Online Meta-Learning for Model Update Aggregation in Federated Learning for Click-Through Rate Prediction

Xianghang Liu[1], Bartlomiej Twardowski[2], Tri Kurniawan Wijaya[2]

[1] Huawei Research Center London,
[2] Huawei Research Center Ireland

Corresponding Author: xianghang.liu@huawei.com

Federated Learning and Click Through Rate (CTR) Prediction

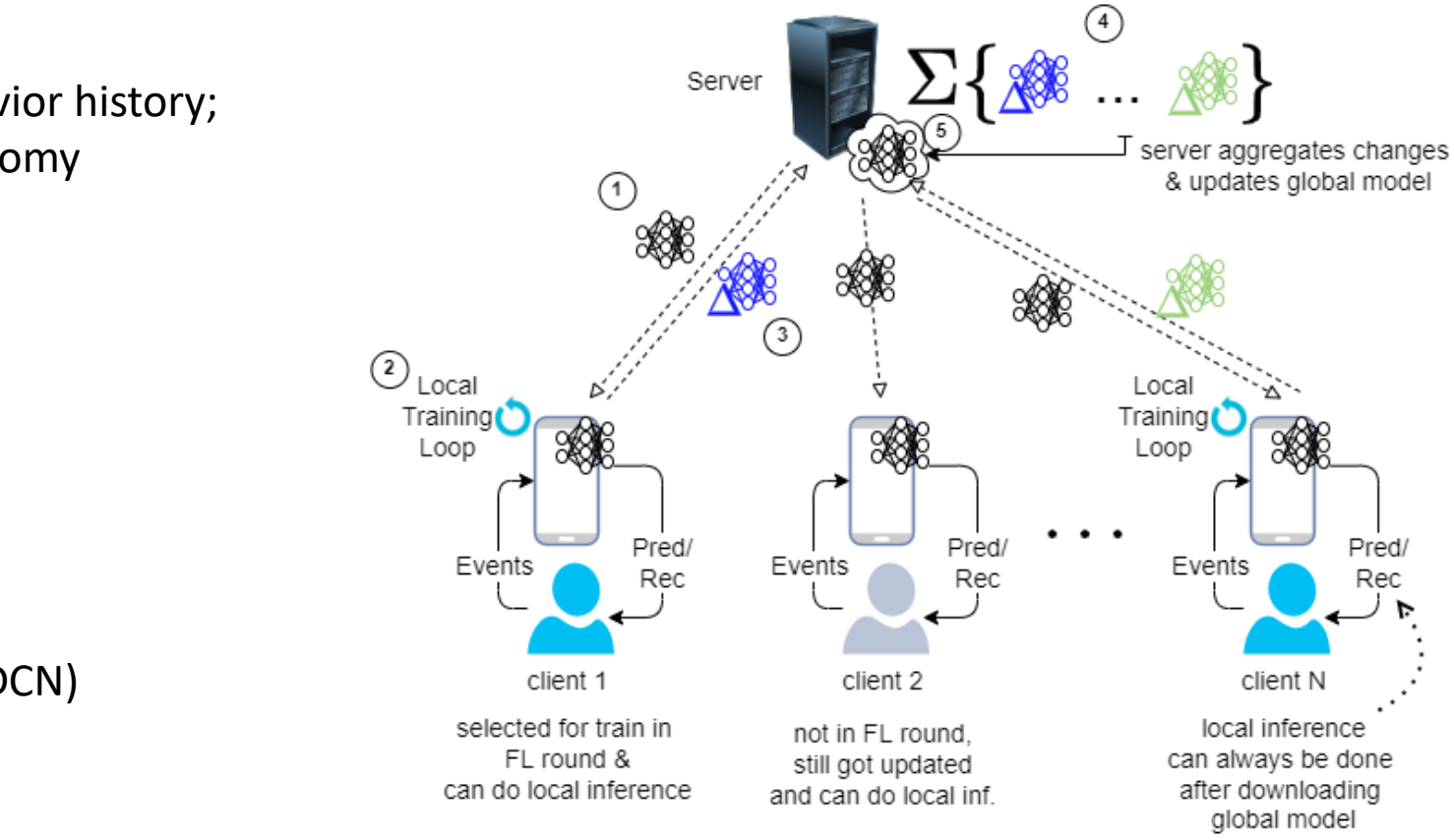
CTR Prediction in Ads Delivery

- Input: features of an ad impression
  - User features: profile and behavior history;
  - Ad features: id, creatives, taxonomy
  - Context features: time, location
- Output: probability of click

