial participation?

To answer each question, we design

(i) an accuracy variation comparison,
(ii) a client removal test, and
(iii) experiments for complexity analysis with different numbers of clients.

1) BASELINE EVALUATION METHODS

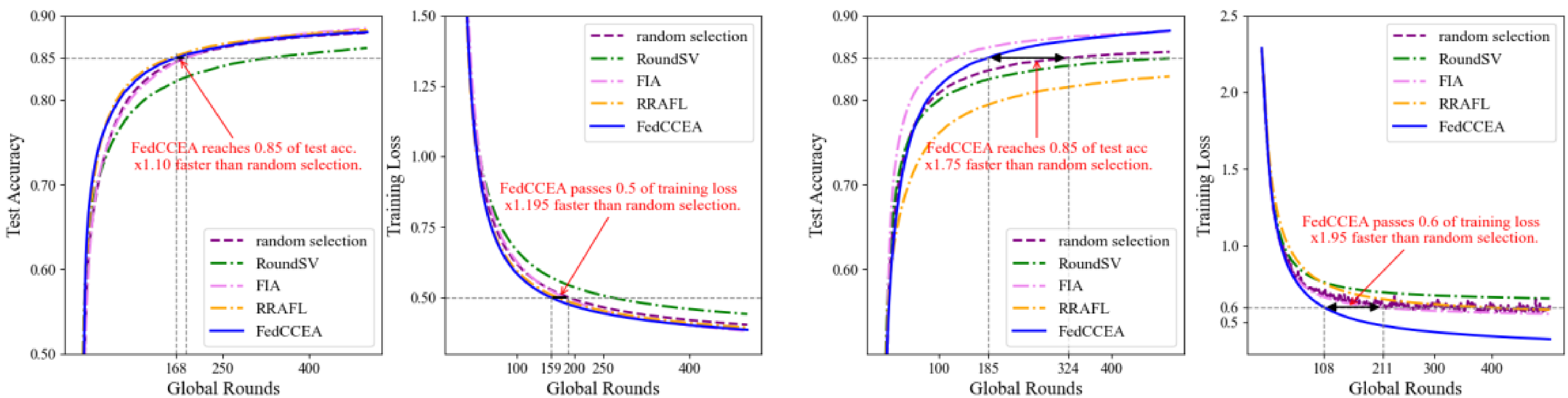
We answer the above questions and prove the strengths by comparing FedCCEA to the three baseline evaluation methods in recent studies.

- RoundSV [8], [40] is an approximation of Shapley Value in FL. We use the permutation-based RoundSV, utilizing Monte-Carlo sampling for SV approximation.
- Fed-Influence in Accuracy(FIA) [25] is a type of FedInfluence measurement metric that simply measures the influence by investigating the effect of removing a client only. The actual FIA value can be obtained from the results of the leave-one-out test.
- RRAFL [28] measures the contribution of cosine similarity between the final global weight vector and current local weight vectors.

Client Selection

Empirical Measurement of Client Contribution for Federated Learning with Data Size Diversification, https://ieeexplore.ieee.org/document/9906094

E. FURTHER EXPERIMENTS
1) Convergence Analysis for Client Selection
2) Client Removal Test with Different Number of Clients
3) Complexity Analysis



(a) MNIST S.1 setting (b) MNIST S.2 setting

FIGURE 10. Training loss and test accuracy performance for 4-client exclusion strategies in (a) S.1 setting and (b) S.2 setting. The difference of convergence speed seems to be trivial between FedCCEA-based client selection and contribution-based selection strategies. However, FedCCEA constantly reaches a convergence point faster than uniformly random selection in both settings.

Client Selection

Empirical Measurement of Client Contribution for Federated Learning with Data Size Diversification, https://ieeexplore.ieee.org/document/9906094

E. FURTHER EXPERIMENTS

1) Convergence Analysis for Client Selection
2) Client Removal Test with Different Number of Clients
3) Complexity Analysis

TABLE 6. Experiment for Complexity Analysis with different number of clients(in seconds) (S : number of simulations, N : number of clients) Empirically, we design a federated setting with 20 samples for each client, 10 rounds, and only implement a single simulation($S = 1$).

