AAAI-22

# Is Your Data Relevant?: Dynamic Selection of Relevant Data for Federated Learning

**Lokesh Nagalapatti**[*1†], **Ruhi Sharma Mittal**[*2], **Ramasuri Narayanam**[3†]

[1] IIT Bombay
[2] IBM Research AI
[3] Adobe Research India
nlokesh@cse.iitb.ac.in, ruhi.sharma@in.ibm.com, rnarayanam@adobe.com

SEOULTECH 서울과학기술대학교 데이터사이언스학과

BK21 FOUR 데이터사이언스와 비즈니스포텐셜

# Introduction

Not all the data owned by each client is relevant to the server's learning objective

-> Federated Learning with Relevant Data($FLRD$)

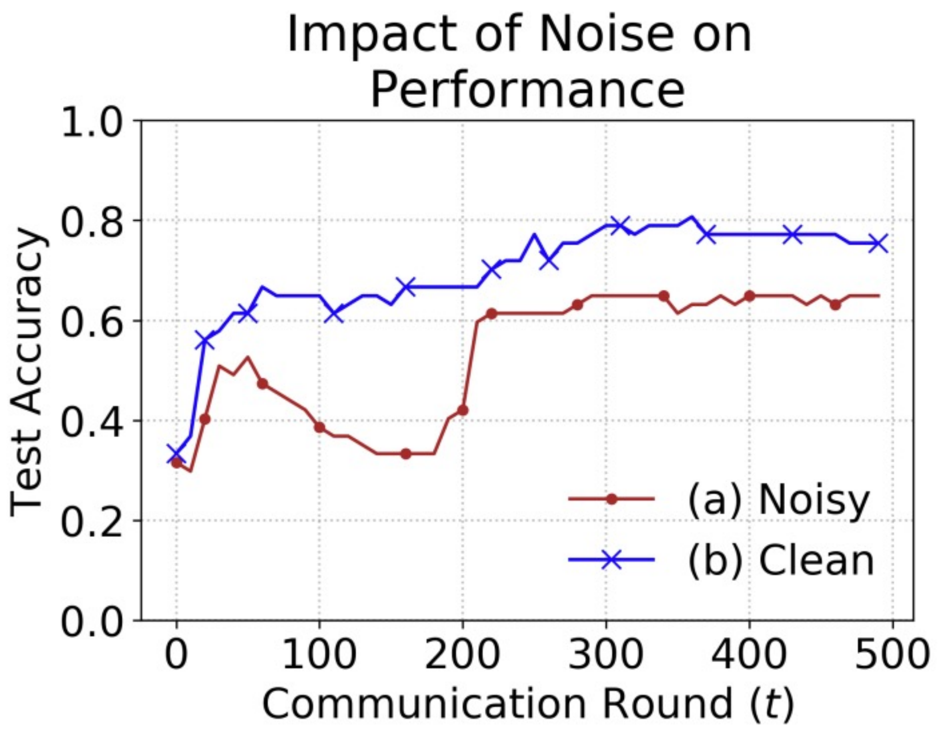
* Relevant Data: data samples that are favorable to Global Model
  - Different across multiple communication rounds
  - Adapt to the dynmaics of FL environment