Model

- Embedding
- Explicit Interactions
- Fully-connected layers
- Biases for calibrations
- Example： Deep & Cross Network (DCN)

Federated Learning



FL Framework

Communication round t

- Local optimization

$$w_k^t = \text{opt-local}(w^t; D_k)$$

- Local model update aggregator

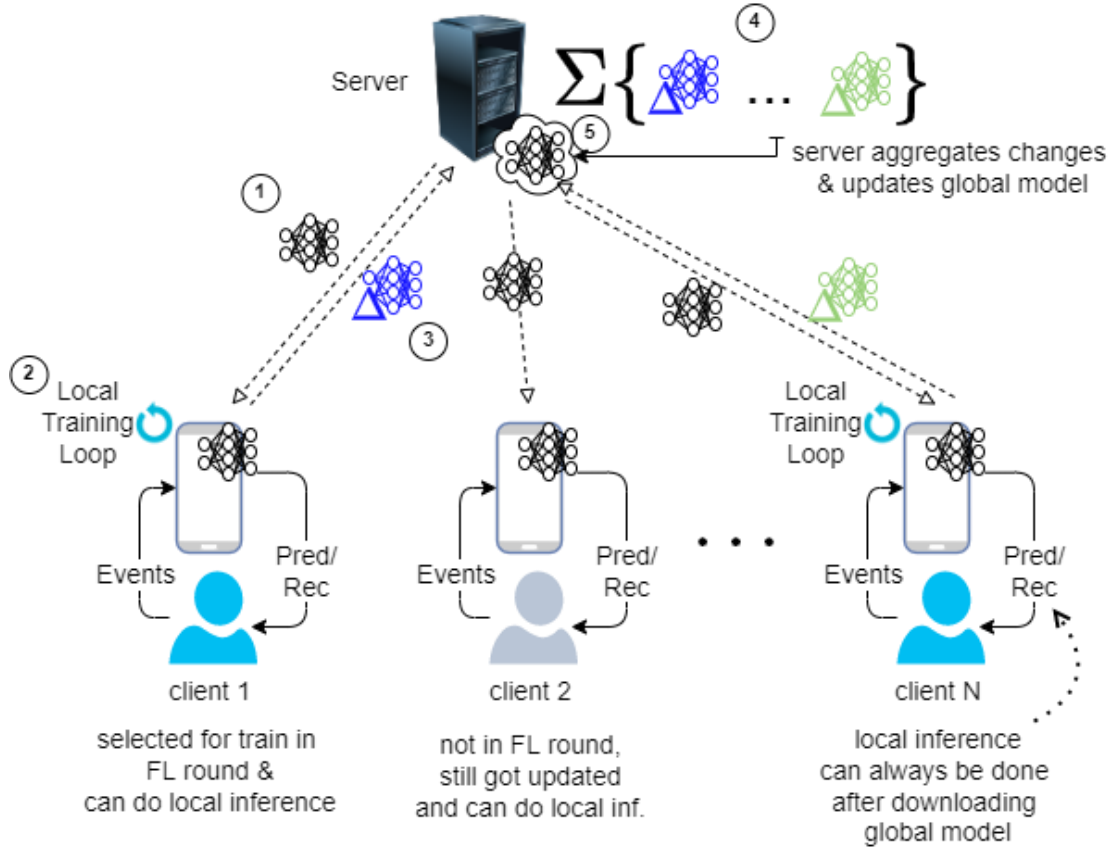
$$\Delta w^t = \text{agg}(\Delta W^t, Z^t)$$

$$\Delta W^t := \{\Delta w_k^t, k \in S^t\}$$

$$\Delta w_k^t := w_k^t - w^t$$

- Server optimizer

$$w^{t+1} = \text{opt-server}(w^t, \Delta w^t).$$



Examples

► **FedAvg** (McMahan et al., 2017)

$$\text{agg}(\Delta W^t, Z^t) = \frac{1}{n^t} \sum_{k \in S} n_k^t \cdot \Delta w_k^t, \tag{4}$$

$$\text{opt-server}(w, \Delta w) = w^t + \Delta w^t, \tag{5}$$

where $z_k^t = n_k$, for $k \in S^t$, n_k is the number of samples for client k, and $n^t = \sum_{k \in S^t} n_k$.