Methods	Complexity (S, N)	Client Number				
		5 ($\times1$)	10 ($\times2$)	20 ($\times4$)	40 ($\times8$)	80 ($\times16$)
FedCCEA	$O(S)$	5.55s ($\times1$)	5.43s ($\times0.98$)	6.02s ($\times1.08$)	5.82s ($\times1.05$)	6.09s ($\times1.10$)
RoundSV	$O(NlogN)$	5.71s ($\times1$)	8.17s ($\times1.43$)	8.32s ($\times1.46$)	24.15s ($\times4.23$)	77.63s ($\times13.60$)
FIA	$O(N^2)$	59.88s ($\times1$)	116.65s ($\times1.95$)	229.72s ($\times3.84$)	488.04s ($\times8.15$)	1284.18s ($\times21.44$)
RRAFL	$O(N)$	9.50s ($\times1$)	10.17s ($\times1.07$)	10.90s ($\times1.15$)	12.87s ($\times1.35$)	16.60s ($\times1.75$)

Client Selection

Oort: Efficient Federated Learning via Guided Participant Selection, https://www.usenix.org/conference/osdi21/presentation/lai
https://github.com/Kwangkee/FL/blob/main/FL%40ClientSelection.md#oort

As a result, data characteristics and device capabilities vary widely across clients. Yet, **existing efforts randomly select FL participants, which leads to poor model and system efficiency.** In this paper, we propose Oort to improve the performance of federated training and testing with guided participant selection.

With an aim to improve time-to-accuracy performance in model training, **Oort prioritizes the use of those clients who have both data that offers the greatest utility in improving model accuracy and the capability to run training quickly.**

Unfortunately, clients may not all be simultaneously available for FL training or testing [44]; they may have heterogeneous data distributions and system capabilities [19,38]; and including too many may lead to wasted work and suboptimal performance [19] (§2). **Consequently, a fundamental problem in practical FL is the *selection of a "good" subset of clients as participants*,** where each participant locally processes its own data, and only their results are collected and aggregated at a (logically) centralized coordinator.

Although random participant selection is easy to deploy, unfortunately,
- it results in poor performance of federated training because of large heterogeneity in device speed and/or data characteristics.
- **Worse, random participant selection can lead to biased testing sets and loss of confidence in results.**

Client Selection

Oort: Efficient Federated Learning via Guided Participant Selection, https://www.usenix.org/conference/osdi21/presentation/lai
https://github.com/Kwangkee/FL/blob/main/FL%40ClientSelection.md#oort

1. **Job submission**
2. **Participant selection:**
- the coordinator enquires the clients meeting eligibility properties (e.g., battery level), and forwards their characteristics (e.g., liveness) to Oort. Given the developer requirements (and execution feedbacks in case of training 2a),
- Oort selects participants based on the given criteria and notifies the coordinator of this participant selection(2b).
3. **Execution**
4. **Aggregation**

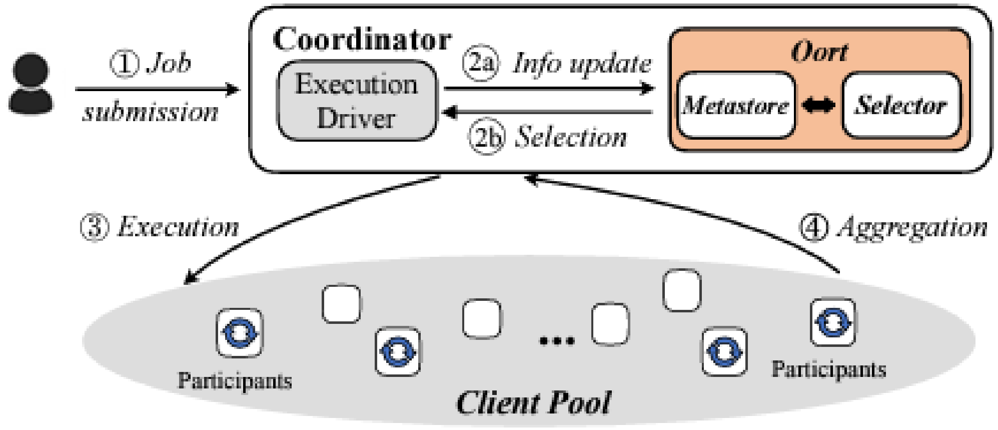
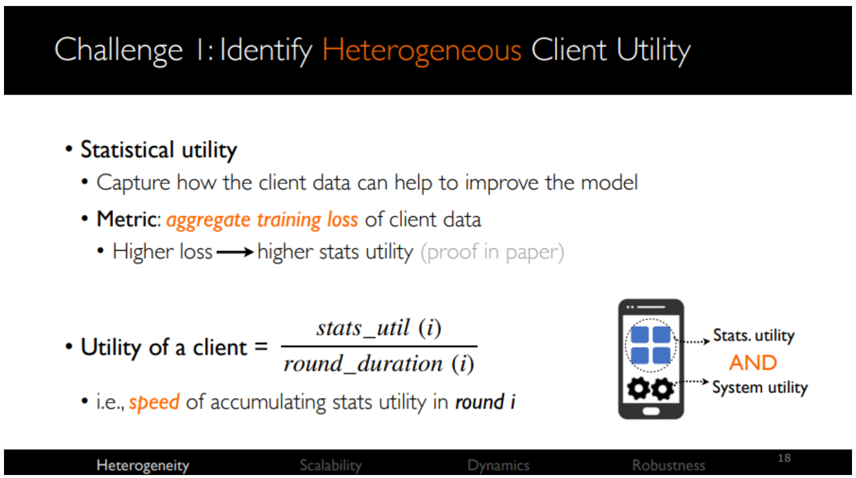


