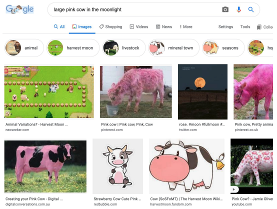
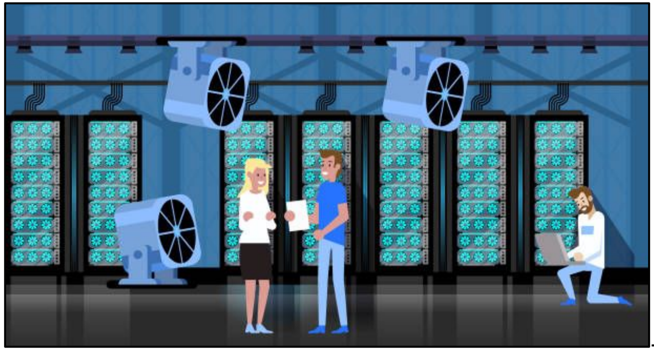
2nd Workshop on Federated Learning for Computer Vision @ CVPR '23
Monday, 19th June 2023 – Vancouver, Canada and *virtually (on zoom)*…

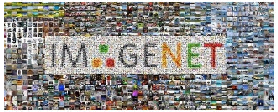
Machine Learning and the Data Center: A Dangerous Dead End

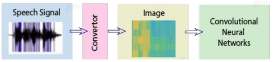
Nicholas D. Lane
University of Cambridge | Flower Labs
@niclane7

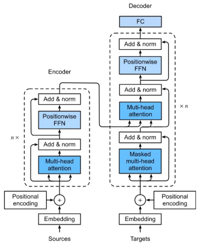
Machine Learning and Data Centers

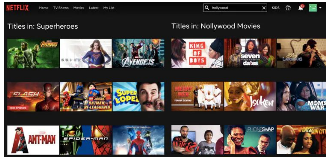












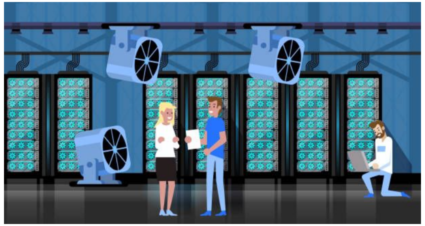
Born gentle
PROUD mothers, please forgive us if we too feel something of the pride of a new parent. For new Philip Morris, today's Philip Morris, is delighting smokers everywhere. Enjoy the gentle pleasure, the *fresh unfiltered flavor*, of this new cigarette, born gentle, then refined to special gentleness in the making.
Philip Morris
Cigarettes
King Size or Regular

a hundred years old!"
● It's a fact—a warm and wonderful fact—that this five-year-old child, or your own child, has a life expectancy almost a whole decade longer than was her mother's, and a good 18 to 20 years longer than that of her grandmother. Not only the expectation
Thank medical science for that.
your doctor and thousands like him…
ing ceaselessly, often with little or no
recognition… that you and yours
joy a longer, better life.
According to a recent Nationwide survey:
More Doctors smoke Camels
than any other cigarette!
NOT ONE but three outstanding independent research organizations conducted this survey. And they asked not just a few thousand, but 113,597, doctors from coast to coast to name the cigarette they themselves preferred to smoke.
The answers came in by the thousands… from general physicians, diagnosticians, surgeons—yes, and nose and throat specialists too. The most-named brand was Camel.
If you are not now smoking Camels, try them. Compare them critically. See how the full, rich flavor of Camel's costlier tobaccos suits your taste. See how the cool mildness of a Camel suits your
CAMEL
THE "T-ZONE" TEST WILL TELL YOU


TRADE MARK
REG. U. S.
PAT. OFF.
NET
Dutch Boy
WHITE LEAD

Why is it a dead end?

(aka why should we innovate away from it?)

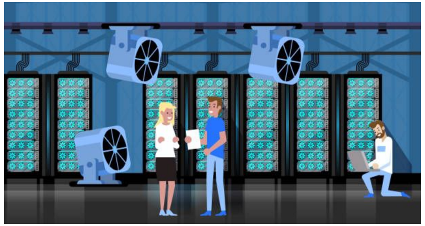


- Reliance on static tiny biased datasets
- Centralization of control
- Energy and carbon footprint
- Safety: Brittleness of systems
- Energy and latency of data to model
- User privacy and control
- Increasing local ML compute

HIPAA
COMPLIANT

Why is it a dead end?

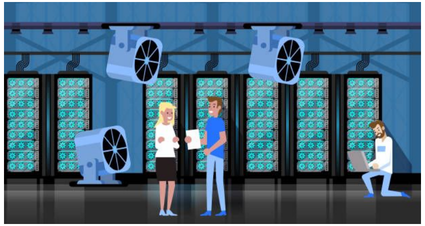
(aka why should we innovate away from it?)



- Reliance on static tiny biased datasets
- Centralization of control
- Energy and carbon footprint
- Safety: Brittleness of systems
- Energy and latency of data to model
- User privacy and control
- Increasing local ML compute

Why is it a dead end?

(aka why should we innovate away from it?)



- Reliance on static tiny biased datasets
- Centralization of control
- Energy and carbon footprint
- Safety: Brittleness of systems
- Energy and latency of data to model
- User privacy and control
- Increasing local ML compute

Model	Hardware	Power (W)	Hours	kWh·PUE	CO_2e	Cloud compute cost
$\text{Transformer}_{base}$	P100x8	1415.78	12	27	26	\$41–\$140
Transformer_{big}	P100x8	1515.43	84	201	192	\$289–\$981
ELMo	P100x3	517.66	336	275	262	\$433–\$1472
BERT_{base}	V100x64	12,041.51	79	1507	1438	\$3751–\$12,571
BERT_{base}	TPUv2x16	—	96	—	—	\$2074–\$6912
NAS	P100x8	1515.43	274,120	656,347	626,155	\$942,973–\$3,201,722
NAS	TPUv2x1	—	32,623	—	—	\$44,055–\$146,848
GPT-2	TPUv3x32	—	168	—	—	\$12,902–\$43,008

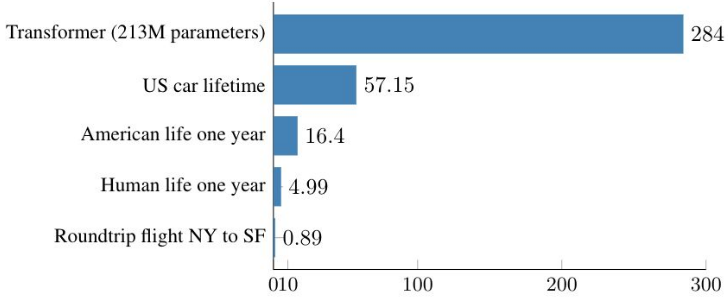
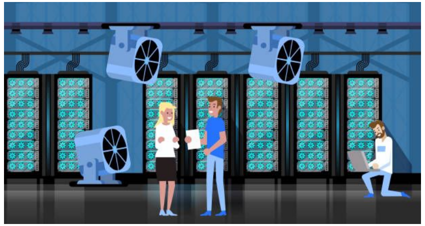


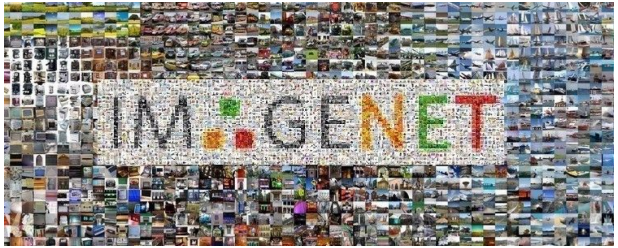
Figure Acknowledgement: "Energy & policy considerations for deep learning in NLP"

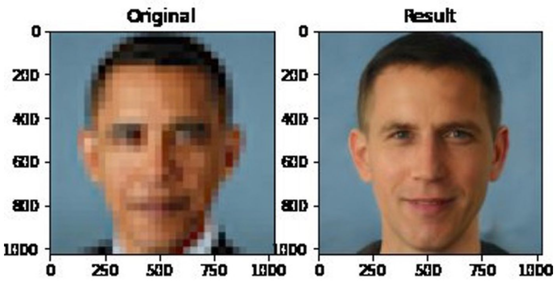
Why is it a dead end?

(aka why should we innovate away from it?)



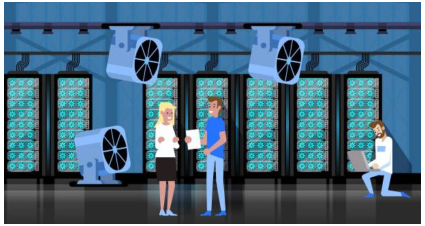
- Reliance on static tiny biased datasets
- Centralization of control
- Energy and carbon footprint
- Safety: Brittleness of systems
- Energy and latency of data to model
- User privacy and control
- Increasing local ML compute



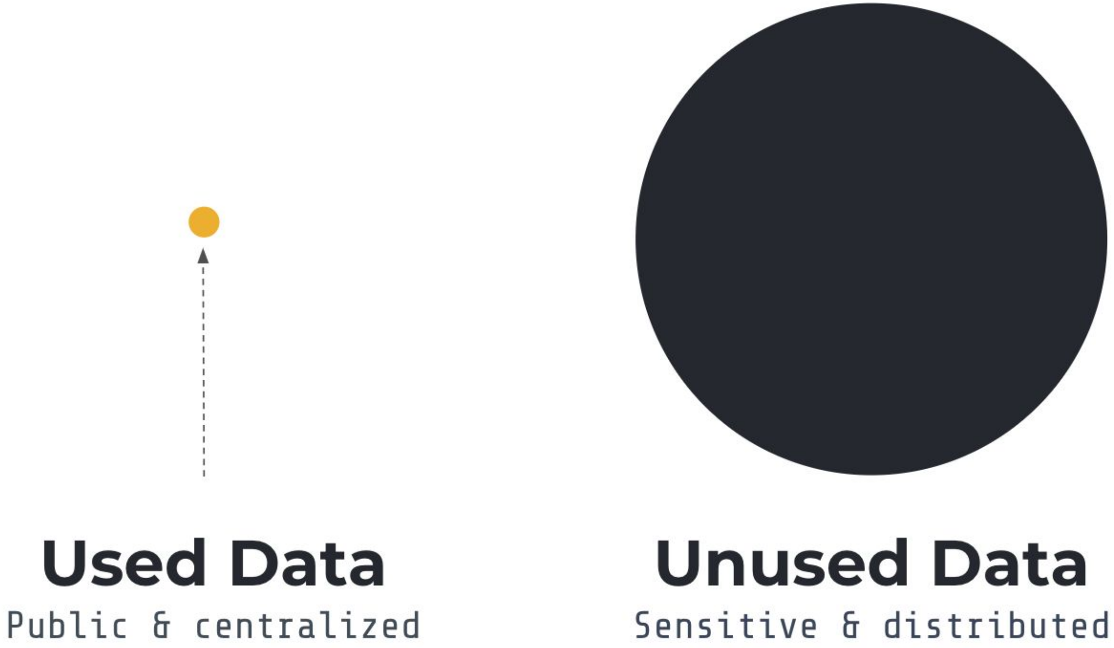


Why is it a dead end?

(aka why should we innovate away from it?)



- Reliance on static tiny biased datasets



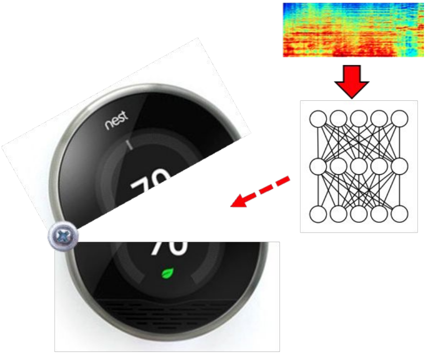
Where to next?

- *(classic)* On-Device ML
- Federated Learning
- Alternative Emerging Learning Paradigms:
  - Self-supervised Local Learning
  - "Zero-Shot" Hybrid Solutions
  - Variations of Foundations Models

Various forms **of** **decentralized** **machine learning**

Where to next?

- *(classic)* On-Device ML
- Federated Learning
- Alternative Emerging Learning Paradigms:
  - Self-supervised Local Learning
  - “Zero-Shot” Hybrid Solutions
  - Variations of Foundations Models



Various forms** of <u>decentralized</u> **machine learning

Where to next?

- *(classic)* On-Device ML
- Federated Learning
- Alternative Emerging Learning Paradigms:
  - Self-supervised Local Learning
  - “Zero-Shot” Hybrid Solutions
  - Variations of Foundations Models

***Various forms* of <u>decentralized</u> machine learning**

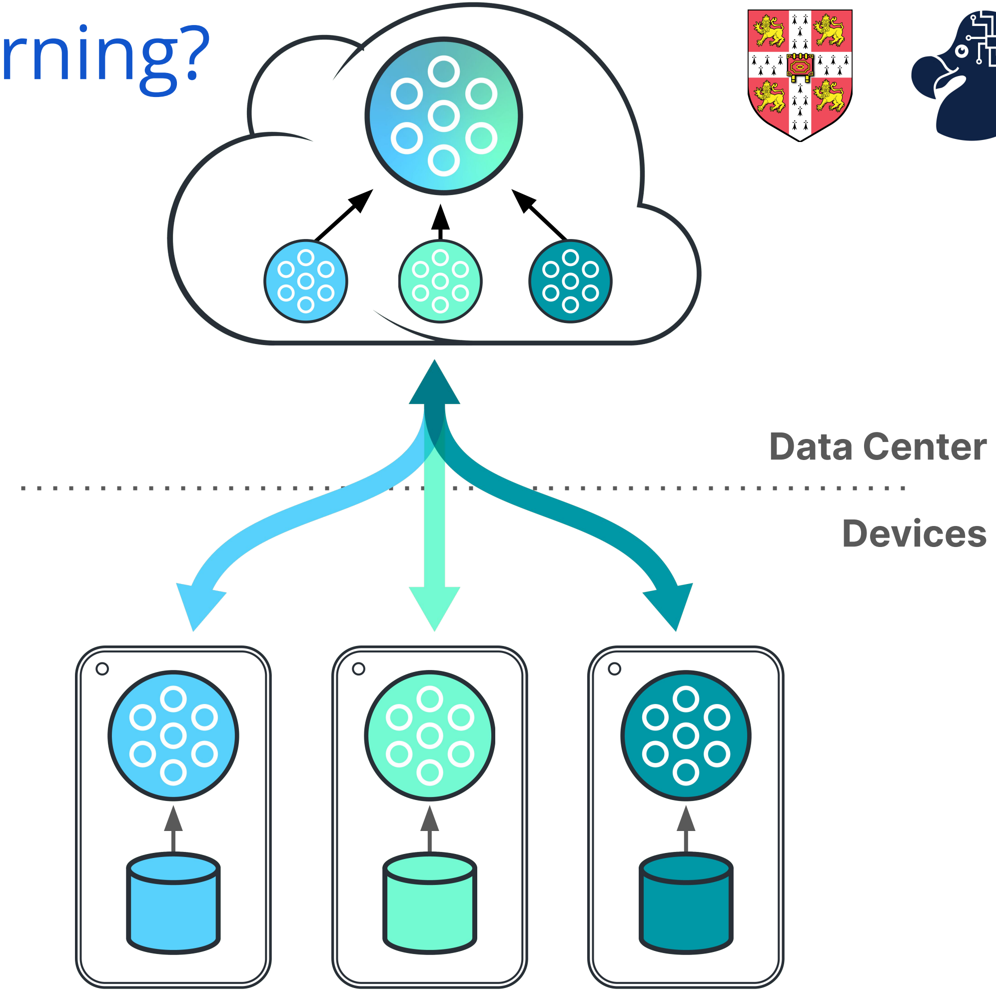
Where to next?