# Motivation Experiment

- Iris dataset
- One server($S$) and two clients($C_1, C_2$)
- FedAvg algorithm

(1) 20% closed-set noise at each client by flippping the labels

(2) Removing the noisy data samples



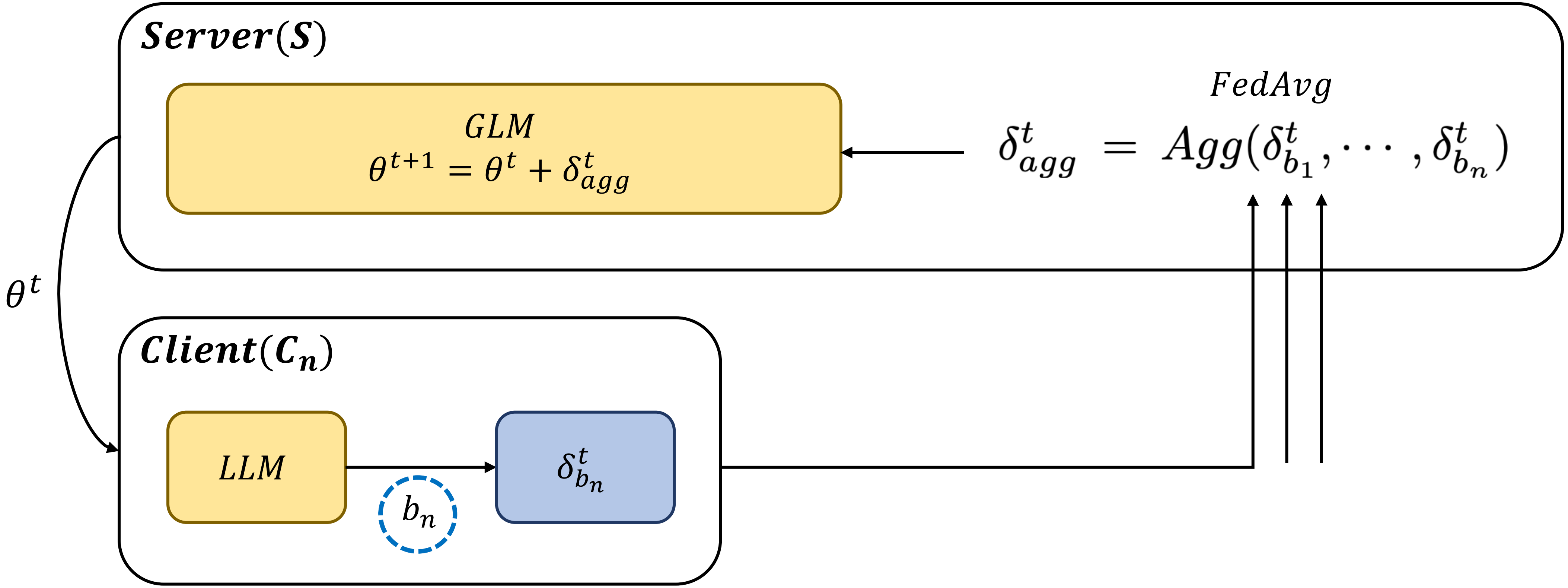
# Problem Definition

1. Server ($S$)
   - $GLM$ : $f_\theta : X \to Y$ to minimize $l(y, f_\theta(x))$
   - $D_V, D_{Test}$ : $IID$ drawn from the target distribution -> used only for test

2. Clients $\{C_1, C_2, \ldots, C_n\}$
   - $LLM$
   - $Local\ Data\ of\ C_i$: $D_i$

# Problem Definition

*At each round* $\boldsymbol{t}$ ...




How client learns a relevance prediction function locally w.r.t the $GLM$'s objective

# Key Challenges

1. The proposed solution should adhere to the privacy constraints imposed by the FL framework

2. Designing a solution that relies only on client's local data may lead to sub-optimalities in the relevance prediction function and thereby it adversely affects $GLM$

3. Relevance value of a data point should vary as a function of time $t$.

# Proposed Method

- Each client learns $RDS$ to predict the relevance score($RS$) [0,1]
- $GLM, RDS$ : deep neural networks

$$GLM : f_\theta : X \rightarrow Y \tag{1}$$

$$RDS_i : g_{i_{\phi_i}} : (X, Y) \rightarrow [0,1] \tag{2}$$

---

- Objective of $FLRD$

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{n} \sum_{(x,y)\in D_i} \underbrace{\boxed{g_{i_{\phi_i}}(x,y)}}_{RS} \cdot \underbrace{\boxed{l(y, f_\theta(x))}}_{Global\ Loss} \tag{3}$$

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{(x,y)\in D_{Test}} l(y, f_\theta(x)) \tag{4}$$

$g_i(x, y)$ to be high for samples which finds better $\hat{\theta}$ closer to $\theta^t$

# Proposed Method

Training $\boldsymbol{GLM}$

1. $RDS_i$ samples mini-batch ($b_i \subseteq D_i$)
2. Fits $LLM_i$ with $b_i$

$$\delta_{b_i}^t = \gamma_i^E - \theta^t$$

$$\delta_{agg}^t = \frac{1}{n}\sum_{i=1}^{n} \delta_{b_i}^t \tag{5}$$

$$\theta^{t+1} = \theta^t + \delta_{agg}^t \tag{6}$$

# Proposed Method

## Training $\boldsymbol{RDS}$

- $RDS_i$ as a local policy network of client $C_i$ to select $b_i$
- Utilize reward signal from server to train $RDS_i$
- Two possible solutions:
    1) Make $\mathrm{D_V}$ public and use it to compute the feed back locally
    2) $\mathrm{D_V}$ is private to server and server computes the feedback on it

