Towards Federated Foundation Models:
Scalable Dataset Pipelines for Group-Structured Learning

Zachary Charles*, **Nicole Mitchell***, **Krishna Pillutla***, Michael Reneer, Zachary Garrett

Google Research

arxiv.org/abs/2307.09619
github.com/google-research/dataset_grouper

FL research is inhibited by data scale

Research datasets for FL are often:

- Small
- Difficult to create and customize
- Unsuitable for LLMs

At foundation scale, FL = training with **group-structured data**

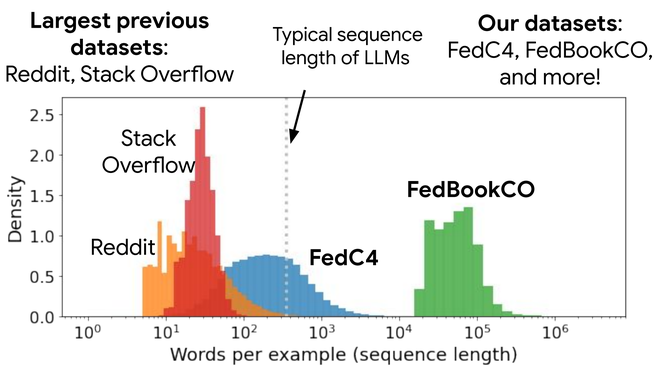
Need for **large-scale**, **group-structured datasets**
+ scalable, flexible and efficient pipelines to create them

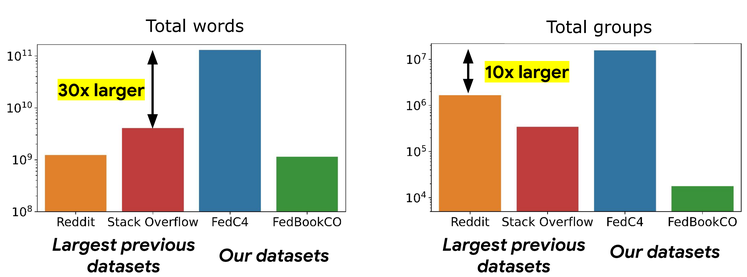
Dataset Grouper - a library for creating group-structured datasets

- **Scalable**: can handle *millions* of clients ✅
- **Flexible**: *any* custom partitioning of *any* TFDS/HuggingFace dataset ✅
- **Platform-agnostic**: works with *any* existing or future platform, e.g. TF, PyTorch, JAX, NumPy, ... ✅

Enables new federated data for LLMs

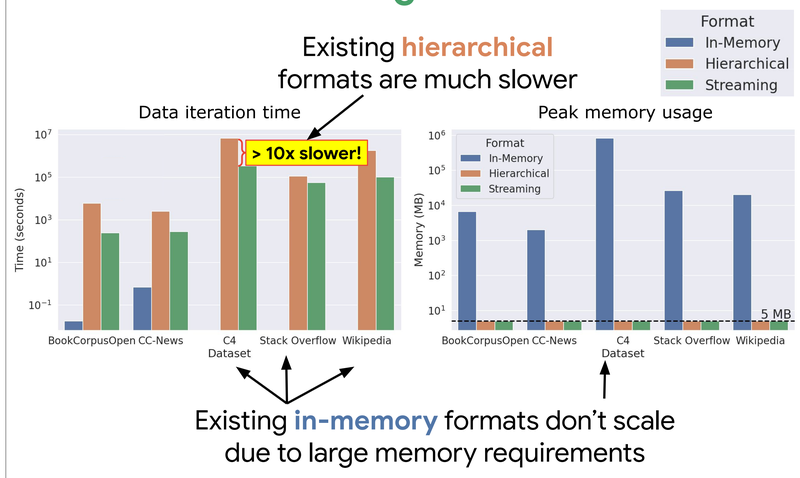
Longer text sequences, more words, more clients





Core Features

1. Scalable streaming data loaders



2. Flexible partitioning of existing datasets

```python
import dataset_grouper as dsgp
import tensorflow_datasets as tfds

dataset_builder = tfds.builder("mnist")
```

Load *any* TFDS or HuggingFace dataset

```python
def get_label_fn(x):
  label = x["label"].numpy()
  return str(label).encode("utf-8")
```

Define *any* partition function

```python
mnist_pipeline = dsgp.tfds_to_tfrecords(
    dataset_builder=dataset_builder,
    split="train",
    get_key_fn=get_label_fn,
    file_path_prefix=...
)

with beam.Pipeline() as root:
  mnist_pipeline(root)
```

Dataset Grouper does the rest!

3. Platform-agnostic group iterators

```python
import dataset_grouper as dsgp

partitioned_dataset = dsgp.PartitionedDataset(
        file_pattern=...,
        tfds_features="mnist"
)
```

Load a partitioned dataset

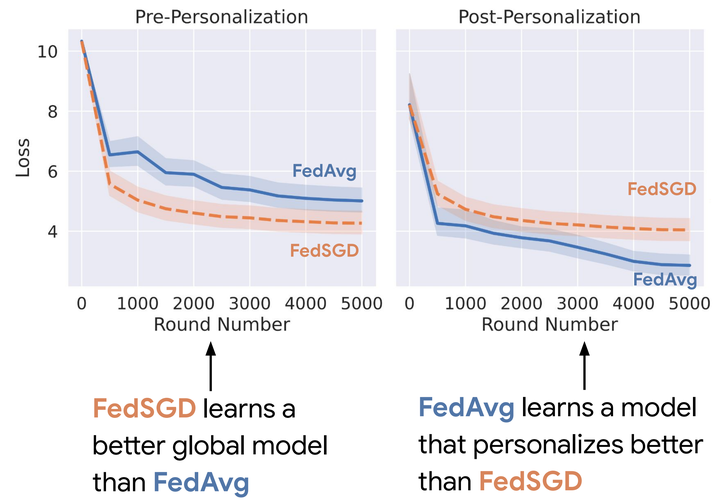
```python
for x in partitioned_dataset.build_group_stream():
    # ds is an iterable of examples.
    for example in x.as_numpy_iterator():
        # Process this example.
```

Use on *any* platform (e.g. TF, Pytorch, JAX, NumPy, ...)

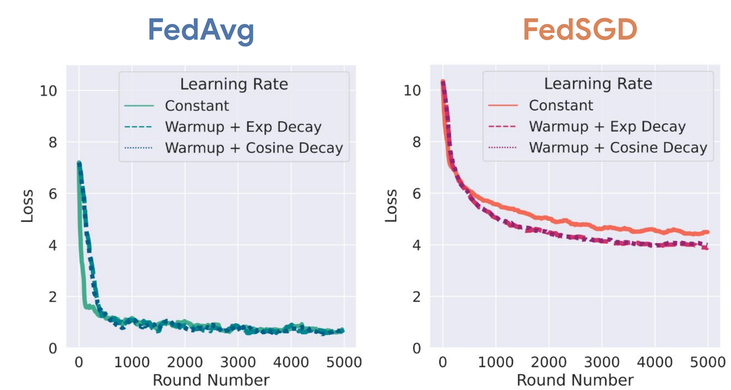
FL simulations at scale

Model: O(100M) and O(1B) transformer
Train: FedC4
Eval: FedBookCO

1. FedAvg is a meta-learner!



2. FedAvg is robust to server learning rate schedules!



Installation:

```
pip install dataset-grouper
```

Pull requests welcome!

Code

Paper