- *(classic)* On-Device ML
- Federated Learning
- Alternative Emerging Learning Paradigms:
  - Self-supervised Local Learning
  - “Zero-Shot” Hybrid Solutions
  - Variations of Foundations Models

***Various forms** of <u>decentralized</u> machine learning*

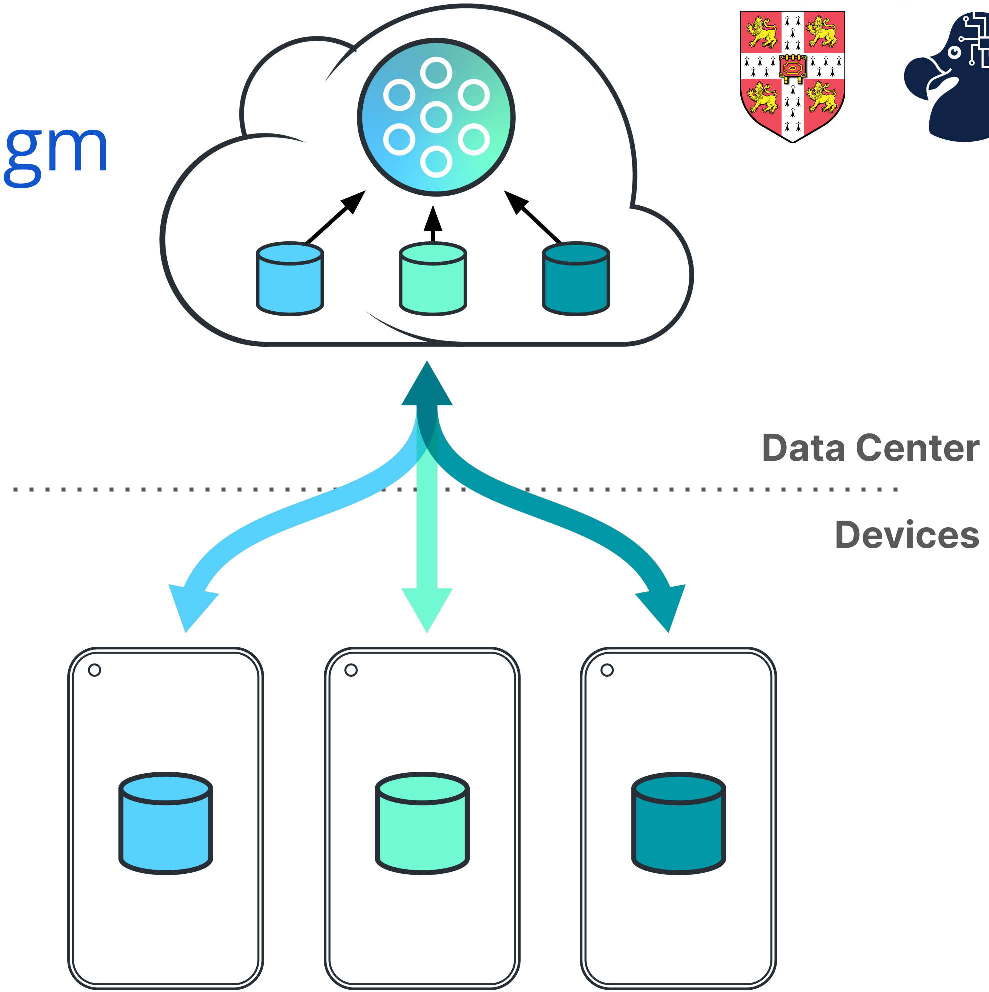
What is Federated Learning?

- Local data, no collection
- Data privacy a priori
- Best availability/latency
- No infrastructure, bring the computation to the data



Comparison to the ML Data Center Paradigm

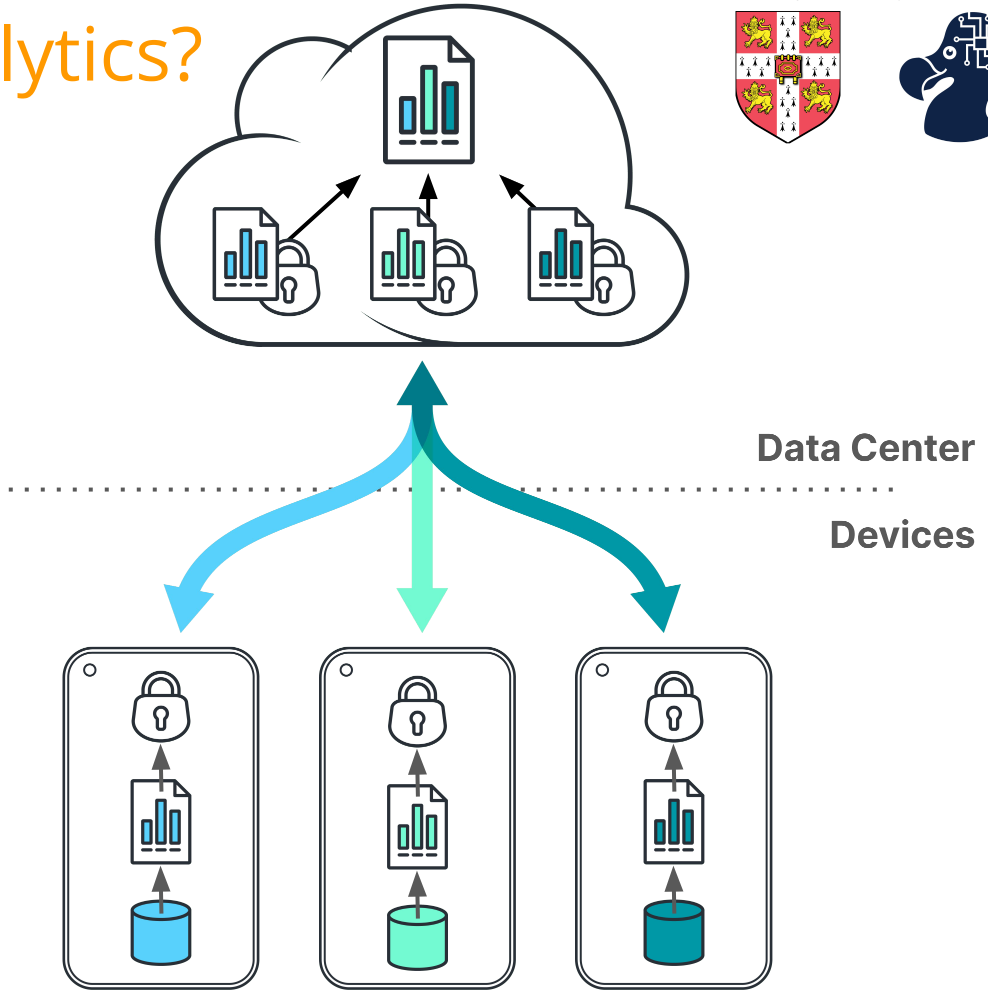
- Regulatory constraints
- System complexity
- Data privacy
- Limited availability, high latency



CaMLSys http://mlsys.cst.cam.ac.uk

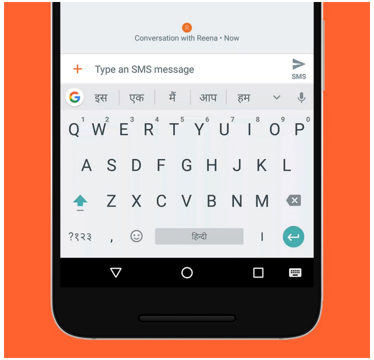
What is Federated Analytics?

1. Compute local analytics
2. Encrypt results via Secure Aggregation Protocol
3. Collect, aggregate, and decrypt results on the server



Federated Examples (Exciting, but niche)

Navigation and Perception of Robots



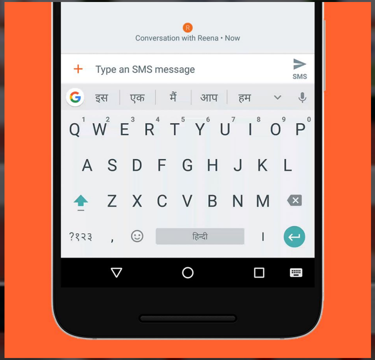
Learning user keyboard behaviors and word selection

Sharing of Sensitive Data between Organizations

Personalization of Speech Recognition

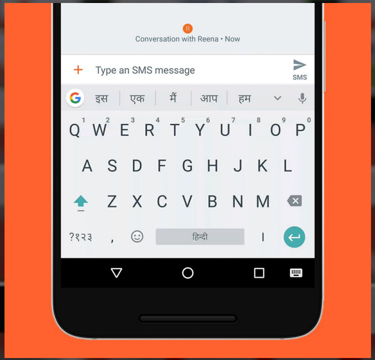
Hurdles to pervasive use of federated methods

- Dependency and availability of labels/supervision
- Inefficiency of communication
- Heavy local compute/memory
- Coping with heterogeneity (clients/devices)
- Challenges of non-I.I.D data (i.e., data heterogeneity)
- Breadth of ML tasks at acceptable accuracy
- Primitive MLOps and tuning capabilities
- Gaps in theoretical understanding and/or empirical best-practices
- ...



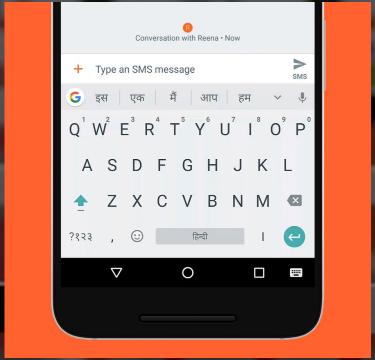
Hurdles to pervasive use of federated methods

- Dependency and availability of labels/supervision
- Inefficiency of communication
- Heavy local compute/memory
- Coping with heterogeneity (clients/devices)
- Challenges of non-I.I.D data (i.e., data heterogeneity)
- Breadth of ML tasks at acceptable accuracy
- Primitive MLOps and tuning capabilities
- Gaps in theoretical understanding and/or empirical best-practices
- ...



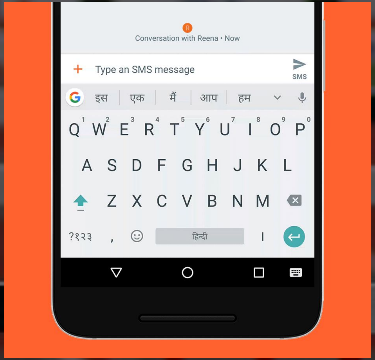
Hurdles to pervasive use of federated methods

- Dependency and availability of labels/supervision
- Inefficiency of communication
- Heavy local compute/memory
- Coping with heterogeneity (clients/devices)
- Challenges of non-I.I.D data (i.e., data heterogeneity)
- Breadth of ML tasks at acceptable accuracy
- Primitive MLOps and tuning capabilities
- Gaps in theoretical understanding and/or empirical best-practices
- ...



Hurdles to pervasive use of federated methods

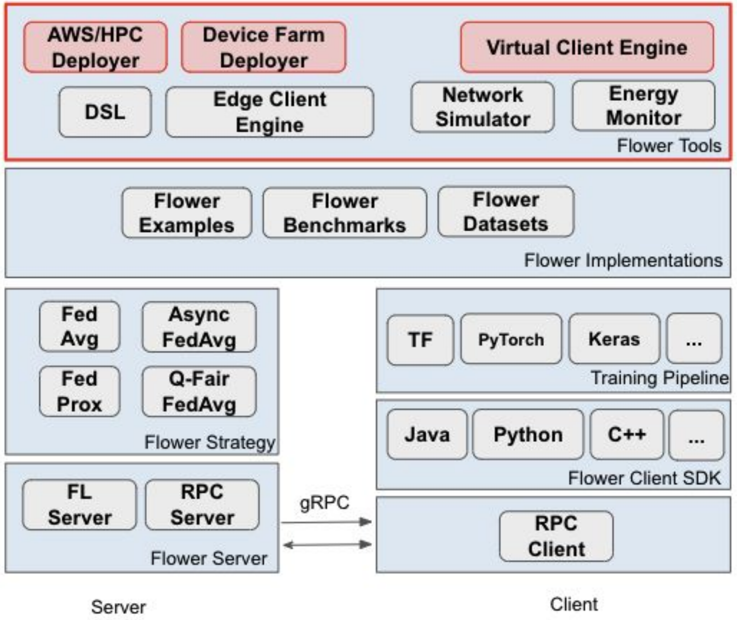
- Dependency and availability of labels/supervision
- Inefficiency of communication
- Heavy local compute/memory
- Coping with heterogeneity (clients/devices)
- Challenges of non-I.I.D data (i.e., data heterogeneity)
- Breadth of ML tasks at acceptable accuracy
- Primitive MLOps and tuning capabilities
- Gaps in theoretical understanding and/or empirical best-practices
- ...



CaMLSys http://mlsys.cst.cam.ac.uk

Flower

http://flower.dev



http://arxiv.org/abs/2007.14390
http://arxiv.org/abs/2104.03042
http://arxiv.org/abs/2205.06117

OPEN SOURCE

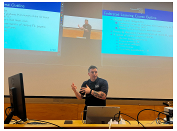
+ Community Driven

Flower

http://flower.dev

- 2.6k GitHub stars
- 34k+ monthly downloads

Flower Summit 2023



Summer of Reproducibility

$100,000 USD; kicking off: July 1st!

CaMLSys http://mlsys.cst.cam.ac.uk

Flower

http://flower.dev

Daniel Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Yan Gao
Javier Fernandez-Marques, Titouan Parcollet, Lorenzo Sani
Pedro Porto Buarque de Gusmão, Nicholas D. Lane

Fed Learning needs to get *Real...*


```
// Number of FL clients = N. Rounds = R

avg_weights = RANDOM_WEIGHTS

For{round in 1...R} {

      client_weights = []

      For{client in C1,C2...CN} {

         w = local_optimize(avg_weights)

         client_weights.append(w)

      }

      avg_weights = federated_averaging(client_weights)

}
```

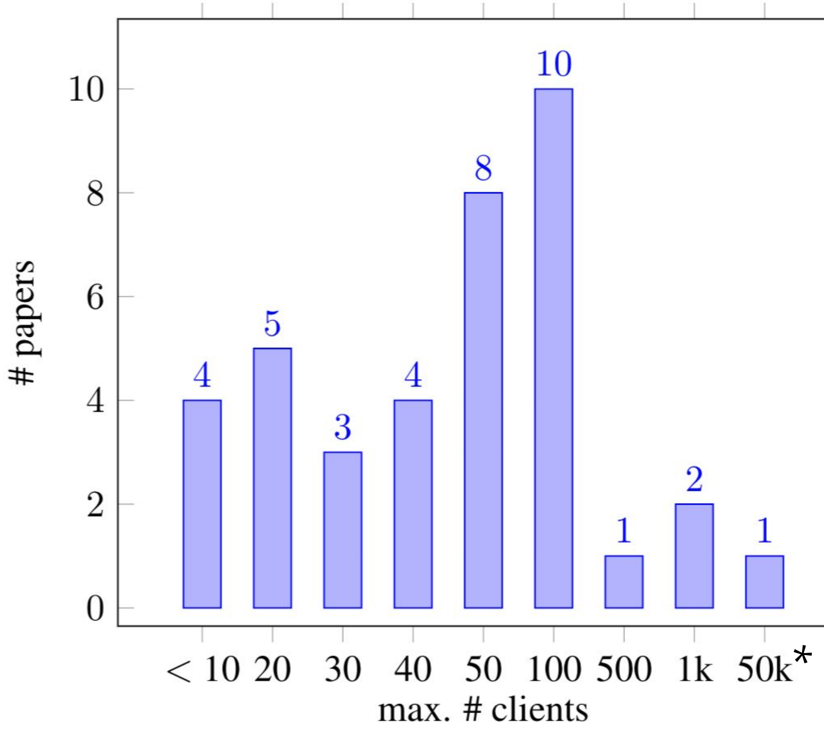
- Networking and Wireless
- Compute Heterogeneity
- Memory Constraints
- Energy Overheads
- Scaling FL Clients

Fed Learning needs to get *Real...*

```
// Number of FL clients = N. Rounds = R

avg_weights = RANDOM_WEIGHTS

For{round in 1...R} {

      client_weights = []

      For{client in C1,C2...CN} {

         w = local_optimize(avg_weights)

         client_weights.append(w)

      }

      avg_weights = federated_averaging(client_weights)

}
```

- Networking and Wireless
- Compute Heterogeneity
- Memory Constraints
- Energy Overheads
- Scaling FL Clients

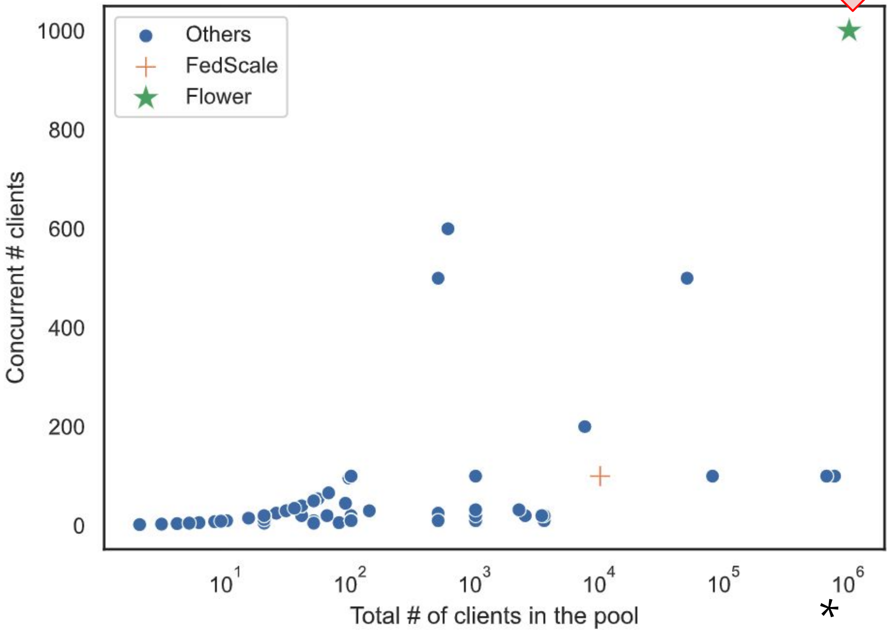
Fed Learning needs to get *Real...*



- Networking and Wireless
- Compute Heterogeneity
- Memory Constraints
- Energy Overheads
- Scaling FL Clients