# Proposed Method

Training $\boldsymbol{RDS}$

1. Receive $(\theta^t, r_i^{t-1})$ from the server
2. Compute $\delta_{bi}^t$ and $\delta_{f\,i}^t$, where $f_i(\subseteq D_i)$ sampled uniformly at random
3. Server computes the reward $r_i^t$

$$r_i^t(b_i) = \mathcal{P}(\theta^t + \delta_{b_i}^t) - \mathcal{P}(\theta^t + \delta_{f_i}^t) \qquad (7)$$

$$\mathcal{P}(\theta) = \frac{1}{|D_V|} \sum_{(x,y)\in D_V} \mathcal{I}(y == f_\theta(x)) \qquad (8)$$

$P(\theta)$ is a performance measure on validation set with parameters of $GLM$ as $\theta$

➢ *If $b_i$ is relevant, then $\delta_i^t$ adds more value to $GLM$ than $\delta_{b_i}^t$*

# Proposed Method

**Utility function**:

- 각 클라이언트 $C_i$는 $\emptyset_i$를 최대화하기 위한 유틸리티 함수를 가지며, 이 함수는 확률 분포 $\pi_i$에 따라 $r_i^t(b_i, f_i)$의 기대값을 최대화하려고 시도합니다.
- $\alpha$는 $D_i$의 부분 집합이며, $\pi_i$는 $D_i$ 의 확률 분포이고 $g_{\emptyset_i}(x, y)$에 따라 계산됩니다.

$$\max_{\phi_i} J(\phi_i) = \mathbb{E}_{\alpha \sim \pi_i}[r_i^t(b_i, f_i)] \quad (9)$$

$$\pi_i(\alpha|D_i) = \prod_{(x,y)\in\alpha} g_{\phi_i}(x, y) \cdot \prod_{(x,y)\in D_i \setminus \alpha} [1 - g_{\phi_i}(x, y)] \quad (10)$$

**Policy Gradients Calculation**:

- 각 클라이언트 $C_i$는 $RDS_i$를 업데이트하기 위해 policy gradients를 계산합니다.
- Policy gradients는 $r_i^t(\alpha)$와 $\pi_i(\alpha|D_i)$를 곱한 합으로 계산됩니다.

$$\nabla_{\phi_i} J(\phi_i) = \nabla_{\phi_i} \sum_{\alpha \in 2^{D_i}} r_i^t(\alpha) \cdot \pi_i(\alpha|D_i) \quad (11)$$