► **FedNova** (Wang et al., 2020)

$$\text{agg}(\Delta W^t, Z^t) = \Big(\sum_{k \in S^t} \tau_k \cdot \frac{n_k}{n^t} \Big) \cdot \sum_{k \in S} \frac{n_k}{n^t} \cdot \frac{1}{\tau_k} \cdot \Delta w_k^t, \tag{6}$$

where $z_k^t = [n_k, \tau_k]$, τ_k is the number of local gradient descent steps on client k. The function opt-server$(\cdot)$ is the same as that of FedAvg, i.e. equation (5).

► **FedAdam and FedAdagrad** (Reddi et al., 2021)

$$w^{t+1} = w^t + \gamma_s \cdot \frac{m^t}{\sqrt{M^t} + \epsilon}$$

$$m^t = \beta_1 \cdot m^{t-1} + (1 - \beta_1) \cdot \Delta w^t,$$

$$M^t = M^{t-1} + (\Delta w^t)^2, \text{ for } \textbf{FedAdagrad},$$

$$M^t = \beta_2 \cdot M^{t-1} + (1 - \beta_2) \cdot (\Delta w^t)^2, \text{ for } \textbf{FedAdam},$$

Learning to aggregate local model updates

- Motivation
  - Weighting of the client updates and client heterogeneity
  - Server learning rate tuning
  - Parameter-wise aggregation
- Aggregation strategy
  - Aware of client heterogeneity
  - Adaptive
  - Parameter-wise
- Learnable aggregation model

$$\Delta w^t[A] = \mathrm{agg}(\Delta W^t[A], Z^t[A]; \theta[A]) := \theta_s[A] \cdot \sum_{k \in S^t} \alpha_k[A] \cdot \Delta w_k[A],$$

$$\text{where } \alpha_k[A] = \mathrm{softmax}(f_\alpha(z_k^t[A]; \theta_\alpha[A])), \forall A \in \mathcal{P}.$$

$f_\alpha(z; \theta_\alpha)$ -- function to compute the client weights

$\mathcal{P}$ -- partition of the weight indices

- (meta) loss function

$$L^t(\theta^{t-1}) = \sum_{k \in S^t} L^t(\theta^{t-1}; D_k^{(q)})$$

$$L^t(\theta^{t-1}; D_k^{(q)}) := \sum_{\{x,y\} \in D_k^{(q)}} \ell\big(h(x; w^t); y\big)$$

- gradient computation

$$\frac{\partial L^t}{\partial \theta^{t-1}}(\theta^{t-1}) = \frac{\partial L^t}{\partial w^t}(w^t) \cdot \frac{\partial w^t}{\partial \theta^{t-1}}(\theta^{t-1})$$

$$g^t = \sum_{k \in S^t} \underbrace{\frac{\partial L_k^t}{\partial w^t}(w^t)}_{g_k^t :=}$$

$$g_k^t = \sum_{x,y \in D_k^{(q)}} \frac{\partial \ell}{\partial w^t}(f(x, w^t), y)$$

$$\frac{\partial L^t}{\partial \theta^{t-1}}(\theta^{t-1}) = \frac{\partial (g^t \cdot w^t)}{\partial \theta^{t-1}}(\theta^{t-1})$$