** Including Simulations and "For-loop" FL Clients*

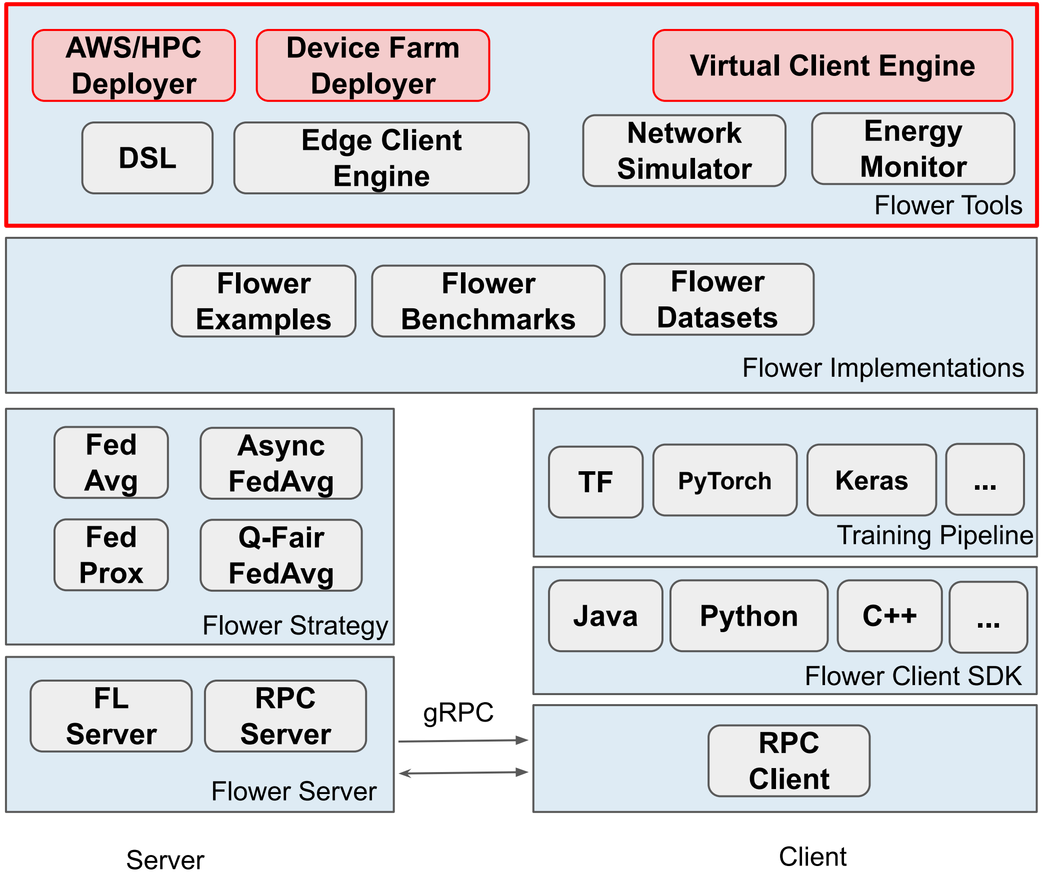
Open Source Scaling of Fed Learning



- Networking and Wireless
- Compute and Memory Heterogeneity
- Non-toy Fully-baked Federated Data
- Scaling FL Clients

** Including Simulations and "For-loop" FL Clients*

What is Flower?



- Networking and Wireless
- Compute Heterogeneity
- Memory Constraints
- Energy Overheads
- Scaling FL Clients

http://flower.dev

The Power of Flower (*more later...*)

```
import flwr as fl

class MyClient(fl.KerasClient):
    def __init__(self, model, ds_train, val)
        self.model = model
        self.ds_train = ds_train
        self.ds_val = ds_val

    def get_parameters():
        return model.get_weights()

    def fit(self, weights, config):
        model.set_weights(weights)
        model.fit(ds_train, epochs=config["epochs"])
        return model.get_weights()

    def evaluate(self, weights, config):
        model.set_weights(weights)
        return model.evaluate(ds_test)

server_address, model, ds_train, ds_test = ...
client = MyClient(model, ds_train, ds_test)
fl.app.client.start_client(server_address, client)
```

1 *Integrate with existing training code*

```
import flwr as fl

fl.app.server.start_server()
```

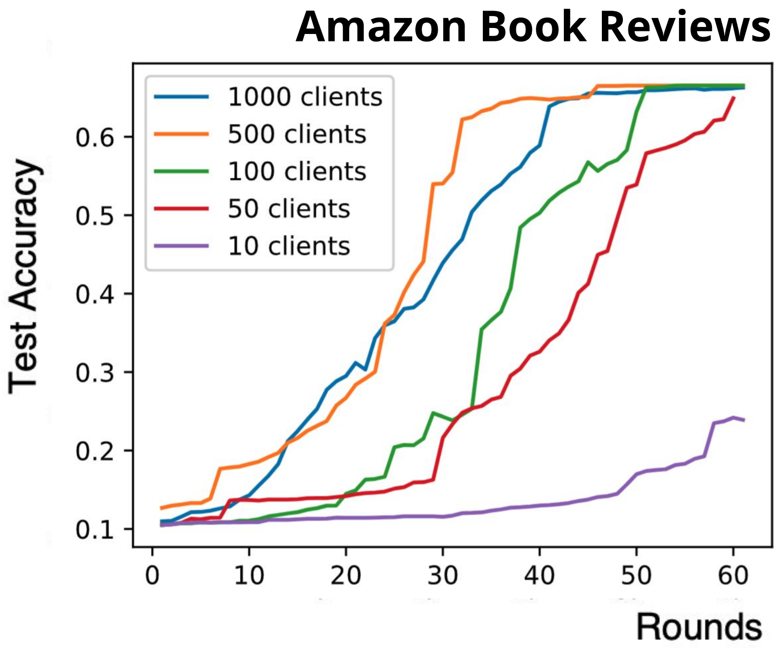
2 *Only one line to start flower...*

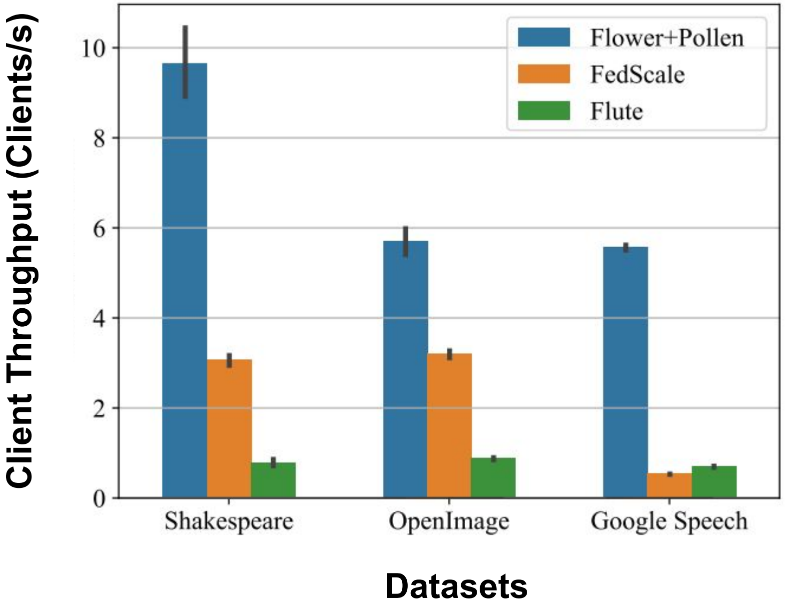
```
import flwr as fl

# Implement your own strategy
class MySotaStrategy(fl.Strategy):
    ...

# Start server w/ custom strategy
strategy = MySotaStrategy(...)
fl.app.server.start_server(strategy)
```

3 *Implement your own FL strategy*

CaMLSys http://mlsys.cst.cam.ac.uk

Extreme Scalability and Training Speed Comparisons

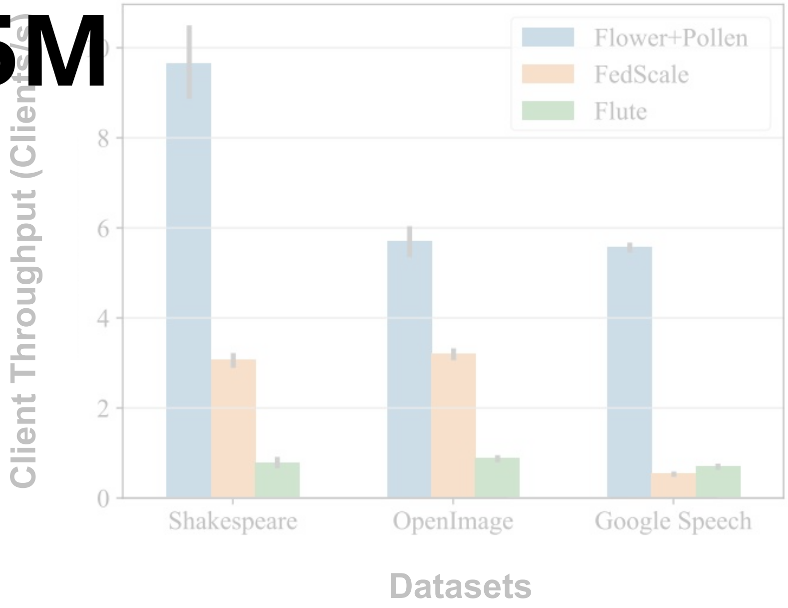




Architecture: DistilBERT

Architectures: { ShuffleNetV2, ResNet 34, 2-cell LSTM }

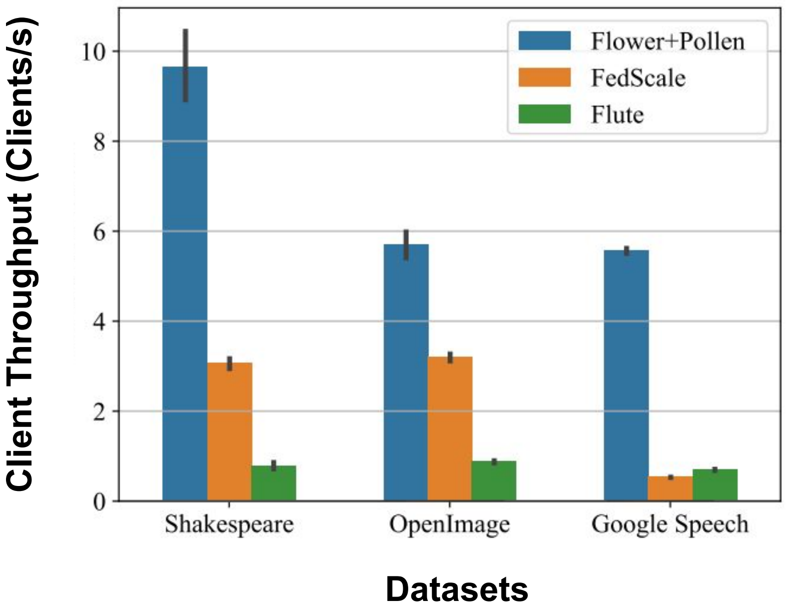
Extreme Scalability and Training Speed Comparisons



Architecture: DistilBERT

Architectures: { ShuffleNetV2, ResNet 34, 2-cell LSTM }

Extreme Scalability and Training Speed Comparisons

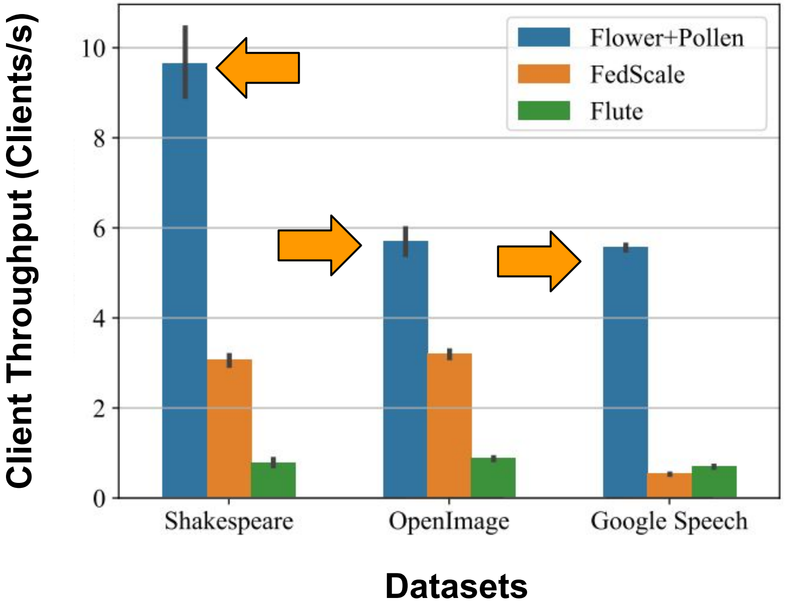


Architecture: DistilBERT

Architectures: { ShuffleNetV2, ResNet 34, 2-cell LSTM }

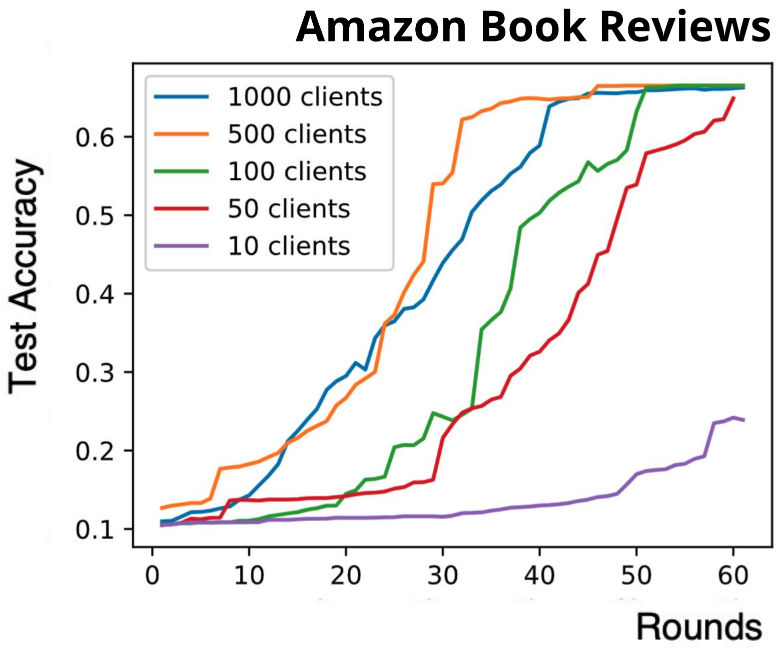
Extreme Scalability and Training Speed Comparisons

