# FLAIM: AIM-based Synthetic Data Generation in the Federated Setting

*Samuel Maddock*[†*] *Graham Cormode*[†‡] *Carsten Maple*[†]

[*] Email: s.maddock@warwick.ac.uk

[†] University of Warwick, [‡]Meta AI

## (Differentially Private) Synthetic Data

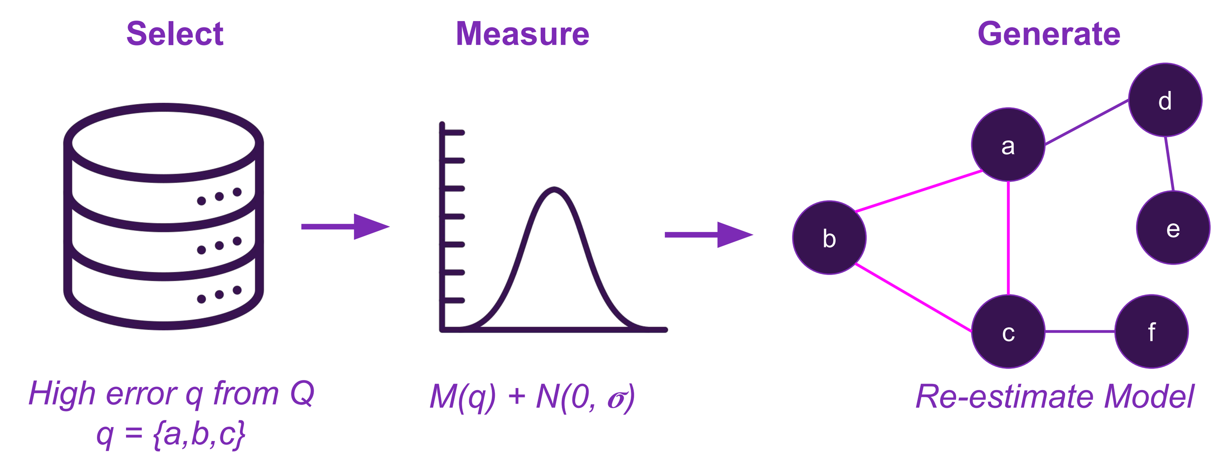
**Goal:** Produce "fake" data with the statistical properties of real data
**Key reason:** Privacy

- Allow general release of data for downstream tasks e.g., training models, analytics
- Many solutions when data is centralised e.g., GANs, LLMs, statistical models
- Methods prone to "memorisation" $\Rightarrow$ often produce verbatim copies of real data
- Prevention via **Differential Privacy (DP)** which adds carefully calibrated noise into training process