# Proposed Method

**Policy Gradients Approximation**:

- Policy gradients를 근사화하기 위해 $b_i$에 대한 하나의 샘플 하위 집합을 사용

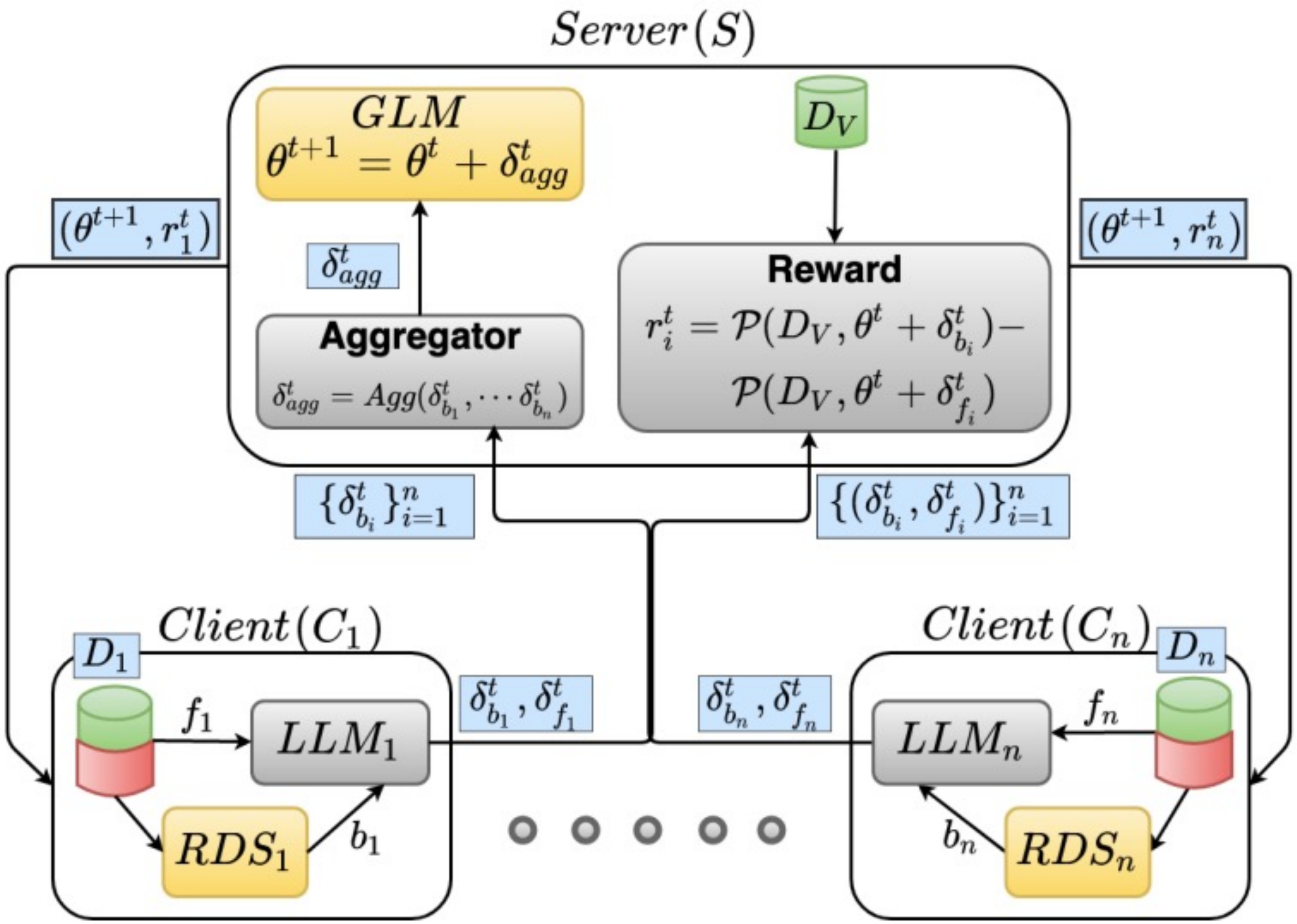
$$\nabla_{\phi_i} \hat{J}(\phi_i) = r_i^t(b_i) \cdot \left[\pi_i(b_i|D_i)\right] \cdot \left[\nabla_{\phi_i} log(\pi_i(b_i|D_i))\right] \quad (12)$$