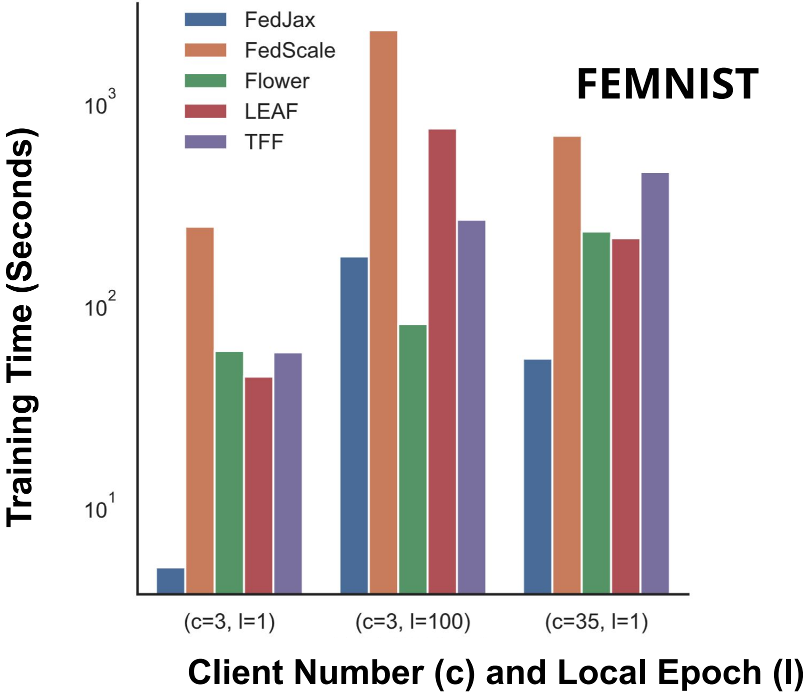
Architecture: DistilBERT



Architectures: { ShuffleNetV2, ResNet 34, 2-cell LSTM }

Extreme Scalability and Training Speed Comparisons



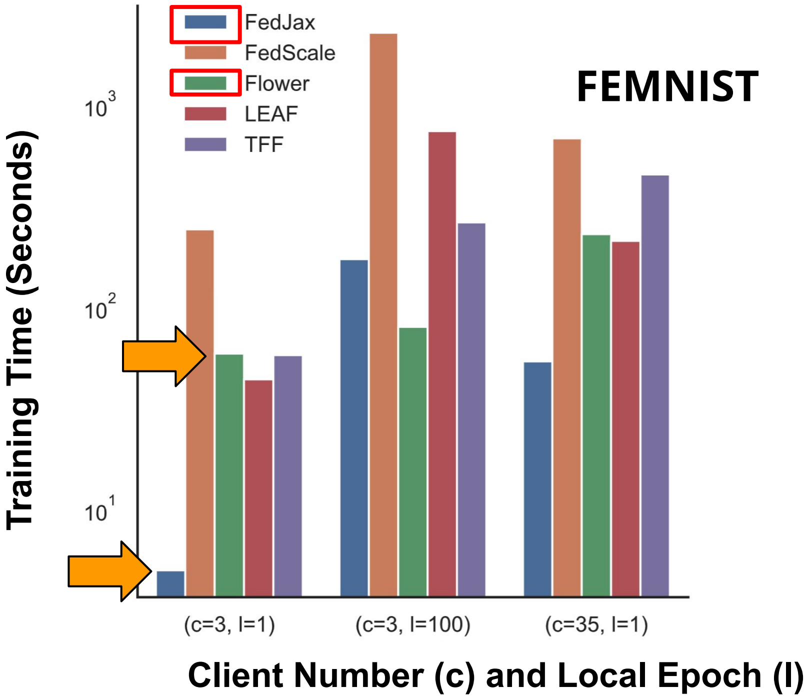


Architecture: DistilBERT

Architecture: ResNet 18

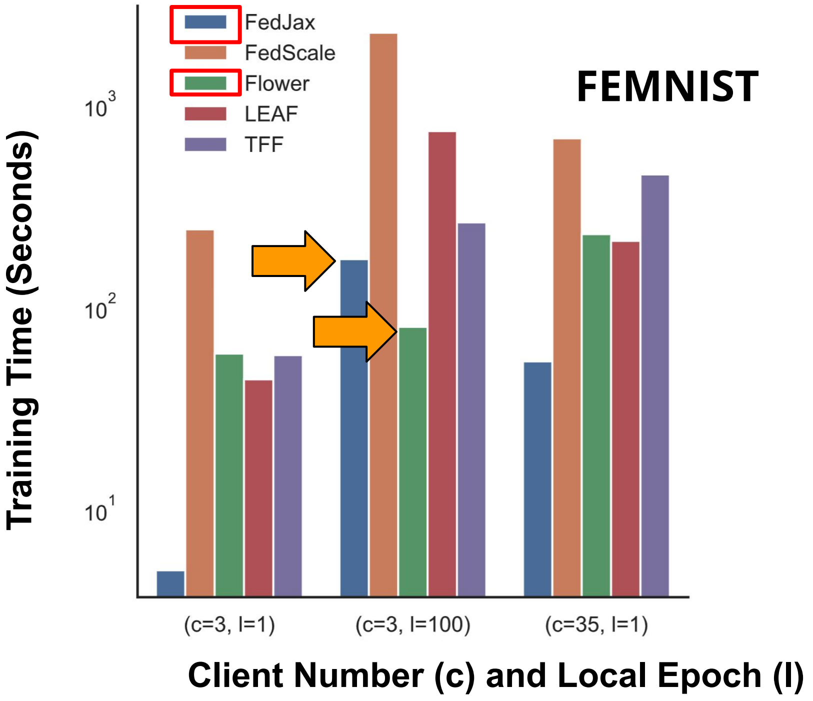
Extreme Scalability and Training Speed Comparisons

Architecture: DistilBERT



Architecture: ResNet 18

Extreme Scalability and Training Speed Comparisons



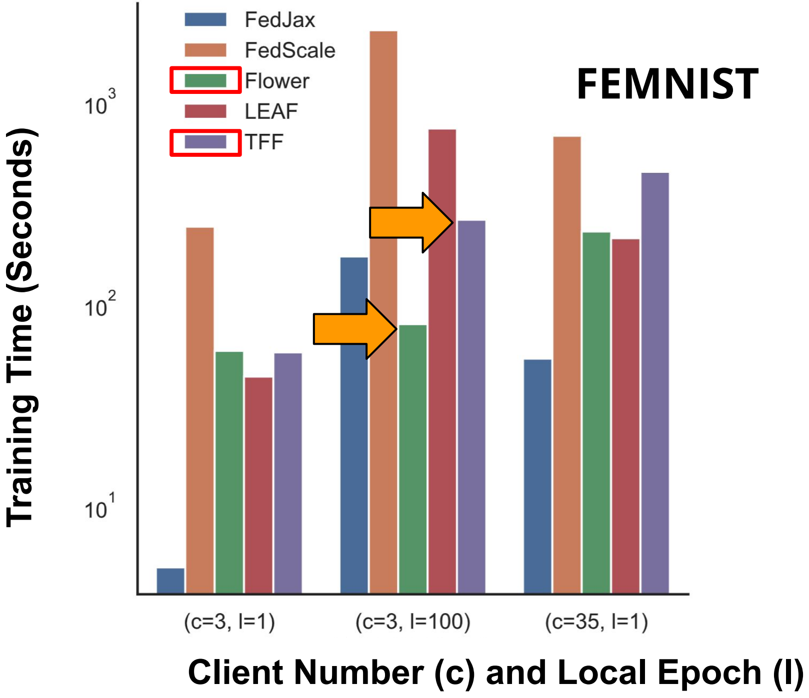
Architecture: DistilBERT

Architecture: ResNet 18

Extreme Scalability and Training Speed Comparisons

CaMLSys http://mlsys.cst.cam.ac.uk

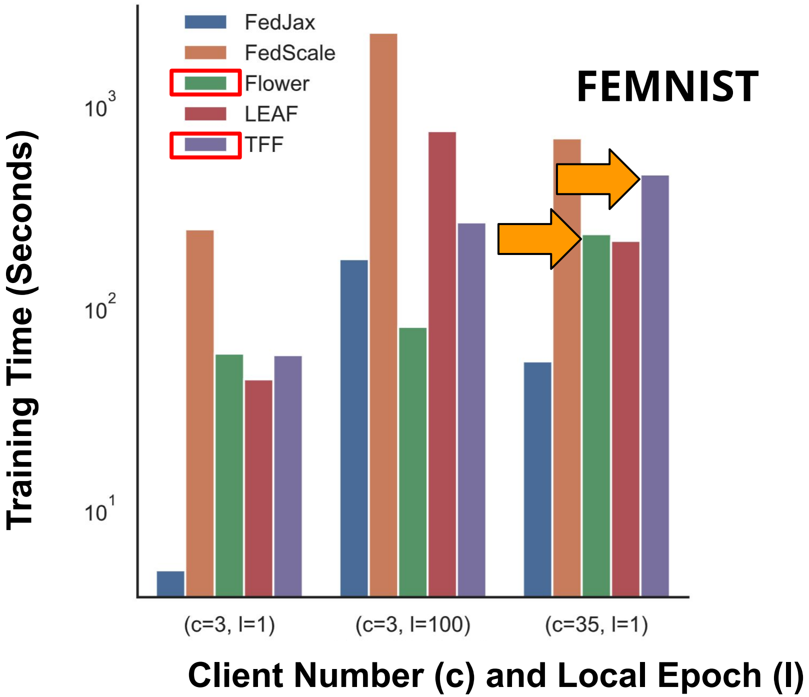
Architecture: DistilBERT



Architecture: ResNet 18

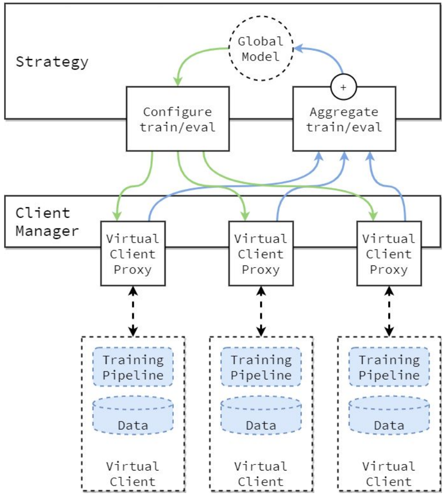
Extreme Scalability and Training Speed Comparisons

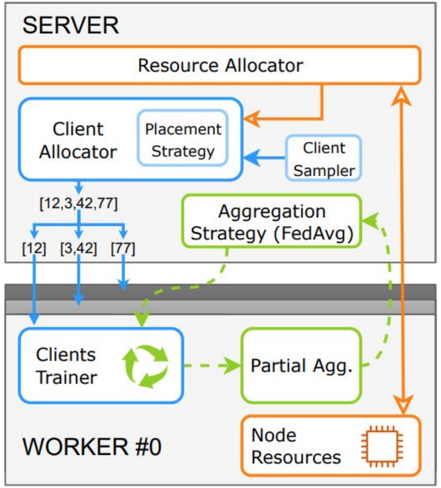
Architecture: DistilBERT



Architecture: ResNet 18

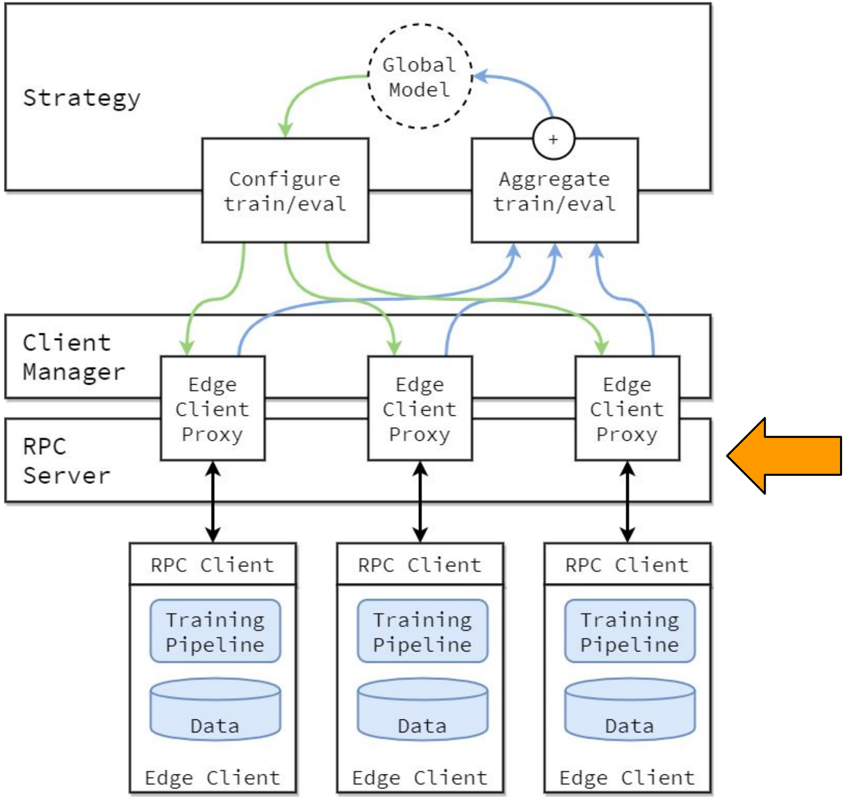
Virtual Client Engine (VCE) + Pollen





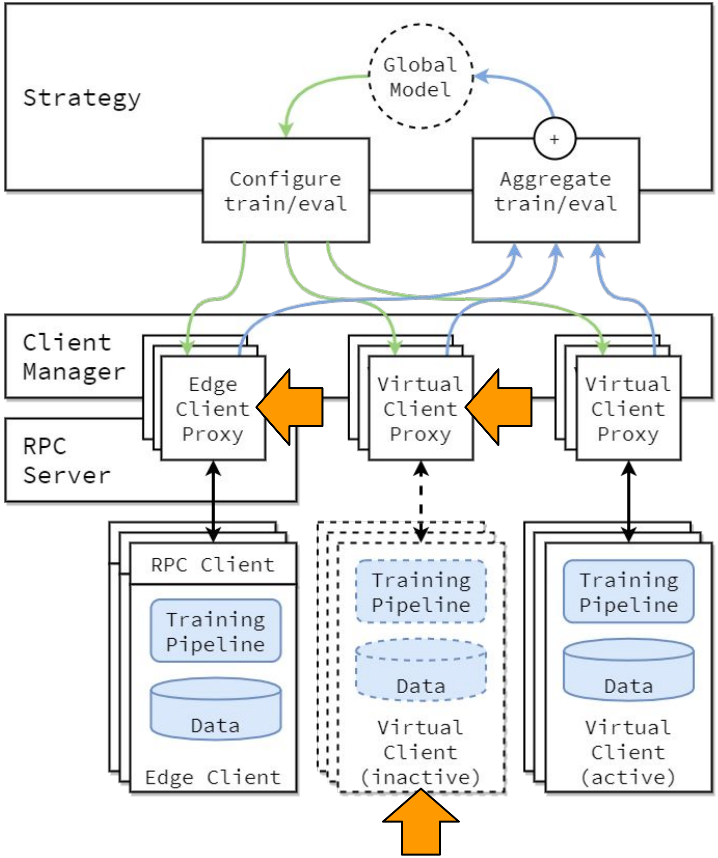
- Clients virtual or devices
- ClientProxy creates clients lazily when they need to perform work
- Resource-aware scheduling w/ latency prediction
- Use case: single-machine or HPC cluster simulation

Edge Client Engine (ECE)



- Clients run in separate processes, potentially on different machines
- Client/server communication over gRPC
- Use case: systems research, production deployment

Mixed Execution: VCE + ECE



- Mixed execution with both virtual and edge clients
- Virtual clients can be active or inactive

The Power of Flower (...*continued*)

- Instantly deploying 1,000++/15M++ client experiments
  - AWS instances / HPC Nodes
  - AWS device farm / Physical devices
- Sensitivity analysis of system resources
  - `{network, wireless, compute, memory}`
- Systematic repeatable comparisons of FL algorithm comparisons
- Launching a functioning FL system

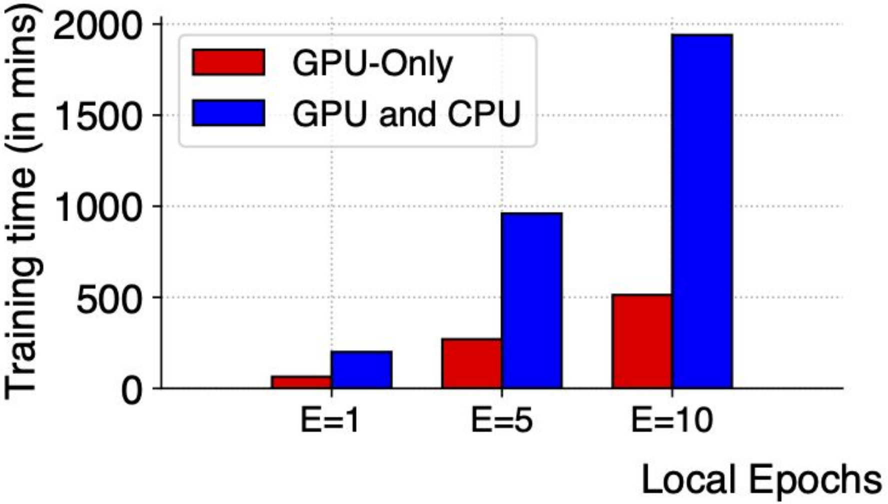
```
Baseline(
    instances=[...],
    server=ServerConfig(
      instance_name="server",
      strategy="fedavg",
      model="resnet50",
      rounds=10,
      sample_fraction=0.1,
      training_round_timeout=3600,
      ...
    ),
    clients=configure_clients(
      instance_names=...,
      num_clients=10,
      ...
    ),
)
```