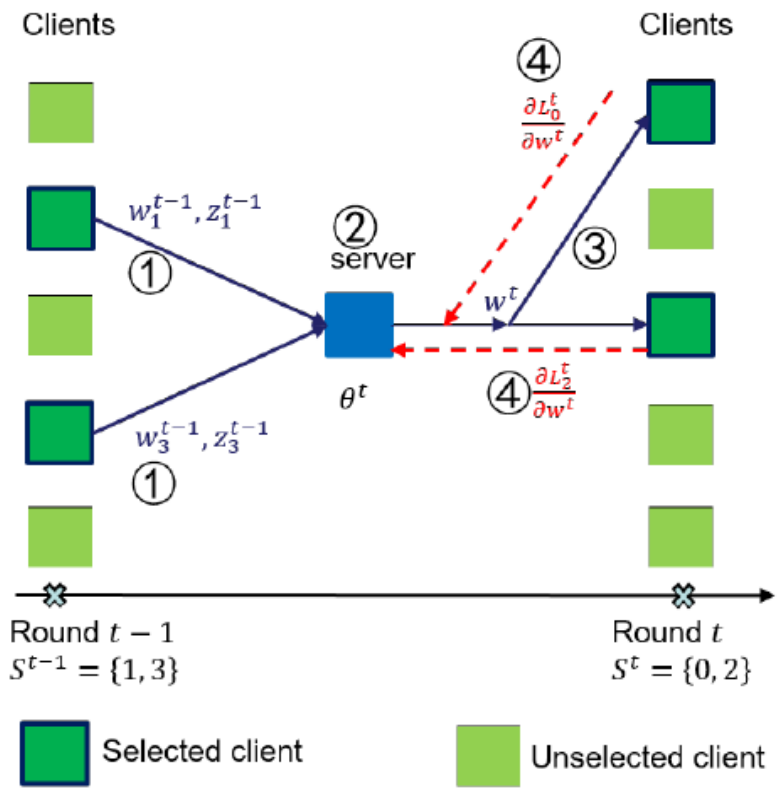


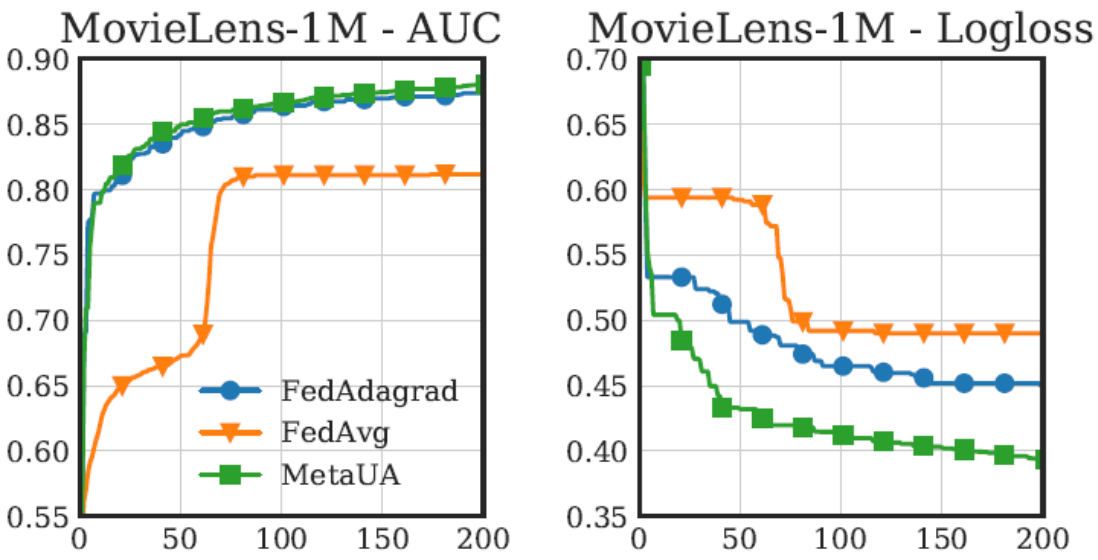
Figure 1: Overview of the gradient computation procedure in MetaUA: ① local model updates and attributes are sent from selected clients; ② local updates are aggregated using meta model; ③ new model is distributed to the selected clients; ④ server receives the gradients from the selected clients and computes the gradients for the meta-parameters.