**Model Update**:

- 클라이언트 $C_i$는 추정된 policy gradients를 기반으로 학습률 $\zeta_i$를 사용하여 $RDS_i$를 업데이트

$$\phi_i = \phi_i + \zeta_i \nabla_{\phi_i} \hat{J}(\phi_i) \quad (13)$$

# Proposed Method



# Three types of noise

1. Attribute noise

   : erroneous values or missing values

2. Closed-set label noise

   : randomly flip labels of data samples

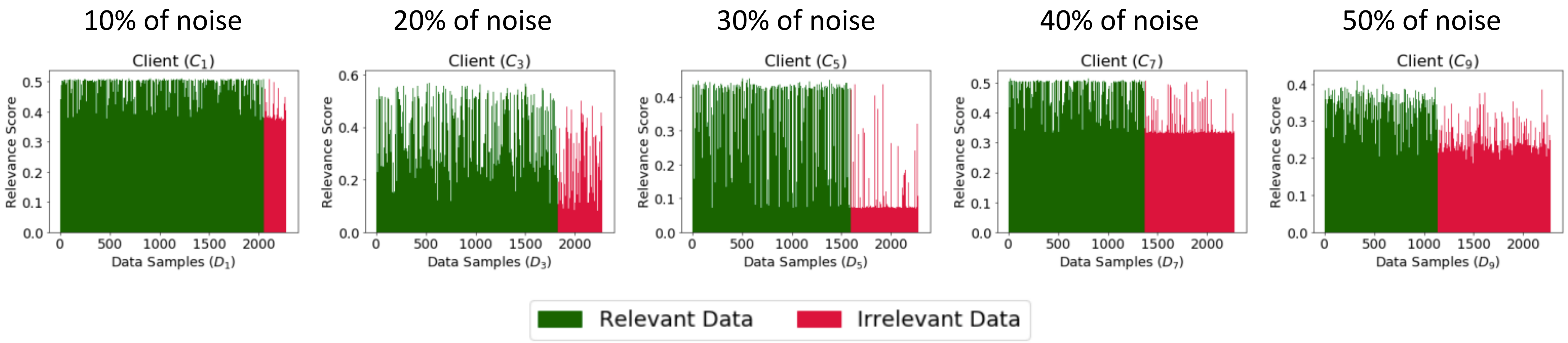
3. Open-set label noise

   : assign out of distribution samples and label them randomly

# Experiments

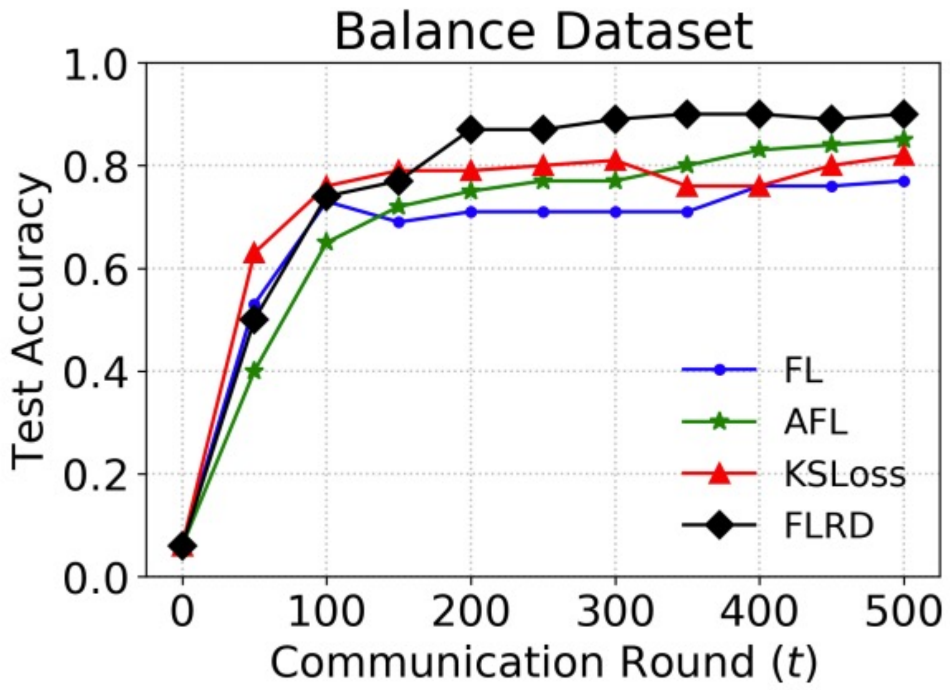
Relevance score$(RS)$ of data samples after 100 communication rounds

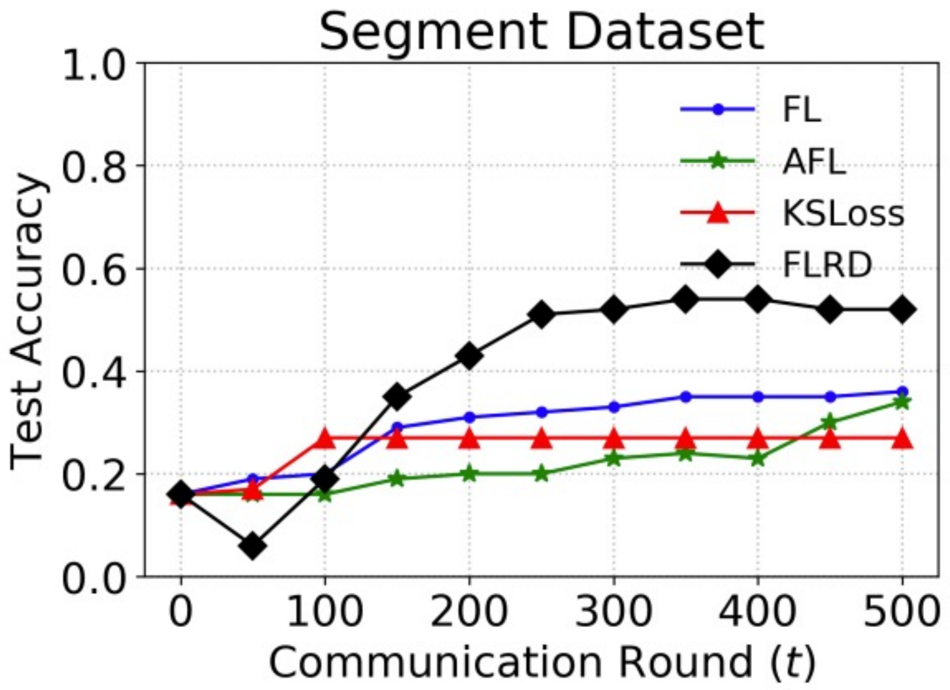
- **Green Bar**: non-noisy data samples
- **Red bar**: noisy data samples