Flower DSL Example

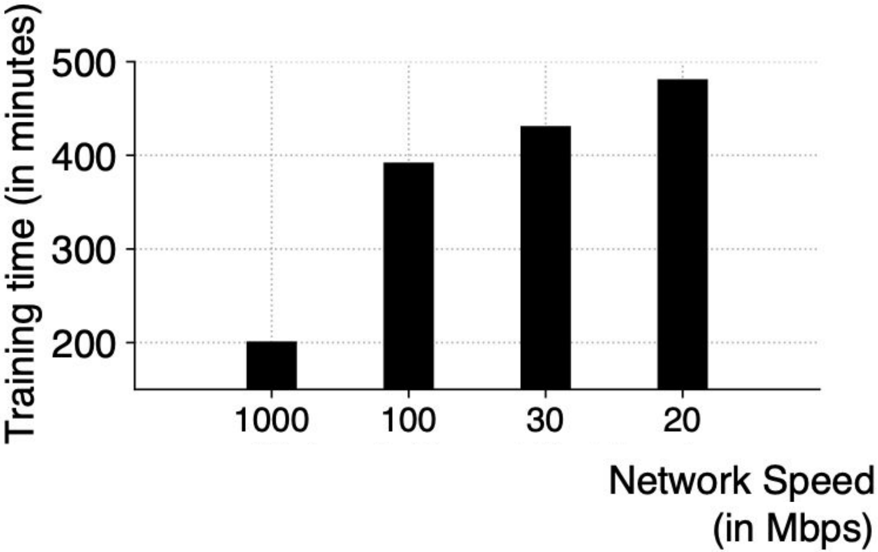
CaMLSys http://mlsys.cst.cam.ac.uk

Increases in FL training time under compute & network heterogeneity

FashionMNIST

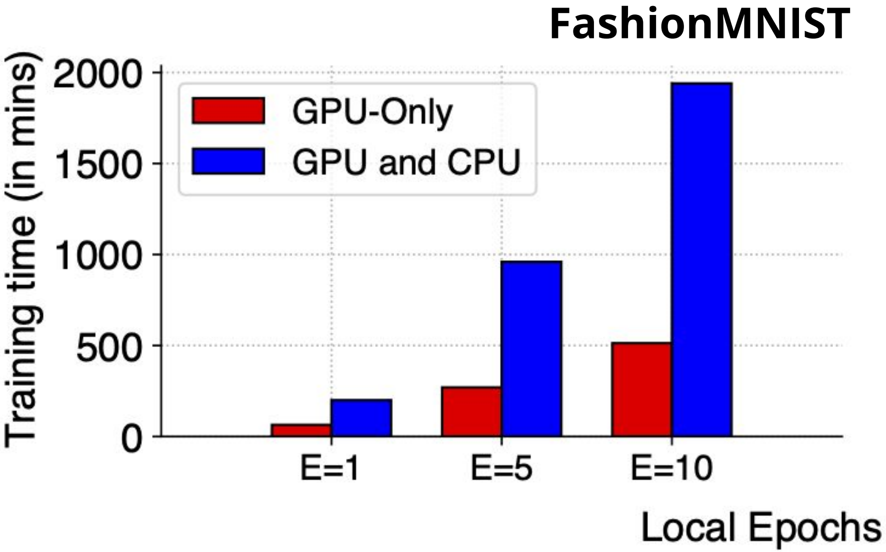


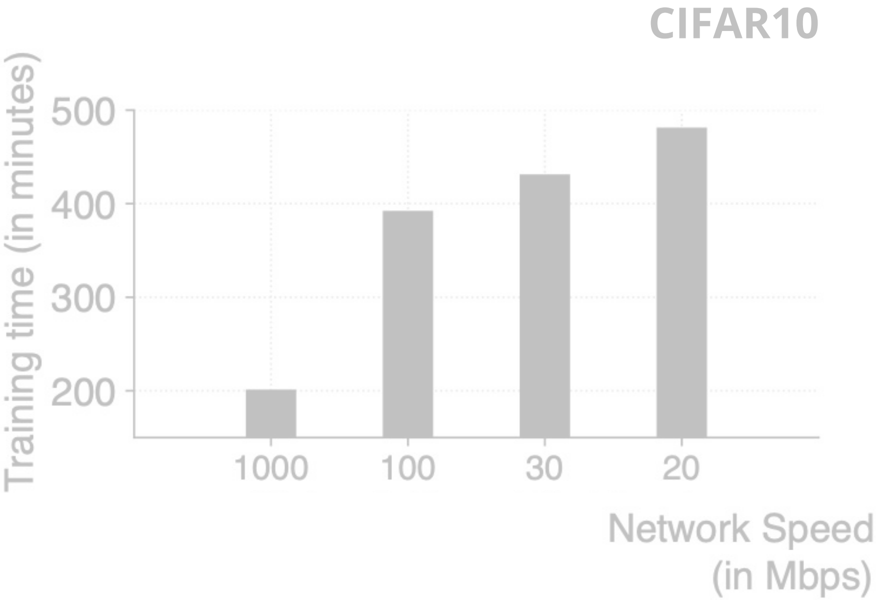
CIFAR10



Architecture: ResNet 50

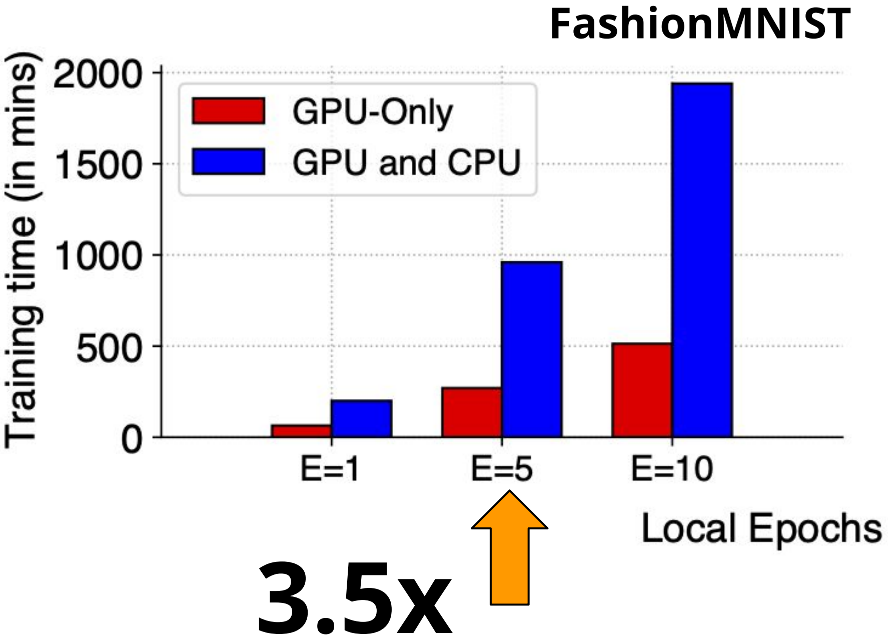
Increases in FL training time under compute & network heterogeneity

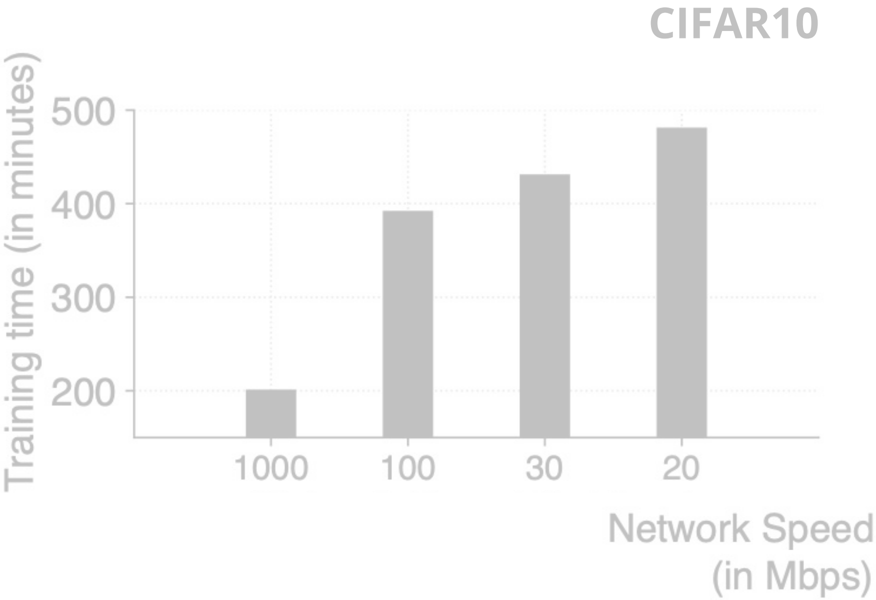




Architecture: ResNet 50

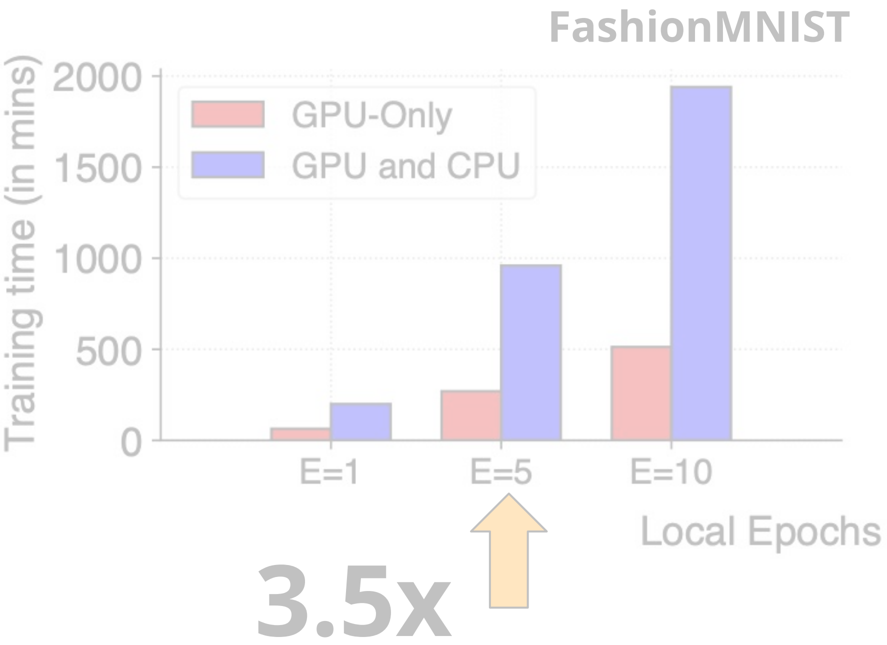
Increases in FL training time under compute & network heterogeneity

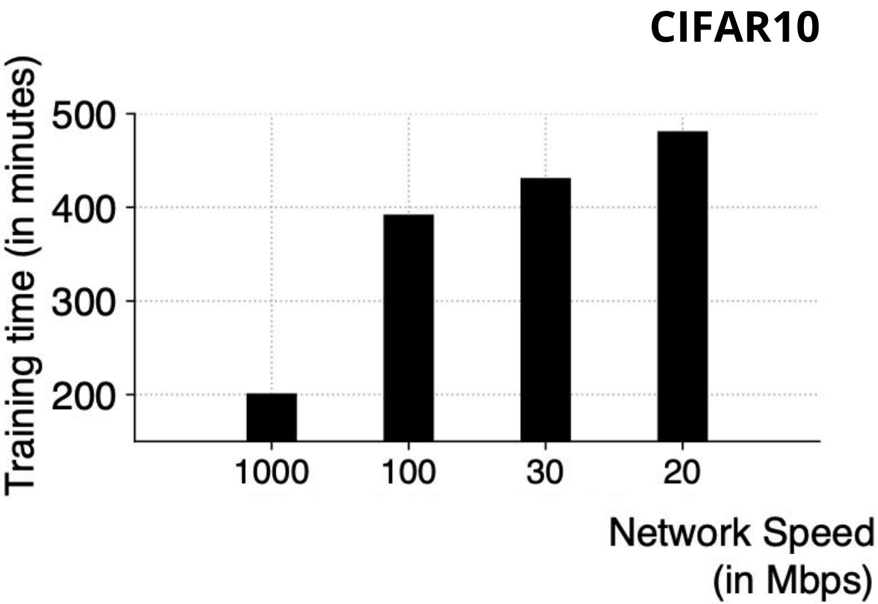




Architecture: ResNet 50

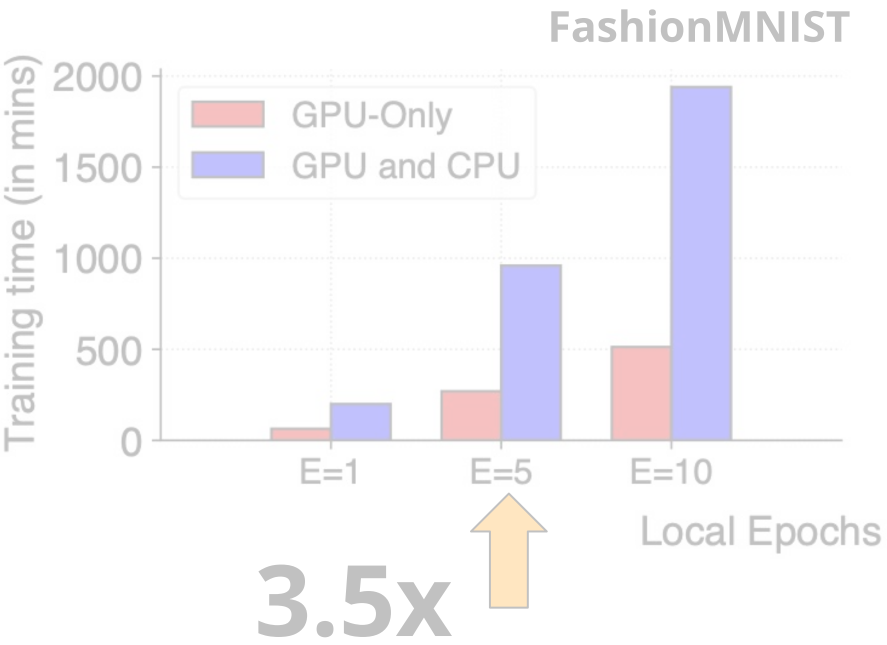
Increases in FL training time under compute & network heterogeneity

