Towards Federated Foundation Models
Scalable Pipelines for Group-Structured Learning

*NeurIPS 2023 (Datasets & Benchmarks Track)*

Zachary Charles
Nicole Mitchell
Krishna Pillutla
Michael Reneer
Zachary Garrett

Google Research
Federated learning research has a small data problem

Research datasets for FL are often:
- Small
- Difficult to create/customize
- Unsuitable for foundation models, especially LLMs

Need for large-scale, group-structured datasets:
- Scalable, flexible and efficient pipelines
Our contributions

Dataset Grouper 🐟

Library for creating group-structured datasets.

- **Scalable:** can handle millions of clients ✅
- **Flexible:** any custom partition function on any TFDS/HuggingFace dataset ✅
- **Platform-agnostic:** works with TF, PyTorch, JAX, NumPy, ...

Federated training of $O(100M)$ and $O(1B)$ parameter models

What happens in FL at LLM-scale?

- FedSGD vs. FedAvg
- Global vs. local performance

```
pip install dataset-grouper
```
Our contributions

**Dataset Grouper**

Library for creating group-structured datasets.

- **Scalable:** can handle millions of clients
- **Flexible:** any custom partition function on any TFDS/HuggingFace dataset
- **Platform-agnostic:** works with TF, PyTorch, JAX, NumPy, ...

**Scalable:** largest federated datasets to-date

`pip install dataset-grouper`
New federated LLM datasets: longer sequences

Largest previous datasets: Reddit, Stack Overflow

Typical sequence length of LLMs

Our datasets: FedC4, FedBookCO

![Histogram comparing sequence lengths of different datasets](image)
New federated LLM datasets: more words & groups

**Total words**

- Reddit: $10^8$
- Stack Overflow: $10^9$
- FedC4: $10^{11}$ (30x larger)
- FedBookCO: $10^9$

**Total groups**

- Reddit: $10^8$
- Stack Overflow: $10^6$
- FedC4: $10^7$ (10x larger)
- FedBookCO: $10^4$
Our contributions

Dataset Grouper

Library for creating group-structured datasets.

- **Scalable**: can handle millions of clients
- **Flexible**: any custom partition function on any TFDS/HuggingFace dataset
- **Platform-agnostic**: works with TF, PyTorch, JAX, NumPy, ...

pip install dataset-grouper
Scalable **streaming** data loaders

Centralized → federated learning

Existing **hierarchical** format is much slower

Existing **in-memory** format doesn’t scale due to its large memory requirement
Our contributions

Dataset Grouper

Library for creating group-structured datasets.

- **Scalable:** can handle millions of clients
- **Flexible:** any custom partition function on any TFDS/HuggingFace dataset
- **Platform-agnostic:** works with TF, PyTorch, JAX, NumPy, ...

Flexible partitioning of existing datasets

pip install dataset-grouper
import dataset_grouper as dsgp
import tensorflow_datasets as tfds

Load any TFDS/HuggingFace dataset

dataset_builder = tfds.builder("mnist")

Any user-defined partition function

def get_label_fn(x):
    label = x["label"].numpy()
    return str(label).encode("utf-8")
```python
import dataset_grouper as dsgp
import tensorflow_datasets as tfds

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)
```
Our contributions

Dataset Grouper

Library for creating group-structured datasets.

- **Scalable**: can handle millions of clients ✅
- **Flexible**: any custom partition function on any TFDS/HuggingFace dataset ✅
- **Platform-agnostic**: works with TF, PyTorch, JAX, NumPy, ... ✅

Platform-agnostic group iterators

`pip install dataset-grouper`
Load a partitioned dataset

```python
partitioned_dataset = dsgp.PartitionedDataset(
    file_pattern=...,  
    tfds_features="c4"  # Or any other TFDS dataset name.
)
```

Platform-agnostic iterators

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

Federated training of O(100M) and O(1B) parameter models

What happens in FL at LLM-scale?

- FedSGD vs. FedAvg
- Global vs. local performance
FedAvg is a meta-learner!

FedAvg learns a model that personalizes better than FedSGD

FedSGD learns a better global model than FedAvg

Model: 128M param LM
Train: FedC4
Eval: FedBookCO
Thank you!

https://github.com/google-research/dataset_grouper
Pull requests welcome!

pip install dataset-grouper
Thank you!