# Experiments

With attribute noise(5%), the performance of $GLM$ is superior to that of other baselines

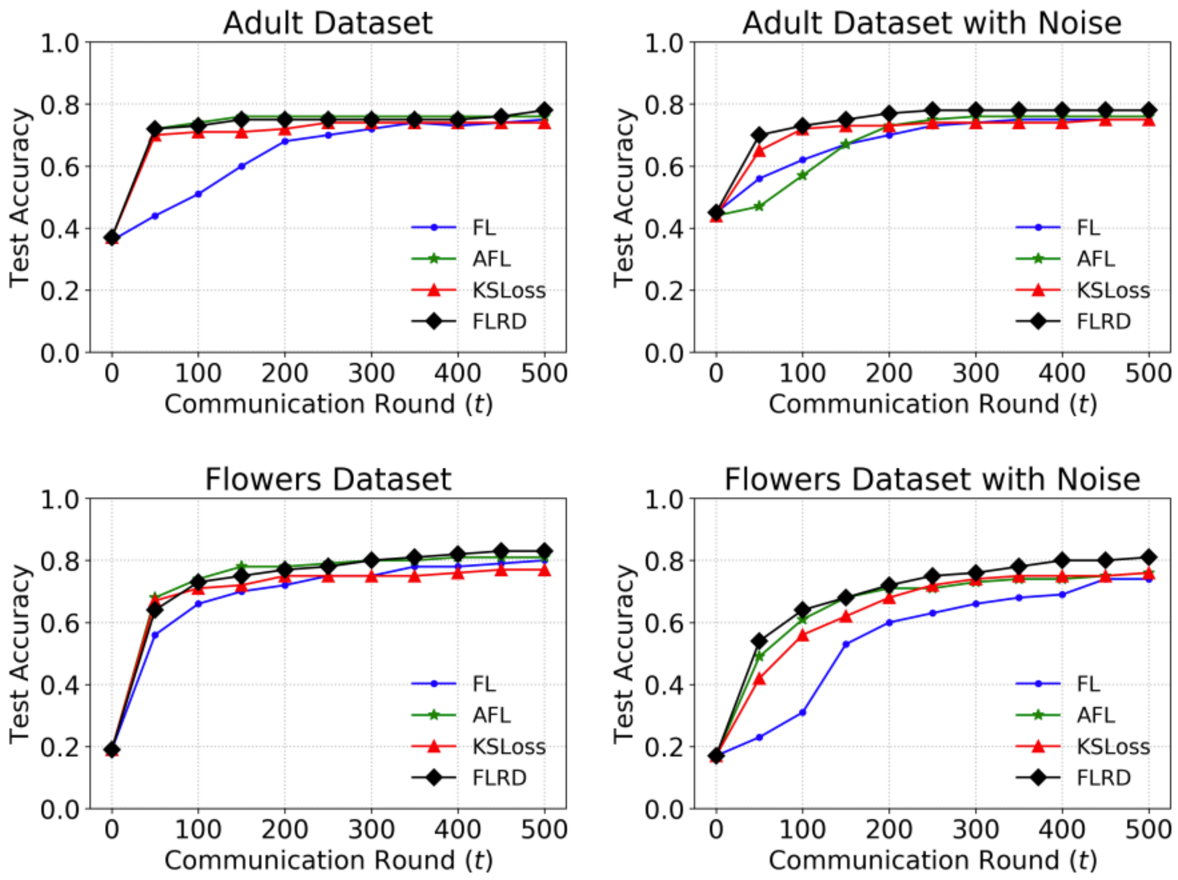


# Experiments

With closed-set label noise, the performance of $GLM$ is superior to that of other baselines with or without noise