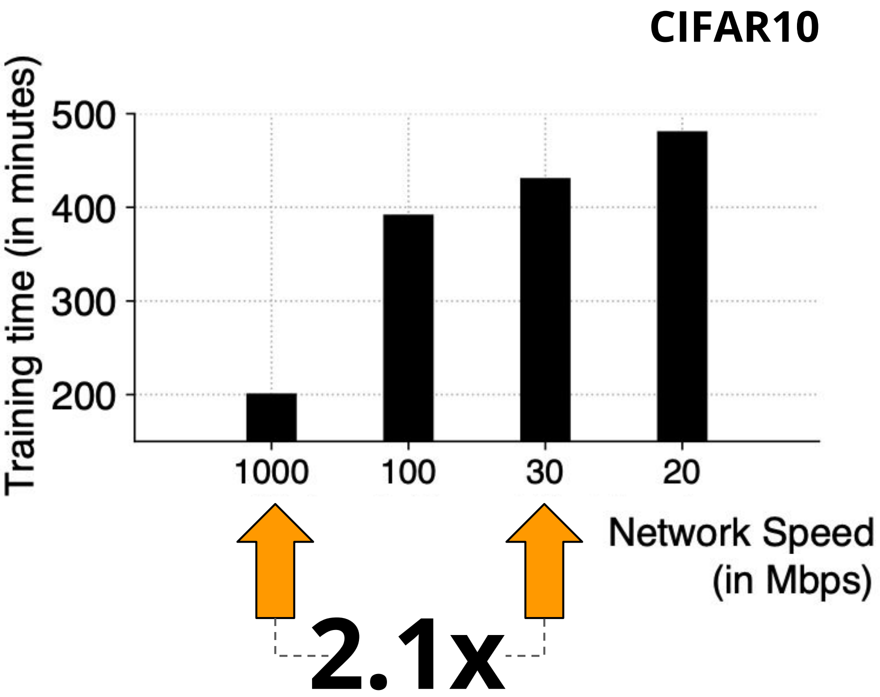




Architecture: ResNet 50

Increases in FL training time under compute & network heterogeneity





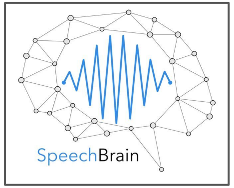
Architecture: ResNet 50

CaMLSys http://mlsys.cst.cam.ac.uk

Samsung AI Cambridge

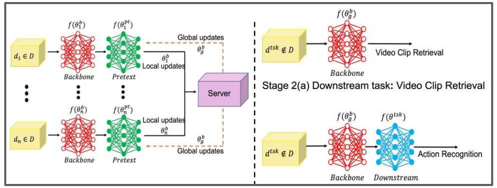
Flower

http://flower.dev



Case Study
Speech Recognition

Case Study
Carbon Footprint



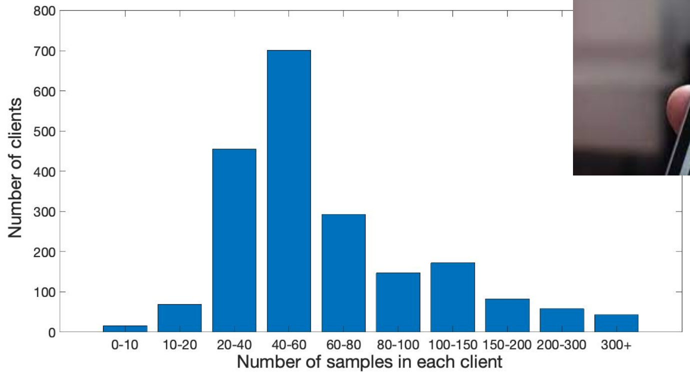
Case Study
Self-Supervised Learning

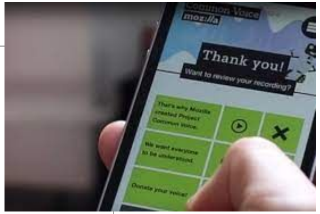
SpeechBrain

Case Study

Speech Recognition

Common Voice as an acoustic dataset with **true** federated properties

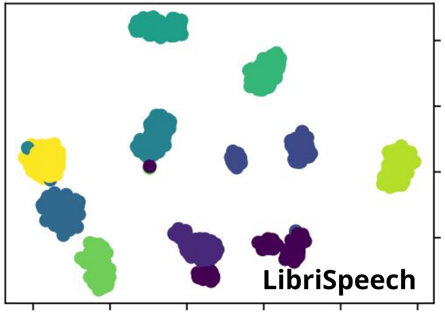


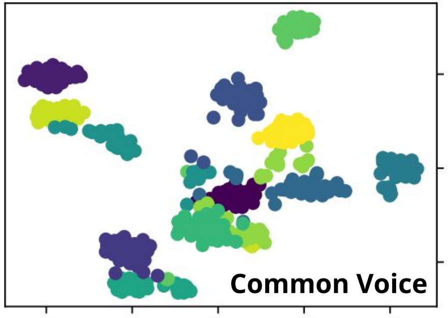


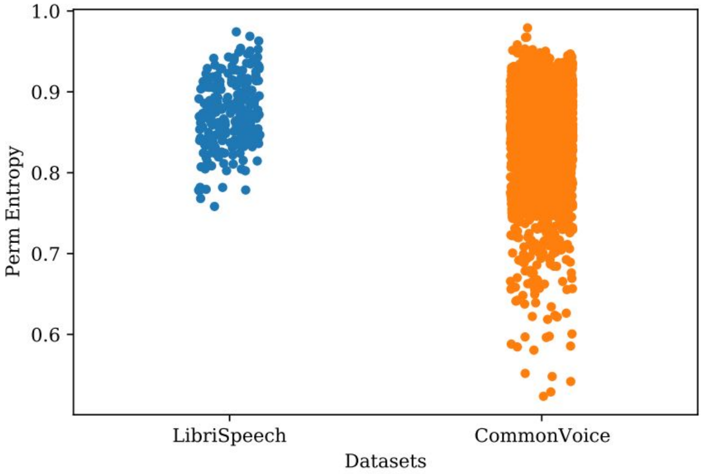
Mozilla Common Voice *(French)*

Set-up

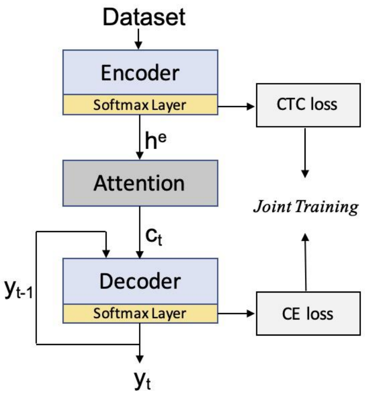
- Training: 4,190 speakers (425.5 hours)
- Test: 4,247 speakers
- Warm-up Model: 117 speakers







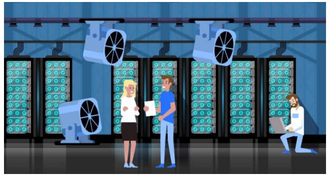
Real-world Federated Acoustic Modeling

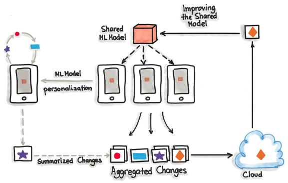


- Model Arch: Seq2Seq attention w/ CTC loss
- WER-based weighting strategy

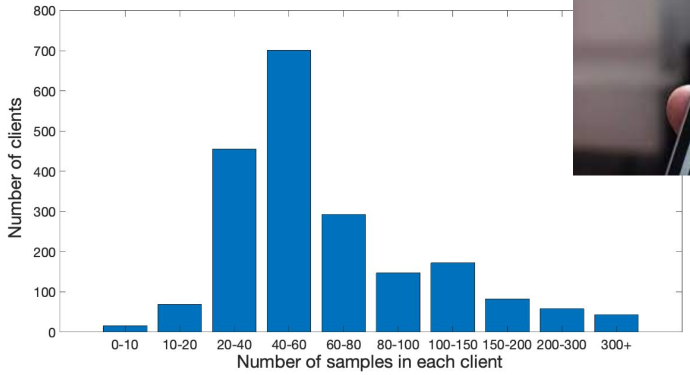
$$\alpha_T^{(k)} = \frac{\exp\left(1 - wer_k\right)}{\sum_{k=1}^{K} \exp\left(1 - wer_k\right)}$$

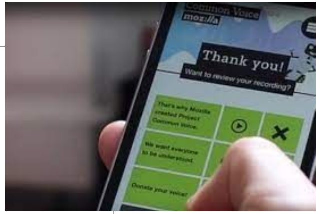
- Warm-up Model: Centralized training as init
- .. many "small" tweaks
  - SGD on clients (e.g., Adadelta worse)
  - LR annealing based on WER validation
  - *Older* FedAvg centralized (e.g., FedAdam worse)





Promising Federated WER under Silo and Device settings

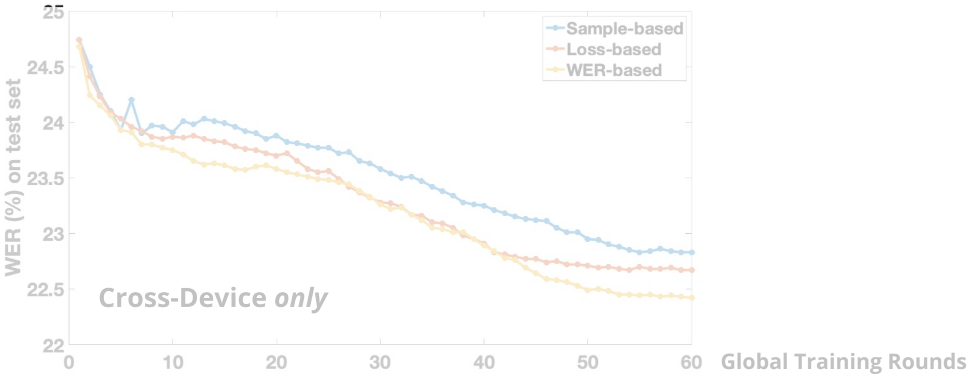




Mozilla Common Voice *(French)*

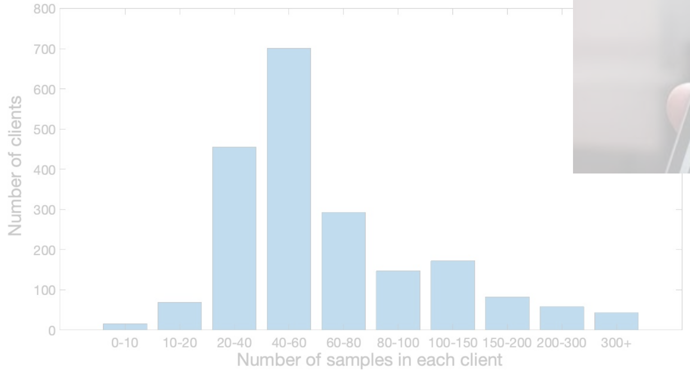
Set-up

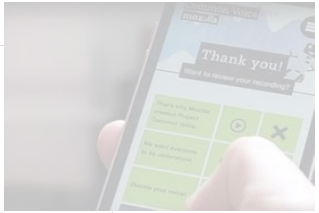
- Training: 4,190 speakers (425.5 hours)
- Test: 4,247 speakers
- Warm-up Model: 117 speakers



	WER (%)
Purely Centralized	20.18
FL Cross-Silo *(10-clients)*	20.99
Cross-Device *(2,000 clients)*	22.42

Promising Federated WER under Silo and Device settings

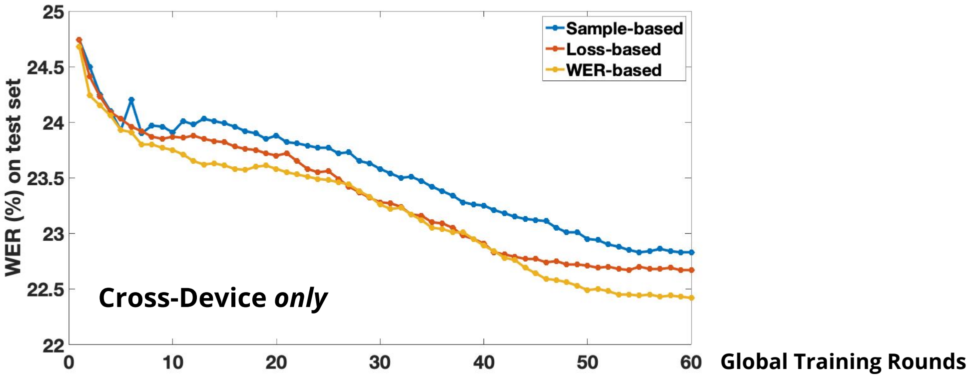




Mozilla Common Voice *(French)*

Set-up

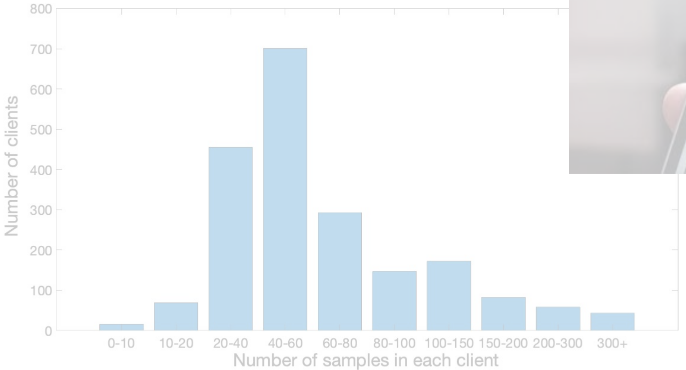
- Training: 4,190 speakers (425.5 hours)
- Test: 4,247 speakers
- Warm-up Model: 117 speakers

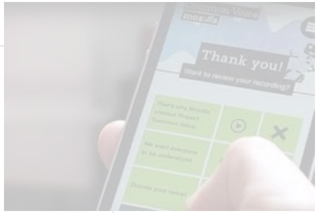


	WER (%)
Purely Centralized	20.18
FL Cross-Silo *(10-clients)*	20.99
Cross-Device *(2,000 clients)*	22.42

CaMLSys http://mlsys.cst.cam.ac.uk

Promising Federated WER under Silo and Device settings

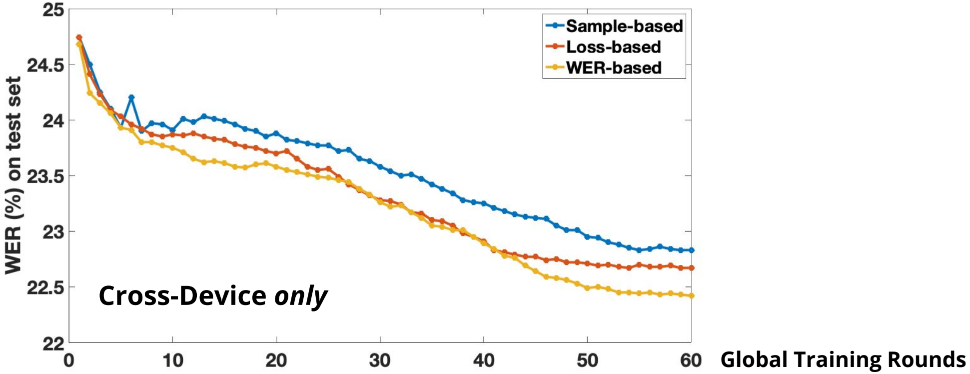




Mozilla Common Voice *(French)*

Set-up

- Training: 4,190 speakers (425.5 hours)
- Test: 4,247 speakers
- Warm-up Model: 117 speakers



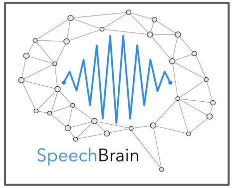
	WER (%)
Purely Centralized	20.18
FL Cross-Silo *(10-clients)*	20.99
Cross-Device *(2,000 clients)*	22.42

CaMLSys http://mlsys.cst.cam.ac.uk

Samsung AI Cambridge

Flower

http://flower.dev

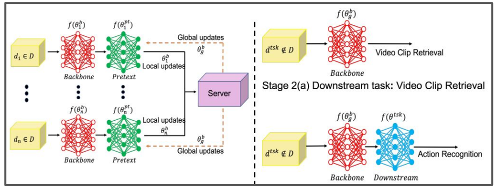


Case Study

Speech Recognition

Case Study

Carbon Footprint



Case Study

Self-Supervised Learning

Tackling Climate Change with Machine Learning
NeurIPS 2020 Workshop

`http://arxiv.org/abs/2010.06537`

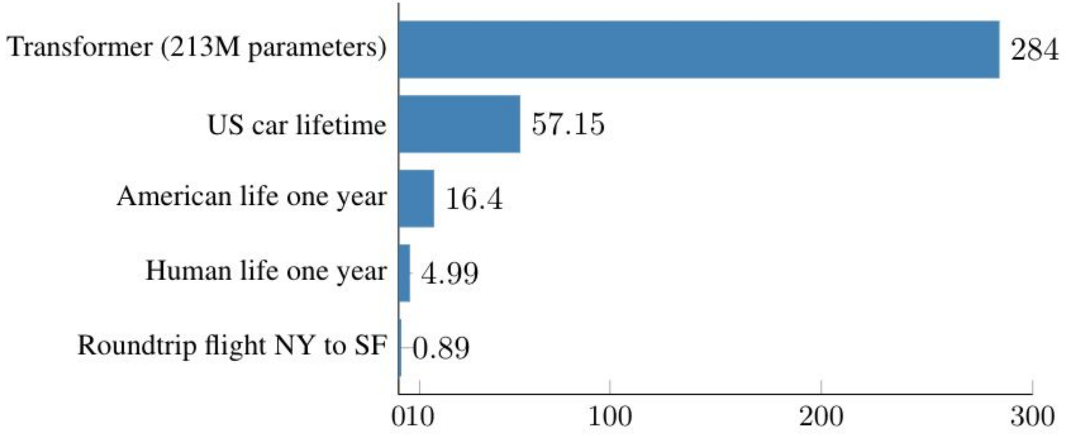
JLMR

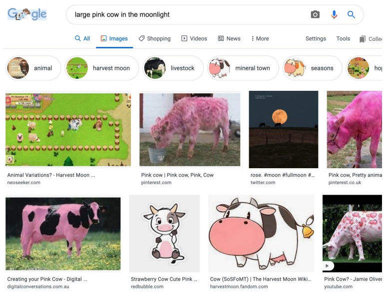
`http://arxiv.org/abs/2102.07627`

Case Study

Carbon Footprint

ML Efficiency is an Environmental Crisis





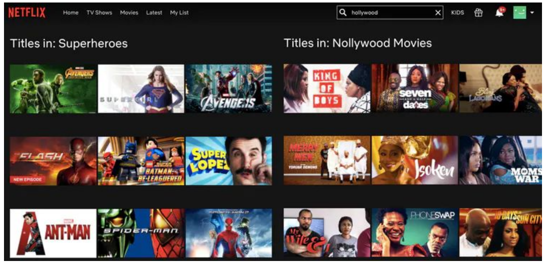
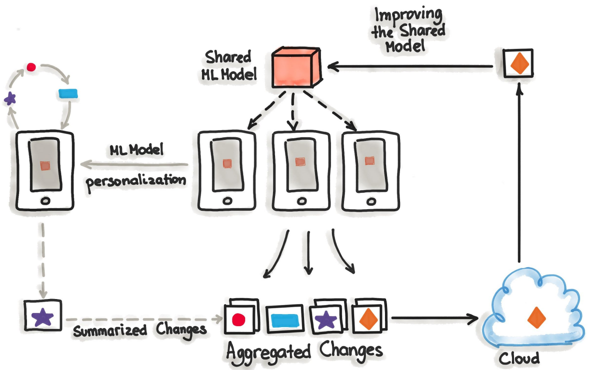
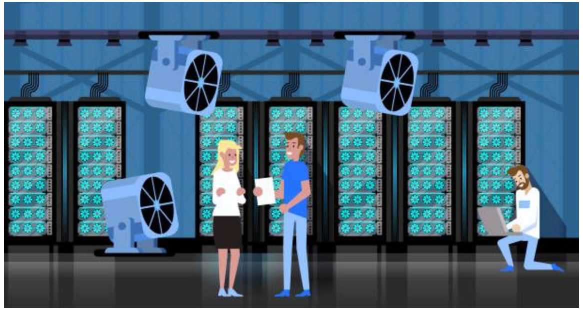