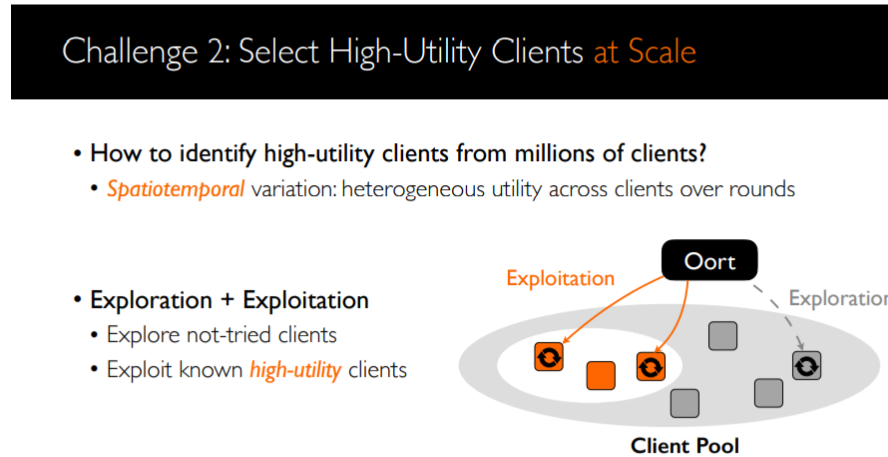
Figure 5: *Oort architecture. The driver of the FL framework interacts with Oort using a client library.*

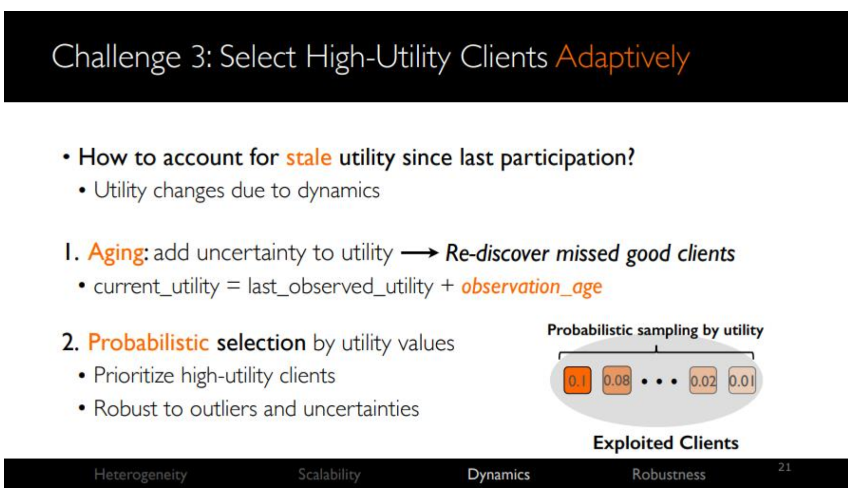
Client Selection

Oort: Efficient Federated Learning via Guided Participant Selection, https://www.usenix.org/conference/osdi21/presentation/lai
https://github.com/Kwangkee/FL/blob/main/FL%40ClientSelection.md#oort