- Random noise percentage at each client $\{5\%, 7\%, \cdots, 25\%\}$

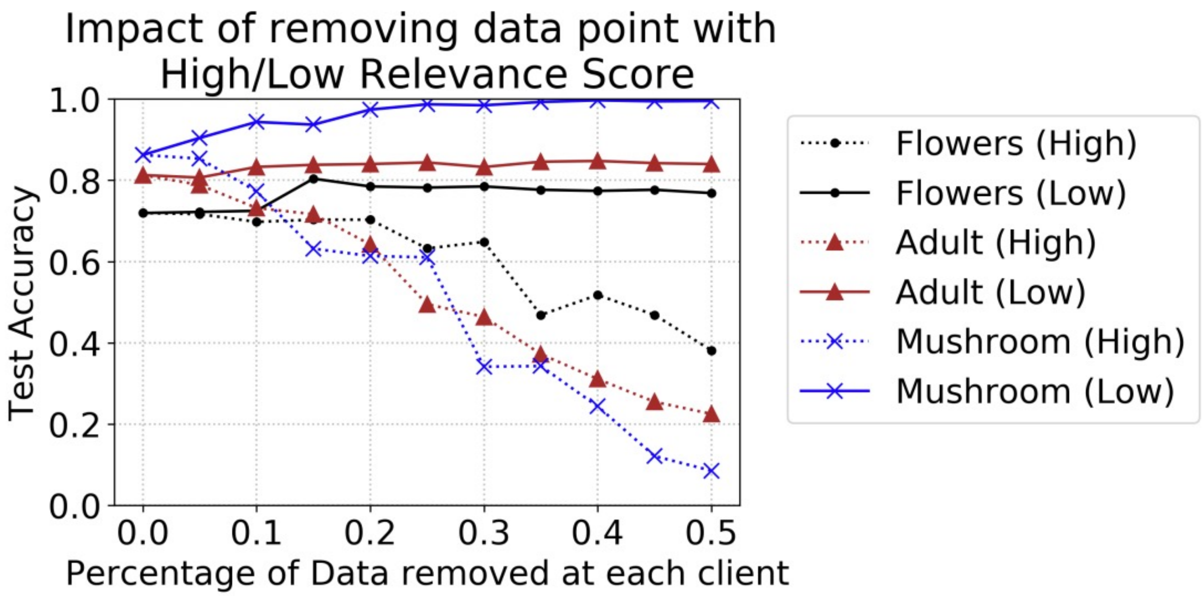



**w/o noise** **w/ noise**

# Experiments

Removing data samples with high relevance scores($RS$) deteriorates $GLM$ performance and with low $RS$ improves it