Figure Acknowledgement: "Energy & policy considerations for deep learning in NLP"

CaMLSys http://mlsys.cst.cam.ac.uk

Carbon Potential of Federated Learning





- No need to cool compute
  - Typical data center PUE: 1.60
- No need to move data
  - e.g., 1 memory operation uses 1000x more energy than 1 compute operation
  - (Note: FL faces its own comms overhead)
- No need for redundant ML training
  - Safe reuse of ML across organizations

CaMLSys http://mlsys.cst.cam.ac.uk

Carbon Footprint of Fed Learning *(method)*

1

Data Center Specifications

GPU

Cooling

Data transfer
& storage

FL Specifications

CPU

Cooling

Data transfer
& storage

2

Country	CO2 *(kg)* / kWh
France	0.079
UK	0.509
US	0.547
China	0.975

3

Flower

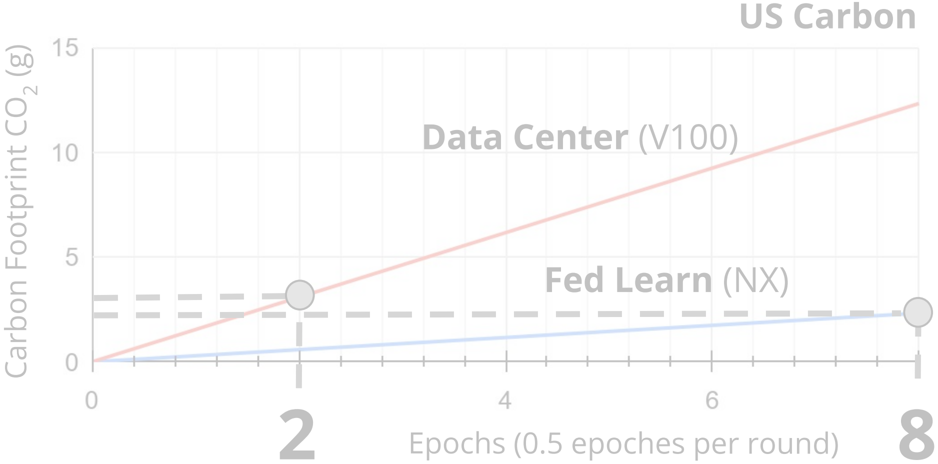
Carbon Footprint of Fed Learning *(illustrative result)*

	ResNet18 at 60% accuracy	
	V100	*FL*
US	3.08	1.85
China	5.49	3.30
UK	2.86	1.72

CO_2 (g)

1.7x

Architecture: ResNet
Dataset: CIFAR10



Data Center Specifications

- Nvidia V100; 250W
- 1 GPU, 48 sec/epoch (V100)
- PUE: 1.67

FL Specifications

- Nvidia Jetson Xavier-NX; 7.5W
- 20.3 sec/round, 1 local epoch
- 5 clients/round
- 16 rounds to 60%

Carbon Footprint of Fed Learning *(illustrative result)*

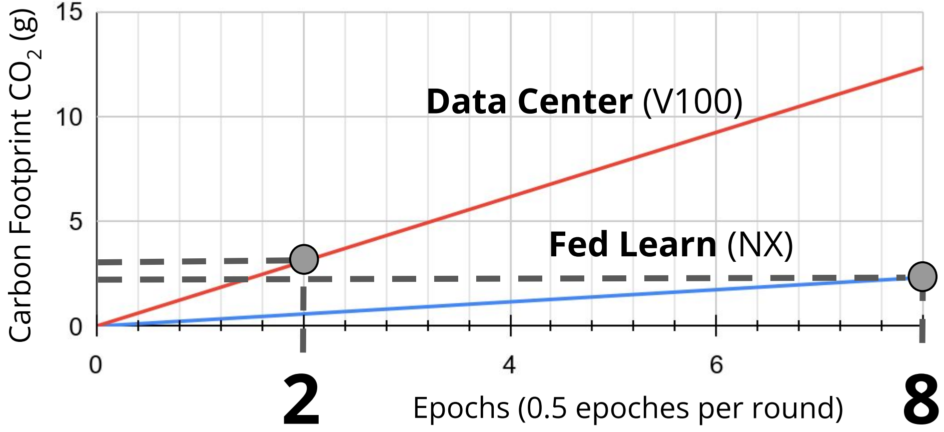
	ResNet18 at 60% accuracy	
	V100	*FL*
US	3.08	1.85
China	5.49	3.30
UK	2.86	1.72

CO_2 (g)

1.7x

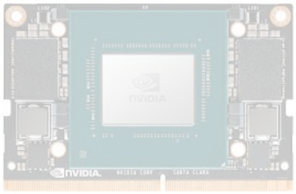
Architecture: ResNet
Dataset: CIFAR10

US Carbon



Data Center Specifications

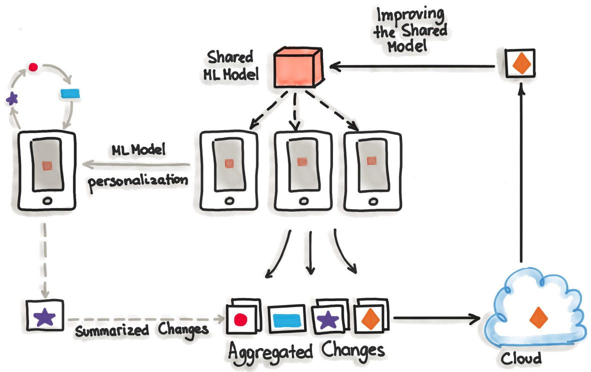
- Nvidia V100; 250W
- 1 GPU, 48 sec/epoch (V100)
- PUE: 1.67

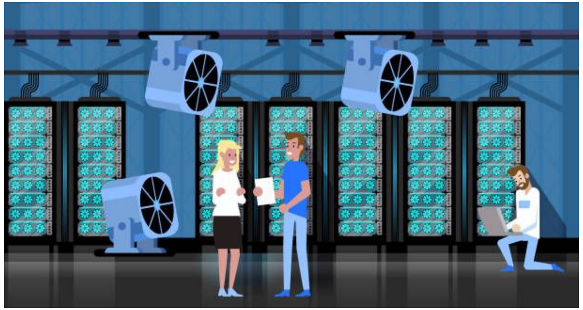


FL Specifications

- Nvidia Jetson Xavier-NX; 7.5W
- 20.3 sec/round, 1 local epoch
- 5 clients/round
- 16 rounds to 60%

What role might FL play in sustainable ML?



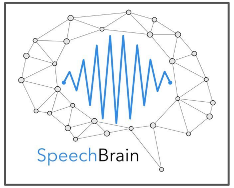


- Selectively replace data centers with FL
  - e.g., speech, vision, hotkey words
- Data center adopting small FL techniques
  - e.g., early-stop centrally, finish w/ FL fine-tuning
- Hybrid data center and FL solutions
  - e.g., FL used to share training effort
- Re-inventing everything around FL

CaMLSys http://mlsys.cst.cam.ac.uk

Flower

http://flower.dev

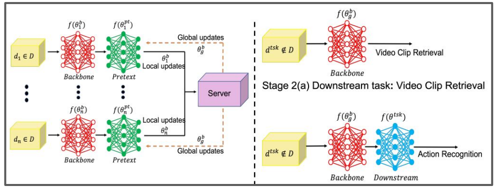


Case Study

Speech Recognition

Case Study

Carbon Footprint



Case Study

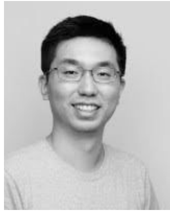
Self-Supervised Learning

EECV 2022

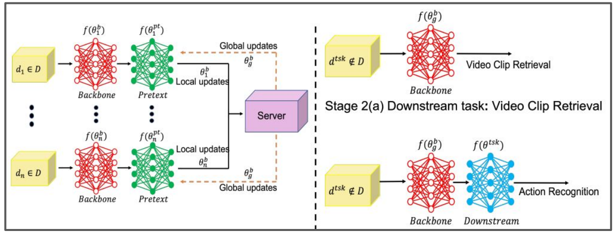
Federated Self-supervised Learning for Video Understanding

https://arxiv.org/abs/2207.01975





Yasar Abbas Ur Rehman, Yan Gao, Jiajun Shen
Pedro Porto Buarque de Gusmão, Nicholas D. Lane



Case Study

Self-Supervised Learning

FedVSSL *(method)*

Algorithm 1 Federated Video Self-supervised Learning (FedVSSL)

Input: $R, M, N, n_m, \eta_s, \eta_l, \alpha, \beta$
Output: θ^b
Central server does:

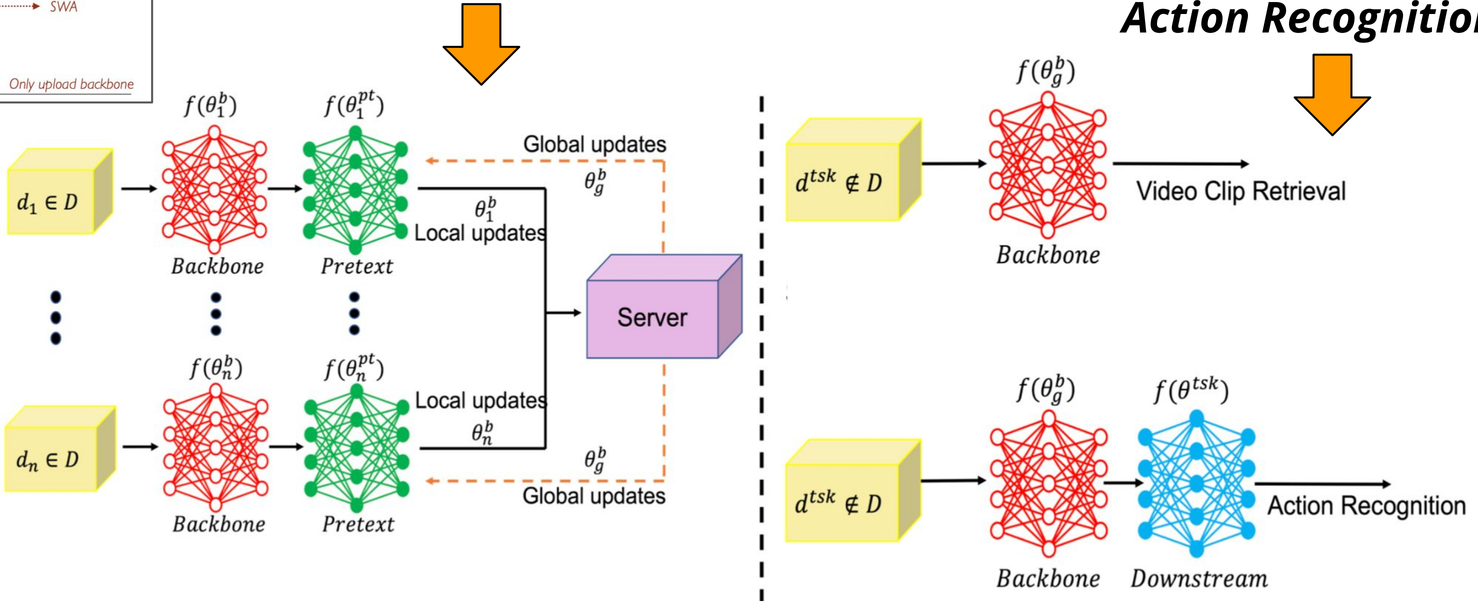
for $r = 1,\ldots, R$ **do**
  Server randomly sample M clients.
  for each m in M **do**
    $\theta^b_m(r), n_m, \mathcal{L}^{pt}_m = \textbf{TrainLocally}(m, \theta^b_g(r))$
  $g^{(m)}(r) = \theta^b_m(r) - \theta^b_g(r-1)$
  $\Delta^{FedAvg}_r = \sum_{m=1}^{M} \frac{n_m}{\sum_{m=1}^{m} n_m} g^{(m)}(r)$
  $\Delta^{Loss}_r = \sum_{m=1}^{M} \frac{\exp(-\mathcal{L}^{pt}_{(m)})}{\sum_{m=1}^{M} \exp(-\mathcal{L}^{pt}_{(m)})} g^{(m)}(r)$
  $\Delta_r = \alpha \Delta^{Loss}_r + (1-\alpha)\Delta^{FedAvg}_r$ → *Weighted combination*
  Update global model weights $\tilde{\theta}^b_g(r) \leftarrow \theta^b_g(r-1) - \eta_s \Delta_r$.
  Compute $\theta^b_g(r) = \frac{(\sum_{i=1}^{\beta} \tilde{\theta}^b_g(r-i)) + \theta^b_g(r)}{\beta+1}$. → *SWA*

TrainLocally $(m, \theta^b_g(r))$:

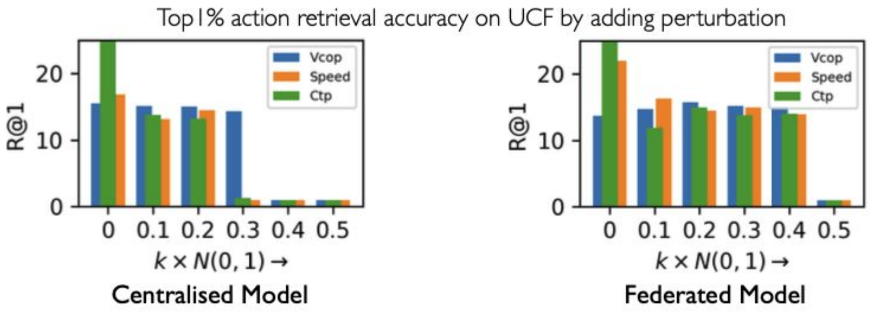
for $k = 1, \ldots, E$ **do**
  $\{\theta^b_m, \theta^{pt}_m\}(k+1) \leftarrow \mathrm{SSL}(\theta^b_g(k), \theta^{pt}_m(k), \eta_l)$
Upload $\theta^b_m, n_m, \mathcal{L}^{pt}_m$ to the server. → *Only upload backbone*

Stage 1
FL Video-SSL Pretraining

Stage 2
Downstream Task
Action Recognition



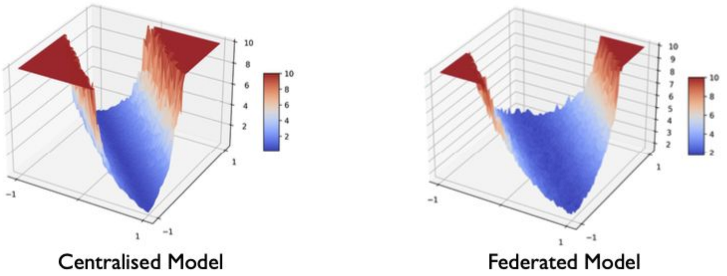
World's 1st Results: FL+SSL+Video



Set-up

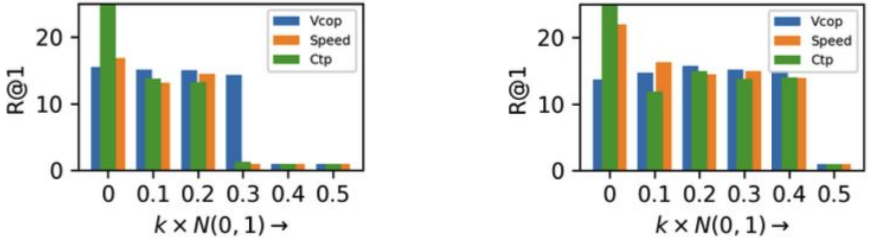
- Model: R3D-18 backbone
- FL Setup: 100 clients, each client 8 classes
- Pre-training Dataset: Kinetics-400
- Downstream Task: UCF-101 & HMDB-51
  - Video Clip Retrieval & Action Recognition

	Retrieval				Fine-tuning		Linear-probe	
	UCF		HMDB		UCF	HMD	UCF	HMD
Method	R@1	R@5	R@1	R@5	Top-1	Top-1	Top-1	Top-1
Centralised	29.00	47.30	11.80	30.10	86.20	57.00	48.14	30.65
FedAvg (Baseline)	32.62	50.41	16.54	35.29	79.91	52.88	45.31	31.44
FedVSSL(α=0, β=0)	34.34	51.71	15.82	36.01	79.91	52.94	47.95	31.12
FedVSSL(α=1, β=0)	34.23	52.21	16.73	**38.30**	79.14	51.11	47.90	29.48
FedVSSL(α=0, β=1)	35.61	52.18	**16.93**	37.78	79.43	51.90	47.66	30.00
FedVSSL(α=1, β=1)	**35.66**	52.34	16.41	36.93	78.99	51.18	48.93	31.44
FedVSSL(α=0.9, β=0)	35.50	**54.27**	16.27	37.25	**80.62**	**53.14**	**50.36**	**32.68**
FedVSSL(α=0,9, β=1)	35.34	52.34	**16.93**	37.39	79.41	51.50	50.30	32.42