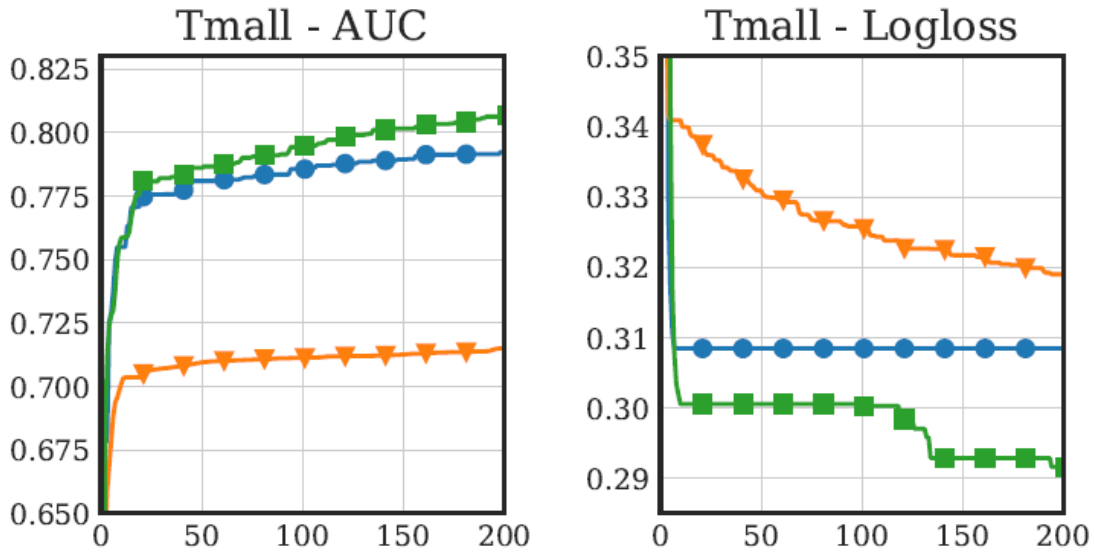
Experiments

- Base model: DCN-v2
- Fed-Adagrad as the server optimizer

Dataset	# Examples	# Features	Vocab Size	# Users	# Items	Density
MovieLens-1M	739012	7	13196	6040	3668	3.34
Tmall	1899378	4	44239	22284	17705	0.48
Yelp	530124	9	34462	22128	12232	0.20
Amazon-CDs	890824	3	31985	15592	16184	0.35

Table 1: Dataset statistics



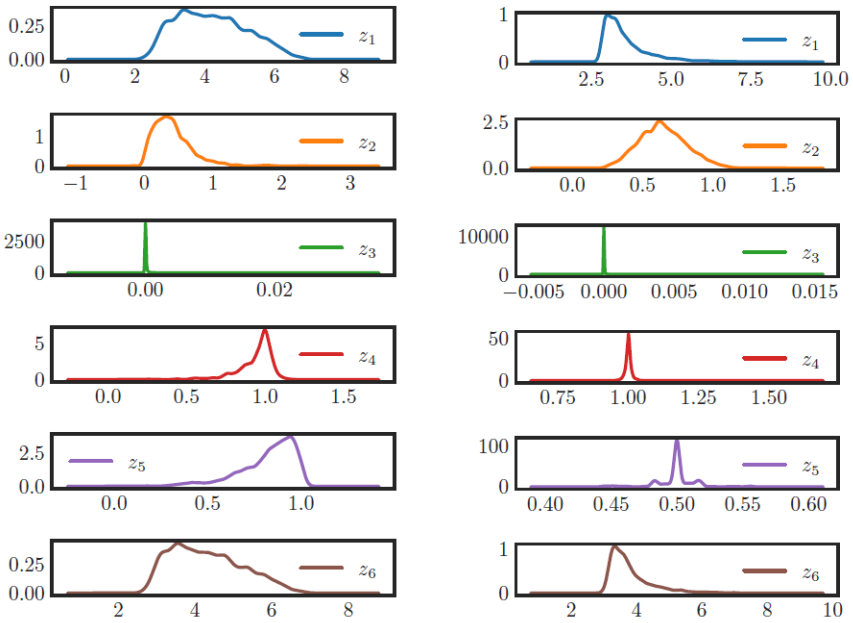


Selection of client attributes

1. Number of samples
2. Local loss values
3. Gradient norm
4. Ratio of loss value reduction after local SGD
5. Positive example ratio
6. Number of unique features in local datasets