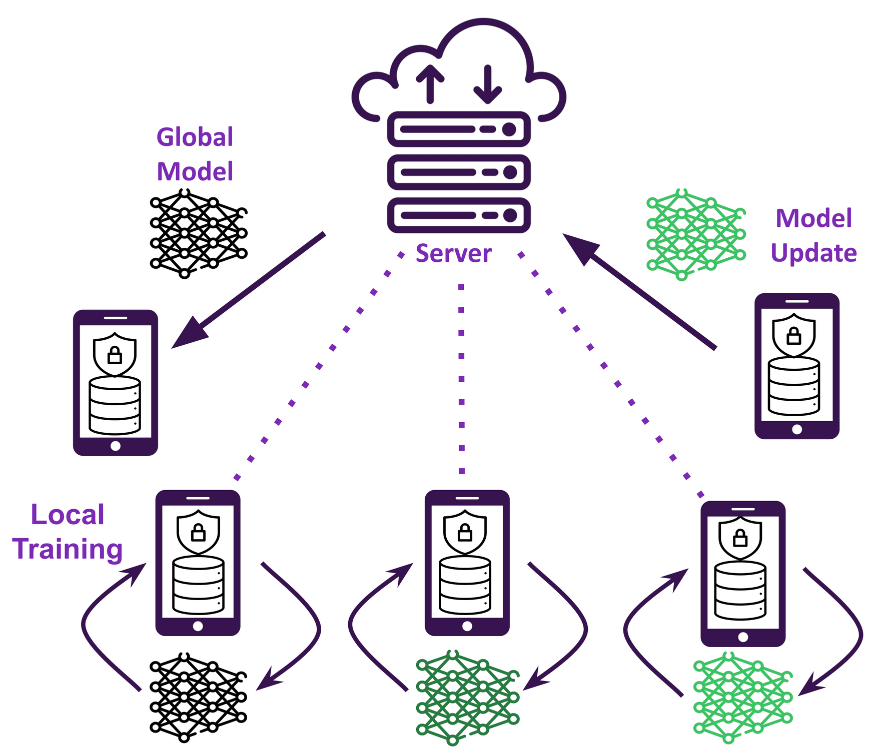
## "Select-Measure-Generate" Approach

Private *tabular* SDG methods follow **"Select-Measure-Generate"** approach. Given a workload of queries $Q$, for $t = 1, \ldots, T$:

1. **Select** query $q \in Q$ with worst error (noisily) with utility scores $u(q)$
2. **Measure** chosen query $q$ under calibrated Gaussian noise
3. **Generate** data and update model to learn (noisy) measured queries




## Our Work: Federated Synthetic Data




**Key Question:** How do we federate AIM? = How to federate "Select-Measure-Generate" paradigm?

**Distributed setting:** All clients participate over a single (or few) rounds. Assume all participants are available.

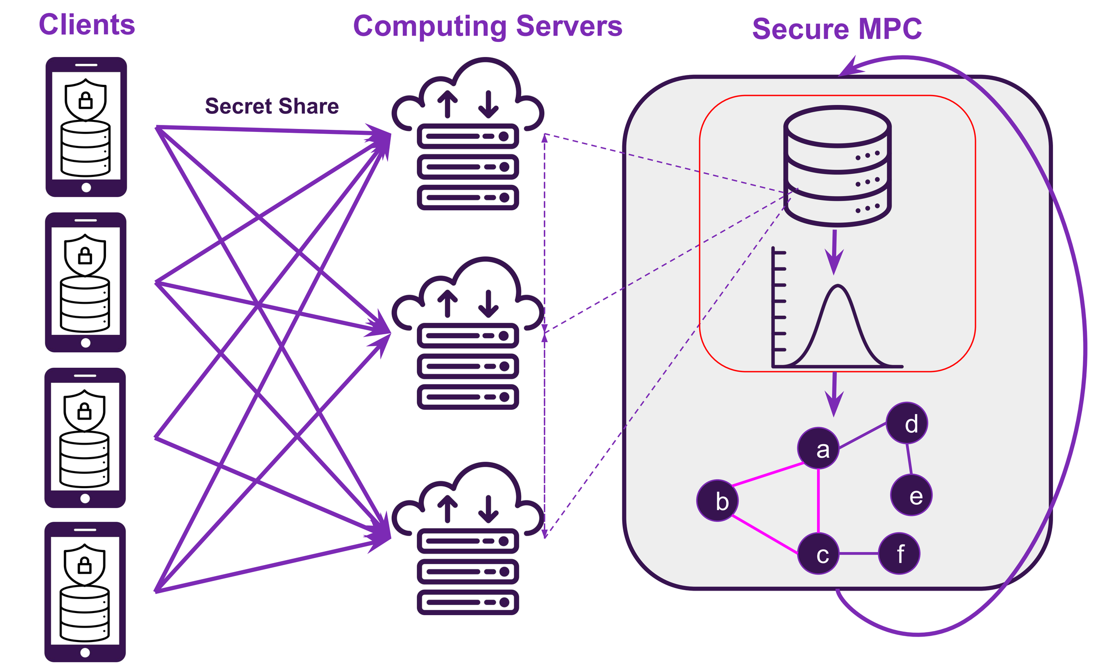
**Federated setting:** Client participation is subset of true population (e.g., dropout, availability) and client data exhibits strong heterogeneity (e.g., distribution or label skew).