- 15% closed-set label noise

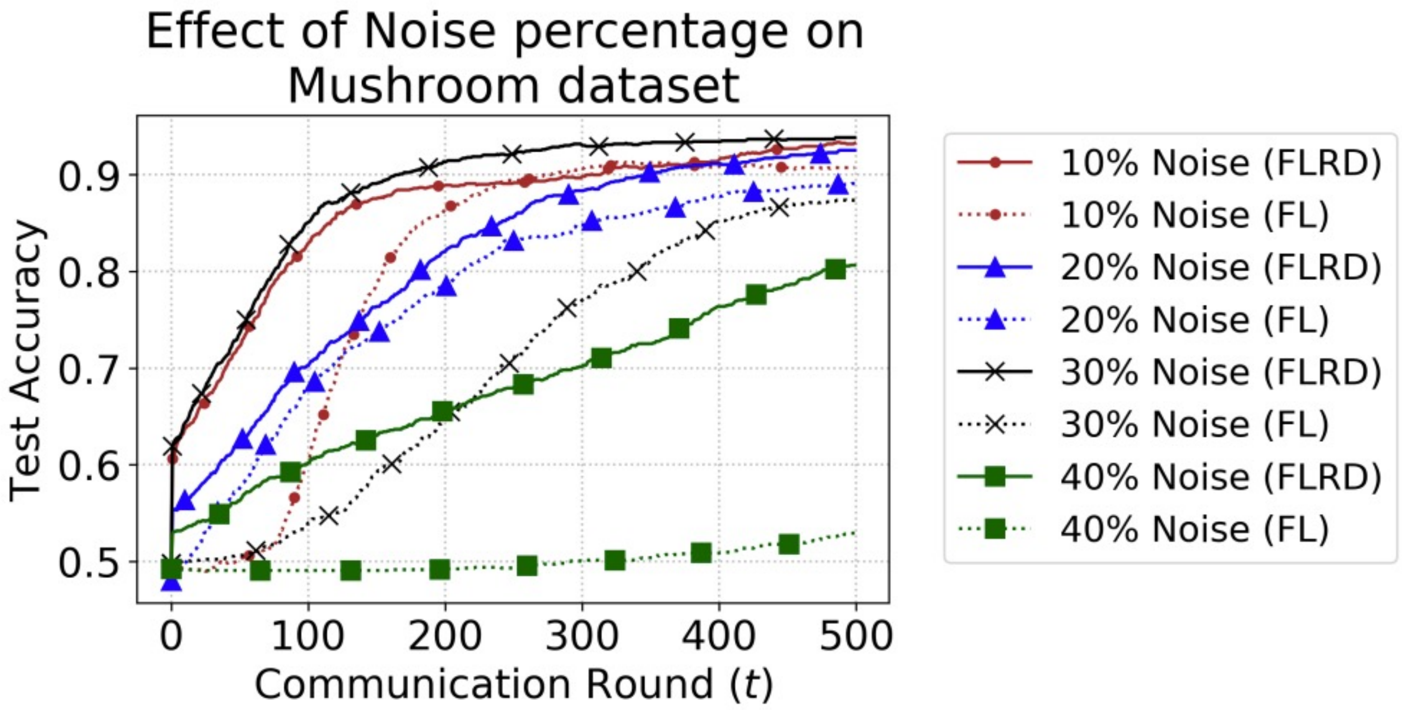



Removing as many as 50% samples with low $RS$ didn't affect the GLM

# Experiments

$FLRD$ is robust to both low and high noisy datasets

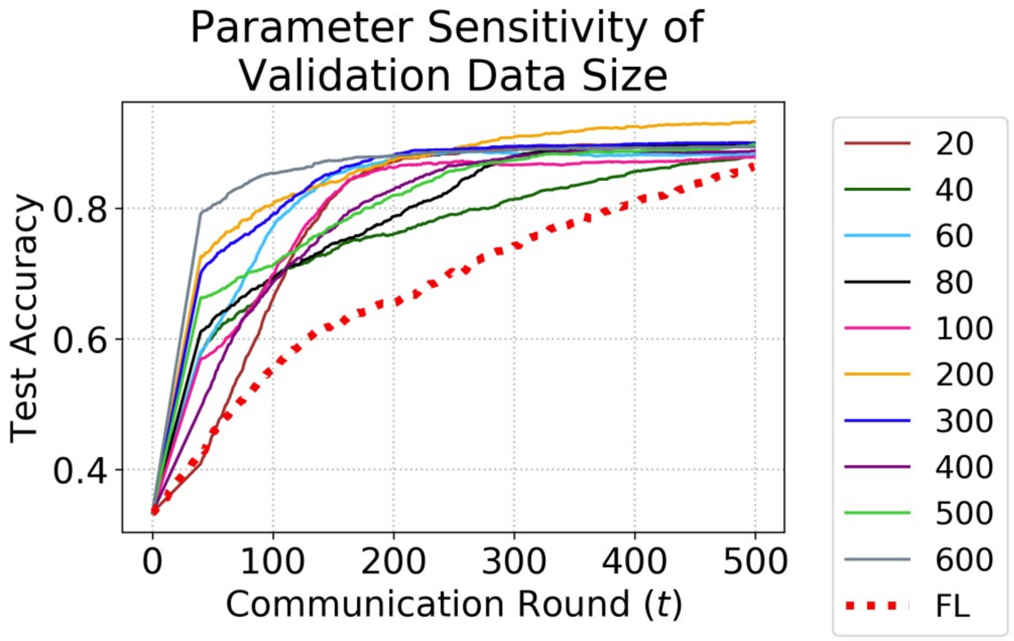
- closed-set label noise



# Experiments

$FLRD$ outperforms $FL$ (McMahan et al. 2017) when validation samples are scarce

- 20% closed-set label noise
- From 20 to 600 size of validation dataset size




Convergence of $GLM$ is fast when size of dataset is large