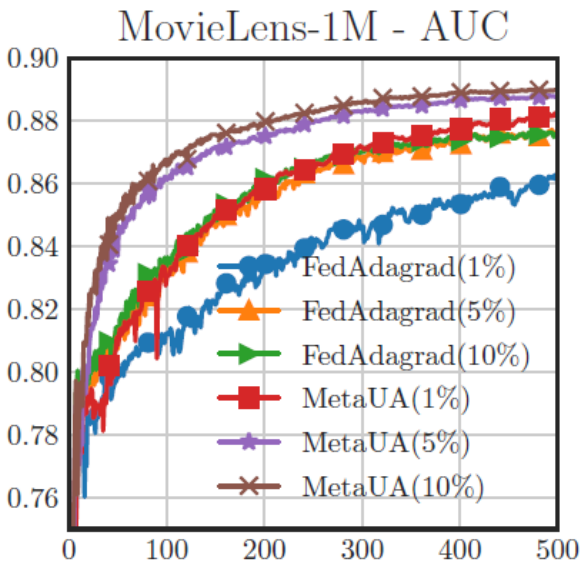
	AUC					Logloss				
rounds method	20	50	100	150	200	20	50	100	150	200
None	0.640	0.761	0.911	0.933	0.943	**0.661**	0.606	0.362	0.298	0.269
z_1	0.700	0.897	0.936	0.949	0.956	0.685	0.399	0.300	0.261	0.241
z_2	0.629	0.637	0.887	0.944	**0.964**	0.667	0.667	0.405	0.275	**0.206**
z_3	0.595	0.690	0.825	0.871	0.894	0.693	0.693	0.676	0.573	0.498
z_4	0.611	0.641	0.815	0.881	0.922	0.679	0.679	0.540	0.419	0.335
z_5	0.637	0.650	0.887	0.926	0.944	0.666	0.666	0.404	0.320	0.275
z_6	**0.712**	**0.899**	**0.937**	**0.950**	0.957	0.684	**0.391**	**0.296**	**0.257**	0.239
All	0.592	0.728	0.808	0.845	0.864	0.691	0.664	0.664	0.664	0.649

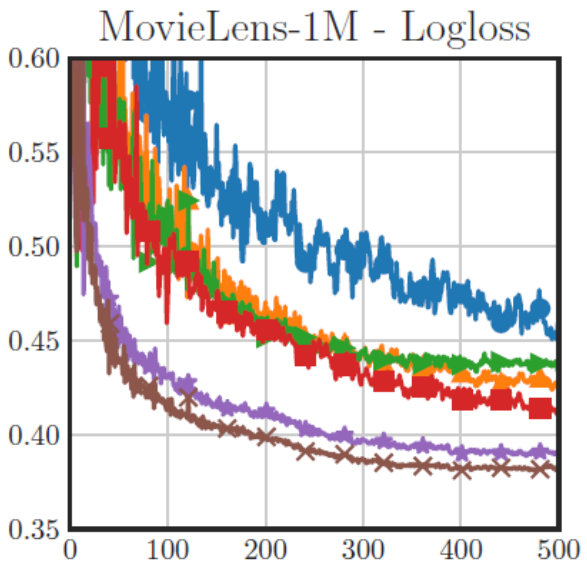
Table 3: Performance of MetaUA evaluated with different set of client's attributes Z on Amazon-CDs dataset.