## Our Work: DistAIM

Prior work by **Pereira et al.** consider running MWEM in a **distributed** setting via **Secure Multi-party Computation (MPC)** with 2/3 computing servers where:

- Each client secret-shares workload answers to computing servers
- Computing servers jointly emulate central algorithm under MPC

We apply this framework to AIM $\Rightarrow$ DistAIM, resulting in $\downarrow$ error but $\uparrow$ overhead




## Our Work: FLAIM

Alternative: FL analog via lighweight MPC $\Rightarrow$ secure-aggregation + noise

- **Select**: Each (available) client performs local "select" steps (under LDP)
- **Measure**: Server uses secure-aggregation and adds noise to measurements
- **Generate**: Post-processing by server over noisy measurements (unchanged)

Avoids (significant) MPC overhead (secure exponential mechanism), BUT problem:

$$u(q; D_k) \propto \|M_q(D_k) - M_q(\hat{D})\|_1 \implies \text{"select" is biased by local heterogeneity}$$

Leads to three distinct approaches:

- NaiveFLAIM - no utility score modification
- AugFLAIM (Oracle) - access to true heterogeneity $\tau_k(q) := \|M_q(D_k) - M_q(D)\|$
- AugFLAIM (Private) - private proxy for heterogeneity $\tilde{\tau}_k(q)$

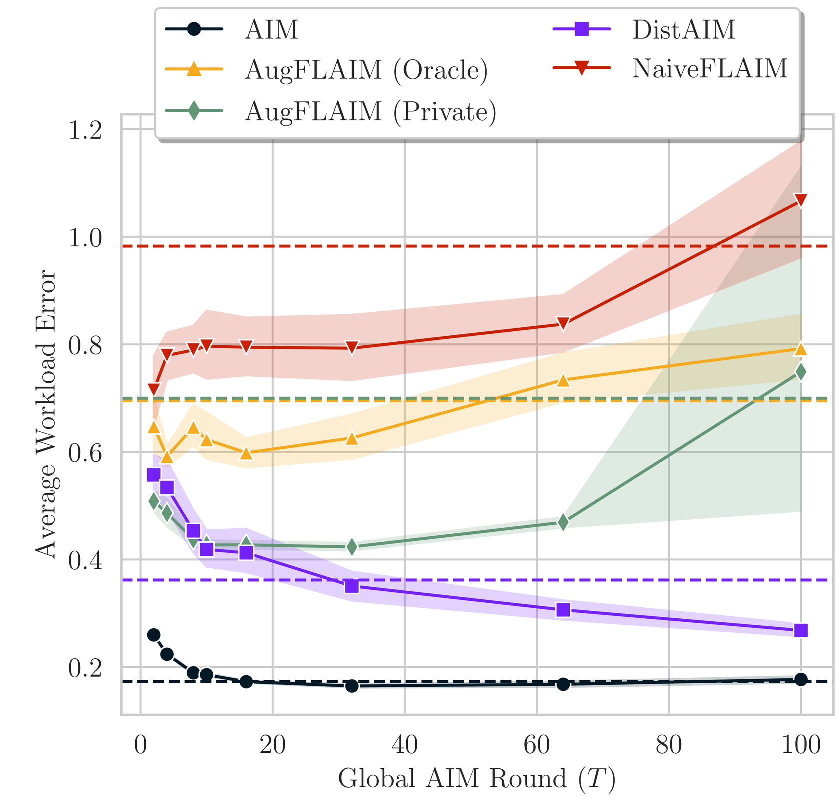
## Evaluation

We measure the **Average L1 error** over the workload, **Negative Log-Likelihood (NLL)** and **Test AUC** of a GBDT trained on generated synthetic data. We federate benchmark tabular datasets to induce heterogeneity in two ways:

- **"Cluster"**: Perform dimensionality reduction (UMAP) on training data and cluster embeddings to form client partitions. Synthetic data model trained on original data.
- **"Label-skew"**: Sample labels from Dirichlet($\beta$) where small $\beta$ results in large skew as in prior FL work.