주요 아이디어: loss-based statistical utility design
주요 아이디어: MAB (Multi-Armed Bandit) problem, exploration-exploitation







Sample Selection

FedBalancer: Data and Pace Control for Efficient Federated Learning on Heterogeneous Clients (ACM MobiSys 2022), https://nmsl.kaist.ac.kr/projects/fedbalancer/
https://github.com/Kwangkee/FL/blob/main/FL%40ClientSelection.md#fedbalancer

Unlike centralized training that is usually based on carefully-organized data, FL deals with on-device data that are often unfiltered and imbalanced. As a result, conventional FL training protocol that treats all data equally leads to a waste of local computational resources and slows down the global learning process.

To this end, we propose FedBalancer, a systematic FL framework that **actively selects clients' training samples.** Our sample selection strategy prioritizes more "informative" data while respecting privacy and computational capabilities of clients. To better utilize the sample selection to speed up global training, we further introduce an adaptive deadline control scheme that predicts the optimal deadline for each round with varying client training data.

For model developers who prototype a mobile AI with **FL without a proxy dataset,** achieving faster convergence on thousands to millions of devices is desired to efficiently test multiple model architectures and hyperparameters [33]. Service providers who frequently update a model with continual learning with FL require to minimize the user overhead with better time-to-accuracy performance [39].

A key objective in FL is to optimize time-to-accuracy performance. FL tasks typically require hundreds to thousands of rounds to converge [13, 38], and clients participating at a round undergo substantial computational and network overhead [21]. Deploying FL across thousands to millions of devices should be done efficiently, quickly reaching the model convergence while not sacrificing the model accuracy. This becomes more important when FL has to be done multiple times, as often the case when model developers prototype a new model with FL without a proxy dataset or periodically update a deployed model to new domain via continual learning or online learning with FL.

Sample Selection

FedBalancer: Data and Pace Control for Efficient Federated Learning on Heterogeneous Clients (ACM MobiSys 2022), https://nmsl.kaist.ac.kr/projects/fedbalancer/

The sample selection of FedBalancer prioritizes more "informative" samples of clients to efficiently utilize their computational effort. **This allows low-end devices to contribute to the global training within the round deadline by focusing on smaller but more important training samples.** To achieve high time-to-accuracy performance, the sample selection is designed to operate without additional forward or backward pass for sample utility measurement at FL rounds. Lastly, FedBalancer can coexist and collaborate with orthogonal FL approaches to further improve performance.

The loss threshold ratio (ltr) enables FedBalancer to start training with all samples and gradually remove already-learned samples. FedBalancer initialize ltr as 0.0 and gradually increases the value by loss threshold step size (lss) as shown in Algorithm 3. Note that the deadline ratio ($ddlr$), which controls the deadline of each round (described in Section 3.3), is also controlled with ltr.

FedBalancer gradually increase loss threshold to remove already-learned samples

- **Round 가 진행될수록, Loss threshold 는 gradually increase**

The intuition of sampling a portion of data from UTi is to avoid catastrophic forgetting [35, 78] of the model on already-learned sample

- We sample $L \cdot p$ samples from OTi and $L \cdot (1 - p)$ samples from UTi where L indicates the number of selected samples and p is a parameter in an interval of [0.5, 1.0]
- L, the length of selected samples, is determined based on the hardware speed of a client
- p is a FedBalancer parameter between 0.5 ≤ p ≤ 1.0.
- -> p 가 클수록, catastrophic forgetting 을 좀 더 걱정한다는 의미.

Client Selection

Yae Jee Cho, https://github.com/Kwangkee/FL/blob/main/FL@CarnegieMellon.md#yae-jee-cho

Towards Understanding Biased Client Selection in Federated Learning,
https://proceedings.mlr.press/v151/jee-cho22a.html
In our work, we present the convergence analysis of federated learning with biased client selection and quantify how the bias affects convergence speed. **We show that biasing client selection towards clients with higher local loss yields faster error convergence.**

To Federate or Not To Federate: Incentivizing Client Participation in Federated Learning,
https://arxiv.org/abs/2205.14840
Figure 2: Aggregating weight qk(w) for any clientk versus the emprical incentive gap Fk(w) − Fk(wb k). The weight qk(w) is small for clients that already have a very large incentive (global much better than local) or no incentive at all (local much better than global), and is highest for clients that are moderately incentivized (global similar to local).