World's 1st Results: FL+SSL+Video

Top1% action retrieval accuracy on UCF by adding perturbation

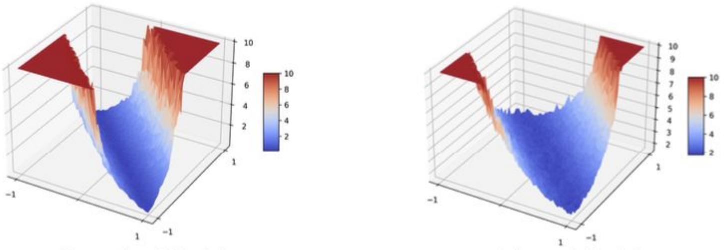


Centralised Model   Federated Model

Set-up

- Model: R3D-18 backbone
- FL Setup: 100 clients, each client 8 classes
- Pre-training Dataset: Kinetics-400
- Downstream Task: UCF-101 & HMDB-51
  - Video Clip Retrieval & Action Recognition

	Retrieval				Fine-tuning		Linear-probe	
	UCF		HMDB		UCF	HMD	UCF	HMD
Method	R@1	R@5	R@1	R@5	Top-1	Top-1	Top-1	Top-1
Centralised	29.00	47.30	11.80	30.10	86.20	57.00	48.14	30.65
FedAvg (Baseline)	32.62	50.41	16.54	35.29	79.91	52.88	45.31	31.44
FedVSSL(α=0, β=0)	34.34	51.71	15.82	36.01	79.91	52.94	47.95	31.12
FedVSSL(α=1, β=0)	34.23	52.21	16.73	**38.30**	79.14	51.11	47.90	29.48
FedVSSL(α=0, β=1)	35.61	52.18	**16.93**	37.78	79.43	51.90	47.66	30.00
FedVSSL(α=1, β=1)	**35.66**	52.34	16.41	36.93	78.99	51.18	48.93	31.44
FedVSSL(α=0.9, β=0)	35.50	**54.27**	16.27	37.25	**80.62**	**53.14**	**50.36**	**32.68**
FedVSSL(α=0,9, β=1)	35.34	52.34	**16.93**	37.39	79.41	51.50	50.30	32.42



Centralised Model   Federated Model

CaMLSys http://mlsys.cst.cam.ac.uk

Flower

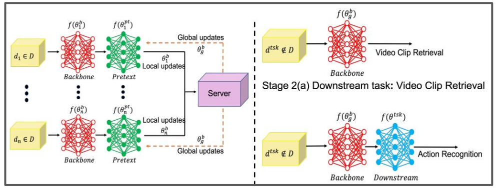
http://flower.dev

Case Study

Speech Recognition

Case Study

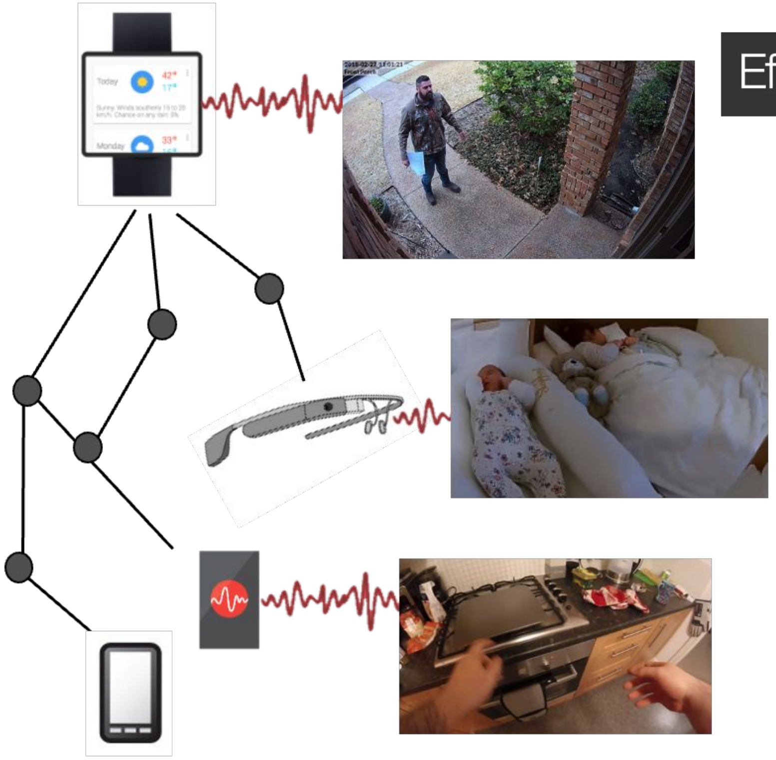
Carbon Footprint



Case Study

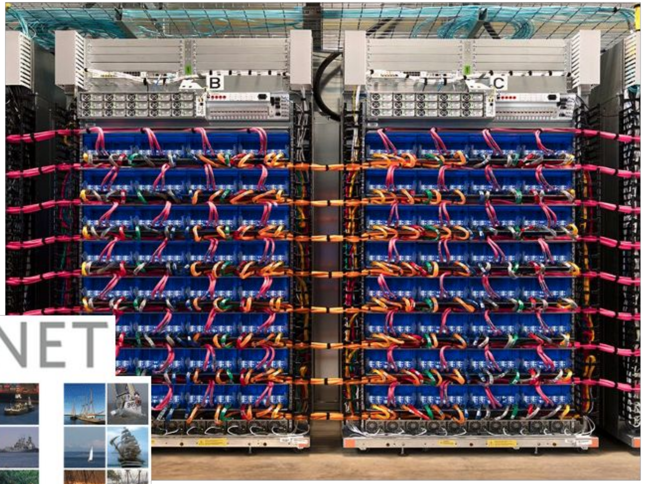
Self-Supervised Learning

Prediction: 3 years from now self-learning starts to push us away from data centers

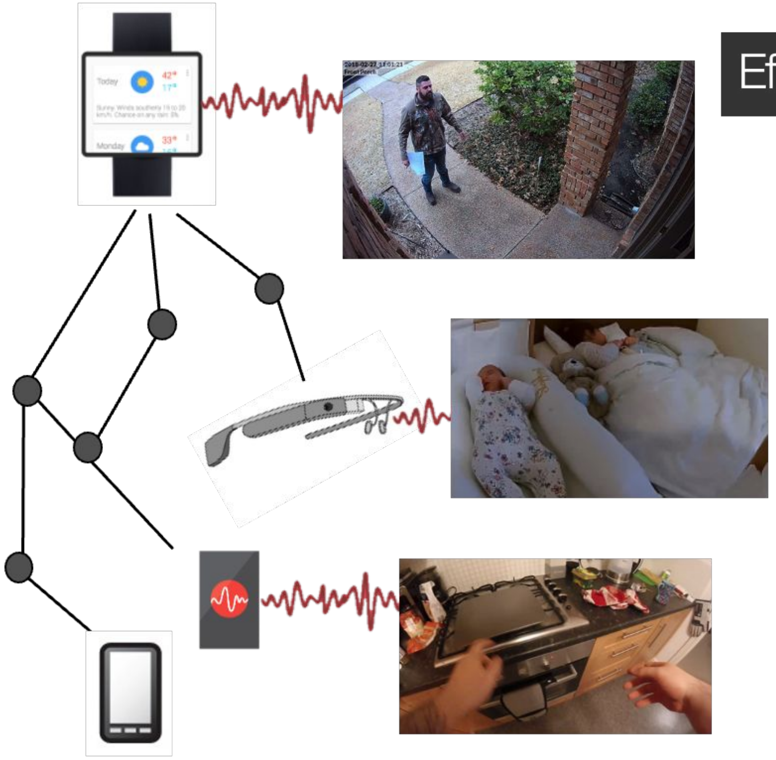


Efficient FL Devices

Data Centers

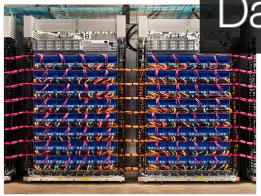


Prediction: 3 years from now self-learning starts to push us away from data centers



Efficient FL Devices

VS

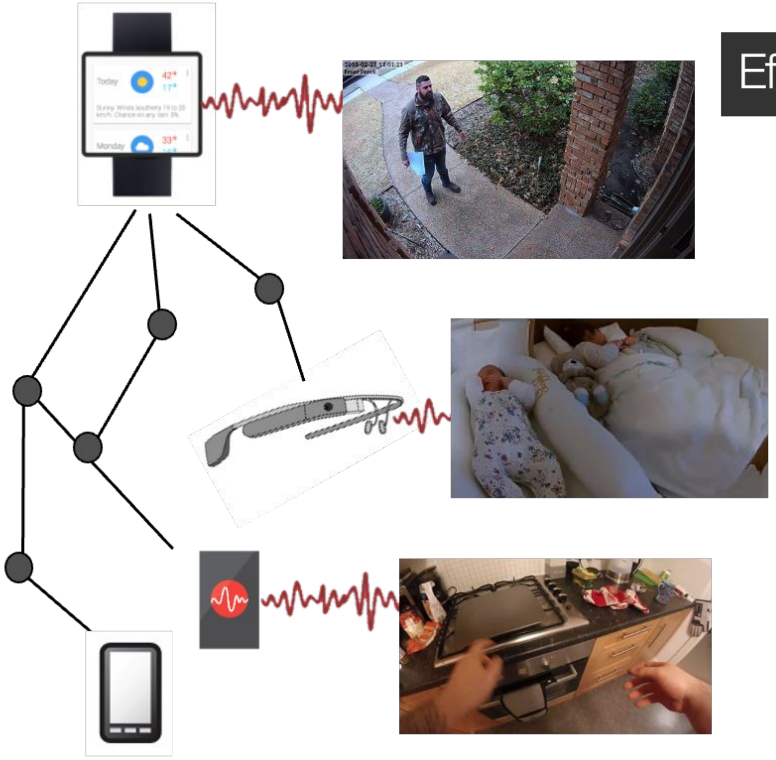


Data Centers

Unfair advantages of Federated SSL

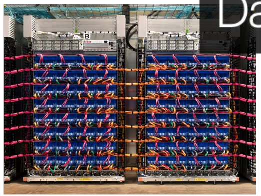
- Data avalanche unaffordable to send to the cloud
- Smaller Domain-shift
- Dynamic Open-world data

Prediction: 3 years from now self-learning starts to push us away from data centers

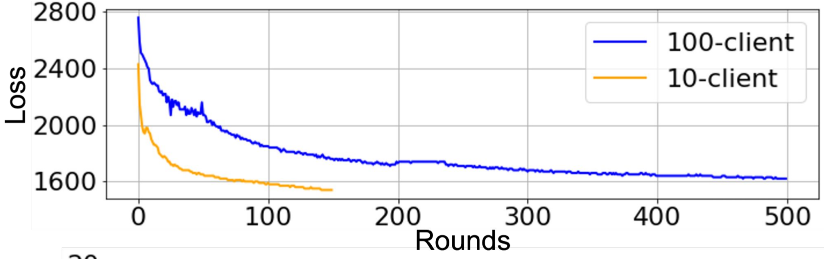


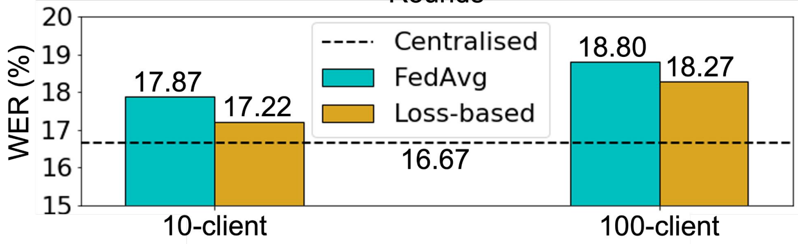
Efficient FL Devices

VS



Data Centers

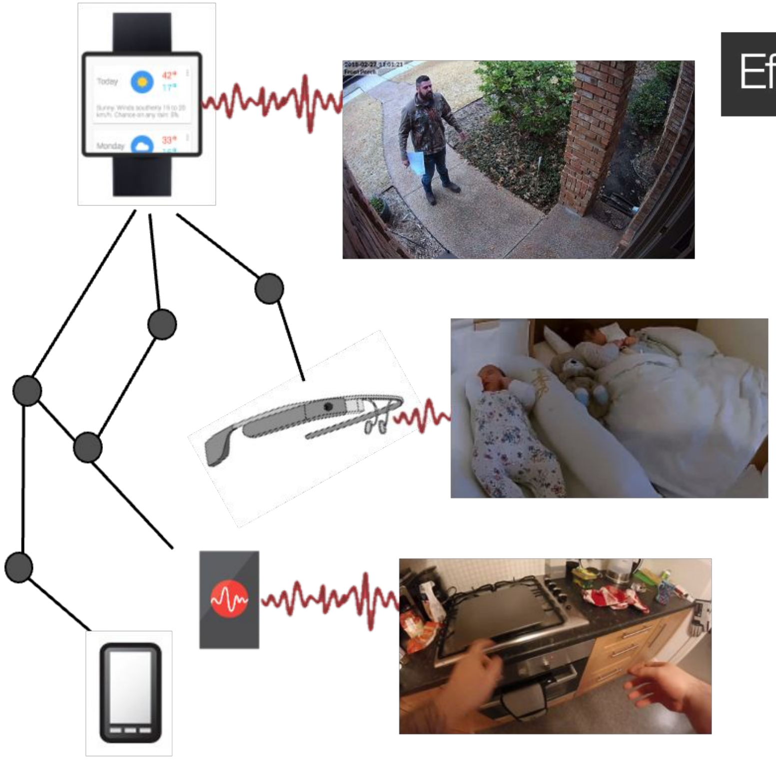




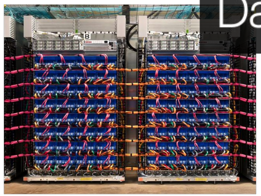
http://arxiv.org/pdf/2204.02804.pdf

Interspeech 2022

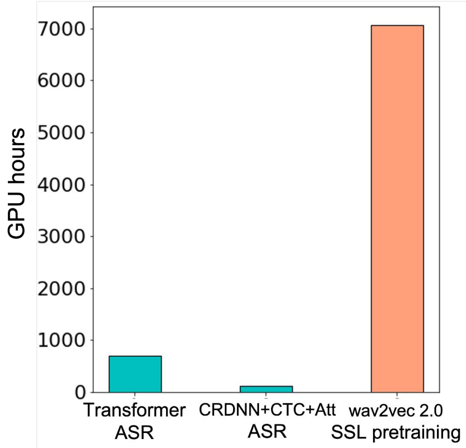
Prediction: 3 years from now self-learning starts to push us away from data centers

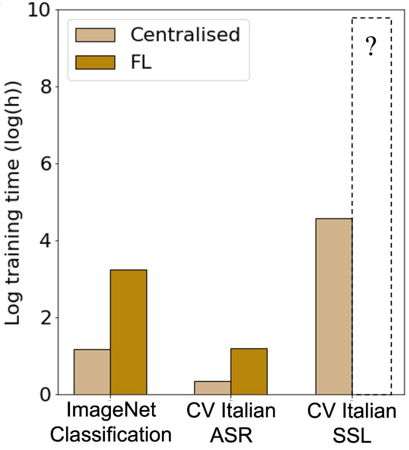


Efficient FL Devices



Data Centers





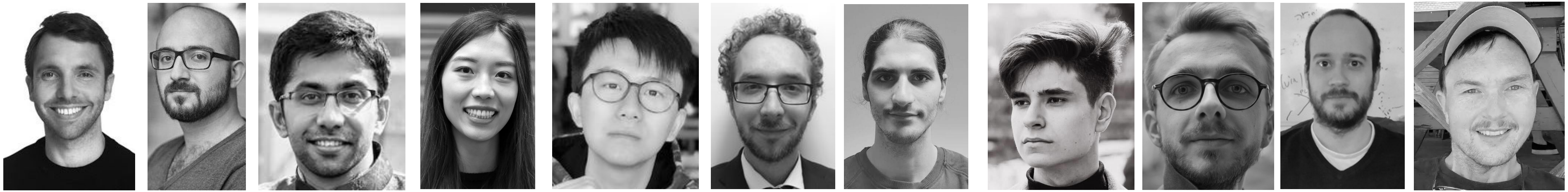
http://arxiv.org/pdf/2204.02804.pdf

Interspeech 2022

CaMLSys http://mlsys.cst.cam.ac.uk

OPEN SOURCE

\+ Community Driven



Questions? Comments?

Flower

http://flower.dev

Nicholas D. Lane

University of Cambridge | Flower Labs

@niclane7