Code available `https://github.com/Samuel-Maddock/flaim`

## FLAIM: Utility



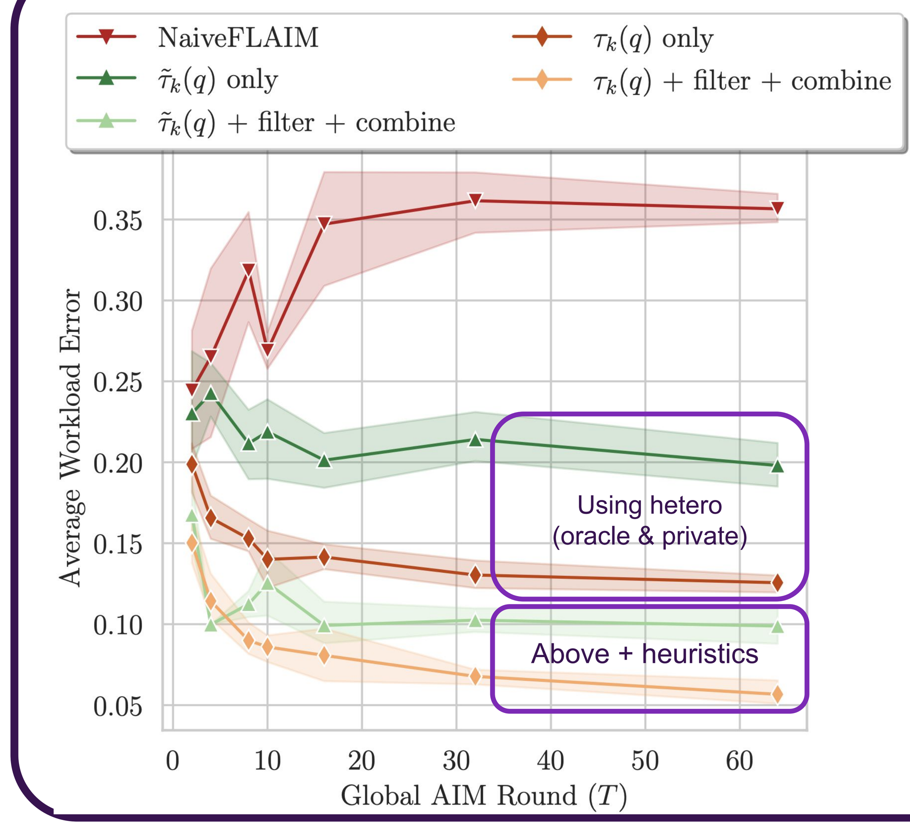

**Key Tradeoff:** Overhead vs. utility. Varying number of global rounds $T$, $\varepsilon = 1$ on Adult.

**AugFLAIM (Private)** error matches DistAIM when T is small

As T increases, DistAIM closes gap with central AIM

At large T, **AugFLAIM (Private)** has large error due to higher per-round noise.

## FLAIM: Ablation




**Why does AugFLAIM (Private) perform so well?**

Study penalisation methods with and without heuristics.

**Findings:** Using heterogeneity to penalise query selections is important in reducing error but estimating 1-way marginals at each round is also key.

## FLAIM: Communication Trade-off

| Dataset | $T$ (↑) | Throughput (↑) | Err (↓) | NLL (↓) |
|---|---|---|---|---|
| Adult | 2× | 1300× (80 / 0.06) | 58% | 11% |
| Magic | 3.2× | 1643× (80 / 0.04) | 20% | 14% |
| Census | 1.5x | 64x (29.6 / 0.46) | 79% | 33% |
| Intrusion | 2.5x | 366x (101 / 0.28) | 82% | 52% |
| Marketing | 2.0x | 97x (18 / 0.19) | 77% | 35% |
| Credit | 1.0x | 167x (93 / 0.55) | 45% | 6% |
| Covtype | 1.25x | 10x (7.6 / 0.76) | 64% | 3% |

Compare DistAIM vs. FLAIM at optimal $T$ for best utility while studying **throughput** = average client sent & received communication:

- If T is **small**, utility of AugFLAIM >= DistAIM
- If T is **large**, utility of DistAIM >= AugFLAIM
- ∀ T, overheads of AugFLAIM <= DistAIM