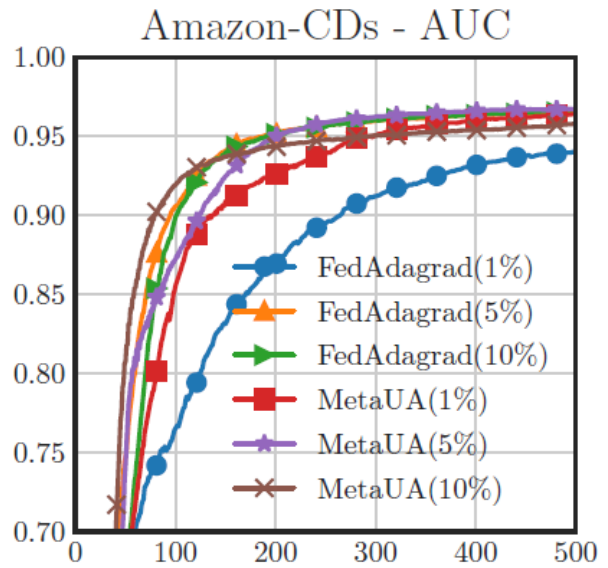


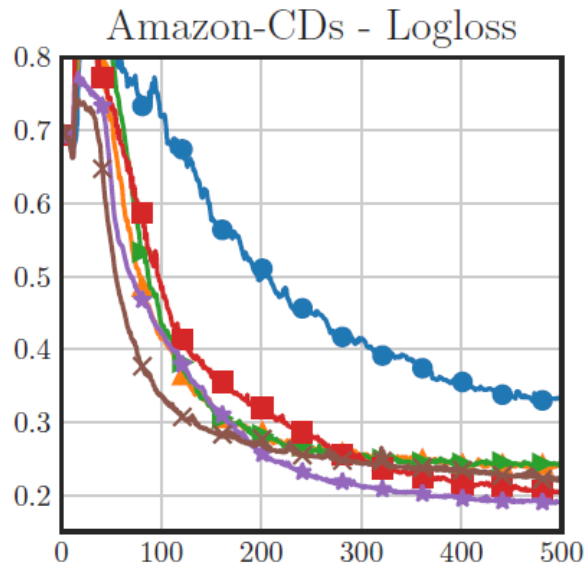
ml-1m Amazon-CD

Different fraction of clients







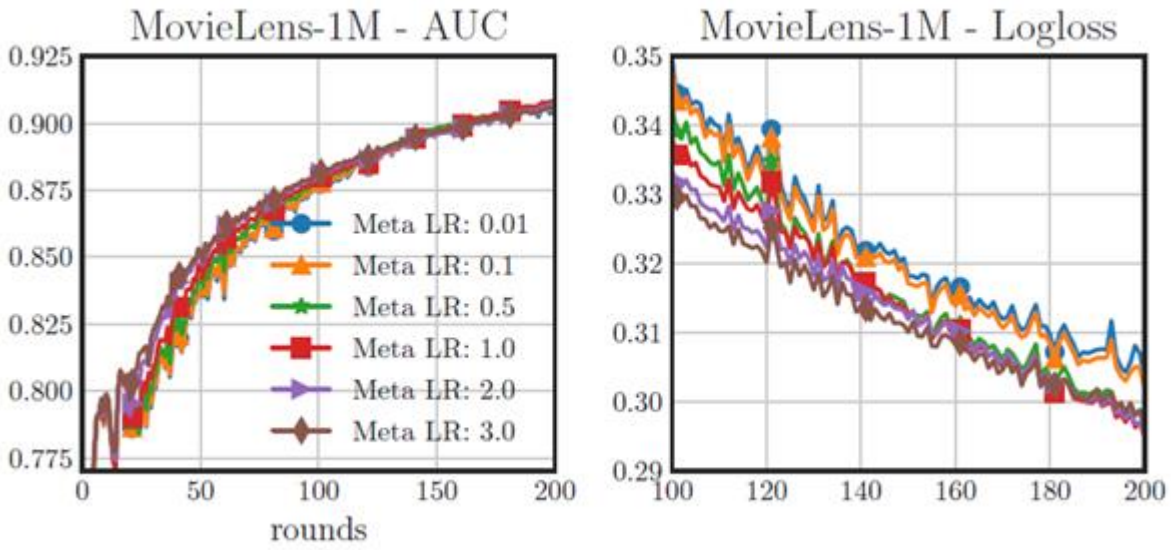


Ablation study

FL round	AUC 50	AUC 100	AUC 200	AUC 400	Logloss 50	Logloss 100	Logloss 200	Logloss 400
no adjustment	0.841	0.864	0.878	0.882	0.502	0.461	0.436	0.429
client weighting	0.844	0.862	**0.880**	0.885	0.454	0.439	0.417	0.413
server lr	0.846	0.865	0.877	0.881	0.484	0.441	0.415	0.413
client weighting + server lr	**0.850**	**0.867**	**0.880**	**0.889**	**0.432**	**0.413**	**0.393**	**0.382**

Table 4: Ablation study

Robust to learning rate



Conclusion

- For federated CTR prediction, we propose a method to learn to aggregate local model updates
  - Aware of client heterogeneity
  - Adaptive in the training process
  - Parameter-wise
- Outperforms the SOTA algorithms